

29
PAGES OF
EXPERT TUTORIALS

web

FREE 115 MINUTES
OF JS VIDEOS

designer™

HTML5 CSS3 jQuery JS JavaScript

@WebDesignerMag /WebDesignerUK

CSS LAYOUTS

We do the maths,
you do the design

SUPERFAST WEBSITES

20 vital tips and techniques to go
faster, get more hits & retain visitors

< CODE >
< SMALLER >
< SMARTER >

HOW TO UNLEASH THE
POWER OF WEB COMPONENTS

CODE WITH
CSS SHAPES

Create unique page designs

UNIVERSAL
JAVASCRIPT APPS

Get building with the next.js framework

Future

ISSUE 259

Digital Edition

GreatDigitalMags.com

GET INSTANT
ANIMATION

Introduce impressive dynamic CSS

CUSTOM
WP MENUS

Make bespoke vertical menus



FROM THE CREATORS
OF RESELLER HOSTING

GET YOUR
FIRST MONTH OF
**UNLIMITED RESELLER
HOSTING** FOR

£1

THEN ONLY
£29.99 A
MONTH

NO CONTRACT
+ 30 DAY MONEY BACK
GUARANTEE

SIGN UP AT
20i.com/design

Reseller Web Hosting on a 100% SSD, Autoscaling Cloud Hosting Platform.

TIM & JONATHAN BREALEY

FOUNDING DIRECTORS OF 20i & INDUSTRY VETERANS



UNLIMITED HOSTING

Includes unlimited SSD web space, bandwidth, websites, user accounts, 1GB MySQL databases, 10GB mailboxes and free SSL certificates.



EASY MIGRATION

Move your domains, websites, email and databases easily with our automated one-stop migration centre.

UNIQUE TO 20i



SSD AS STANDARD

We only fit top of the range solid-state drives in our servers to ensure blazing fast speeds and exceptional reliability.



AUTOSCALING

Our platforms scale resources on demand to ensure your website stays online and fast, even during the busiest times.



MULTIPLE PLATFORMS

Host websites on performance-tuned Linux, Windows and WordPress optimised platforms.

UNIQUE TO 20i



100% BRANDABLE

Give your customers their own customised control panel. It's sleek, packed with features, and user-friendly.

UNIQUE TO 20i

VISIT US

20i.com/design

EMAIL US

sales@20i.com

CALL US

0333 344 2733

Welcome to the issue

THE WEB DESIGNER MISSION

To be the most accessible and inspiring voice for the industry, offering cutting-edge features and techniques vital to building future-proof online content



Steven Jenkins

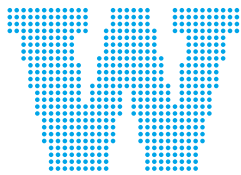
Highlight



The dream is of course to rely on our reputation alone to draw aspirational work through the door

Hungry for success in the digital arena, **Web Designer** talks to Salad. **Page 34**

Web Components = smarter HTML



Web Components have been touted as the future of HTML every since they were introduced in 2011.

Finally, six years and many tweaks later, browsers are finally catching up, making them an attractive proposition for web designers and developers.

Without Web Components, the very nature of HTML, CSS and JavaScript does not lend itself to developing in a modular way.

However, with Web Components users can create custom fully functional HTML elements. So, what else does it offer? Smaller code bases, inheritance of functionality, meaning easier

maintainability, easier-to-understand code and less testing. Check out our feature on page 42 for more insight and how to build your own custom component.

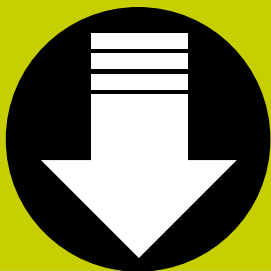
Another stalwart of web design, CSS, is finally getting to grips with the art of layouts. Thanks to Flexbox and the hot new topic CSS Grid, designers and developers are going to get the freedom they've always wanted. Turn to page 66 to get up close and personal with both and find out how to start building your perfect layout.

Elsewhere, we have a host of tips and techniques to speed up your site for both desktop and mobile. Learn how to get unique page designs with the help of CSS Shapes, build better navigation for WordPress and code universal JS apps. Enjoy the issue.

Web Components are a way of creating modular, custom components for use in web apps, in a manner that blends seamlessly with 'vanilla' HTML

Follow us on Twitter for all the news & conversation @WebDesignerMag

Visit our blog for opinion, freebies & more www.gadgetdaily.xyz



FREE - exclusive with this issue

247 Designer resources

Video Tuition - 115 minutes of expert video guides on Polymer.js from Pluralsight (www.pluralsight.com)

Assets - 200 Futuristic light leaks from SparkleStock (sparklestock.com)

- 38 Vintage noise textures from Slogan Wanted



www.filesilo.co.uk/webdesigner

This issue's panel of experts

Welcome to that bit of the mag where we learn more about the featured writers and contributors...

☛ New features enable us to define our own fully functional HTML and encapsulate the supporting code for simple delivery onto a page ☛



Simon Jones

Simon is a software engineering director in financial services, leading a team of full-stack developers. He comes from a background in computer science and UX design. This issue he delves into the world of HTML future and shows how to use Web Components for better builds. **Page 42**

Steven Roberts



Steven is a front-end developer working at Better Brand Agency. He studied Creative Digital Media at Teesside University where he fell in love with CSS. This issue he shows how to use Flexbox and Grid to get perfect layouts. **Page 66**



Tam Hanna



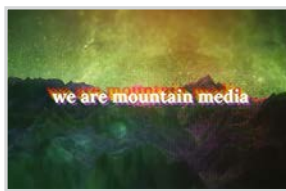
Tam's development career has seen him wrestle with a variety of stacks and languages. This issue he brings together 20 tips, tools and techniques to ensure that your sites are fast loading, keeping your visitors happy. **Page 74**



Mark Shufflebottom



Mark is a Professor of Interaction Design at Sheridan College of Advanced Learning. This issue he is creating CSS animations that are inspired by the Animatable project and that react to the user. **Page 52**



Jemmima Knight



Jemmima leads a full-stack scrum team developing enterprise-scale, customer-facing apps. This issue, with the help of Angular 2, she shows how to add a simple parallax effect to a single-page site. **Page 80**

Leon Brown



Leon is a freelance web developer and trainer who assists web developers in creating efficient code for projects. This issue he reveals a host of techniques, as seen on the top-class websites featured in our Lightbox. **Page 14**

Neil Pearce



Neil is a web and UI designer from Essex. He has been contributing to the magazine for many years. In this issue he demonstrates how CSS shapes can be used to create unique layouts in your projects. **Page 56**

Sarah Maynard



Sarah is a freelance WordPress developer, writer and digital artist based in Wiltshire, UK. In this issue she uses plugins, functions and CSS to make a simple easy-to-use vertical sidebar for a WordPress sidebar. **Page 60**

Matt Crouch



Matt is a front-end developer based in London. This issue, he takes a look at what goes into universal JS apps and how to build one with Next.js, making use of React components along the way. **Page 86**

Got web skills?
We're always looking for the hottest web-design talent. Email webdesigner@imagine-publishing.co.uk with examples of your creative work

web designer

Future Publishing Ltd
Richmond House, 33 Richmond Hill
Bournemouth, Dorset, BH2 6EZ
☎ +44 (0) 1202 586200
Web: www.gadgetdaily.xyz
www.greatdigitalmags.com
www.futureplc.com

Editorial

Editor Steven Jenkins
steve.jenkins@futurenet.com
☎ 01202 586233

Designer Harriet Knight
Photographer James Sheppard
Senior Art Editor Will Shum
Editor in Chief Amy Hennessey

Contributors

Harvey Wheaton, Craig Murphy, Leon Brown, David Howell, Simon Jones, Mark Shufflebottom, Neil Pearce, Sarah Maynard, Mark Billen, Steven Roberts, Tam Hanna, Jemmima Knight, Matt Crouch, Philip Morris, Phil King

Advertising

Digital or printed media packs are available on request.
Commercial Sales Director Clare Dove
Advertising Manager Mike Pyatt
☎ 01225 687538
michael.pyatt@futurenet.com
Account Director George Lucas
george.lucas@futurenet.com
Advertising Sales Executive Chris Mitchell
chris.mitchell@futurenet.com

International

Web Designer is available for licensing. Contact the International department to discuss partnership opportunities.
Head of International Licensing Cathy Blackman
☎ +44 (0) 1202 586401
cathy.blackman@futurenet.com

Subscriptions

For all subscription enquiries:
webdesigner@servicehelpline.co.uk
☎ 0844 848 8413
☎ Overseas +44 1795 592 878
www.imaginesubs.co.uk
Head of subscriptions **Sharon Todd**

FileSilo.co.uk

Assets and resource files for this magazine can be found on this website.
Register now to unlock thousands of useful files.
Support filesilohelp@imagine-publishing.co.uk

Circulation

Circulation Director Darren Pearce
☎ 01202 586200

Production

Production Director Jane Hawkins
☎ 01202 586200

Management

Finance & Operations Director Marco Peroni
Creative Director Aaron Asadi
Editorial Director Ross Andrews

Printing & Distribution

Printed by Southernprint Ltd, 17-21 Factory Road
Upton Industrial Estate, Poole, Dorset, BH16 5SN

Distributed in the UK, Eire & the Rest of the World by
Marketforce, 5 Churchill Place, Canary Wharf, London, E14 5HU
☎ 0203 787 9060 www.marketforce.co.uk

Distributed in Australia by Gordon & Gotch Australia Pty Ltd,
26 Rodborough Road, Frenchs Forest, New South Wales 2086
☎ + 61 2 9972 8800 www.gordongotch.com.au

Disclaimer

The publisher cannot accept responsibility for any unsolicited material lost or damaged in the post. All text and layout is the copyright of Future Publishing Ltd. Nothing in this magazine may be reproduced in whole or part without the written permission of the publisher. All copyrights are recognised and used specifically for the purpose of criticism and review. Although the magazine has endeavoured to ensure all information is correct at time of print, prices and availability may change. This magazine is fully independent and not affiliated in any way with the companies mentioned herein.

If you submit material to Future Publishing via post, email, social network or any other means, you automatically grant Future Publishing an irrevocable, perpetual, royalty-free licence to use the material across its entire portfolio, in print, online and digital, and to deliver the material to existing and future clients, including but not limited to international licensees for reproduction in international, licensed editions of Future Publishing products. Any material you submit is sent at your risk and, although every care is taken, neither Future Publishing nor its employees, agents or subcontractors shall be liable for the loss or damage.

© 2017 Future Publishing Ltd
ISSN 1745-3534



Future is an award-winning international media group and leading digital business. We reach more than 57 million international consumers a month and create world-class content and advertising solutions for passionate consumers online, on tablet & smartphone and in print.

Future plc is a public company quoted on the London Stock Exchange (symbol: FUTR).
Chief executive Zillah Byng-Thorne
Non-executive chairman Peter Allen
Chief financial officer Penny Ladkin-Brand
Tel +44 (0)1225 442 244
www.futureplc.com



1,500
FOOTBALL PITCHES
EVERY DAY!

Did you know that European forests, which provide wood for making paper and many other products, have grown by 44,000km² over the past 10 years? That's more than 1,500 football pitches every day![†]

Love magazines? You'll love them even more knowing they're made from natural, renewable and recyclable wood



[†] UNFAO, Global Forest Resources Assessment 2005-2015.

Two Sides is a global initiative promoting the responsible use of print and paper which, when sourced from certified or sustainably managed forests, is a uniquely powerful and natural communications medium.

There are some great reasons to [#LovePaper](#)
Discover them now,
[twosides.info](#)



contents

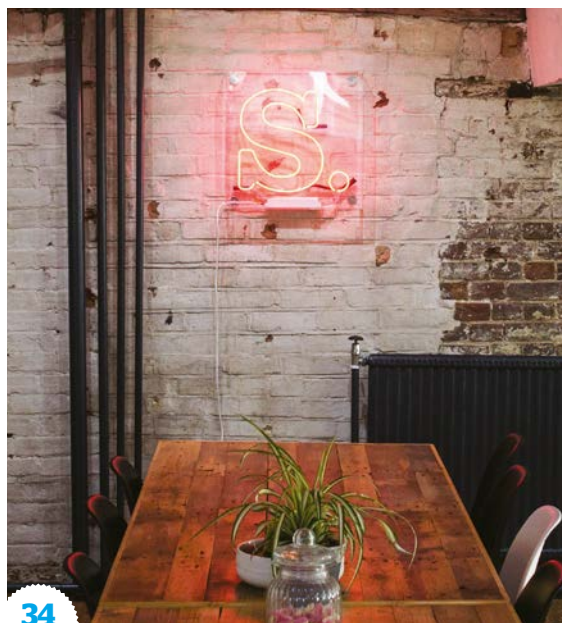
Cutting-edge features, techniques and inspiration for web creatives

Chat with the team and other readers and discuss the latest tech, trends and techniques. Here's how to stay in touch...

✉ webdesigner@imagine-publishing.co.uk 🐦 @WebDesignerMag 🖱️ www.gadgetdaily.xyz

Quick look...

- 08 What skills do you need?**
The industry reveals the tools and skills that they are using in 2017
- 10 Webkit: The best must-try resources out there**
Discover the libraries and frameworks that will make your site a better place to visit
- 11 Should designers learn to code?**
CodeClan CEO Harvey Wheaton suggests designers need to learn the basics of web development to be better at their job
- 14 Lightbox**
A showcase of inspirational sites and the techniques used to create them
- 26 Auto suggestion**
If you thought car paint sites were boring, think again. We find out how Red Collar changed people's opinion
- 34 Hungry design**
Creativity and an understanding of the digital space is a core strength that underpins every piece of work Salad produces. We find out more
- 42 Code smaller smarter HTML**
Take a look at how Web Components work, where they can be used and how to build your very own
- 66 CSS layouts. We do the maths, you do the design**
Discover the power of Flexbox and CSS Grid and find out how to create any design you want
- 74 Get superfast sites**
20 vital tips and techniques to go faster, get more hits and retain visitors
- 92 Course listings**
Want to start learning online? Check out what courses are out there with this list
- 94 Hosting listings**
An extensive list of web hosting companies. Pick the perfect host for your needs
- 98 Next month**
What's in the next issue of **Web Designer?**



34

ProFile: Salad

Understanding creativity and the digital space



14

Lightbox: FIFTEEN

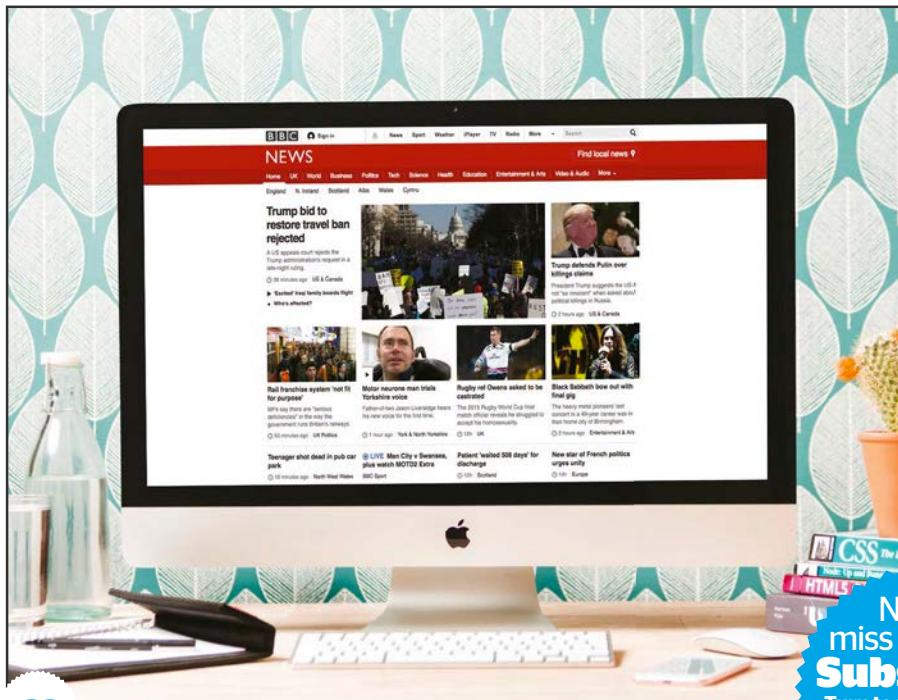
Getting clever with animation and interaction

FileSilo

96 Get the latest must-have resources and videos

A comprehensive collection of free designer resources!

- 115 minutes of expert Polymer.js video guides from Pluralsight
- 200 Futuristic light leaks
- 38 Vintage noise textures



66

CSS Layouts: Get to grips with CSS Grid and learn to build with Flexbox

<header>

The tools and trends to inspire your web projects

08 Getting the right web dev skills in 2017

What development skills are you going to learn this year?

10 Webkit: The best must-try resources out there

Need a new framework or library? Then this is where you should start

11 Comment: Harvey Wheaton

CodeCler's CEO suggests that designers could produce better designs by learning the basics of web development



Learning the basics of code can enhance so many things when designing a new website

Harvey Wheaton

Never miss an issue
Subscribe
Turn to page 32 now
USA readers turn to page 85 for an exclusive offer

Visit the **WEB DESIGNER** online shop at
www.imagineshop.co.uk
for back issues, bookazines and DVDs

<tutorials>

Web gurus take you step-by-step through professional techniques

52 Create multiple CSS animation effects

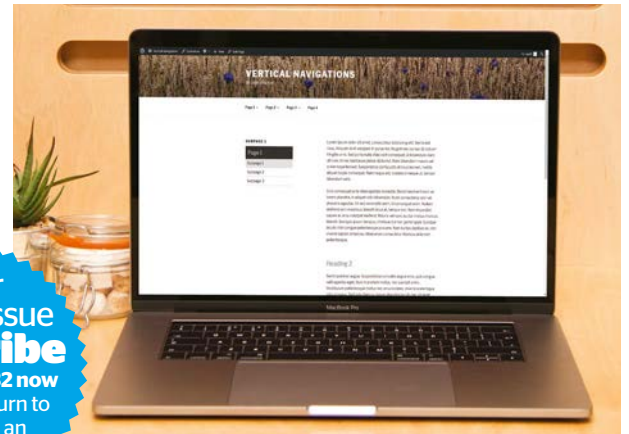
Make your site stand out from the crowd with stunning CSS animation effects that will make your content pop

56 Build unique designs with CSS

Learn how use the power of CSS Shapes to create multiple shapes for layout and presentation

60 Code custom menus for WordPress

Content is king, and by using vertical menus you can help readers navigate your pages quickly



web workshop

50 Code a drag-and-slide button

www.banxico.org.mx/mibanxico

Introduce animation and interactivity

64 Create an animated transition

hlebdom.ru

Add interest to splash screens

web developer

74 Get superfast sites

20 vital tips and techniques to go faster and get more hits

80 Create a parallax effect with Angular 2

How to implement a basic parallax scroll effect

86 Build a universal React site

Produce a lightweight React-powered site that is server-rendered by default with the help of the Next.js framework

Header

The tools, trends and news to inspire your web projects.

What web skills should you learn?

Industry experts reveal what web developers need to learn to stay relevant in 2017

There's no shortage of libraries, frameworks and languages to learn. It can be difficult for web designers and developers to choose their progression path. There are so many, it is simply impossible to learn them all. Your head would explode and you would never master any. What's the better choice? Learning a host of languages or mastering a few. We think mastering a few is the way forward. But even mastering a few doesn't mean that designers and developers can simply stop learning. Languages, libraries and frameworks are constantly being updated, meaning that users need to keep up. If you don't keep up with the latest update, you will eventually get left behind and still be creating code that is no longer supported. So, it makes sense to keep up to date with what's happening with your favourite languages, libraries and frameworks. This is pretty obvious, but the

real dilemma comes when it's time to learn something new. It might be that you have a new job, or it might be that the program of choice is no longer supported. A new job can bring with it new opportunities, but learning the right language, library or framework may bring with it new opportunities.

So, what do the pros have to say about what you should be considering in 2017? A recent report from Packt, entitled 'Skill Up: Developer Talk', takes a look at what web developers need to stay relevant. It gets comment from the industry, revealing what to learn. Matt Frisbie, a software engineer at Google and bestselling Packt author of *AngularJS Web Application Development Cookbook*, was asked, 'What tools and techniques should web developers learn in 2017?' 'Web developers will want to learn ES6 in 2017, and I'll also give a nod to TypeScript here,' he replied. 'React has a great foothold and there appears to be an

ever increasing demand for React developers. So, if you are looking at getting another notch on your tool belt, learning React may be a good choice. Last but not least, I suggest taking a look at Isomorphic JavaScript, which renders the JavaScript on both the client and server. It provides several advantages over how things were done 'traditionally.'

Benjamin Jakobus, a software engineer and co-author of *Mastering Bootstrap 4*, was asked what he thought were the most exciting up-and-coming tools and tech in web development. He replied: 'Vue.js seems to be gathering momentum quickly... Personally, I am excited about where Slack is going. The tool has really transformed how I work and collaborate, and the Slack team has made great strides over the past 12 months. On a frameworks front, the ones that I keep a keen eye on are Angular 2, Ember.js and MEAN.' Frisbie suggested that 'the new ECMA specifications, and further adoption of Web Components were going to have a big impact on web development.'

This brings us neatly to another question asked in the report: 'Should developers move from AngularJS to Angular 2?' Frisbie replied, 'It depends on the needs of the developer... Performance. Mobile friendliness. Reactive patterns and architecture. Platform independence. Angular 2 brings all these things to the table. For someone looking to put together a simple prototype, any platform that lets them move the fastest is the one to go with. But for someone looking to build a scalable platform built around upcoming high-octane web tech, Angular 2 is certainly the way to go.'

Learning the right languages and frameworks will only increase your chances of getting the job you desire. Check out the full report at bit.ly/214MjQx.

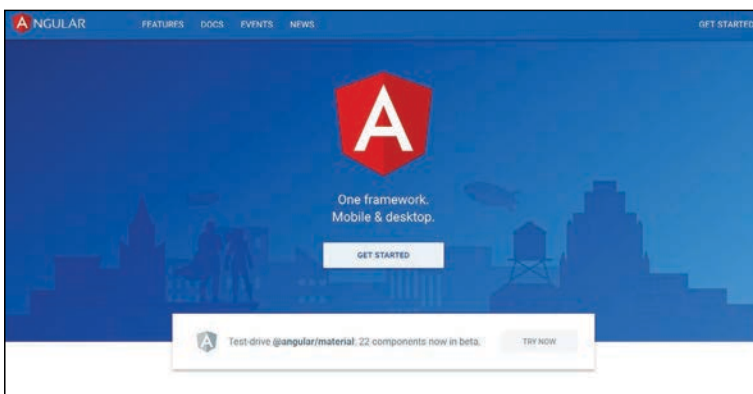
Get Web Designer digital editions

Get your hands on a collection of the best contemporary techniques

Need an instant hit of **Web Designer**? Then check our digital specials. **The Web Apps Handbook Volume 2** and **The Developer's Guide to JS** reveal the best tools and methods to build apps with web technologies and learn how to get started with ES6 and NPM.

Don't forget **Animate with HTML & CSS** and **3D and the Web**. We cherry-pick a collection of the best animation- and 3D-related techniques to add interest to any project. Plus, check out **20 Quickfire HTML & CSS Techniques**. How can you get your hands on a digital edition? Head to bit.ly/1hsGYgl, download the free Web Designer app and find them waiting for you as an in-app purchase.

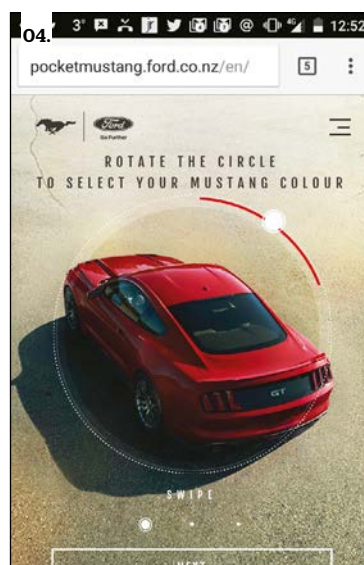
“If you are looking at getting another notch on your tool belt, learning React is a good choice”



Should you make the move from AngularJS to Angular 2?



Sites of the month



01. VERMONT Holding

vermont.eu
3D animations, smooth transitions, text effects and neat audio.

02. north2

north2.net/about-us.html
Smart interaction and colour palette engage the user with the About Us page.

03. RTRN

rtrn.nl
It's yellow and black and has some smart animation. What's not to like?

04. Pocket Mustang

pocketmustang.ford.co.nz/en/
Turn your mobile into a Ford Mustang and show off your drifting skills.

Graphics Business card

bit.ly/2kWtn8Y
Beautifully crafted business cards perfectly complemented by its colour palette



Colour picker Free Shattered Sword

bit.ly/2jXy4zL

#8FC73E

#C1D82F

#7DC242

#AFD136

#C1D82F

Typesetter Rig Shaded

bit.ly/2kPCaHv
A 3D type family with a geometric sans serif at its heart. The font is designed as a framework to support a range of 3D effects.



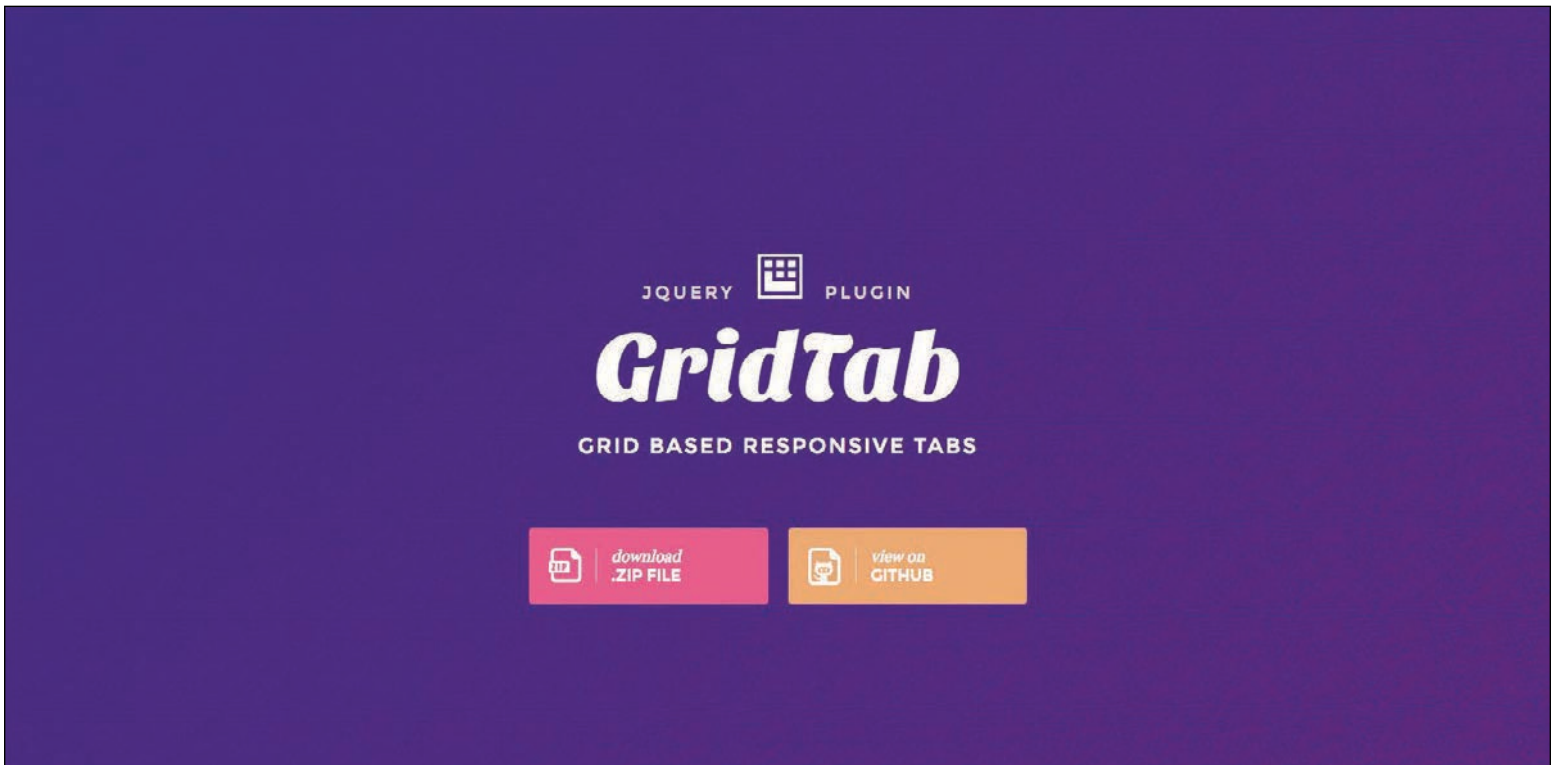
WordPress Hodges

bit.ly/2kHYq8k
A modern multipurpose theme that offers a different design for business, portfolio, agency, creative and more



webkit

Discover the must-try resources that will make your site a better place



GridTab

gopalraju.github.io/gridtab

With a clue in the name, GridTab is a lightweight jQuery plug-in that creates grid-based responsive tabs. Being responsive means that the setup works on all screens. GridTab supports two layout modes: grid and tab. By default, it uses the grid layout. Click a tab in the grid and it opens, like an accordion, to reveal additional information.



Vanilla-tilt.js

micku7zu.github.io/vanilla-tilt.js

A smooth 3D tilt JavaScript library forked from the jQuery version, Tilt.js. It is smooth, lightweight and offers neat title effects.



Service Mocker

service-mocker.js.org

Tagged as 'the next generation frontend API mocking framework', Service Mocker is based on the service workers API for in-browser processing.



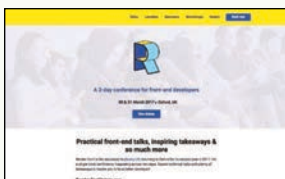
KUTE.js

thednp.github.io/kute.js/index.html

KUTE.js is a JavaScript animation engine that is hot on performance, memory efficient and has a bunch of tools for custom animations.

TOP 5 Web conferences – March 2017

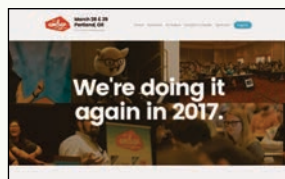
Get yourself a seat at the biggest and best conferences coming your way soon



Render Conference

2017.render-conf.com

A single-track two-day conference. Includes technical talks with plenty of takeaways.



EmberConf 2017

emberconf.com

A three-day conference with training, sessions and activities with the Ember Core Team and community members.



DIBI

dibiconference.com

Are you looking to shape the future of the web? An expert collection of speakers at 'Design It; Build It' are here to help.



React London

react.london

A single-day event packed with talks by talented people from the community. Discover the newest advances in React.



jDays

jdays.se

jDays brings together software engineers around the world to share their experiences in web, UX, Agile and digital trends

Should designers learn to code?

CodeClan CEO says web designers should learn the basics of web development

Harvey Wheaton

CEO at CodeClan

codeclan.com

W

eb design and web development. Two entirely different professions often lumped together by those who believe, or maybe hope, they

are the same thing.

Designers and developers have different priorities, but their common goal is the same – to create a fantastic, engaging, fully functional website or application delivering excellent user experience.

Nowadays, however, we are seeing an increasing number of ‘designer developer’ hybrids. It’s becoming more common for designers to dabble in the development process in a bid to understand how it works. And the fact is that web designers who choose to ignore at least the fundamentals of web development could potentially be worse off for it.

Learning the basics of code can enhance so many things when designing a new website, not least the valuable ‘designer developer’ relationship.

An understanding of development can improve the lines of communication between yourself and the developer you are working with, and effective communication makes for a more productive team. Knowing and using a vocabulary that developers understand helps communication flow as designers can better express their creative intentions.

Even the most basic knowledge of how the development process works can help a designer understand the needs of a developer, and this understanding can help designers be realistic about what’s achievable from the development side.

A grasp of programming can also save time as designers can focus on making creations compatible with the development of a site or application. This also gives a designer the chance to channel energy and focus on the parts of design that are not in the developer’s control.

Learning code can also open up a whole new realm of design possibilities. If a designer can understand why a developer has used a certain language or particular framework, they can use this to their



“ Learning the basics of code can enhance so many things when designing a new website ”

advantage from a design perspective when creating a stunning website or application.

And it shouldn’t be forgotten that designers choosing to upskill and enhance their understanding of how development works will also greatly boost their employability. A designer should see upskilling as a chance to broaden their skill set, adding another string to their bow, and improving what they can offer a potential employer or client.

You never know, you could uncover a hidden talent or even become one of those ‘designer developer’ hybrids – a valuable asset in today’s market!

It’s never been easier to upskill, whether you are a designer keen to know more about programming, or even a developer eager to tap into your creative side. There are facilities all around us to help us learn. We can broaden our skill set at the click of a button.

The web is a fantastic resource and is bursting at the seams with opportunity. Online forums like Stack Overflow and Reddit are brilliant ways to learn the basics and to feed off the inspiration and experience of other like-minded

people. Sites like these are more than just websites: they are a community. And better yet, they welcome passionate designers with open arms.

The web is also rich with educative tutorials and webinars. Take Codecademy, for example, which teaches development through fun and interactive online tutorials. And of course, sites like YouTube provide a great opportunity to learn the basics.

Coding ‘boot camps’ such as CodeClan’s are also a fantastic way to learn in an intense and accelerated way. Developers and designers can choose from an abundance of courses aimed at teaching everything from coding the basics, to front- and back-end development, to a variety of different programming languages. These courses can be a great way for designers to understand how development works, to learn how developers think and to arm themselves with

the relevant skills and knowledge, all in a supportive learning environment.

The value of a local community as a source for designers to upskill mustn’t be underestimated, as this can be a fantastic way for designers to learn and liaise with people with similar passions. Here in Edinburgh, for

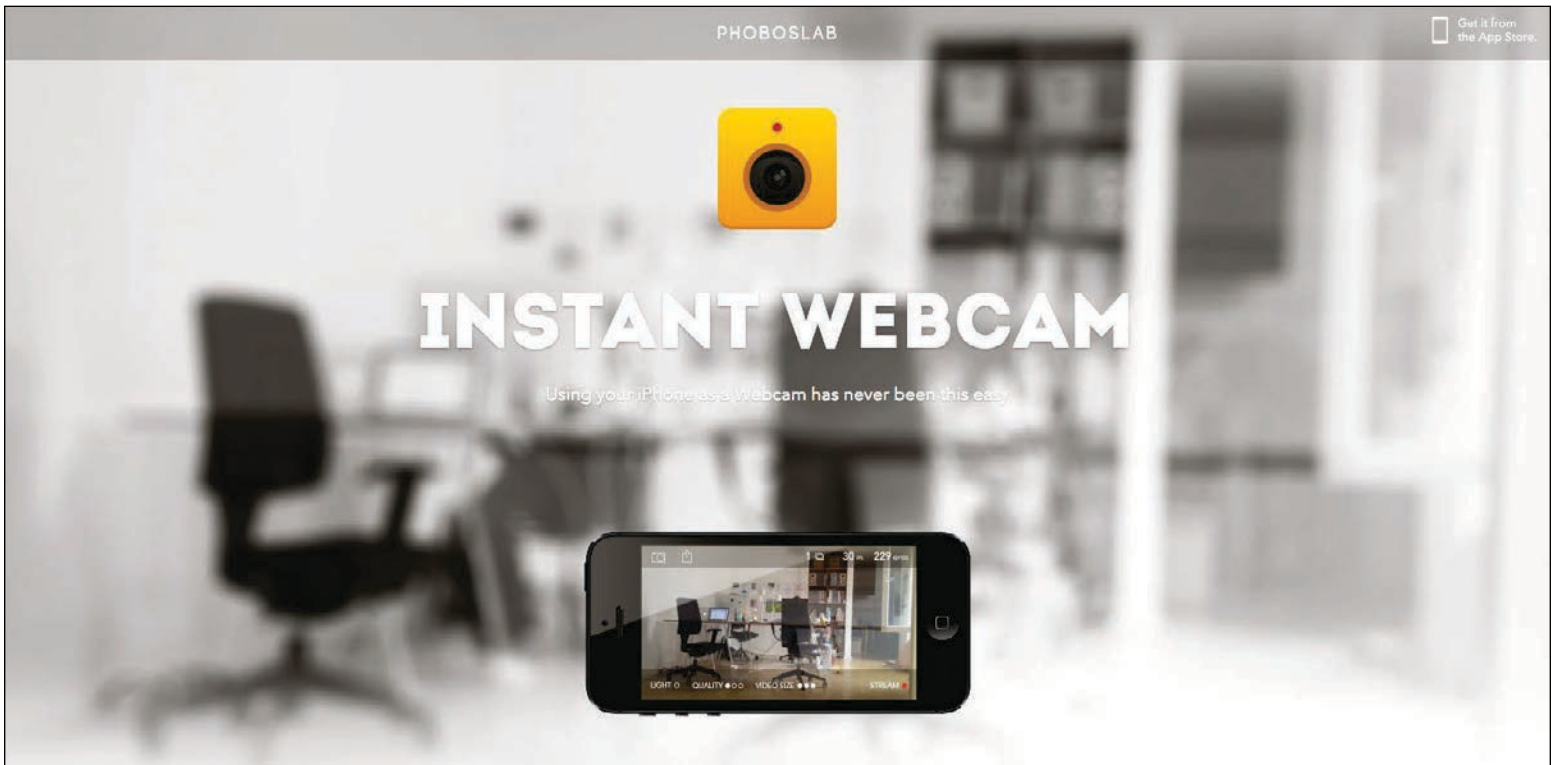
example, we have the web design and development meet-up group, a social group that meets regularly to share key topics, trends, methodologies and, most importantly, learnings from both sides of the design/development gulf.

The opportunity to learn can present itself on a day-to-day level too. Knowledge and an understanding of code can be built up through the likes of paired-programming between designers and developers. Paired-programming is an agile programming method that usually sees two developers work together on the same project. Not only does working in teams increase learning and understanding, but it also encourages faster, more efficient communication.

Some believe that designers are the architects of the web and developers are the builders. But nowadays, an increasing number of designers and developers are rather sensibly blurring the lines in a bid to learn both. So, to code or not to code?

webkit

Discover the must-try resources that will make your site a better place



Instant Webcam

instant-webcam.com

Want to use your iPhone as a webcam? Instant Webcam lets you stream fullscreen video to multiple devices in the same Wi-Fi network with ease. Encoded video is sent via Web Sockets to your web browser, which decodes the stream in JS and paints to a Canvas.



Lottie

github.com/airbnb/lottie-android

Lottie is a mobile library for Android and iOS. It parses Adobe After Effects animations exported as JSON and renders them natively on mobile.



WebGazer.js

webgazer.cs.brown.edu

An eye-tracking library that uses common webcams to infer the eye-gaze locations of web visitors on a page in real time.



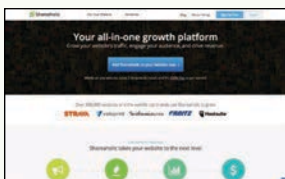
Intercooler.js

intercoolerjs.org

This is tagged as 'AJAX with attributes'. Use declarative HTML attributes to add AJAX to your application. Check the examples to see what Intercooler can do.

TOP 5 Tools to boost social media

Build up your brand and get yourself shared and seen



Shareaholic

shareaholic.com

Add 1-click sharing of content from your site to social media. Analyse data, add follow buttons and create a campaign.



Hootsuite

hootsuite.com

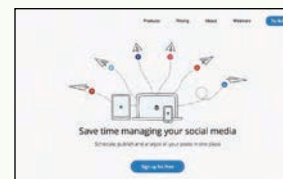
Get the tools to manage all your social profiles and automatically find and schedule effective social content.



Socialloomph

socialloomph.com

A free option lets you schedule tweets, track keywords, and save and reuse drafts. The Pro option offers even more.



Buffer

buffer.com

Managing your social media effectively isn't easy. Buffer lets you schedule, publish and analyse all posts in one place.



Bitly

bitly.is/2kiwxjP

Get yourself a branded short domain and start using it in your social media to get your brand out there.



Snapchat : A new tool for marketing

Snapchat isn't just for teenagers: it's emerging as a leading platform in B2B

Craig Murphy

Managing director at ALT

altagency.co.uk



napchat started life with teenagers using it as a bit of fun, sending each other funny images/videos that automatically deleted themselves after ten seconds. In the early days, Snapchat got a bit of bad press when people were sending sensitive material to each other via the platform, but since then marketers have jumped on board and started to use it wisely.

Snapchat has over 100 million users, many of whom fall into the 18-34 age range, so if that is your demographic, it's a great place to be seen by them.

Firstly, the platform should be used as part of your wider plan to distribute your content. The Snapchat stories last for up to 24 hours, so it's easy to string together multiple ten-second snaps that tell a story and drive engagement.

Secondly, Snapchat is showing incredible levels of engagement and attention. Why? Because a snap only lasts for ten seconds when sent privately, when

people receive snaps they are more focused and their attention is held better as they know that after ten seconds the content is gone forever.

So, how can you leverage Snapchat for your business with those two points above in mind?...

- Snap access to live events: If you're running live events, Snapchat offers a way to create hype and engagement by providing a view of what's going on at your event.
- Speak 1-on-1 with your audience: If you follow Gary Vaynerchuck, for example, you will have seen his snaps and felt the depth and personal connection that can be delivered through a snap. By going deep into subjects and providing value on your industry, you can connect far better with your audience. Far too many people post spam - go deep and provide value!
- Leverage snap takeovers: One way to grow your audience is to take part in what's known as a 'snap takeover', which means you can partner up with an influencer who has access to your market and share your

snaps through their Snapchat account and reach their audience. This not only increases your reach but allows you to grow your snap following - this is one of the quickest ways to get scale through your Snapchat activities.

- Provide teaser content: A lot of successful models, beauty bloggers and fitness gurus have used this method. They provide teaser videos of ten seconds and drive traffic from the snap to their sign-up pages. This is a great way to boost sales.

Those are just four simple ways that you can leverage Snapchat marketing.

There are lots of creative ways to use the platform. For inspiration and general industry knowledge, we advise you follow Gary Vaynerchuck, Mark Suster, everlane.com and McDonald's on Snapchat - all of whom are pushing the platform to grow their audience, drive engagement, provide depth and create product awareness in very creative ways.



SKULLMON

FIFTEEN

www.agency15.com

Designer: Agency 15 - www.agency15.com

Development technologies HTML5, jQuery, selectivizr, SVG

“New York agency *FIFTEEN* evokes the spirit of perspective, perception and rhythm with this enigmatic online home”

KEY | submit



#D34953



#E4B46C



#D87E32



#470D05

abcABC 123456789

Above

Futura BT Medium, by Paul Renner for Bitstream, is the only font style used throughout the site, mainly for menu links.

Create a video background

Capture the attention of visitors and guide their eye to the content you want them to focus on

1. Page document

The first step is to initiate the page document, which consists of the HTML document container, along with the head and body sections. The head sections contains descriptive information such as the title, along with resource links - i.e. the CSS stylesheet. The body section will be used to store the visible content created in the next step.

```
<!DOCTYPE html>
<html>
<head>
<title>Video Background</title>
<link rel="stylesheet" type="text/css"
href="styles.css" />
</head>
<body>
<main>
*** STEP 2
</main>
</body>
</html>
```

2. Document content

The document consists of a video for the background, along with text content - placed inside the main tag. As different browsers support different video formats, the video is a container for sources that the browser can decide to choose from. The foreground content is kept separate inside the main container, allowing us to keep control from CSS.

```
<video autoplay muted loop>
<source src="http://www.w3schools.com/
html/mov_bbb.mp4" type="video/mp4">
<source src="http://www.w3schools.com/
html/mov_bbb.ogv" type="video/ogg">
</video>
<main>
<h1>Heading</h1>
<nav>
<a href="#">Option</a>
<a href="#">Option</a>
<a href="#">Option</a>
</nav>
```

```
</main>
*** STEP 3
```

3. Initiate CSS stylesheet

With the HTML content now in place, the next step is to initiate the CSS stylesheet. Create a new file called "styles.css", in which you will start to enter the CSS rules. The rules for this step ensure that the HTML document and its body display to cover the full width of the screen. It's important that the background is transparent so that the video is visible.

```
html,body {
display: block;
width: 100%;
height: 100%;
margin: 0;
padding: 0;
font-family: Helvetica, sans-serif;
background: transparent;
color: #fff;
}
```

4. Video positioning

We reference the background video as the video tag - but you might want to apply an ID if you're using multiple videos. The video is placed using fixed positioning and responsive resizing through min-width and min-height to ensure that it covers the full page. Additionally, z-index is used to ensure the video displays below any page content, while transform with left and right attributes are used to position the video from the centre.

```
position: fixed;
top: 50%;
left: 50%;
min-width: 100%;
min-height: 100%;
z-index: -9999;
transform: translateX(-50%)
translateY(-50%);
background: #000;}
```

5. Article content styling

The main container is used for controlling how the page content is displayed over the background video. To make sure the text is readable, we apply a semi transparent background using rgba colouring. The width is set to 75% of the screen width, with a margin set to position it in the centre.

```
main {
background: rgba(0,0,0,0.5);
```

```
padding: 2rem;
width: 75%;
height: 100%;
margin: 0 auto 0 auto;
}
```

6. Navigation container

The remaining part of the page to define the styling of is the navigation section. We start this with the rules for the navigation container - this is the nav tag that contains the navigation links. This is set to display as full width with a padding of 2 characters on all sides. To ensure that links appear in the middle, we apply a text-align of "center".

```
nav {
display: block;
width: 100%;
padding: 2em;
text-align: center;
}
```

7. Navigation links

We reference the navigation links as the first level elements inside the navigation container through "> *". This allows you to change items from links to other elements that have Javascript based event listeners for interaction. These elements are set to appear with spacing through margin and padding, along with a changing border when hovered over.

```
nav > *{
color: #fff;
text-decoration: none;
padding: 1em 2em;
margin-right: 1em;
border: 0 dashed #fff;
transition: border .5s;
}
nav > *:hover{
border-bottom: .5em solid #c00;
}
```

Visitors: Please pay attention

Use video background to make website visitors pay attention to content areas you want to promote as a higher priority.



LOSE YOURSELF IN A WORLD OF

Vinyl

FIND YOURSELF IN
OXFAM'S ONLINE SHOP

oxfam.org.uk/shop



OXFAM



RET
TO NA



Rainforest Foods

www.rainforestfoods.com/experience/#!/

Designer: Immersive Garden - www.immersive-g.com

Development technologies HTML, CSS

“The transition effect helps content sections to appear more connected to each other”



#F5E0B2



#DF7B49



#AB916A



#6AA248

abcABC 123456

Above

Qanelas by Radomir Tinkov is a modern sans serif font and appears in Bold and Black forms

Create a morphing content effect

Introduce text content in a way that looks like it is being typed in real time

1. Page Document

Initiate the project with the creation of the HTML document structure. This consists of the head for storing descriptions and links to external resources, while the body section contains the visible page content. In this example, we also have the main container within the body - which is where the content elements for steps 2 and 3 will be placed.

```
<!DOCTYPE html>
<html>
<head>
<title>Title Morph</title>
<link rel="stylesheet" type="text/css"
href="styles.css" />
</head>
<body>
*** STEP 2
</main>
</body>
</html>
```

2. Navigation Content

The navigation is made from a nav container, which contains a series of anchor links. These links reference elements on the page through the use of the hash character in the href attribute. This is important to make sure that the browser doesn't try to load an external page when clicked.

```
<a href="#first">Option</a>
<a href="#second">Option</a>
<a href="#third">Option</a>
</nav>
*** STEP 3
```

3. Article Content

An article container is used to store the article content - each of which will be stored inside a section. Each section has an ID attribute, which is used by the navigation links and CSS to show the targeted content. For this example, content is limited to titles that are to be presented, but you should add more content.

```
<article>
<section id="first">
<h1>First Title</h1>
</section>
```

```
<section id="second">
<h1>Second Title</h1>
</section>
<section id="third">
<h1>Third Title</h1>
</section>
</article>
```

4. Initiate Styling

With the HTML now complete, the next step is to initiate the CSS stylesheet. Create a new file called "styles.css", in which you will enter this step's style rules. This first step of styling will set the HTML document and its body to display across the full browser window without any border margin or padding. The default background will be set to black and font text set to white.

```
html,body{
display: block;
width: 100%;
height: 100%;
margin: 0;
padding: 0;
font-family: Helvetica, sans-serif;
background: #000;
color: #fff;
}
```

5. Article Content Styling

The main article content will be positioned inside the article container, with each section appearing as they are referenced through the URI. Firstly, articles are placed using absolute positioning that's relative to the article container. Each section has zero opacity visibility by default. When a section is selected via the URI, its opacity changes to become fully visible - this is animated through the activation of the transition attribute.

```
■ article{
```

```
position: relative;
}
article section{
position: absolute;
top: 0;
left: 0;
opacity: 0;
transition: opacity 1s;
}
article section:target{
opacity: 1;
z-index: 9999;
}
```

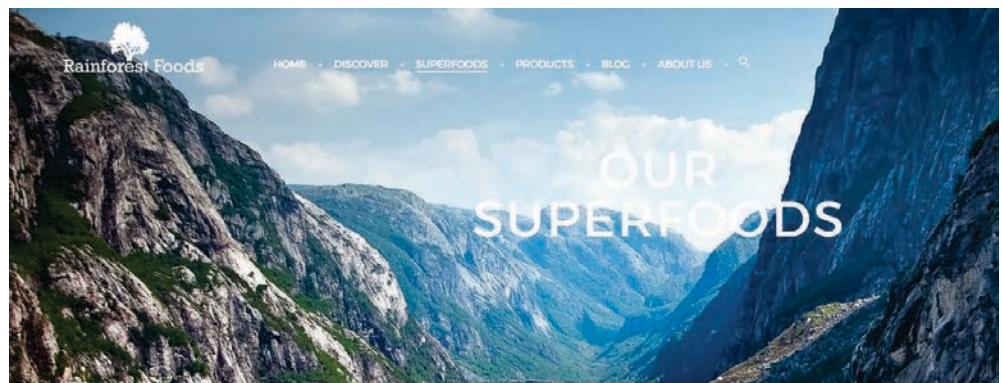
6. Page Navigation

The last step is to define the navigation, which is made from a nav container and the associated links. The nav container is set to fit the full width of its parent container, with its content displayed in the centre. Each "a" tag inside this will have padding and margin for spacing, with an additional border appearing when they are being hovered.

```
nav{
display: block;
width: 100%;
padding: 2em;
text-align: center;
}
nav a{
color: #fff;
text-decoration: none;
padding: 1em 2em;
margin-right: 1em;
}
nav > a:hover{
border-bottom: .5em solid #c00;
}
```

Segmenting Content

Avoid the need to store separate content on individual pages, which in turn require pages to be loaded between navigation interactions.





Recycle your magazine and seven days later it could come back as your newspaper.

www.recyclenow.com



The possibilities are endless.

MARCH SALE 25% OFF
#ZeroCarbonHosting

PROMO CODE: WED3

0800 808 5450
netcetera.co.uk/hosting

Slash your internet, TV & phone bills!

**One quick call.
Thousands of deals.**

We do the hard work for you:

- 1 Call free and speak to our London based impartial experts.
- 2 We review over 6,000 deals from the top UK providers to find the right one for you - with our Lowest Price Guarantee.
- 3 Our experts do the hard work, answer your questions and even arrange your installation!

We're proud to be the only service to be accredited by Ofcom for TV, broadband and home phone comparison



LOWEST PRICE GUARANTEE

Save up to £408* per year

83% of our customers save**



Call us free and we'll do the hard work!

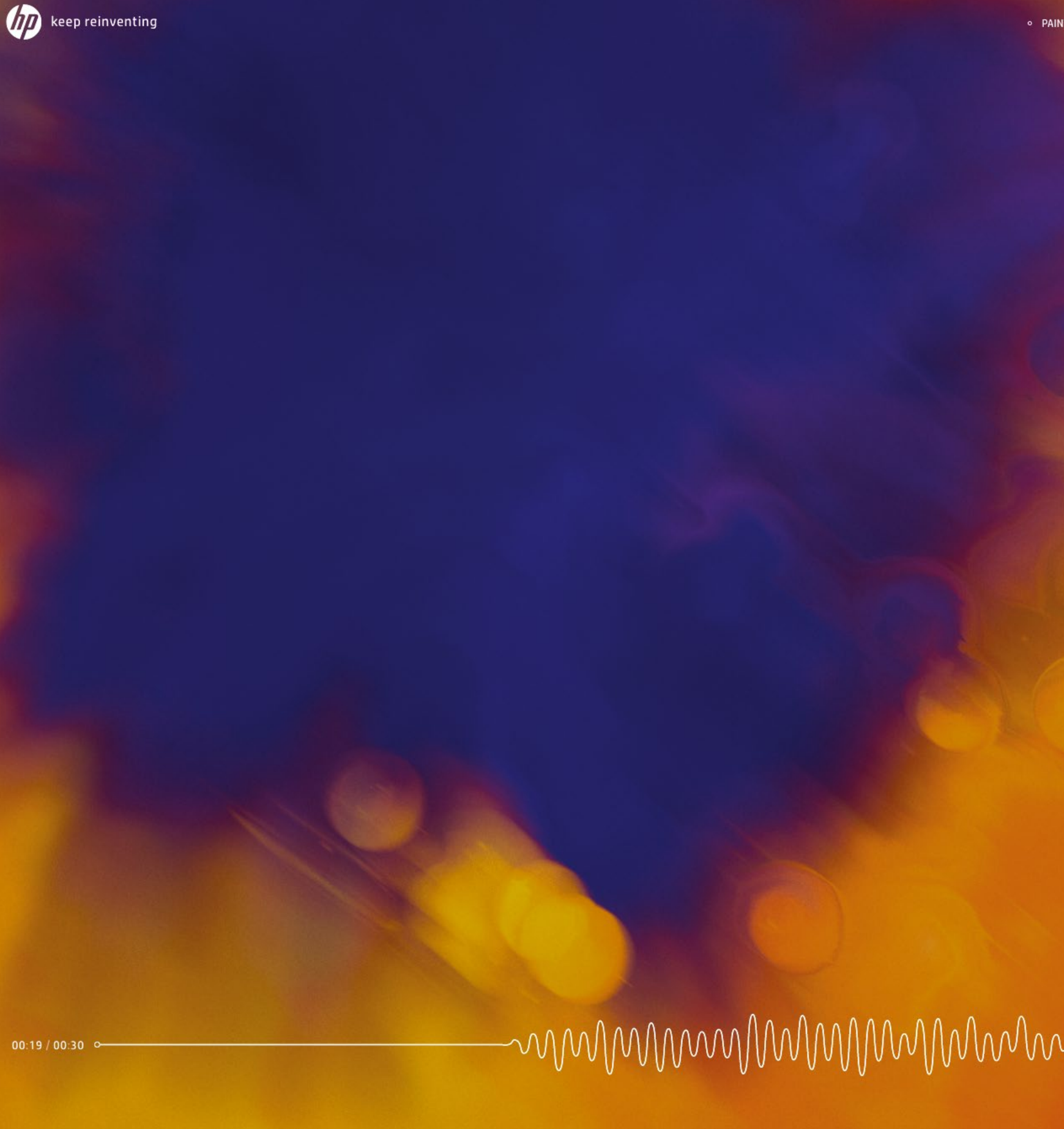
0800 466 1053

www.techradarstores.co.uk/digital/save Or text "DEAL" to 81400 and we'll call you

Experts available:
 Monday-Friday 8am-8pm,
 Saturday 9am-5.30pm,
 Sunday 10am-5.30pm

simplifydigital

*20% of Simplifydigital customers received an average saving of £408. Data based on 1,538 records between 1st of Jan 2010 to 8th Nov 2010. ** Out of 445 consultations between 01/01/2010 and 03/05/2010, 368 Simplifydigital customers saved money by signing up to our recommended deal. Lowest Price Guarantee Terms and Conditions apply - see www.simplifydigital.co.uk/lowest-price-guarantee for full details. Standard SMS network charges apply.



00:19 / 00:30 ◦

HP - Sound in Color

www.hpsoundincolor.com

Designer: Resn - www.resn.co.nz

Development technologies jQuery, WebGL, GSAP, Web Audio API

“HP invites us to ‘reinvent giving’ with sound, generating unique colourful artworks before gifting the results to a loved one”



#00006E



#CA8220



#430085



#A83331

abcABC

Above

Unsurprisingly, the site makes exclusive use of HP's own font starting with HP Simplified Regular.

abcABC

Above

The HP Simplified Light typeface is used across straplines and subsequent text throughout the site.

Create a microphone notification popup

Create a notification container that tells the user to activate a microphone

1. Page document

Start the project by defining the HTML document structure. This includes the HTML document container, along with its associated head and body section. The head section contains descriptive information and links to external resources – the CSS stylesheet and JavaScript code in this case. The body section will contain the visible content elements created in the next step.

```
<!DOCTYPE html>
<html>
<head>
<title>Microphone Notification</title>
<link rel="stylesheet" type="text/css"
href="styles.css" />
<script src="code.js" type="text/
javascript"></script>
</head>
<body>
*** STEP 2
</body>
</html>
```

2. Content elements

The page's content elements will consist of a container for the available messages, along with a button to activate the microphone detection. To keep the page maintainable, we are placing the available notifications within the HTML – meaning that designers who want to make changes don't need to modify any JavaScript.

```
<div id="notify">
<span data="detected">Click here to
activate microphone</span>
<span data="failed">Your browser doesn't
support microphone capture.</span>
</div>
<button id="activate">Activate
Microphone</button>
```

3. JavaScript check function

With the HTML now complete, create a new file called 'code.js' to store your JavaScript code. This step creates a JavaScript function that detects whether the 'getUserMedia' functionality is available from the browser. If this functionality is detected, the function will return

true; if not, it will return false instead.

```
function checkGetUserMedia(){
var result = false;
if(navigator.getUserMedia || navigator.
webkitGetUserMedia ||
navigator.mozGetUserMedia || navigator.
msGetUserMedia)
result = true;
return result;
}
```

4. Popup activation

Two event listeners are required to activate the notification popup. Firstly, we must wait until the page has loaded – otherwise the code won't be able to find the button. Secondly, we must apply a 'click' event listener to the button. When the button is clicked, the corresponding event listener function will decide whether to apply a value of 'detected' or 'failed' to the notification container. The corresponding presentation rules are left to the CSS stylesheet.

```
window.addEventListener("load", function()
{
document.querySelector("#activate").
addEventListener("click", function(){
if (checkGetUserMedia()) {
document.querySelector("#notify").setAttri
bute("data","detected");
} else {
document.querySelector("#notify").
setAttribute("data","failed");
}
});
});
```

5. Initiate CSS

The JavaScript code is now complete, so create a new file called 'styles.css' to define the visual presentation rules. The first step in this file is to set the document and its body to fit the full size of the available page. Font style, colour and background properties are also set here.

```
html,body{
display: block;
width: 100%;
height: 100%;
margin: 0;
padding: 0;
font-family: Helvetica, sans-serif;
background: rgb(21, 34, 47);
color: #fff;
}
```

6. Notification container

The notification container is set with relative positioning to ensure inner elements can be positioned in relation to it if required. Additionally, we set the font size to be three times larger than the default, along with associated padding. A visible border is also set to fit around the content.

7. Up arrow

The notification container needs to appear to point towards where the microphone activation notification will appear. We achieve this by using the 'before' element that CSS can apply to the container. This element will use its borders to appear as a triangle. Its absolute positioning combined with the relative positioning of its parent means that we can place the element with specific coordinates in relation to where the notification appears.

```
#notify::before{
content: "";
position: absolute;
top: -1em;
left: 0;
width: 0;
height: 0;
border-style: solid;
border-width: 0 2em 1em 2em;
border-color: transparent transparent #fff
transparent;
}
```

8. Notification appear

A set of rules are required to define the conditions for when the notification container and its message elements are to appear. Firstly, we set a rule to say that the container and its elements are not displayed by default. Secondly, we state that the container is displayed when a 'data' attribute is applied. Its inner messages are also displayed when the container data attribute matches their data attribute.

```
#notify,
#notify > *{
display: none;
}
#notify[data],
#notify[data="detected"] >
[data="detected"],
#notify[data="failed"] > [data="failed"]{
display: block;
}
```

Fallback notification

Older browsers will not provide access to mic features, hence this popup notification can also be used as a graceful fallback feature.

EXPLORE THE TECH INSIDE

www.gadgetdaily.xyz

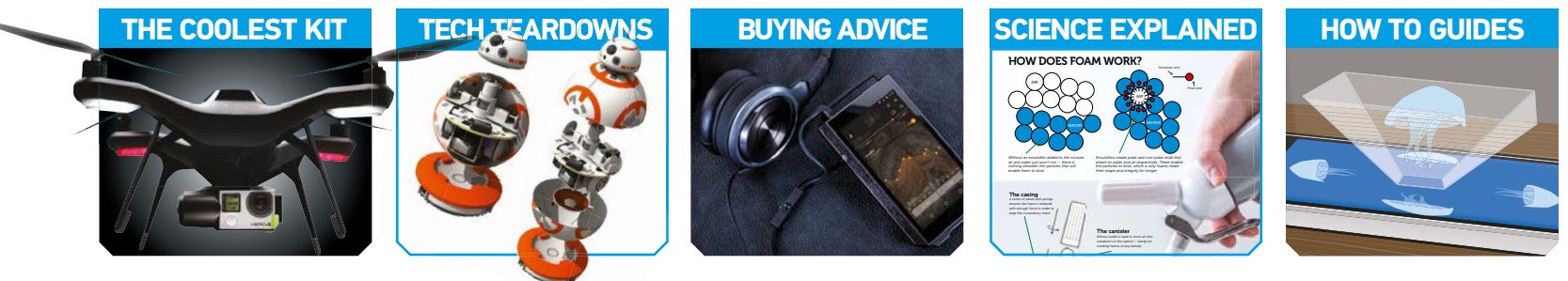


Gadget™

Available
from all good
newsagents and
supermarkets

ON SALE NOW

■ **ULTIMATE CAR TECH** ■ **NINTENDO SWITCH HANDS-ON VERDICT** ■ **BEST WEARABLE GEAR**



BUY YOUR ISSUE TODAY

Print edition available at www.imagineshop.co.uk

Digital edition available at www.greatdigitalmags.com

Available on the following platforms



facebook.com/gadgetmagdaily twitter.com/gadget_magazine

AUTO SUGGESTION

project

Auto Repair Materials
(ARM)

web

ARM-AUTO.DE

agency

Red Collar

web

redcollar.
digital

follow

@redcollars

**project
duration**
3

**no. of
people
involved**
6

**total man
hours**
220



ACRYLIC CLEARCOAT



PUTTY



ACRYLIC PRIMER



IF YOU'VE EVER THOUGHT THAT SITES FOR CAR PAINT AREN'T EXCITING THEN YOU WEREN'T ALONE. RUSSIAN AGENCY RED COLLAR WAS DETERMINED TO CHANGE PEOPLE'S OPINION

For any modern business operating in today's market, no matter what they sell, marketing their services or products online is essential. Savvy customers increasingly look to the internet as their first port of call when it comes to shopping around, surveying purchasing options and researching brands before they buy. Now, it's fair to say that companies know this and invest often eye-watering marketing budgets into establishing a digital presence that encompasses websites, social media outreach, targeted advertising campaigns and more. However it's also not unreasonable to say that some brands are more marketable in this space than others, especially when they inhabit fairly niche market sectors. This is when branding agencies and consultants really earn their crust, bringing enough expertise and imagination to deliver slick, engaging experiences for communicating an appropriate message. Which brings us to Germany's Auto Repair Materials and the story behind the build of its new website. A respected manufacturer of industrial paint since 1995, ARM specialises in car refinishing

products and has been supplying wider European, American and north African territories for the past five years. "Paints for cars is a pretty boring theme," shrugs the team who would take up the task of making it a significantly less boring theme. "And the client wanted us to build them just another paint manufacturer site, but that's not what we're about, so we proposed the creation of something interesting, that could show the paints in a different way." Quite a refreshingly candid admission from the folks at Red Collar, the digital agency ARM hired to add some gloss. Hailing from Voronezh, Russia, the team cites "creative vision, technologies and digital experience" as core principles behind the work it produces, so no surprise they had broader ambitions for what <http://arm-auto.de> could be. "We wanted to design not an ordinary paint manufacturer site, but instead to impress customers and make the site memorable using bright colours and interactive animated elements." So, you mean an experience not akin to watching (car) paint dry? Joking aside, it's time to find out how Red Collar put a stylish coat across ARM's eCommerce shoulders.

Personnel



DENIS LOMOV
Creative Director



SERGI MATVEEV
Designer



ANTON USTINOV
Front-end Developer



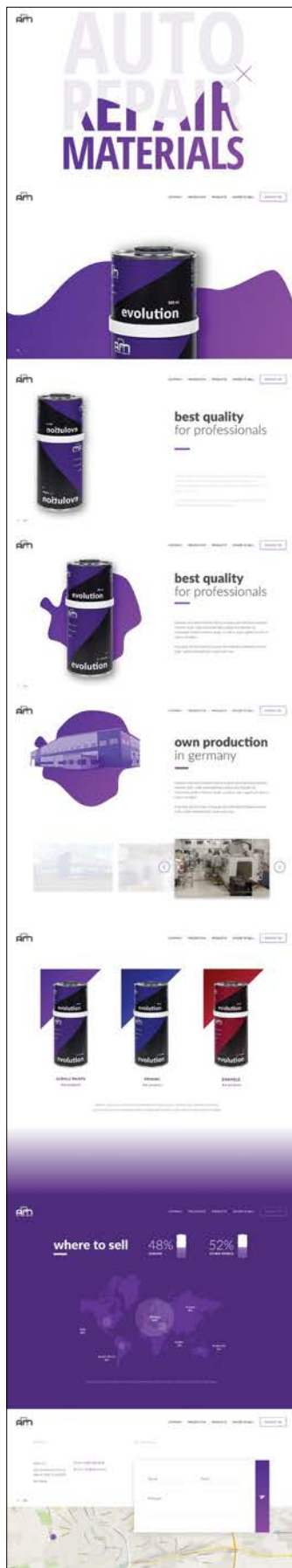
BOGDAN LOTAREV
Back-end Developer



IRINA BORZUNOVA
3D Modeller



LENA KUDAEVA
Photographer



Different strokes

"We started thinking about the main idea of the website just after the contract was signed," begins Creative Director Denis Lomov when asked about the very first perceptions Red Collar had at the outset. "The biggest question we had was how to make people love car paint and how it can be interesting and beautiful. We gathered our managers, designers and front-end developers to start brainstorming." Of course it had already been established that the team would not be satisfied keeping things simple and uninspiring. Red Collar was determined to throw out the old conventions and traditional approaches for marketing a specialist product like ARM's, with the client complicit in extending this beyond the confines of just a new website. "Initially we came up with some fairly difficult and at the same time quite funny ideas. For instance, we wanted to make a video of the factory, to show processing, and use real person photos with products and so on. But after a while we realised that the best way to emphasise a company's image is to show the product's face. We thought that the paint can itself is a product face, and asked the client to send us photos of the can so we can use it on the website not only as a part of a product catalogue. From doing this however, we also decided to rebrand the whole company and the great thing was that the client agreed!" So from choosing to use imagery of ARM's physical retail product on the site, Red Collar realised that the existing paint cans weren't attractive or distinctive enough and needed a revamp too. Decisions on the colour of the new can would then eventually inform the colour palette of the website, with the can itself eventually taking centre stage. "We wanted to make the website stand out from others, to show that this is the company that will make your car great. This all led to the decision to render the can and put it on the main page."

Talking concepts

When discussing the early stages, the team is quick to emphasise the importance of teamwork. The commitment to do something "non-typical and bold" would require that kind of cooperation, not just within the Red Collar team but between them and the client too. "Of course it is really important to always keep in touch with the client and to tell them how the process is going so they do not have any worries about it," continues Denis. "After the initial brief we asked the client to send us some products so we could make some photos of them and to see how we can use cans in our work. The next step was a presentation of a design concept. Typically, we use wireframes and black and white colours to present an idea to a client but in this case we changed our principle. We prepared a



colourful mock-up with real product photos and showed it to the client. It was very important to approve the main idea of the site so we could move on." This early talking was pivotal however, with the realisation that ARM was slightly freaked out by the bold concepts being presented. But as confident producers on the project, fired up by the direction things could go, the Red Collar guys were able to allay any fears by demonstrating stronger proofs of concept. "After that we moved on and finished enough design so that the next time we presented it, we also offered to render a 3D model of the product and proposed the future animations with 360-degree product rotation and dynamic waves moving by the mouse cursor. We discussed every detail of the project, including all those animations and interactions. The important thing here was that before showing the design to our client we

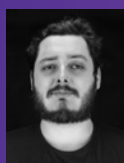




had briefed our developers so we knew every idea being presented to the client could be realised.”

Gloss then primer?

Considering the nature of this project, it's perhaps not totally surprising that the front-end work took priority and preceded any back-end development. The design team assigned to the visual stuff comprised three people including a photographer, 3D-modeller and indeed the front-end designer. “First of all our designer created sketches of the proposed website and we showed them to the client. After all the sketches were approved we started our work on the paint can package redesign because we understood that the old one did not suit at all. After this our photographer shot one hundred photos of the new product so the ▶



Site highlight

WE ASK FRONT-END DEVELOPER ANTON USTINOV FOR HIS THOUGHTS ON WHAT PART OF THE PROJECT THE TEAM CONSIDERS A HIGHLIGHT...

“The website itself is already a standout feature for us. It is absolutely unique from the websites that everyone is used to seeing when they think of those for similar products. However one of the main killer features of the website is the gradient wave triggered on mouse and page scroll. In the second place, and in my own opinion, is also the can twist animation the main page”

A different spin

One of the overriding factors associated with the early design work in this project was just how strong Red Collar's sense of direction was. It seems that initial market research acted as a powerful springboard for inspiring a vision for a website experience that didn't follow the crowd. “The idea came from the brainstorming,” insists Creative Director Denis Lomov. “First of all, we made a competitor analysis before a target audience and then task analysis. We went through dozens of sites looking for good solutions and interesting decisions. Unfortunately, most of

the existing car paint manufacturer sites were too similar to each other. All of them had small photos, bad typography and dull colours, which seemed to us to be too boring and not interesting at all. It was very important for us to create something absolutely different and we wanted to show to the world that even such an uninteresting thing could look great.”

Conceding that car paint was a challenging sell for imaginative online marketing, the team took it upon themselves to mix things up. So how would that fresh

thinking translate into a concept the market would embrace?

“As we said we wanted to stand out from the competitors and present the product. That was really important for us to show the product from an unusual side. The target audience mostly consists of car paint and coatings dealers, car repair shops and other industry professionals. They are not used to seeing websites in a way that we were going to show them. So we think that we managed to find a balance between a bold site while at the same time being understandable for the target audience.”



3D-modeller could make a render of it using the actual size package and can. As we mentioned, we don't show the design to the client until we discuss everything with our developers. This is because we want to be sure that everything we show to the client can be realised to avoid a situation when a small detail then needs lots of work." The rendered paint can was certainly one of those elements where prior consultation was necessary to be confident of delivery in the time available. Within the context of the broader front-end development, the can render as an asset was therefore made especially for it and in a way the designer wanted. "He had to write the script that rolls the paint can and changes the render frame-by-frame, while also going with page scroll and resize. Then for the button animations and paints triangles, the backgrounds were drawn across three stages." With the surface gloss applied, Red Collar's equally talented back-end developer could get to work on what was a fairly straightforward task in that regard. Typically in terms of technologies the team favours a Russian content management system called UMI.CMS for this sort of work. The platform has an English version also, while apparently offering greater protection from hacking over equivalent platforms such as WordPress. "As far as the back-end development is concerned we should say that there were not any problems. It is a really simple project for our team considering that there was only a small catalogue of products to serve to the site. One of the most important things we did do was in setting up the static cache for a faster downloading of the pages."

Time to shine

If the challenges associated with the back-end work were mercifully minimal, Red Collar admits that the prospect of launch was more anxious. For a three-month project commanding the attentions of just six people, the team had decided to reach for a much more ambitious outcome than might have been expected. Although this had been fuelled by unshakable confidence from day one, when asked about the complexities running up to launch, Denis reveals some last minute wobbles. "Well, of course, one of the main worries we had was that people would not understand and would not accept this project (at launch). It is so untypical for the industry and so we were risking failure because of our bold approach to

this project." Such hesitancy is understandable, particularly when the whole purpose lives or dies on having a positive effect on marketing ARM's business. If the target market was confused in any way, from individual customers to retailers alike, then the impact could be underwhelming. So how were these fears addressed and were they warranted in the end? "We made several tests and people were impressed," Denis breathes a sigh of relief. "We showed it to different dealers, sellers of the paint and coatings product, and people liked it. Then after the website was launched we've had lots of other companies come to us asking if we could make them exactly the same or something that would be really close to it, as bold as it is. Similarly, the client too has had lots of new clients come to them after the site was launched and released."

"After the website was launched we've had lots of other companies come to us asking if we could make them exactly the same or something that would be really close to it"

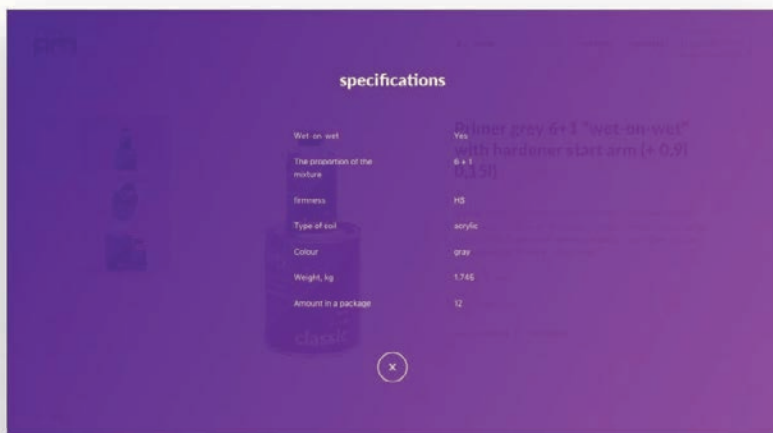
CONTACT US

CONTACT US

CONTACT US

Animations are everywhere. UI elements add finer details to the overall look and feel of the site

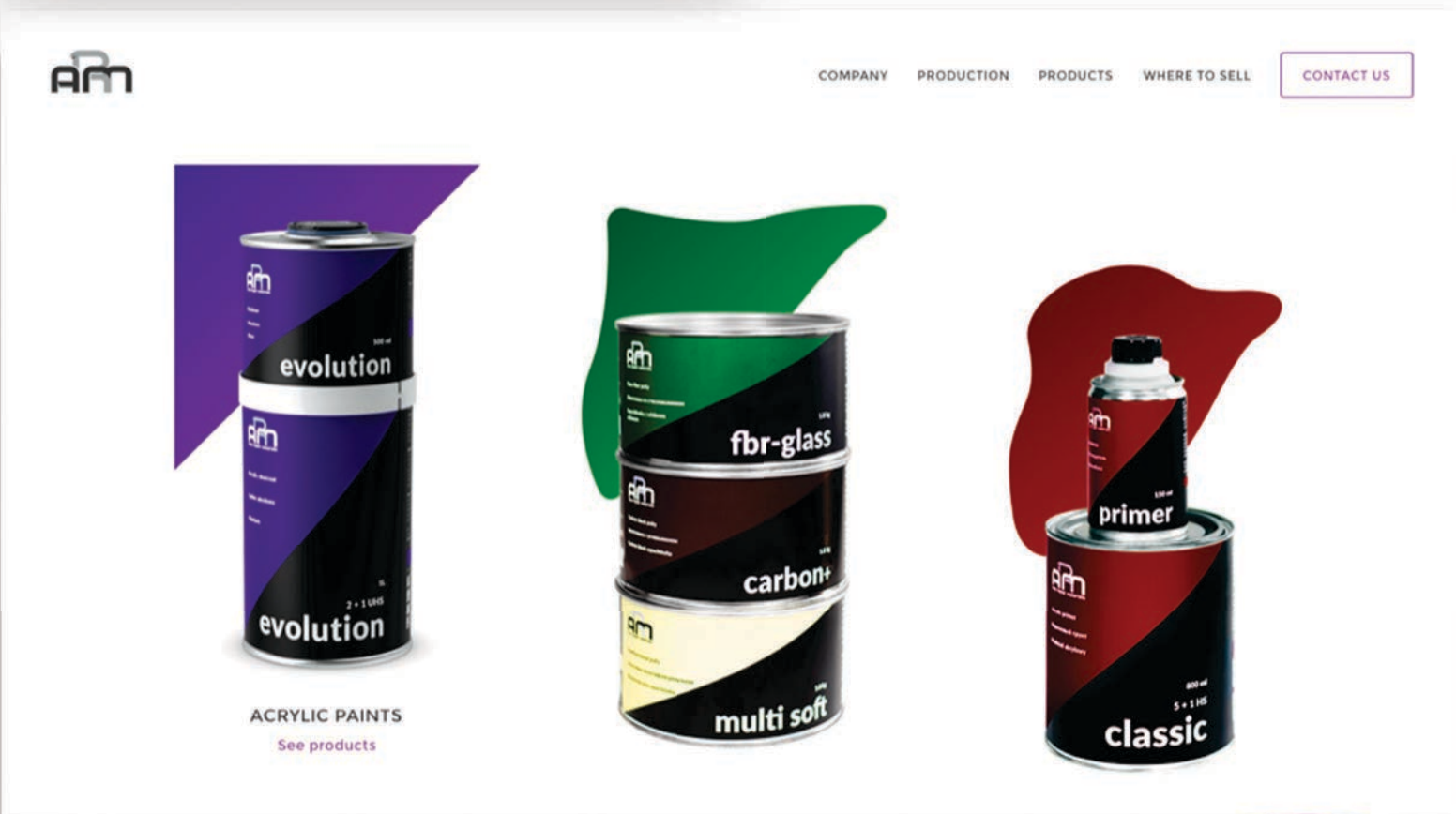
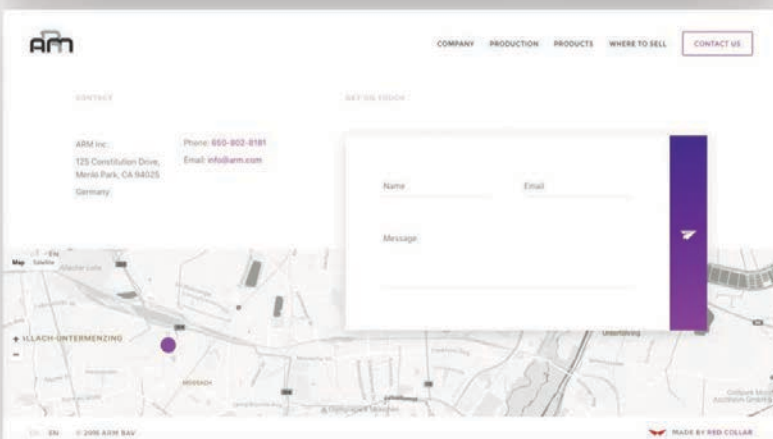




Adding extra sheen

"After all the work is done we sign with the client and act of acceptance," explains Denis when asked about the handover process. That passing over of a completed site is so often ignored as a vital phase, especially when the client must use and manage the content once the thing goes live. "After that we give them all the accesses, to the administrative panel via the creation of a separate admin account, and to different services we used in the website if needed such as YouTube, Yandex.Metrica and so on. We also send them all the documents, source files and backup copies." However the supply of all the required assets isn't the extent of things. Red Collar, like many digital

agencies, typically offers a broader service for monitoring a website post-launch and the prospect of tweaking things where necessary. "Normally after launch we offer to do the analysis, to see metrics and to understand what we can do better. We never give up on our projects once finished and always help our clients, so on this project we are still in that process and it is too early to make any conclusions. But for some big projects with catalogues and difficult administrative panels we always offer full website support which includes design, development and content changes if needed. It is hard to say right now if this website will need our help in future, but we always take care of our websites."



~~£5.99~~
£4.49
per issue



Subscribe and save 25%*

Every issue, delivered straight to your door



Never miss an issue

13 issues a year, and you'll be sure to get every single one



Delivered to your home

Free delivery of every issue, direct to your doorstep



Get the biggest savings

Get your favourite magazine for less by ordering direct

What our readers are saying about us...

"I've been a reader of Web Designer Magazine since the early days."
@sixrevisions via Twitter

"Love the latest magazine that I purchased today"
@navigation_web via Twitter

"My favourite magazine!!!"
@eduardomurillo via Twitter

Pick the subscription that's right for you

**MOST
FLEXIBLE**

Subscribe and save 25%

- ✓ Automatic renewal – never miss an issue
- ✓ Pay by Direct Debit

Recurring payment of £26.95 every six issues,
saving 25% on the retail price



Instruction to your Bank
or Building Society to pay
by Direct Debit

Originator's reference
5 0 1 8 8 4

Name of bank

Address of bank

Account Name

Postcode

Sort Code

Account no

Please pay Imagine Publishing Limited Direct Debits from the account detailed in this instruction subject to the safeguards assured by the Direct Debit guarantee. I understand that this instruction may remain with Imagine Publishing Limited and, if so, details will be passed on electronically to my Bank/Building Society. Banks & Building Societies may not accept Direct Debit instructions for some types of account

Signature

Date

**GREAT
VALUE**

One year subscription

- ✓ Great offers, available world-wide
- ✓ One payment, by card or cheque

A simple one-off payment ensures you never miss
an issue for one full year. That's 13 issues,
direct to your doorstep

☐ **UK** £62.30 (saving 20% on the retail price)

☐ **Europe** £70 ☐ **USA** £80 ☐ **Rest of the world** £80

Pay by card or cheque

Pay by Credit or Debit card

☐ **VISA** Visa ☐ **Mastercard** Mastercard ☐ **Amex** Amex

Card number

Expiry date

Pay by Cheque

I enclose a cheque for

£

Made payable to
Imagine Publishing Ltd

Signature

Date

Your information

Name

Telephone number

Mobile number

Email address

Address

Postcode

Please post this form to
Web Designer Subscriptions, 800 Guillat Avenue, Kent Science Park,
Sittingbourne, Kent ME9 8GU

Please tick if you do not wish to receive any
promotional material from Imagine Publishing Ltd
☐ By post ☐ By telephone ☐ By email

Please tick if you do not wish to receive any promotional material
from other companies ☐ By post ☐ By telephone
☐ Please tick if you DO wish to receive such information by email

Order securely online www.imaginesubs.co.uk/wed

Enter the promo code **PS17** to get these great offers



Speak to one of our friendly
customer service team

Call **0844 848 8413**

These offers will expire on
Saturday 31 May 2017

Please quote code **PS17**

Calls cost 7p per minute plus your telephone company's access charge

*This offer entitles new UK Direct Debit subscribers to pay only £26.95 every 6 issues. Offer code PS17 must be quoted to receive this special subscription price. New subscriptions will start from the next available issue. Details of the Direct Debit guarantee are available on request. Subscribers can cancel this subscription at any time. This offer expires 31 May 2017

salad

Hungry Design

Creativity and an understanding of the digital space is a core strength that underpins every piece of work Salad produces. Passion and a masterful command of today's creative tools ensures this agency is leading where others follow



who Salad Creative

what Branding, Design, Digital, Packaging, Marketing, Strategy, Development, Social Media, Brand Development, Brand Communications

where 19 The Quay, Enefco House, Poole, Dorset, BH15 1HJ

web saladcreative.com

key clients

Clipper Round the World Yacht Race

- AECC
- Greene King
- Silver & Green
- Tea India

Founded in 2001 by Ravensbourne fashion design graduate **Arabella Lewis-Smith** and entrepreneur and founder of popular sports brand **Animal**, **Ian Elliot**. The idea to launch

Salad was one of a handful of very different life choices that intrigued the pair.

Originally the shortlist included setting up a creative agency by the coast, progressing creative careers in their own fields in the city or moving abroad. Early in their respective careers, Arabella and Ian struggled to find businesses in Dorset that offered the same level of creativity they knew from London, thus spawned the idea of Salad.

With a potent cocktail of entrepreneurial and creative intuition, Bella and Ian founded the agency with a desire to deliver impactful and imaginative creative solutions and impeccable customer service in tandem.

Today, Salad is an increasingly integrated brand, design and digital agency with services spanning digital marketing, social strategy and campaign planning.

"I would love to tell a deep and meaningful story about how the name Salad came to us in an epiphany or a significant event in our lives," said Arabella Lewis-Smith, Founder and Managing Director. "The truth is, back in the beginning we probably didn't have any perspective of how big our business could grow to become."

"Of course, we wanted to be successful and grow but there's always an element of humbleness at the start. So, we tried to come up with names that would work for bands and up popped Salad which made us laugh. It's catchy and it gets your attention because it's unusual and that's pretty much why we landed on it."

"The name of an agency is definitely significant as I think it needs to represent your vision, your team, your culture (where possible) but it also needs to entice people to approach you. Salad works well for us because in-part we have a very food-centric culture. We have a real passion for working with food brands and it shows that we don't take ourselves too seriously either. It's fun, it's friendly and it begins to tell a story about who we are."





Industry Insight

Arabella Lewis-Smith
Founder and Managing Director

"My vision for the future of Salad is one where we work with as many like-minded clients as possible. The clients who get the most from us are those who trust in our abilities and give us the freedom to follow our creative intuition where it counts."



Arabella continued: "As for the URL, while we've always been called 'Salad' the creative element was reflective of our initial branding and design offering. As time has gone on, this has remained relevant and continues to apply to our way of thinking and approaching work. It also ensures we don't get confused visitors looking for a caterer! All our other touchpoints name us as Salad to avoid confusion. The name came first, though I expect that as more agencies come to fruition, entrepreneurs will be increasingly forced to start with a URL search."

The websites of agencies are often the first point of contact with a new client. The approach an agency can take with their own site is varied. Arabella explained Salad's approach: "It's absolutely crucial. As is the case for our clients, our website is typically the first or second touch point that our

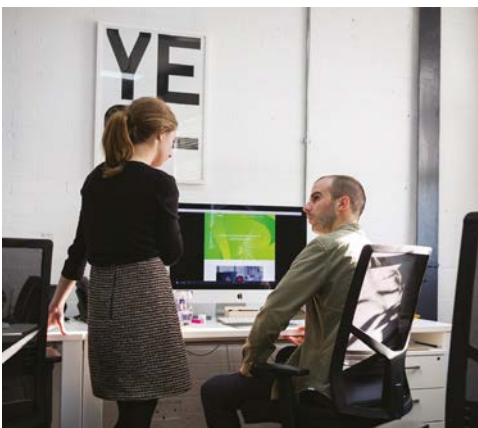
customers or future employees will interact with. Just like a face-to-face meeting or the exchange of business cards, our website is the first opportunity we have to communicate, connect and make an impact. Attention spans are getting shorter when it comes to our interaction with digital and content. A

website must make the right impression and quickly. The second layer to this is that with web design and development on our menu, our website acts as an opportunity for clients to experience our

"The dream is of course to rely on our reputation alone to draw aspirational work through the door"

work first-hand.

"Too frequently, business-related tasks fall to the bottom of the pile as client work comes in the door and we've had to work hard over the years to ensure that we shake off this habit and use our time intelligently. We launched our latest website in the summer of 2015 to reflect the more integrated nature of the way we do business and there will be



Hall & Woodhouse, Path to Prizes app

hall-woodhouse.co.uk

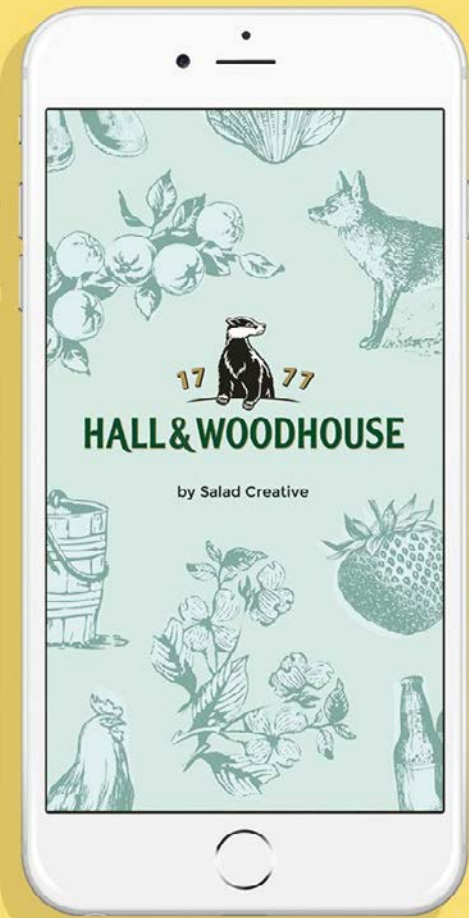
Although hugely successful over its four-year run, Hall & Woodhouse was ready to move on from its annual printed Pub Trail campaign. It returned to us to develop a new concept in line with its objective to increase pub footfall. We combined the practical functionality of a pub directory with a game-inspired loyalty scheme to create a single app for smartphone users.

By checking into multiple pubs, the user can progress around a fictional map and gamble their rewards to win even better prizes. As part of a campaign including printed materials

and a responsive website, the app encouraged Hall & Woodhouse customers to explore the world beyond their local pub and appreciate the variety of locations on offer.

"Salad have been fantastic to work with on this project. They helped me plan and present the concept to our internal stakeholders, before managing the entire app creation from design to implementation, all of the time working with us to ensure everything met our needs. Their work has been nothing short of faultless."

Lesley McIlroy, Marketing Manager, Hall & Woodhouse.



further creative changes in the coming year. It's an evolving exercise of decluttering and featuring more impressive, attention-grabbing visuals with a stronger focus on the effectiveness of our work. We let it speak for itself."

Salad has an impressive array of clients that have benefited from the expertise that has been bought to their projects. But how does Salad attract new clients? Fleurie Forbes-Martin, Marketing and Communications Manager explained: "The dream is of course to rely on our reputation alone to draw aspirational work through the door but that's our long-term goal. We have a steady flow of enquiries that come into us, largely thanks to the work people have seen us create. We have an appetite to continue to work with local as well as national and international clients. The team would relish more opportunities to work with household names and that's a focus of mine. Having recently joined Salad, I'm a much-overdue dedicated marketing and new business capacity for the business."

Fleurie also outlined the approach Salad is now taking to expand its client base: "In my experience, most agencies of our size try to balance these

responsibilities around their day-to-day work. But as we continue to grow, that's not something we can afford to consider 'secondary' and we want to make the most of the exciting opportunities that come our way. It's my responsibility to be proactive when it comes to all things 'brand Salad' - PR, marketing, awards et al.

"I also invest time throughout the year to network in different circles (for which I have a penchant anyway). Directors of industry are notoriously busy people but Salad is ultimately a people business and so safeguarding time to build and nurture relationships is critical to me. I wouldn't want it any other way and I think Arabella's decision to lead the agency in this people-oriented way resonates with other business owners. We make the effort where some do not.

"Naturally we enjoy pitches because they're an opportunity for us to collaborate and think big (when we have a detailed brief). But they are always a significant investment so we have to pick and choose carefully. On the flipside, we're always focused on improving the service we deliver our existing clients over the duration of our relationships.



Stag & Hen thestagandhenexperience.com

The Stag and Hen Experience is a nationally recognised party organiser. Offering over 80 destinations worldwide, with hundreds of activity and accommodation variations, the business is operated solely online. This makes the effectiveness of its website vitally important.

The Stag and Hen Experience had suffered a dramatic drop in enquiries following a change to Google's search algorithm, which excludes non-responsive websites from certain mobile searches.

They were also experiencing a static conversion rate as the website was old and not especially intuitive. The brief was therefore to develop a fully responsive and intuitive website and to serve two main objectives: to improve visitor levels and conversion rates from visit to enquiry.

The new website provides a solid and stylish platform for the business's varied offerings. Striking design and considered, intuitive navigation is complemented by carefully positioned calls to action. The use of video on the homepage immediately engages with the visitor, and the fully customisable content throughout the site allows Stag and Hen to promote its choice of experience quickly easily from the CMS.

Since launch there has been a transformational improvement of its key metrics - visitor levels are up by 35 per cent and conversion rates have improved by 20 per cent, which has resulted in over 150 per cent improvement in sales revenues.



We understand that there's more to be said of focusing your time on making clients happy than chasing 'shiny new ones'. The grass isn't always greener after all."

Agencies also need to think about how the clients they work with can influence their own businesses. Arabella explained how Salad approach the 'business' of being a creative agency: "It's a balance but the short answer is yes, we do choose who we work with very carefully. My vision for the future of Salad is one where we work with as many like-minded clients as possible. The clients who get the most from us are those who trust in our abilities and give us the freedom to follow our creative intuition where it counts.

"The reality is that we have a variety of different clients with unique needs who buy into our team in very different ways. Sometimes we are the lead creative partner or lead digital partner but we still have clients who use us on a project-by-project basis. We're extremely proud to have such a variety

"When it comes to the typical projects we handle, our average turnaround time is probably 12 weeks"

of sectors in our portfolio, giving us the ability to work with so many different people.

"We've made the decision to be industry general, rather than hiding the fact we're not niched. However, we are growing as a business and through that growth we are adopting new talent and experience that will continue to influence the kind of work we can and want to win. We are continuing to build a team that thrives on larger, more integrated design, digital and marketing projects. Our preference will always be long-lasting, trusting relationships with clients who utilise us as a studio of 'partners', rather than short-term suppliers."

Striving for an individual voice in the digital marketplace has always been a central component of Salad's driving force. Is there a project that encapsulates the Salad ethos? "Towards the end of

TIMELINE

<div>Year 2001</div> <div>No. of employees 3</div> <div>Founded with a focus on delivering an open, honest and down-to-earth experience.</div>	<div>Year 2002</div> <div>No. of employees 3</div> <div>Win Bar Med and become the lead agency partner for the first time.</div>	<div>Year 2006</div> <div>No. of employees 5</div> <div>Launch first professional agency website (with animations!).</div>	<div>Year 2008</div> <div>No. of employees 6</div> <div>Ups and moves shop to an amazing creative space with an up-close view of the sea.</div>	<div>Year 2010</div> <div>No. of employees 8</div> <div>Merge with agency Skigo. Adam Morland joins to head up the new digital dept.</div>
--	--	--	---	--



2015 we invested some time in understanding what makes us unique and makes us tick," said Arabella. "We defined our values as ambition, craftsmanship, effectiveness and cutting edge. This aim to communicate that we are ambitious for clients and want them to succeed. The quality of our design underpins everything we say and do and we exist to create the advantage for clients with work that will always be smart, insightful and informed.

"We now use these four bold business traits as a means of connecting with clients that value these characteristics. Our work with Tea India probably best demonstrates our values in practise very nicely. The Tea India team (part of Keith Spicer) came to us to help them solve a design challenge.

"The popularity of chai had soared in recent years and Tea India knew there was potential to grow their market share. Having developed two new flavours to add to the range, they needed to develop their brand, revisit their packaging designs and better equip the brand to compete in an increasingly competitive market. The brief was to create a more evocative aesthetic and inspired by an eclectic mix of Indian sign and language writing styles, we created a collection featuring textured fonts, hand drawn graphics and a vibrant colour palette. As a result of the new look, Tesco and Sainsbury's stocked the newest product in over 1,200 stores and Waitrose also agreed to stock two products in over 290 stores. Design is such a subjective part of what we do but with an increasing focus on making the beautiful effective, it's up to us to invest intelligent planning into the process of our work to guarantee the best possible output for our clients that will achieve their objectives."

With such a diverse range of clients, how each new project is approached means understanding the client's needs, but also how this will be delivered. "When it comes to the typical web design projects we handle, our average turnaround time is probably 12 weeks, give or take," said Sophia Tagliavore, Account and Studio Manager.

"Having said that, one of our biggest projects last year took 1,700 hours over 13 months to complete in three delivered stages.

"I'm responsible for ensuring our projects finish on-time and on-budget. A large-scale project will be assigned an Account Director, a dedicated designer



Industry Insight

Fleurie Forbes-Martin
Marketing and Communications Manager

"We have a steady flow of enquiries that come into us, largely thanks to the work people have seen us create before. We have an appetite to continue to work with local as well as national and international clients. I know the team would relish more opportunities to work with household names and that's a focus of mine."



Year 2014
No. of employees
10

Bella becomes majority owner of Salad and lays out ambitious plans.



Year 2014
No. of employees
10

Enter their first ever awards and win RAR awards for Best Website.

Year 2016
No. of employees
15

Increases expertise with hire of talented Strategy Director, Daniel Ward-Murphy.

Year 2016
No. of employees
17

Open doors to its beautiful new loft office expansion and make space for new hires...

Think Logic

thinklogic.co.uk

ThinkLogic is a specialist Lenovo reseller with a strong emphasis on quality of service and engaging customer support.

ThinkLogic quickly built a good relationship with Lenovo and achieved Gold Reseller status. It needed to push growth further, by winning larger clients and drive an increase in repeat business from existing customers.

Starting with an evolution of the brand identity, Salad proceeded to redesign the front-end of the website, including look and feel, key messaging, and responsive design.

A restructured navigation includes a new mega-nav featuring large, eye-catching images to aid the UX and ensure customers can easily find what they need. The site has been expanded with new sections including case studies and third-party reviews, to emphasise credibility and expertise.

The Magento platform has been upgraded. Salad integrated a smart and

automated CNET product delivery system, which ensures greater efficiency in the management of the catalogue.

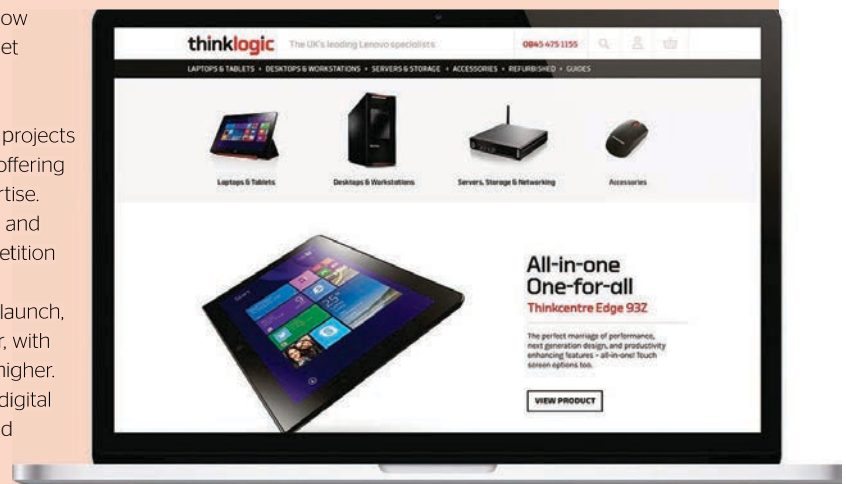
We also added the ability to attach files and videos to products in the CMS, giving ThinkLogic the edge when it comes to offering a greater wealth of information about their products. Finally, the site now works seamlessly across all tablet and mobile devices, with truly responsive functionality.

The new ThinkLogic website projects the image of a serious partner offering added value and genuine expertise. Larger customers are engaged, and feedback tells us that the competition does not like it.

In the three months since re-launch, bounce rate is 16 per cent lower, with pages per session 28 per cent higher. Salad continue to implement a digital strategy to drive more traffic and optimise conversion rates.



Mobile commerce is now a focus for all retailers. Salad ensured that the new site for ThinkLogic was engaging and easy to use on mobile devices



Agency Breakdown

- 1.....Founder/Managing Director
- 1.....Digital Director
- 1.....Strategy Director
- 1.....Finance Director
- 1.....Marketing & Comms Manager
- 1.....Digital Marketing Manager
- 1.....Design Manager
- 3.....Creatives
- 2.....Developers
- 3.....Account Managers
- 1.....Producer
- 1.....Office Manager



(or two), a dedicated developer and the consultation of our Strategy Director and Digital Marketing Manager.

"Generally speaking, every client benefits from the expertise of the entire development team as they frequently share insights to tackle challenging problems together. All work is also overseen by our Digital Director to ensure it's of the highest quality, committed to our values of craftsmanship and cutting edge.

"Finally, with a talented digital marketing manager in our ranks, all web builds receive her input from the very beginning. This is vital to building and protecting each site's reputation with Google. She consults during the wireframing stage right through to the formation of calls-to-action, content usage and headings etc.

"Each of our projects are divided into key phases. Planning, design, client review (there are a few of these throughout), build, test, sign-off, go-live.

"Needless to say, we have evolved with the times to consider the best way to measure the success of our work. The proposition we offer our clients is 'beautifully effective' and it's therefore critical that we demonstrate the impact of our efforts, beyond looking great."

Adam Morland, Digital Director also outlined the extensive toolset that Salad uses: "Our design team predominantly use Adobe Creative Suite although

increasingly we're using Sketch for digital design. When it comes to development we all use Sublime as our main text editor. We're also big fans of CodeKit which we use for compiling, live multi-device browser refreshing and optimising assets.

"Our preferred platform for development is Laravel upon which we have built our own CMS. Laravel has some killer features out of the box that we love - Blade, Eloquent, RESTful Routing, and database migrations and seeding that actually works! For eCommerce we typically use Magento CE. It's got all the features a shop owner would expect from a platform and has a fantastic marketplace of extensions.

"With the now essential requirement for everything to be responsive, HTML5/CSS3 have an important role in building websites which can cater for all viewport sizes. With HTML5/CSS3 now being 'the norm' our dependence on JavaScript, and in particular jQuery, lessens.

"Simple animations and transitions can be done without the need for JavaScript. Browsers are in a much better place than they were a few years ago so in truth, post IE8, we don't really need JavaScript to mop up after us. It'll be exciting to see how these technologies will evolve over the coming years and as ever we hope that web browsers can keep up allowing us to use these new features in the real world sooner rather than later!"

Adam also explained how new technologies are impacting how Salad works: "We're always excited to see what Taylor Otwell (creator of PHP framework, Laravel) will come up with next! We've recently used Lumen on a project, and we also use Forge to manage deployments. Spark looks like a real time-saver and we're just waiting for the right project to come in that could use it. We recently found Moltin which is an eCommerce API that looks likely to solve the headache of cross-platform eCommerce. Web-based IDEs are getting increasingly more impressive and feature rich. Services such as Codenvy and Codeanywhere seem to be leading the way. At the other end, Codeshare is just great for those quick brainstorming sessions with other developers who aren't in the room."

An agency is effectively only as good as the people it employs. What qualities do you look for in a prospective employee and what advice would you give to anyone looking to take a step into the industry? "Attention to detail, passion for what they do, knowledge and confidence in their field and of

"Browsers are in a much better place than they were a few years ago"

course they have to have that special 'Salad' quality," said Fleurie.

"I would go as far as to say that Salad is a culture as much as a business. Our team our incredibly invested in our success and working together to achieve this. In terms of advice for getting going in our industry, I would say stay curious and throw yourself into situations where you can meet interesting people. It's almost impossible to quantify but I cannot even begin to understand how meeting one person introducing me to another has opened all kinds of doors. Be generous with your time."

And what does the future hold for Salad? Arabella concluded: "The future is extremely exciting. The last few years have seen a new 'era' for Salad and the direction we're taking the business. We are ambitious to expand, though we are mindful that this can only happen at a sustainable rate. This year has seen us expand our working space, which is now probably one of the most unusual and beautiful agency office spaces in Dorset."

Also, with a number of hires last year, we will be honing in on what our clients value most about us. Then fulfilling our invigorated marketing strategy to ensure we open ourselves up to bigger and better opportunities. Strategy, content, social and branding are areas that will continue to grow and flourish."

salad

saladcreative.com




Founders
Arabella Lewis-Smith

Year founded
2001

Current employees
17

Location
Bournemouth + London

Services
Strategic Branding
Visual Branding
Web Development & Innovation
Campaigns
Digital Marketing



CODE SMALLER, SMARTER HTML

Web Components have matured and are gaining serious momentum. We take a look at how they work, where they can be used, and show how to build our very own

At their heart, Web Components are a way of creating modular, custom components for use in web applications, in a manner that blends seamlessly with 'vanilla' HTML. Imagine importing a library of pre-made UI components onto your page, then using their features like any other HTML tag. This would enable us to quickly build complex functionality from a large set of prefabricated HTML elements.

In C we can use `#include`, then invoke third-party functions as if they were our own. In Java, we can import libraries and use their pre-built classes and methods. But traditionally the web has not had such a modular approach, relying on workarounds and frameworks to enable component-based development. Yet as web applications become more complicated and prolific, the need for such a model grows.

Libraries providing reusable widgets for the web have existed for some time, but they have gained nowhere near the popularity of, say, jQuery or Bootstrap. Why? Because without Web Components, the very nature of HTML, CSS and JavaScript does not lend itself to developing in such a modular way.

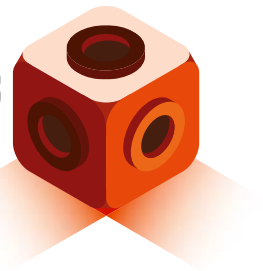
Web Components introduce a set of features to the HTML5 specification which allow us to do just this. The new features enable us to define our own fully functional HTML elements, and encapsulate the supporting code for simple delivery onto a page and efficient reuse.

The Web Components APIs are in the process of being introduced to all major browsers, and there are already a number of high-traffic sites using them in production. Google and Mozilla in particular are championing Web Components as a major leap forward. Libraries are also emerging to make the process of working with the APIs as simple as possible. Now, more than ever, is an excellent time to begin to get familiar with how to create your own components.



REDUCING THE AMOUNT OF CODE WRITTEN AND GROUPING FUNCTIONALITY TOGETHER ULTIMATELY REDUCES THE AMOUNT OF CODE THAT NEEDS TO BE MODIFIED AND TESTED WHEN ANY GIVEN FUNCTIONALITY NEEDS CHANGING LATER

Simon Jones
Software engineering director



WHY BUILD WITH COMPONENTS?

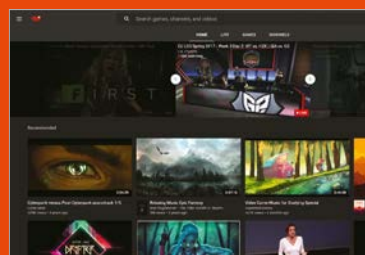
Web Components are one of several attempts to bring component-driven development to the web. Component-based code design has been a mainstay of other realms of software development for many years, and underpins leading design principles such as service-oriented architecture (SOA). When we talk about separation of concerns, loose coupling and encapsulation, we're ultimately talking about components.

So what are these benefits? Applications built using these principles can sometimes have less code. This is enabled by inheritance and reuse. Code is written in a generic way and used multiple times by components that inherit its functionality and behaviour. They're more maintainable because individual modules or components should

encapsulate related functions, meaning the code structure is much easier to navigate and understand. Reducing the amount of code written and grouping functionality together ultimately reduces the amount of code that needs to be modified and tested when any given functionality needs changing later. Another advantage is the ability to apply proper scoping so that fewer conflicts emerge when multiple external libraries are used.

These are all recognisable as benefits that would be more than welcome in web dev, as they are hard to achieve with basic HTML, CSS and JS. What we too often find is that as you scale up a page the code starts to get harder and harder to keep track of, and components can help us manage this.

WEB COMPONENTS IN ACTION



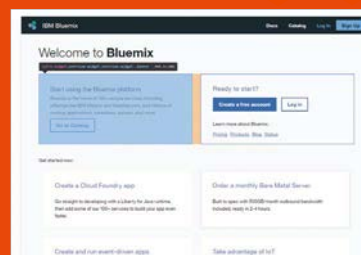
YouTube Gaming gaming.youtube.com

Google uses Polymer, with almost every visible UI element encapsulated in a custom component.



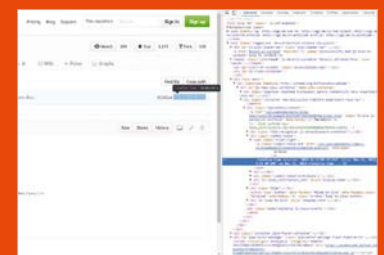
NET-A-PORTER net-a-porter.com

NET-A-PORTER employs several reusable components for common parts of its pages, such as item prices.



IBM Bluemix console.ng.bluemix.net

Many sections on the homepage of IBM's cloud platform offering are built as reusable widgets.



GitHub bit.ly/2lkMjPp

Of course GitHub has a component tucked away! The `<relative-time>` is used to show the timing of commits.

THE KEY CONCEPTS OF WEB COMPONENTS

Four main W3C standards come together to give us the ability to create and manage reusable components

CUSTOM ELEMENTS

Custom elements enable modular code to be embedded in an app in a clean, manageable way

HTML is both powerful and human-readable, but its native syntax is limited. Custom elements break through this limitation by enabling the creation of completely new HTML tags, or modification of existing ones.

With the Custom Elements specification, elements which perform reusable functions in multiple places can be named appropriately. Take the native HTML5 `<video>` tag, for example, which neatly encapsulates a complex functionality and can be reused anywhere. Similarly straightforward tags can now be created for any purpose that is needed – perhaps anything from a simple `<spinning-wheel>` element to display a loading animation, to a reusable fully featured `<media-player>` widget. This alleviates some of the difficulty of complicated pages which often require large, incomprehensible sets of nested `<div>` tags.

Use of custom HTML syntax was actually possible before the introduction of the Custom Elements specification. It was straightforward to use a non-standard tag, style it with CSS and attach JavaScript to it. However, these unrecognised tags inherit the `HTMLUnknownElement` interface, which brings significant limitations. As a result, they were rarely put to use in a real-world setting.

Custom elements can be defined as follows:

```
customElements.define("my-element",
  MyElement);
```

This enables a `<my-element>` tag to be used anywhere within the document. The naming of custom elements always requires a hyphen, to differentiate them from HTML's native tags.

But how do we add functionality to it? This is done by defining the `MyElement` class which extends `HTMLElement` and describes the content, styling and functionality. For this to become really useful, we need to understand other parts of the Web Components specification – most notably, Shadow DOM and HTML Templates.

It's worth noting that the current Custom Elements specification, v1, was preceded by v0 with some fundamental differences. Many articles online still reference the v0 implementation. Keep a lookout for this, and ensure you're always working with v1.

SHADOW DOM

Ideally the implementation of a custom element should be completely hidden from, and inaccessible to, the remainder of the page. This ensures readability, since the HTML, CSS and JavaScript associated with the element are kept independent of the rest of the page. It also avoids conflicts, since there is no risk that one piece of code will accidentally affect an identically named element, class or variable elsewhere.

This is particularly important as we begin to combine custom elements from different sources, with no way of knowing whether their naming conventions will conflict. Taking the `<video>` tag once more, the implementation is invisible and none of its constituent objects such as buttons or sliders appear in the main page's DOM tree. This means that we do not have to worry about naming conflicts with any of our own code, nor that our own CSS may affect the styling of parts of the component.

This concept of 'encapsulation' is a core feature of many common programming languages, but is not native to web development. HTML, CSS and JavaScript are normally global in scope, meaning that when combined together on the page, there is nothing stopping any piece of code interacting with any other piece of code, even when residing in

Safely encapsulating Web Components would be made difficult by the standard global scoping of HTML, CSS and JS. Shadow DOM has the answer

separate libraries. Over the years a number of solutions have emerged to work around this, but Shadow DOM offers a more comprehensive and flexible approach than any of its precursors.

Shadow DOM solves this problem by enabling us to explicitly create a separate DOM tree which does not form part of the main document's DOM, yet still renders on the page. CSS rules and JavaScript functions in the main document do not interact with elements of the shadow DOM.

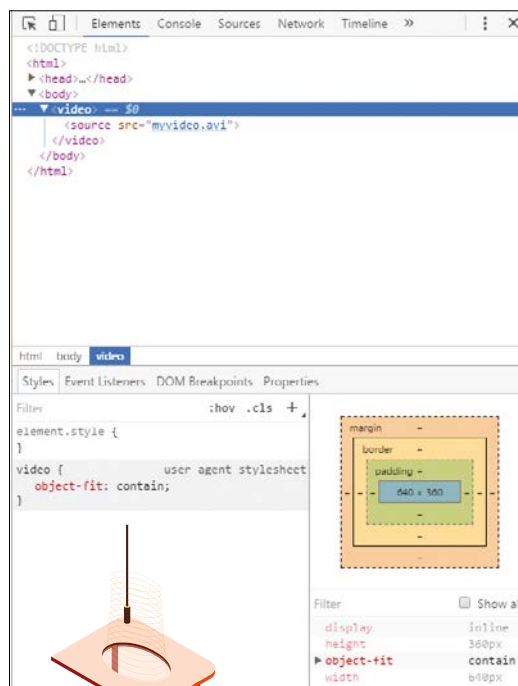
Conversely, if we embed CSS within the shadow DOM, it is scoped within it and does not interact with the main document.

To create a shadow DOM, we create a 'shadow root' and attach it to a 'host' element in the main DOM:

```
var shadowRoot = element.
  attachShadow({mode: "open"});
```

We now have the ability to add HTML content to this shadow root. You'll want to create the root in 'open' mode – if we used 'closed', this would prevent external JS interacting with the DOM.

Not all types of element can host a shadow DOM. The W3C Shadow DOM spec defines which can and can't.



Left

The `<video>` tag, along with many other HTML5 elements, uses shadow DOM to encapsulate its functionality. When inspecting the code, we see none of its constituent UI elements

Top left

This is because by default Chrome won't show the shadow DOM at all. Enable it in DevTools Settings by checking 'Show user agent shadow DOM'

Top right

Now if we look again at the `<video>` tag, we can see that it hosts a shadow root, with a myriad of other elements such as media controls contained within the shadow tree

HTML TEMPLATES

Templates provide the ability to create invisible, inactive portions of HTML which can be used at a later stage



At its heart, the concept of templating is to allow something to be recreated again and again with minimal effort and guaranteed accuracy. The introduction of a `<template>` tag to HTML enables us to do just this, by defining inert chunks of code which can be used at a later stage.

Creating a template is as simple as wrapping the tags around some HTML, with a named identifier for each:

```
<template id="my-template">
  
</template>
```

Within the `<template>` tags, we can include any HTML allowed within a `<head>` or `<body>` tag. When the browser parses this code, it will not render the image or any other content within the tags. It will also not execute any script. Any DOM queries will not return elements of the template.

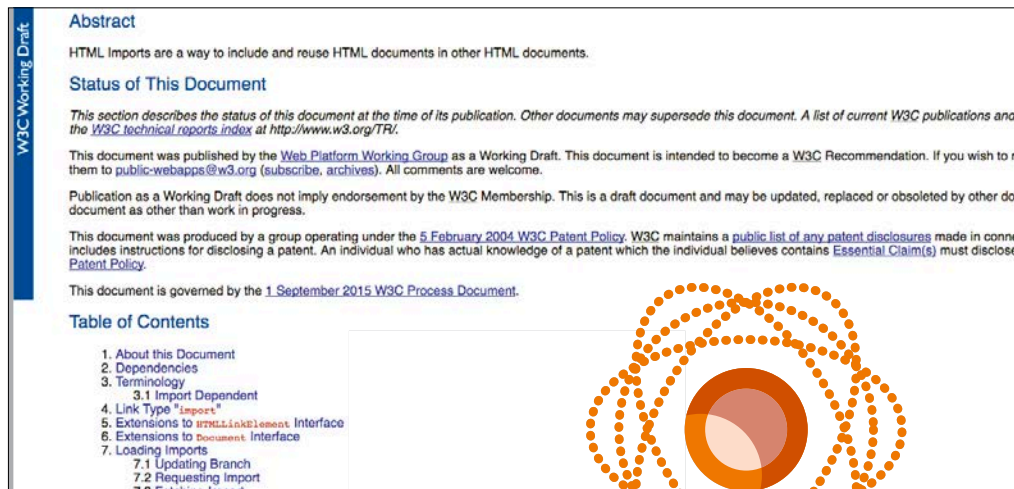
To use a template, we create a copy of it and then append it to the main document's DOM. For example:

```
var template = document.querySelector("#my-
template");
var copy = document.importNode(template,
content, true);
document.body.appendChild(copy);
```

If we wished, we could also make changes to the template before creating the copy, by running `querySelector` on its `content` property. This allows customisation of the template each time it is used.

Templating becomes particularly useful when combined with imports. We may not want resources to be fetched and scripts to be run as soon as a component is imported. Therefore, defining the component's HTML as a template enables the component to be imported into a page without being activated before it is needed.

“CREATING A TEMPLATE IS AS SIMPLE AS WRAPPING THE TAGS AROUND SOME HTML, WITH A NAMED IDENTIFIER”



HTML IMPORTS

The lack of a straightforward way to load HTML content from one page into another has long been conspicuous. Enter HTML Imports as a simple yet powerful solution to the problem

Importing HTML is probably the simplest of the Web Components specifications to understand, but has some nuances that can be difficult to get to grips with. Importing CSS or JavaScript from their own external files has always been straightforward with `<script>` and `<link>` tags, but there is no direct equivalent to load HTML itself. Until recently, we were limited either by the constraints of `<iframe>` or the need to use JavaScript.

The syntax for an HTML import is extremely straightforward:

```
<link rel="import" href="my-component.
html">
```

This will load the content of **my-component.html**, along with all its dependencies, and execute any JavaScript. It will not immediately render HTML in the imported file, but instead makes its DOM accessible to manipulate from the main page's JavaScript.

There is a little trickery required to access the new DOM:

```
var link = document.querySelector('link[rel=
"import"]');
var content = link.import.
querySelector('#my-component');
```

This provides us with a content variable with the imported content of **my-component.html** ready for use. If we wanted to add it to the main document, we could then use:

```
document.body.appendChild(document.
importNode(content, true));
```

It's worth noting that as a security measure, HTML

Imports enforce a same-origin policy. To import an HTML document from a different domain to the one your page is hosted on, you need to enable Cross-Origin Resource Sharing (CORS). Significantly, however, this also causes some browsers such as Chrome to decline imports from the local file system, necessitating the use of a server to test out the new standard.

However, once imports are working, we can go even further. An import will include all CSS or JavaScript contained or referenced within the HTML file being imported. This is where it becomes really powerful, since it provides the ability to deliver an entire component via a single import statement.

Consider a library which has CSS and JavaScript files. In the past, this would most likely be imported onto a page like this:

```
<link rel="stylesheet" href="my-component-1.
css">
<link rel="stylesheet" href="my-component-2.
css">
<script src="my-component.js"></script>
```

But importing a library shouldn't be this difficult, nor should it require us to understand the function of each file comprising that library. With HTML Imports, the library can provide a simple HTML wrapper, placing all three of these lines into **my-component.html**. We can then import this HTML file with a single statement.

Currently HTML Imports are the least widely supported part of the Web Components specifications. More on this later. In the meantime, we can test whether the browser supports them: `if ("import" in document.createElement("link")) {...}`

CREATE YOUR FIRST COMPONENT

Get hands-on with Web Components by creating your own custom element

01. Before we start

Let's start by giving some thought to what we're trying to achieve. We're looking for a component which stands alone, is available for use through a single import statement, and is usable as a custom tag. We want to cleanly encapsulate its functionality, so it should be loaded into the shadow DOM. We should follow the usual good development practices, so we'll use templates to keep markup and styles separate from scripting. The syntax of our component will be as follows:

```
<slideshow-component src=[...]>
</slideshow-component>
```

...where `src=[...]` holds a list of image files to be included in the slideshow.

02. The component template

The first thing we'll create is a simple HTML template for the component's appearance. In `slideshow.html`, we add:

```
<template id="slideshow-template">
<link rel="stylesheet" href="slideshow.css">
<div id="slideshow-div">
<p>Welcome to my slideshow!</p>
<img id="slideshow-image" src=""
width="800" height="auto">
</div>
</template>
```

The `img src` attribute is empty for now as we'll populate it by script. Also note the `id` attribute of the template, which we'll use later to import it.

03. The component styling

You'll notice we also include a CSS reference within the template. This becomes important later, as it ensures we can apply it here without affecting the document we import the component into. Let's create some simple styling in `slideshow.css`:

```
p {
```

```
color: rgb(169,169,169);
font: 48px verdana;
text-align: center;
margin: auto;
}
img {
display: block;
margin: auto;
}
```

04. Scripting a custom element

Let's introduce some scripting to define a custom element for our component. In `slideshow.html`, we add:

```
<script src="slideshow.js"></script>
```

By placing this in the component's HTML file, we

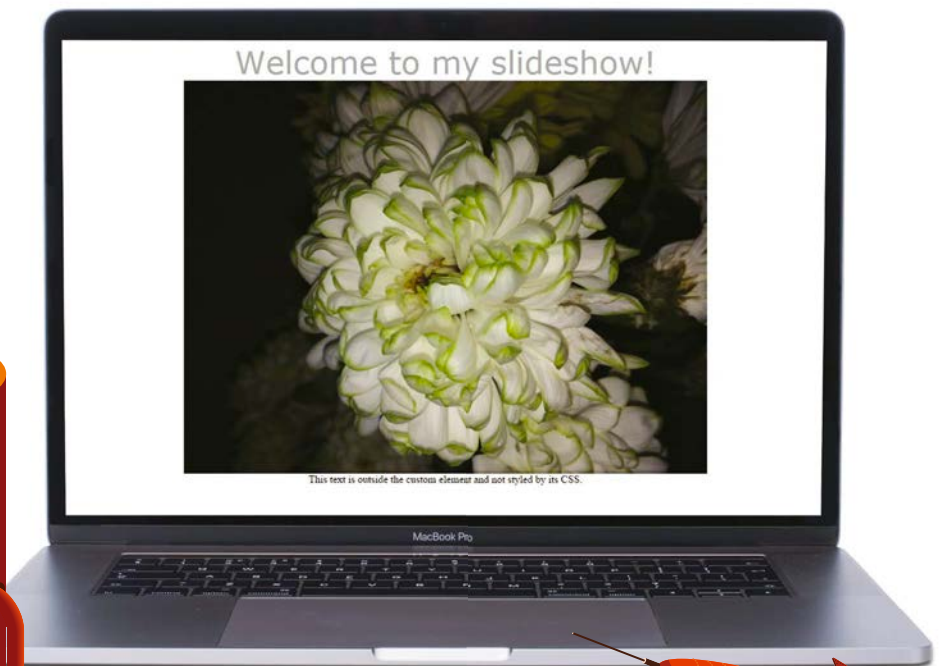
ensure that everything can be bundled through a single import statement. We will not need to import scripts separately to use the component.

Within `slideshow.js`, we'll create the shell of a custom element:

```
class SlideshowComponent extends
HTMLElement {
constructor() {
// Invokes the constructor of the parent
class; this is mandatory
super();
}
}
// Define the custom <slideshow-component>
element using the class above
customElements.define("slideshow-component",
SlideshowComponent);
```

05. Importing the template

Our next step is to ensure that when the custom element is created, it instantiates the HTML



ESSENTIAL RESOURCES TO READ AND WATCH

Google Developers Web Fundamentals

bit.ly/2lu0A96

Google has been at the forefront of the Web Components drive for several years, with Chrome leading in browser support. Eric Bidelman, a developer at Google, has written some excellent articles covering the latest v1 version of the standards.

An Introduction to HTML Imports

bit.ly/2kRH3Ss

While conceptually simple, HTML Imports can be one of the more troublesome parts of a Web Component. Matt West walks through them in detail, and provides guidance on where to look to solve the same-origin issue.

What's New In Shadow DOM v1

hayato.io/2016/shadowdomv1

Shadow DOM is arguably the most complex of the Web Components specifications, and has evolved significantly since its v0 implementation. Hayato Ito provides a detailed breakdown of what's changed.

Web Components and Polymer

bit.ly/2kuVuv5

Talks by Taylor Savage and Monica Dinculescu at the Chrome Dev Summit 2016 give some great context on why Web Components are useful, how to take advantage of them. The video also covers Google's Polymer library.



template we defined, otherwise it would not appear on the page! To do this, we'll add some additional functionality to the bottom of the constructor we just created.

Remember, we're providing a list of images in the `src` attribute of the element:

```
// Read the src attribute of the element
to obtain a list of images
var imageArray = JSON.parse(this.
  getAttribute("src"));
var currentImage = 1;
```

We obtain the template itself by finding the appropriate `<link>` element on the main page, then searching for our template by id:

```
// Import the slideshow template from our
component file
var importedDocument = document.querySelec
tor('link[rel="import"]').import;
var template = importedDocument.
  querySelector("#slideshow-template");
// Create a copy of the slideshow template
ready to use
var copy = document.importNode(template.
  content, true);
```

06. Creating shadow DOM

So we have the template's content. Remember that we don't want to add it directly into the main document. To maintain encapsulation, we need to build a shadow DOM tree with our custom element as the host. This requires a little more work in the element's constructor after the template has been retrieved:

```
// Create a Shadow DOM root with this
component as the host
var shadowRoot = this.attachShadow({mode:
  "open"});
// Add the copied template to the shadow
DOM
shadowRoot.appendChild(copy);
```

At this point, the content of our template will render on the page when the `<slideshow-component>` tag is used, but will not form part of the main document DOM tree.

07. Building out functionality

Our element still doesn't do much. In fact, that `` tag is still blank! We need to add a few additional functions to the constructor to make the element interactive. Firstly, we need a method to update the image:

```
// Display the image at the specified
position in the array
function displayImage(imageNumber) {
  currentImage = imageNumber;
  shadowRoot.getElementById("slideshow-
    image").src = imageArray[imageNumber
    - 1];
}
```

Note that the update is made in the shadow DOM.

Then we'll build two more simple functions which call this:

```
// Display the next image
in the array, or go back to
the beginning
function nextImage() { ...}
// Pick a random image from the
array to display
function randomImage() {...}
```

08. Finishing the component

To round off our component script, let's ensure that the interactivity works as expected. We want to add an event listener for click events to trigger a change in image, and ensure that a random image is displayed when the element is first created:

```
// When we click the component, move to
the next image
this.addEventListener('click', e => {
  nextImage();
});
// Display a random image when the
component is first loaded
randomImage();
```

We now have a fully functional component. It consists of an HTML template, some simple CSS, and a JavaScript file defining custom element behaviour.

09. Putting it to use

Using the component is surprisingly easy. We import it through a single line:

```
<link rel="import" href="slideshow.html">
```

This statement will also load all CSS and JavaScript referenced in the HTML file, and execute the JavaScript to register the custom element. However, because of the same-origin policy, you'll need to test this on a server. If you don't have one available, move the content of your `slideshow.html` file into the page you're testing with.

From there, the custom component works exactly as expected:

```
<slideshow-component src=["img/
  image1.JPG","img/image2.
  JPG","img/image3.JPG"]>
</slideshow-component>
```

We can also test out the scoping of CSS to ensure it does not affect the rest of the page, by adding below it:

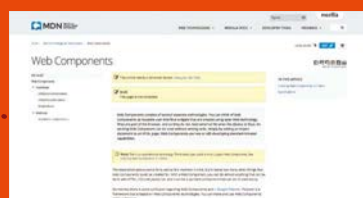
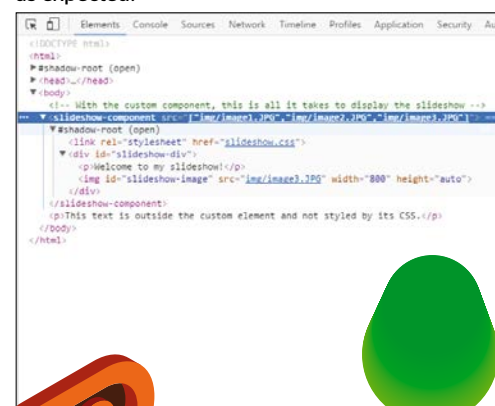
```
<p>This text is outside the custom
element and not styled by its
CSS.</p>
```

THE POSSIBILITIES
ARE ENDLESS!

So there you have it - your first Web Component! Play around and see what else you can make it do, or try to build your own from scratch.

10. In the browser

Let's finish up by taking a look at the page in DevTools. We can see the custom element with a shadow DOM hierarchy below it, exactly as expected!



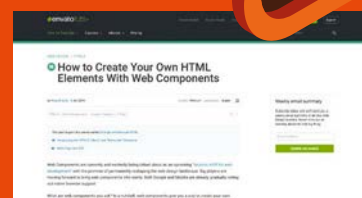
Mozilla Developer Network mzl.la/2kKMxCM

MDN provides details of the Web Components specifications, including links to the latest W3C working drafts. A number of tutorials and examples are also available.



HTML5 Rocks bit.ly/2kW4euA

Google developer Eric Bidelman strikes again with some more great articles covering other parts of the Web Components specifications, complete with examples.



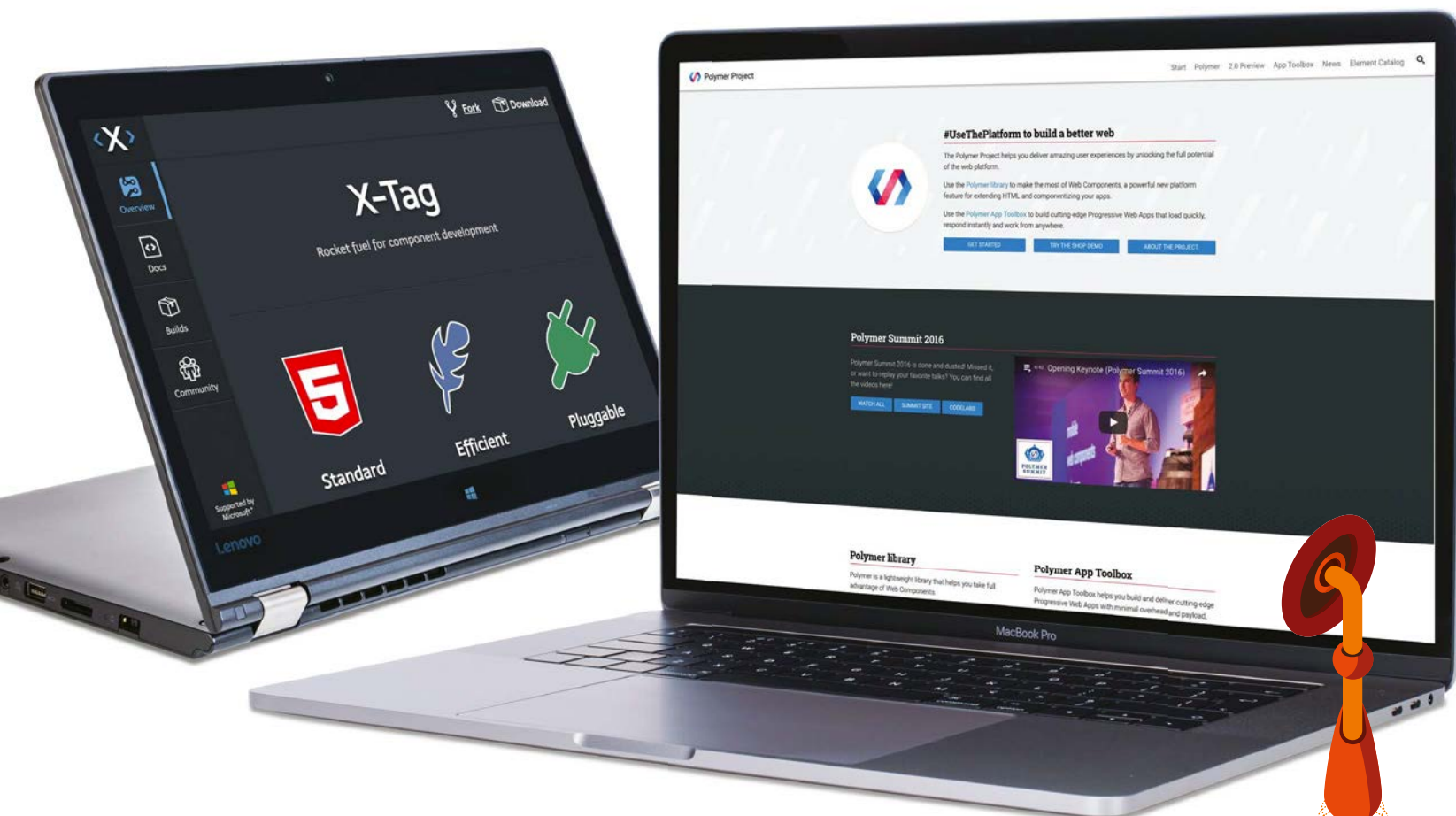
How to Create Your Own HTML Elements bit.ly/1mysVXJ

Kezz Bracey provides a back-to-basics walkthrough of Web Components, rich with examples, and explains how all the key specifications fit together.



Using Web Components to Build PWAs bit.ly/2k6PrtD

From Progressive Web App Summit 2016, Eric Bidelman talks us through the creation of a custom element with code examples.



BROWSER SUPPORT AT A GLANCE






Take note before you get started, to avoid confusion and frustration

Web Components aren't new. In fact, they were first introduced back in 2011, generating a great deal of excitement. However, like any new specification, it has taken time for browsers to implement the new technologies.

Things have come a long way over the last six years, with updates and refinements to the specifications, and polyfills gradually being

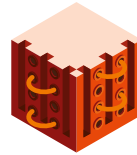
replaced by native browser support. Google has led the way, with Chrome consistently providing the most complete Web Components support. The real question, however, is whether Web Components are ready for prime time. Given their slow adoption, many have asked whether we should still be pursuing these specifications. The answer is yes, but we aren't quite there yet. Templates are

generally well-implemented; custom elements and shadow DOM are in progress; however, imports lag behind and there is some question around whether Firefox will ever implement them. The site www.webcomponents.org maintains the latest information on browser support, so keep an eye out. For now, to learn and experiment, you'll probably want to stick with Chrome or Opera.

					
CUSTOM ELEMENTS	✓	✓	✗	✗	✗
SHADOW DOM	✓	✓	✗	✓	✗
HTML IMPORTS	✓	✓	✗	✗	✗
HTML TEMPLATES	✓	✓	✓	✓	✓

WHAT ABOUT MOBILE?

Mobile support for Web Components lags a little behind desktop browsers, but is catching up. Currently, Chrome for Android is the only mobile browser with fully fledged support (Chrome on iOS uses the same rendering engine as Safari).



THE PROS & CONS: POLYMER & X-TAG

Manipulating the Web Components APIs directly can be cumbersome, particularly as our components grow more complex. This is where libraries such as Polymer and X-Tag come in. Both provide a lightweight layer of abstraction which simplifies the creation and use of Web Components without losing their benefits.

Polymer is supported by Google and X-Tag by Microsoft. They share a common goal, but there are key differences. X-Tag offers features only related to custom elements, whereas Polymer also supports Shadow DOM and HTML Imports.

POLYMER

```
PolymerElement = Polymer ({
  is: "polymer-element",
  created: function() {
    alert("Created a Polymer element!");
  }
});
```

The Polymer function is used to register a new custom element. We pass it a prototype, with an `is` property which defines the name of the custom element. Instances can then be created using `document.createElement`.

Crucially, both Polymer and X-Tag make use of polyfills to ensure a consistent cross-browser experience while native support for Web Components is under development.

The developers of X-Tag say this focus on custom elements makes it simpler to work with, useful for smaller projects. But if you want to fully benefit from the other features of Web Components, Polymer, which is arguably the more prolific of the two at this time, could be the way to go.

As a starting point, let's take a look at how a new element is registered using each library:

X-TAG

```
xtag.register("xtag-element",{
  lifecycle: {
    created: function() {
      alert("Created an X-Tag element!");
    }
  }
});
```

Creating an element in X-Tag is similarly simple, using the `xtag.register` function. We can define similar functionality to that we added with Polymer through lifecycle callbacks.

ANGULAR OR WEB COMPONENTS?

Hold on, doesn't Angular.js also give us the ability to create custom HTML elements? So are they the same? There is certainly an overlap in features, but the two do not conflict as emerging web development standards. In fact, one of the primary drivers for such a large shift in approach for Angular 2 was to make the framework use the Web Components specification. This is because Angular is meant to be a full-scale framework, using a combination of modules and components to organise functionality.

A key difference between a site built on Angular and a site not built on Angular (both making use of Web Components) is the performance implications Angular doesn't implicitly rely on HTML5 support in the browser, resulting in an overhead of code to support the framework functionality. Web Components expect the browser to have these features available and will always try to use them natively, offering reduced overhead and better performance.

If you're building a fully fledged, single-page app and need to ensure cross-browser support, Angular is probably the way to go for now. However, the Web Components spec will outlast any current framework and so a library like Polymer or X-Tag may be a more future-proof option.

POLYFILLS

Each major browser has its own rendering engine, and for new features such as Web Components these may be at different stages of development. Polyfills provide the code required to flatten the cross-browser landscape. Like the paste used to fill cracks in a wall, polyfills fill the cracks in the functionality available in your browser so that a developer can treat them all (almost) the same. Where a browser doesn't provide some needed HTML5 functionality natively, polyfilling provides the missing parts needed to ensure your site still works the same way.



FUTURE DEVELOPMENTS

Web Components have fantastic potential, but there is still some way to go. What's next?

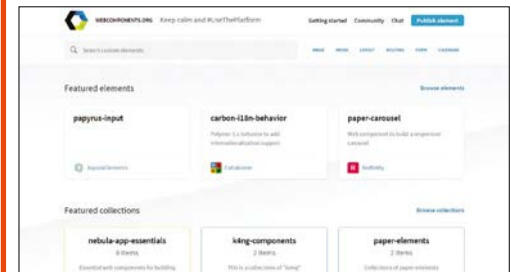
Web Components are an interesting study in the challenges of making major changes to HTML specifications. With the reliance on browser makers to implement the technology, it takes time for new features to reach maturity. The good news is that browser standardisation is now much more consistent than it once was, and Web Components are being backed by the right players. It's also clear that they offer a more elegant solution to key problems facing web developers than previous, often improvised solutions.

It's encouraging that there are major production sites live today using Web Components, and in particular Polymer. All the evidence indicates that this continues to be the direction the industry will

move. Over the coming years, expect to see custom element and shadow DOM support extended across all the main desktop and mobile browsers. The future of HTML Imports is less certain, but this is not a critical element of the specification and there are other solutions to the problem it addresses.

With a growing trend away from desktop sites and native mobile apps, in favour of web apps, Web Components are likely to come into their own alongside several other emerging web technologies. Expect more focus on bringing the maturity of native software development to the web. For developers, now is the time to begin exploring and adopting, to get ahead of the curve.

COMPONENT COLLECTIONS



Web Components

webcomponents.org

As well as an extensive source of reference, webcomponents.org offers an element registry with a wide range of user-created elements. Well worth taking some time to explore.

Custom Elements

customelements.io

Another large catalogue of user-created elements, ranging from simple UI features to complete frameworks designed for some of the latest technologies.

Strand

strand.mediamath.com

Strand is a library of UI components built on Polymer, designed for consistency and interoperability. Expect to see more collections like this emerge in the future.

web workshop

Code a drag-and-slide navigation button

As seen on www.banxico.org.mx/mibanxico

The main menu

The menu is hidden off screen and when the user clicks the burger icon, each of the links bounces onto the screen one at a time.

Extra movements

The movement of the characters' heads on the screen updates as the user moves the slider over to the right.

Circular transition

When the user has dragged the slider a circular transition moves out from the centre of the screen while the new content is loaded into the screen.

Alternative navigation

As an alternative to dragging and sliding the arrow, there are a number of small spots that will also serve as a navigation clue with a simple click.

Drag and slide

The navigation works by dragging the arrow over to the right until it hits the smaller circle. In the background, the dotted line animates over to the right to emphasise the arrow's movement.



Create a drag-and-slide navigation button

↓ **DOWNLOAD TUTORIAL FILES** www.filesilo.co.uk/webdesigner

EXPERT ADVICE

Let the user decide

Experimental navigation such as the dragging of the slider on the Mi Banxico site is great to see, but not everyone viewing the site will want to do a click and drag when they are used to just clicking a button. Secondary navigation is always important to ensure your audience can easily navigate to their liking.



ne of the best sites seen in the Web Designer offices for a long time is the Mi Banxico site. The site has bright, bold

colours, and as the styling for the site these change depending which section of the site the user is on. The boldness seems to be a recurring theme within the ethos of the site. The illustration work for the site is all created in 3D with the site heading in 3D lettering and then the illustrated character work that is featured on each page. The illustrations are in the styling of a bright Pixar movie which works perfectly with the bold backgrounds. There is even a boldness to the navigation

with the user required to drag a slider of an arrow inside a circle across to another point, all the while the dotted line is animating in the direction that the user has to drag. Once the user moves the slider the illustrated elements on the page also start to move and they react to the drag position of the user. Even the main menu doesn't just move onto the screen in the conventional animated way. Instead each menu item and background colour, which is colour coded, bounces onto the screen, completely grabbing the users attention. The way all of these elements work together shows the great integration of animation and interaction.



<comment>
What our experts think of the site

Three focus points

"The style of Mi Banxico is the result of great teamwork. In this project I focused on three main things: smooth shapes with rounded characters, backgrounds and fonts; a vibrant colour palette; and finally the sharpness of the image, achieved with the use of vector images."

Alejandro Tello Pérez, Art Director, Founder & Partner at D10studio

Technique

1. Drag slider

Creating a draggable arrow like on the Mi Banxico site is quite complex and can be done a number of ways. This solution will use CSS to get the animated dashed line and JS to add the drag effect. Open 'start.html' and add the code to the body.

```
<div id="holder">
  <div id="border"></div>
  
  
  <div id="cover"></div>
</div>
```

2. Creating the line

Now add the CSS for the border element. This will create a dashed background so that we can animate the movement of the lines. Different browsers will require the right prefixes for the gradient.

3. Moving the line

Here the animation keyframes are set so that the line continues to move across the screen and this helps the user to know exactly what to do with the arrow. In the previous step's code, it was set to move infinitely so that it continuously animates.

```
@keyframes border-move {
  0% {
    background-position: 0px 0px;
  }
  100% {
    background-position: 300px 0px; } }
```

4. Adding the interactions

Move to the script tags at the bottom of the document and with this code it gets the position of the holder element. This will be different depending on screen size. HTML ID elements are cached and event listeners added.

```
var offsets = document.
  getElementById('holder').
  getBoundingClientRect();
var left = offsets.left;
var cover = document.getElementById('cover');
var drag = document.getElementById('drag');
drag.addEventListener('mousedown', mouseDown,
  false);
window.addEventListener('mouseup', mouseUp,
  false);
```

5. Up and down

The following code checks whether the mouse is pressed or not; if it is, the 'divMove' function is used as a listener to the mouse movement. When released the position is checked so that if dragged to the other circle it will trigger a page load.

```
function mouseUp(e) {
  window.removeEventListener('mousemove', divMove,
    true);
  if (e.clientX - left > 280) {
    window.location.href = 'https://www.google.com';
  } }
function mouseDown(e) {
  window.addEventListener('mousemove', divMove,
    true); }
```

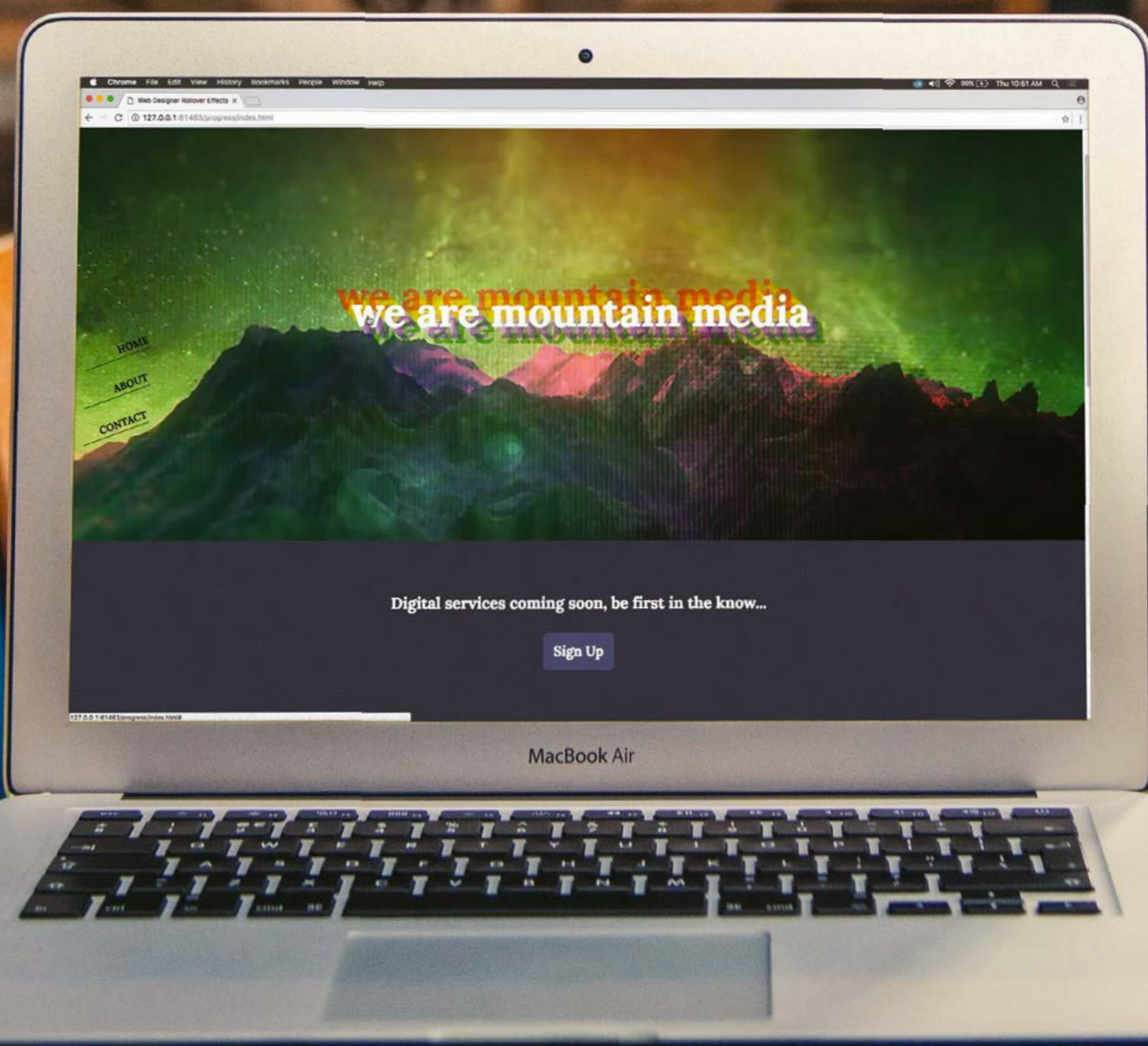
6. Moving the mouse

The next code is only called when the mouse is pressed and moving the object. It updates the element position by taking off its original position. A div named 'cover' is made wider to hide the dotted animation behind.

```
function divMove(e) {
  if (e.clientX >= left && e.clientX <= (left
    + 320)) {
    drag.style.left = (e.clientX - left) + 'px';
    cover.style.width = (e.clientX - left) +
    'px'; } }
```


Create multiple CSS animation effects

Make your site stand out from the crowd with stunning CSS animation effects that will make your content pop



In this tutorial, we have been influenced by the Animatable project (leaverou.github.io/animatable) which isn't a library but just a demonstration of all the effects that can be achieved when the designer animates just one property of CSS in their project.

While this tutorial occasionally delves into animating multiple properties, you'll be surprised from that site exactly what is achievable by animating just one property.

The project will set up a one-page website that will contain a fixed navigation section on the left and we'll add rollover animations to this. Then the main content will feature a number of hover animations for the user as they start to interact with the content on the page.

An important point to remember in any web project is that every part of the page - whether it's text, imagery or any other content - is an interface that causes the users to move deeper into the site. The very best sites make their interactive content really stand out to their users by making it react and animate when the user rolls over their links.

As all of the content in the project is animation, this is all controlled via CSS. So in order to focus on just this important content, the HTML is already written so that all you have to work on is adding the CSS that animates that content.

1. Opening up

From the project files, drag the **start** folder into your code editor and open the page **index.html**. The first step will be to connect the typeface that will be used and assign the stylesheet that will contain all of the design for the project. Add the code shown into the head section of the **index.html** page.

```
<link href="https://fonts.googleapis.com/
css?family=Lora:700" rel="stylesheet">
<link rel="stylesheet" href="css/design.
css">
```

2. Page basics

Switch over to the **design.css** file, which is located in the **css** folder. The first CSS line will just remove the default browser margins and padding on all elements in the page. Following that, the page will be styled up with the correct typeface, background and font sizing.

```
* { margin: 0; padding: 0;}
html, body
{
width: 100%;
height: 100%;
font-family: 'Lora', serif;
font-size: 1.1em;
line-height: 1.5em;
text-align: center;
background-image: url(../img/grey.png);
}
```

3. Page navigation

The page navigation is a very important part of every site and here the navigation is being made fixed, so that it sticks to the left-hand side of the screen as the user navigates up and down. The list elements are just having their bullets removed.

```
nav {
position: fixed;
z-index: 200;
left: 10px;
top: 35%;
}
ul {
list-style-type: none;
}
```

4. Defining the angled links

Each of the menu links is going to be tilted so they are slightly rotated upwards on their right-hand edge. This step adds that and also ensures that each element has sufficient space to move when the user rolls over the link.

```
.tilt {
width: 130px;
color: #000;
text-align: right;
display: block;
transform: rotate(-20deg);
margin-top: 40px;
transition: all 0.5s ease;
transform-origin: 0 0;
border-bottom: #000 1px solid;
}
```

5. Rollover transition

As the user hovers over the tilted link, it will move upwards slightly, using the transform-origin property to move it. The background and text will also just change colour so that it highlights the interaction further.

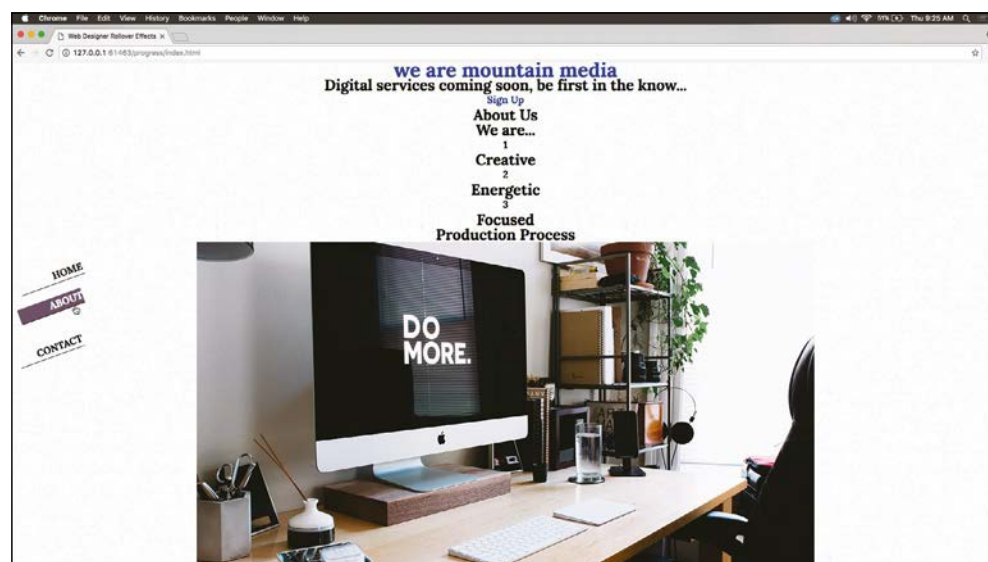
```
.tilt:hover {
transform-origin: -30% -30%;
background: #715871;
color: #fff;
}
a {
text-decoration: none;
}
```

6. The top of the page

At the top of the page, there will be a large header section containing an image and a rollover effect on the text heading. Continue to add the CSS code which will position the header and text with the background image.

Setting up transitions

The transition setup always follows these parameters - `transition: CSSproperty timing optionalTransitionType;`. Use 'all' for CSSProperty to change multiple CSS properties at once.



Top left

The main title is styled up with a rollover effect on the text, this uses CSS keyframes and the background shadow to get the multi text effect

Top right

The sign-up section is styled with a dark background. The transition of the button causes it to expand dramatically with the drop shadow movement on the text

Left

The design of the navigation is in place with it fixed to the left side of the screen. Hovering over the text will transition the link upwards and change the colour

Tutorials

Create multiple CSS animation effects

```
header {
  height: 80%;
  background: url(../img/bg.png) no-repeat
  center center fixed;
  background-size: cover;
}

header h1 {
  position: relative;
  top: 50%;
  font-size: 4em;
}
```

7. Heading transition

The heading is still in the default link colour, so here the colour is changed to white to allow it to stand out. The hover over effect will use animation keyframes named 'multicol'; these will play back and forth infinitely, taking one second to do so.

```
header h1 a {
  color: #fff;
}

header h1 a:hover {
  animation-name: multicol;
  animation-duration: 1s;
  animation-direction: alternate;
  animation-iteration-count: infinite;
}
```

8. Adding the keyframes

Here, the keyframes are defined for the 'multicol', which is short for multicoloured. These use a number of background shadows to add different colours to the text. Try viewing this in the browser and seeing the effect in

action by hovering your mouse over the title.

```
@keyframes multicol {
  0% {
    text-shadow: 0 0 #fff;
  }
  100% {
    text-shadow: 0 0 rgba(255, 255, 255, 50),
    -30px -30px 0 rgba(255, 0, 0, 0.4), -20px
    -20px 0 rgba(255, 165, 0, 0.4), -10px
    -10px 0 rgba(255, 255, 0, 0.4), 10px 10px
    0 rgba(238, 130, 238, 0.4), 20px 20px 0
    rgba(75, 0, 130, 0.4), 30px 30px 0
    rgba(2, 92, 2, 0.4);
  }
}
```

9. Sign up section

Below the header, there will be a sign-up section on the page. The correct spacing is given for this with the padding, but a coloured background with a semi-transparent PNG is added over the top of this. The button will have a hover effect that adds padding to the side of it, so that it expands wider to fill the space.

```
.signup {
  padding: 100px 0;
  background: #38394a url(../img/cartographer-
  new.png);
  color: #fff;
}

#sign:hover {
  text-shadow: 20px 20px 12px rgba(0, 0, 0,
  .5);
  padding: 20px 300px;
  background: #e16762;
}
```

10. Placing the button

The button is turned into a block element, but still inline and is given a small round corner on the button. The background is set and a drop shadow added that will also be transitioned with the hover of the mouse.

```
#sign {
  display: inline-block;
  margin-top: 40px;
  padding: 20px;
  border-radius: 10px;
  background: #4e5273;
  color: #fff;
  font-size: 1.3em;
  text-shadow: 0 0 black;
  transition: all 0.5s ease;
}
```

11. Defining the space

The next part on the screen is an about section. This will have three panels placed side by side; each will have an animated effect added to a circle at the top of these panels. The CSS here just defines the headings and that it will be 1200 pixels wide.

```
#about {
  width: 1200px;
  margin: 100px auto;
}

#about h2 {
  margin-bottom: 30px;
}
```

12. Clearing the decks

An alternative colour for the headings is added to the design and then an important styling is added. The three panels in this section will be floated side-by-side. The section here states that at the end of the about section there should be a block that has no content and clears the floated panels.

```
h2.alt {
  color: #666;
  font-weight: 100;
}

#about::after {
  content: "";
  display: block;
  clear: both;
}
```

CSS transforms

Transforming using CSS has many options - in 2D alone, there is scaling, translating, rotating and skewing. So it's worth experimenting with these to find how to apply them.



Right

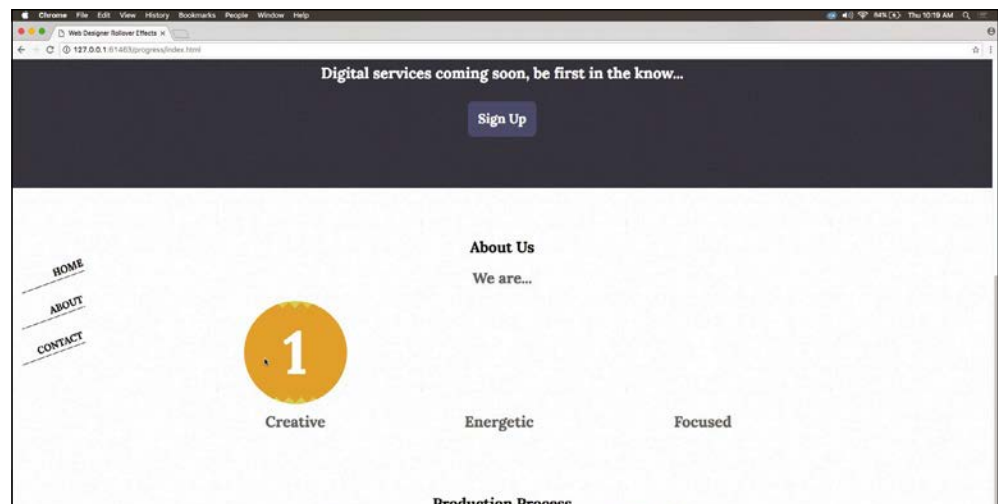
The first rollover section adds a change of background colour to reveal the jagged opening edges, which looks like jaws opening!

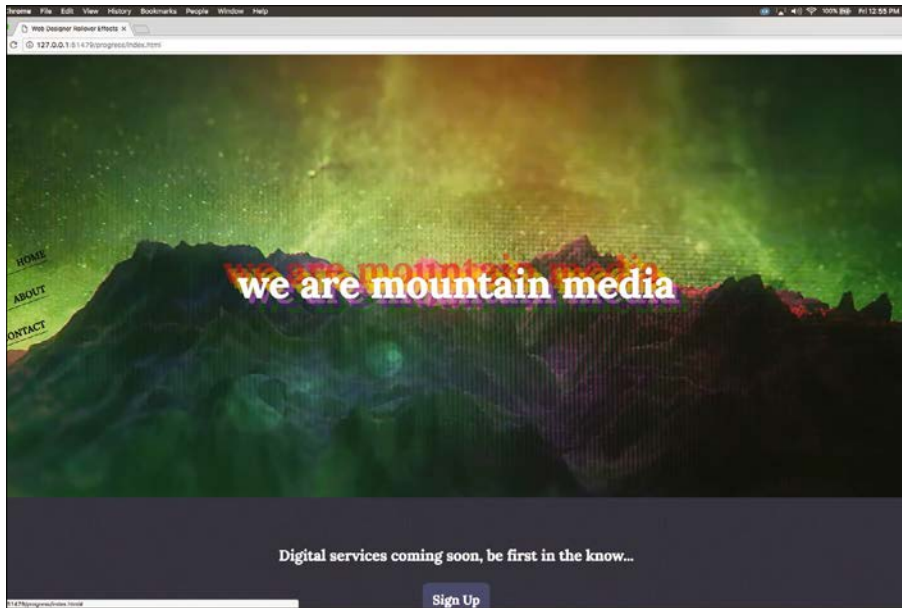
Top left

The second rollover is visible here; this is actually mid transition and the final positioning looks different from this

Top right

The final rollover section shows how to rotate and move an image in 3D. This is mid transition with the image rotating and zooming towards the user





Getting to grips with keyframes

CSS keyframes can look a little overwhelming on your first glance at the code, but they actually run on a fairly simple system. First of all, to call the keyframes to your selector, simply add the name of the keyframes - animation-name: example;.

Now, as that is called, you need to create the keyframes. To work with the previous code, you would add:

```
@keyframes example {
  from {background-color: #f80;}
  to {background-color: #ff0;}
}
```

It's also possible to put the % in instead of from and to. So this would directly relate to 0% and 100% and, of course, that gives options to add keyframes at 30% or wherever they would be needed.

In order to add timing, you could use - animation-duration: 2s; You can also add an iteration count, direction and delay to this, and build up your animation.

13. Three panels

The panel class is created now; this gives each panel a 30% width, which when added together gives 90%. A padding on each side of the panel of 1.5% takes the total up to 99% width, fitting neatly into the space allocated with 1% to spare!

```
.panel {
  width: 30%;
  padding: 0 1.5%;
  float: left;
}
```

14. Blobbing out

The blob class defines a 200 by 200 pixel circle at the top of each panel. There is no background colour here, because each will have its own colour and rollover effects. The number is vertically aligned in the centre of this blob, using line height.

```
.blob {
  width: 200px;
  height: 200px;
  border-radius: 50%;
  margin: 0 auto 20px;
  font-size: 6em;
  line-height: 200px;
  vertical-align: middle;
  color: #fff;
}
```

15. First circle

The first circle is given a jagged background effect which will only be visible on the rollover because the background colour is the same as the jagged effect. An image is constructed out of the gradient, but it won't look much like a gradient when the results are visible!

```
.one {
  background-color: #d2cf65;
```

```
background-repeat: repeat-x;
background-size: 24px 300px;
background-image: linear-gradient(135deg,
  #d2cf65 25%, transparent 25%), linear-
  gradient(45deg, transparent 75%, #d2cf65
  75%), linear-gradient(45deg, #d2cf65 25%,
  transparent 25%), linear-gradient(135deg,
  transparent 75%, #d2cf65 75%);
transition: all 0.5s ease;
background-position: 3px 0, 3px 0, 15px
  -100px, 15px -100px; }
```

16. First hover

As the user moves their mouse over the first circle, the background colour changes and the jagged effect moves to reveal the effect. Save this and view it in the browser to see the effect working.

```
.one:hover {
  background-position: 3px -70px, 3px -70px,
  15px -30px, 15px -30px;
  background-color: #dda048; }
```

17. Second circle

The colour and background are defined for the second circle here. Again there is a gradient involved, but this time there is a repeating radial gradient which isn't visible until the rollover effect, when it will be revealed to the user.

18. Second hover effect

In order to get the rollover effect working for the second, it just requires the size of the background gradient changing, but it's dramatic to see so test this out. There's also the background colour added for the third circle.

```
.two:hover {
  background-size: 150px 150px;
}
.three {
```

```
background-color: #9e6088;
box-shadow: none;
transition: all 0.5s ease; }
```

19. Third hover effect

The last circle will have a much simpler effect added to it using shadows. This time the box-shadow property is used and it's added to the inside of the circle so that an inner shadow is added. Here, the background for the production section is also set.

20. Final effect

In the production panel is a large image; this has the id of 'spin'. Here, the CSS resizes the image and sets the transform of the object so that it is rotated 90 degrees. This makes the image invisible, but the rollover will rotate it into view.

```
#spin {
  width: 600px;
  display: block;
  margin: 80px auto;
  transform: perspective(550px) translateZ(-
  500px) rotateY(90deg);
  transition: all 0.5s ease;
```

21. Final hover effect

Here, the CSS states that when the user's mouse is over the 'roll' section, add this CSS to the spin section, which rotates the image back into view and brings it closer on the z axis so that it zooms in. Save and view in the browser to see this effect.

```
.roll:hover > #spin {
  transform: perspective(550px)
  translateZ(150px) rotateY(0deg);
}
footer {
  margin-top: 50px;
  padding: 50px; }
```


Build unique layouts with CSS Shapes

Learn how use the power of CSS to create multiple shapes for layout and presentation





One of the best features of CSS is that it reduces the use of images in web design and enables web designers to create more abstract, geometric

layouts, beyond simple rectangles and squares. Not only does this give us more control of our page layouts, but also it helps to improve performance due to there not being so many HTTP requests. Magazines and newspapers have always enjoyed elegant ways for arranging content layout, such as wrapping text around, or inside, non-rectangular shapes. Rectangles inside other rectangles – this is what our webpages have always been made of. The new CSS Shapes specification is changing that, and its goal is to provide web designers with a new way to alter how the content flows inside and around arbitrarily complex shapes – something we've never been able to do before.

In this tutorial, we will take a look at several ways we can use shapes for layout, such as wrapping text around a circle, or creating more complex shapes such as a polygon. Then we will go through several other shapes that we can create. So open up your text editor and let's get in shape!

1. Setting up

In this guide we will just be using a text editor and your browser. So the first thing to do is create a new folder directory on your computer with a new .html file and .css file. Then open them both up in your favourite text editor (we're using Sublime 3) and then we can begin coding.

2. The HTML

In your HTML document we will add in a div with a class name of 'box'. Then, within the box div, we will add in another div with a class name of 'circle'. Underneath that, add as much content as you want; we'd say at least two paragraphs with a hundred-plus words in each.

```
<div class="box">
<div class="circle"></div>
<p>Some content....</p>
<p>Some content....</p>
</div>
```

3. Background image

For us to really get a feel of how this effect will take shape (pun intended), it would be a good idea to jazz up the page a bit. So once you've styled your page elements such as nav bar and footer, go ahead and add a nice background image to the body.

```
body {
background: url('lake-bg.jpg') no-repeat
center top;
font-family: Verdana;
line-height: 22px;
size: 14px;
overflow: hidden;
}
```

4. The shape-outside property

The CSS Shapes specification provides us with new CSS properties which include shape-outside and shape-margin. The shape-outside property controls how content will wrap around a floated element's bounding-box. Typically this is so that text can reflow over a shape such as a circle, ellipse or a polygon.

```
.circle {
float: left;
shape-outside: circle(50%);
width: 300px;
height: 300px;
}
```

5. Clip the path

As you will have noted in the last step, we can't actually see our circle, but we can see the text wrap around it. So let's change that by giving it a white background. Then we need to use the clip-path: property which allows us to specify a specific region of an element to display, rather than showing the complete area. Without it you'll get a square instead of a circle.

```
.circle {
clip-path: circle();
background-color: #f5f5f5;
margin: 20px;
}
```

6. Adding a profile image

As you can now see, using the shape-outside property makes the text float around our white circle and gives us so much more creativity. We can use this in a number of ways – one would be to use it for a profile image, say. So, within the 'circle' rule, add in a background image. Then we will finish this example by pushing down the box element by 50px.

```
circle {
background: url('profile.jpg') no-repeat
center;
clip-path: circle();
background-color: #f5f5f5;
margin: 20px;
}
.box {
margin: 50px 0;
}
```

7. Creating a polygon shape

Beyond simple shapes like circles and rectangles, we can also create polygons: shapes consisting of multiple sides, such as a pentagon or a hexagon. To create one, we use the CSS polygon() function, which takes multiple comma-separated values to set the shape's coordinates.

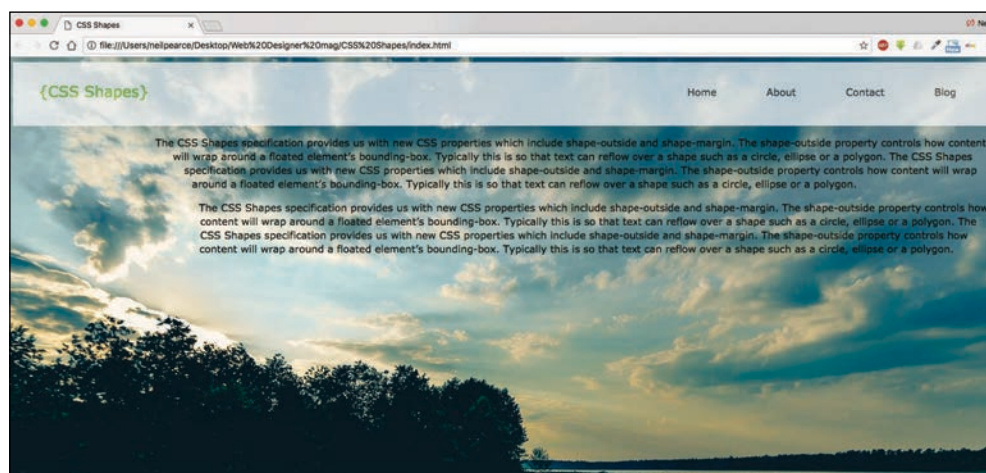
```
<div class="shape pentagon"></div>
<blockquote>
<p>Blockquote content</p>
</blockquote>
```

8. The pentagon

Creating a polygon shape manually is not as easy as creating a circle. In this CSS, we have added five coordinates to the polygon() function, thereby generating a pentagon shape. Adding another coordinate to the

Using shape-margin

Wrapping content around a shape can make it rub too closely against the element. You can add spacing around the shape with the shape-margin property.



Top left

The skeleton HTML has been added, ready now to begin adding in the CSS

Left

The background image has now been added along with a navigation bar and footer

Top right

The first shape we created is a circle that will be floated over to the left

Tutorials

Build unique layouts with CSS Shapes

function will turn the shape into a hexagon; one more for a heptagon, another for an octagon, and so on.

```
.pentagon {
  background-color: #333;
  shape-outside: polygon(32px 126px, 163px 32px,
    293px 127px, 244px 282px, 82px 281px);
  clip-path: polygon(32px 140px, 163px 32px,
    293px 140px, 244px 282px, 82px 281px);
  shape-margin: 20px;
  width: 500px;
  height: 500px;
}
```

9. Make the blockquote more appealing

Now that we have a pentagon shape sitting within our blockquote, we can get a little bit creative with this by adding some more styling and include another image within the pentagon to make it look like this is the person making the quote. So add this code in your HTML file, making sure you use an image of your own.

```
<section class="shape-box">
  <div class="shape pentagon">
    </div>
    <blockquote>
      // content goes here
    </blockquote>
  </section>
```

10. Adding the blockquote content

Keeping to our creative plan, go ahead and create a quote symbol in Photoshop (or other image editor) that is about 50 × 40px and export it as a PNG. Then hop back

into the HTML file and add the following markup inside the .shape-box class.

```
<div class="quote"></div>
<blockquote><p>In geometry, a pentagon is any
  five-sided polygon or 5-gon.</p>
<cite><a href="https://en.wikipedia.org/wiki/
  Pentagon">Wikipedia</a></cite>
</blockquote>
```

11. Styling the blockquote

We can now position our new quote symbol using CSS and position it to the left of our blockquote with the opacity turned down. For us to be able to position this correctly, we need to make sure the .shape-box class is positioned relative and with a width of 800px.

```
.shape-box {
  position: relative;
  width: 800px;
  border: 1px solid #fff;
  margin-top: 100px;
  left: 500px;
}
.quote {
  background-image: url('quote.png');
  position: absolute;
  top: 20px;
  left: 100px;
  height: 40px;
  width: 49px;
  opacity: .2;
}
```

12. Finishing the blockquote

Floating the .shape class to the left will make sure the text is positioned on the right-hand side. Then we can style our profile image and position it using relative positioning, and centre it using both left and right.

```
.shape {
  float: left;
  margin: 30px;
```

```
}
.shape img {
  position: relative;
  left: 30px;
  top: 30px;
}
```

13. More shapes

Now that we have got a good grasp of how we can use CSS shapes to create simple layouts, we will take a look at other shapes we can implement, using percentages too. So, change the pentagon shape-outside and clip-path properties as shown below. You'll now have a shape called a 'rhombus'.

```
shape-outside: polygon(50% 0%, 100% 50%, 50%
  100%, 0% 50%);
clip-path: polygon(50% 0%, 100% 50%, 50%
  100%, 0% 50%);
```

14. Creating a comment bubble

Using the polygon function, we can create some very handy shapes. How many times have we created tooltip shapes or comment bubbles using a graphics app, such as Photoshop? Well, this is how we can create a comment bubble in pure CSS.

```
clip-path: polygon(0% 0%, 100% 0%, 100%
  75%, 75% 75%, 75% 100%, 50% 75%, 0% 75%);
```

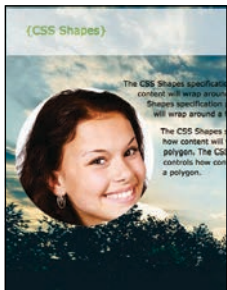
15. Right and left chevron

Using these coordinates, we can easily create a right and left chevron. These are very handy for pointing or directing the user to a different bit of content on the page. You can also play with the percentages to create smaller, more narrower chevrons.

```
clip-path: polygon
  (100% 0%, 75% 50%, 100% 100%, 25% 100%, 0%
  50%, 25% 0%);
clip-path: polygon
  (75% 0%, 100% 50%, 75% 100%, 0% 100%, 25%
  50%, 0% 0%);
```

Browser support

Even though the support for CSS shapes is getting better, it would certainly be wise to add your browser prefixes to your CSS.



Top left

The profile image will sit within our circle and the text will now flow around it

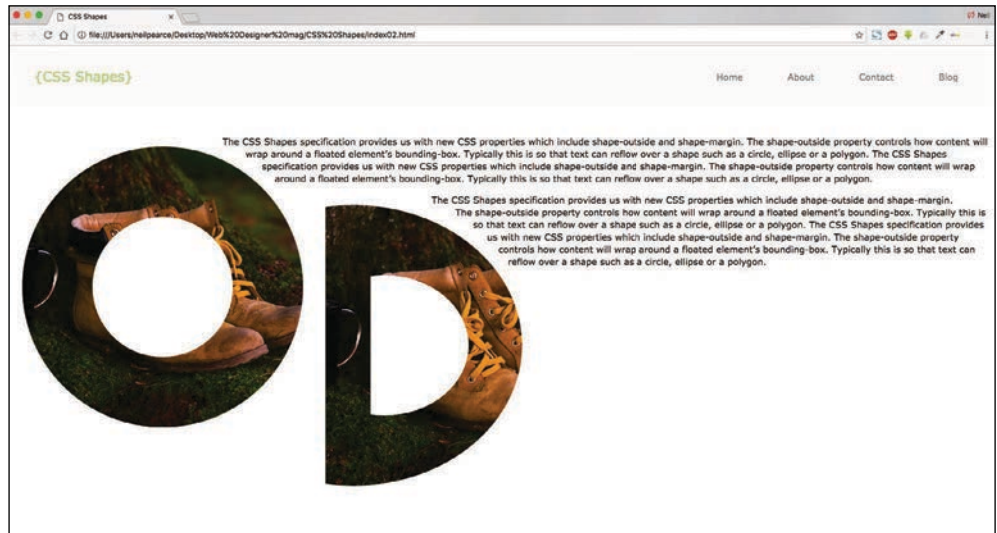


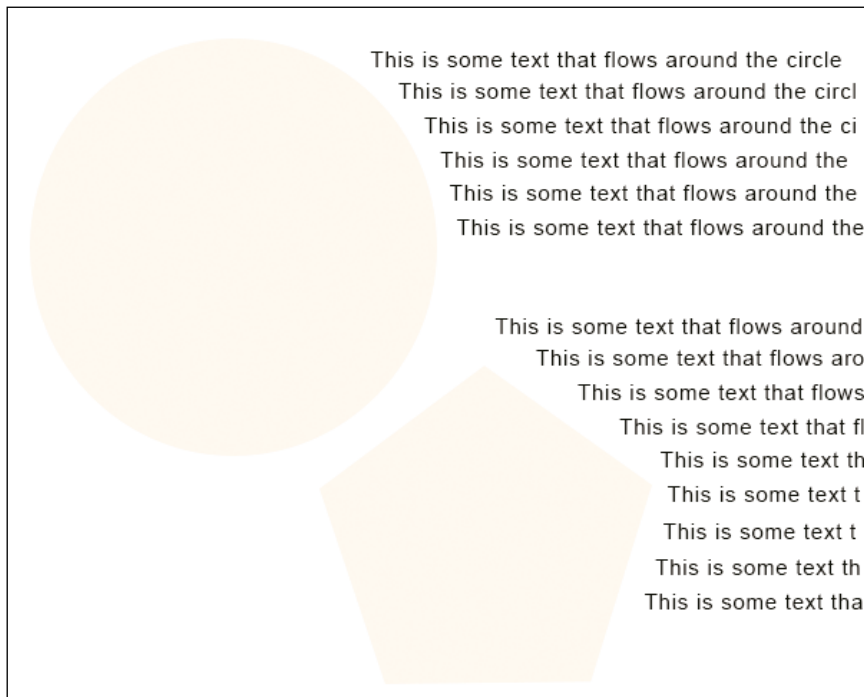
Top right

By using the polygon shape, we can create some really useful elements such as a profile image for a blockquote

Right

Creating letters such as 'O' and 'D' is pretty simple using CSS Shapes





The shape property and polygon function

Every element on a webpage is rectangular, and is seen as a rectangle by every other element on the page. So, even if you change the shape of an element - by rounding it, for example, using border-radius - the content around the element will still flow as if the element is rectangular. Using the shape-outside property, you can change that, allowing content to wrap around a specified shape, instead of having it wrap around the old rectangular one.

```
■ shape-outside: circle(50%);
```

If circles and ellipses are too limiting, the polygon shape function opens up a world of options. This function takes in a set of points that form the polygon, each point defined by x and y coordinates. The format is polygon(x1 y1, x2 y2, ...), where you specify pairs of x y coordinates for each vertex (point) of a polygon. The minimum number of pairs to specify a polygon is three, a triangle, which are separated by a comma.

```
■ polygon(0 0, 0 300px, 300px 600px);
```

Vertices are placed on the coordinate system. For responsive polygons, you can use percentage values for some or all of the coordinates.

```
■ polygon(0 0, 0 100%, 100% 100%);
```

16. The ellipse

Ellipses look like squished circles and they are defined as ellipse(rx ry at cx cy), where rx and ry are the radii for the ellipse on the X-axis and Y-axis, while cx and cy are the coordinates for the centre of the ellipse. Using the ellipse function, you have the option of creating longer arced layouts than you would using the circle() function.

```
.shape element{
  shape-outside: ellipse(150px 300px at 50%
  50%);
  width: 300px;
  height: 600px;
}
```

17. Combining shapes

In the next few steps, we will look at combining two shapes to create a text effect. The shapes we'll use will be a circle and an ellipse. The idea here is to create the letters 'O' and 'D'. Now, you might not need these particular letters, but you'll get a decent idea of what you can achieve using CSS shapes. The first thing to do is add in some additional HTML.

```
<div class="circle">
<div class="sm-circle"></div>
</div>
```

18. Circle inside a circle

Using the same circle rule we created in an earlier step, we first need to make sure this is also positioned relative. Then we will create a small white circle and by using absolute positioning, we can centre that within the bigger circle.

```
.circle {
  float: left;
```

```
position: relative;
background: url('boots.jpg') no-repeat center;
shape-outside: circle(50%);
width: 500px;
height: 500px;
background-color: #999;
clip-path: circle();
margin: 20px;
}

.sm-circle {
  background-color: #fff;
  position: absolute;
  width: 250px;
  height: 250px;
  clip-path: circle();
  top: 125px;
  left: 125px;
}
```

19. Creating the letter D

Same as before, we first need to add some additional HTML placed within the main ellipse class. Then we can go ahead and create the first ellipse, which can be shaped accordingly using the percentage values. The first two percentage values represent the height and width while the other two represent its positioning (the centre).

```
<div class="ellipse">
<div class="sm-ellipse"></div>
</div>

.ellipse
{
  position: relative;
  float: left;
  margin-left: -100px;
```

```
background: url('boots.jpg') no-repeat center;
shape-outside: ellipse(60% 50% at 10% 50%);
width: 500px;
height: 500px;
shape-margin: 10px;
clip-path: ellipse(60% 50% at 10% 50%);
background-color: #f5f5f5;
margin: 20px;
}
```

20. The Inner D

To finish off our letter effect we need to add in the small ellipse shape, which again will be placed in the centre of the large ellipse using absolute positioning. We don't need to change the percentage values since we want to create the exact same shape as the large ellipse.

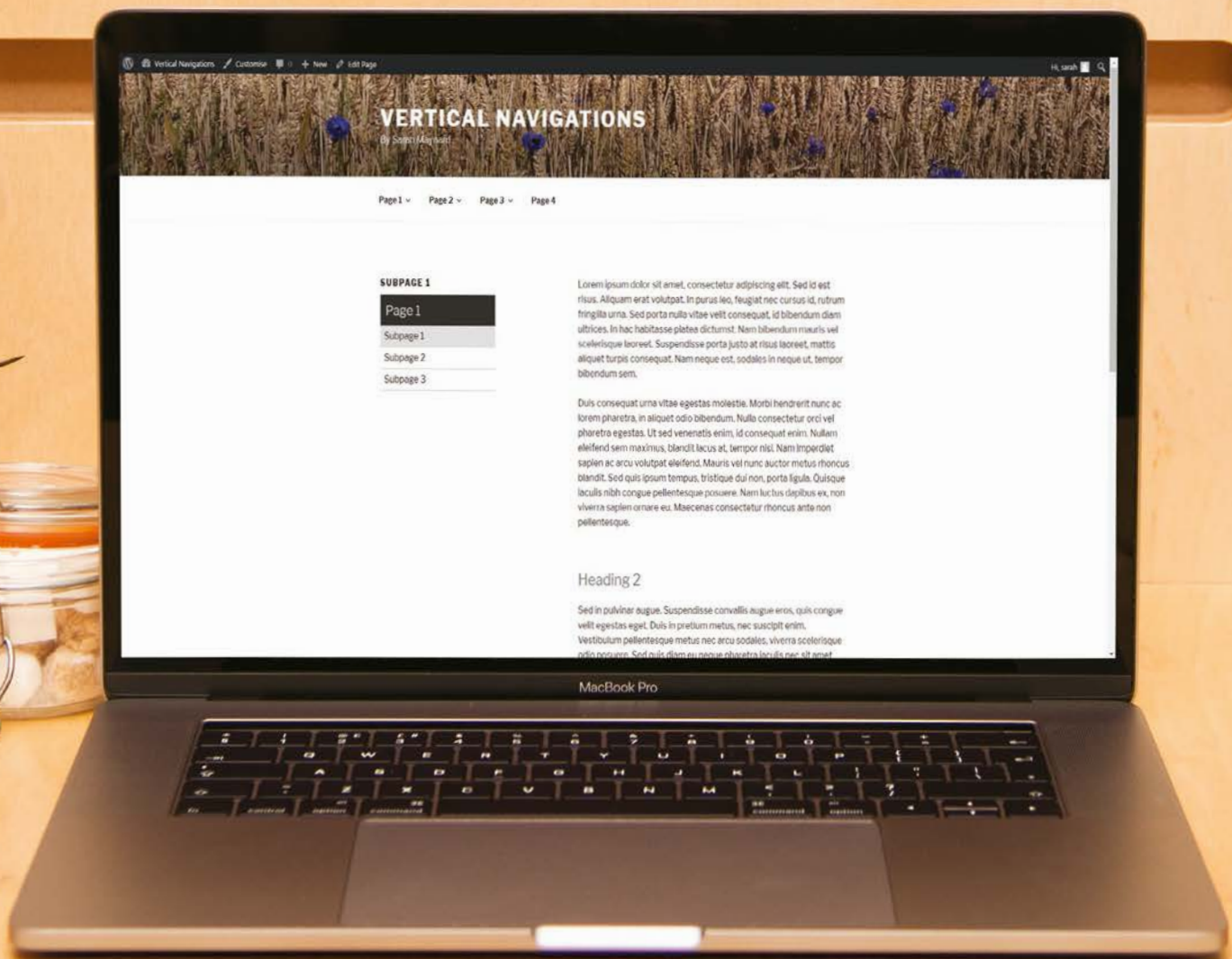
```
.sm-ellipse {
  background-color: #fff;
  position: absolute;
  width: 250px;
  height: 250px;
  clip-path: ellipse(60% 50% at 10% 50%);
  top: 125px;
  left: 80px;
}
```

21. Taking things further

In a web where content is mostly trapped in simple boxes, CSS Shapes provides us a way to break out of the box and create expressive layouts. In this tutorial we covered quite a few of the basics of CSS shapes. However, we recommend you take things further by experimenting with the polygon function to see how it can enhance the content presentation of your pages.

Code custom menus for WordPress

Content is king, and by using vertical menus you can help your readers navigate your pages quickly



There is a strong trend for horizontal navigation in modern web design. Vertical navigation was popular when the web was primarily viewed on desktop and insights into user experience were lacking. These days, a horizontal navigation system is key for primary links, but in the right circumstances vertical navigation can still pull its weight. On text-heavy sites, those with deep navigation, menus with many categories, or when design needs suit, a vertical menu in the sidebar can complement the main menu and assist readers with quick and easy navigation.

There are many ways to create submenus in WordPress, from basic (using a pre-made plugin) to the advanced (writing your own walker class). For a quicker but less PHP-orientated solution, you can use CSS instead to turn a full menu into the current page's submenu by targeting WordPress's default `current-menu-item`, `current-menu-ancestor`, and `current-menu-parent` classes.

This tutorial covers how to make a simple single-level submenu to complement your main navigation using the WordPress Menu and a pre-made function, displaying the subpages of the current page in the sidebar with no superfluous HTML. Style and enhance your new submenu with CSS and jQuery for a sleek navigational aid.

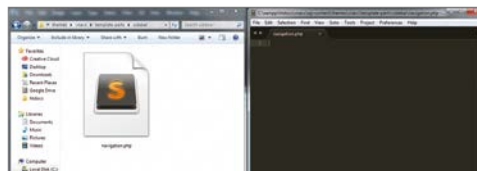
1. Setting up the menu

Use the WordPress menu options to create your menu. For this tutorial, we will be using the main menu and echoing the relevant pages into the sidebar, so you don't need to duplicate it in the admin panel. Make sure that all of your required pages are included and organised into submenus.



2. Create the menu placeholder

Create a blank PHP file in your theme, where we'll build the menu. In this example we're using a child theme of Twenty Seventeen, so our new file is saved in the theme folder under `/template-parts/sidebars/navigation.php`.



3. Include the menu placeholder

This vertical menu will be supporting the content; so, for the Twenty Seventeen theme, it would sit best beneath the page title in the left-hand sidebar. Use the code below to include your new menu file in your existing template files. Ensure the slug matches your include file path, minus the .php suffix.

```
<?php get_template_part('template-parts/
sidebar/navigation'); ?>
```

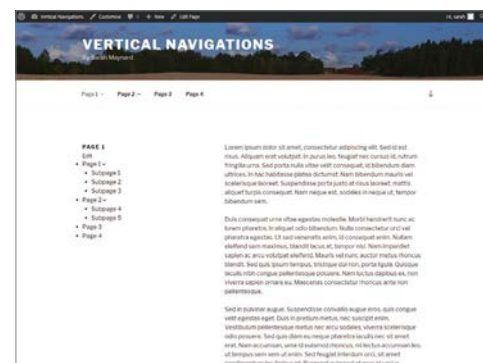
4. Include the menu code

Inside your `navigation.php`, include the default menu code. One of the simplest ways is to copy the top menu code from elsewhere in the theme. The `theme_location` will be the same as the main menu, but the `menu_id` can be unique in order to target your menu with CSS later.

```
<?php wp_nav_menu( array(
'theme_location' => 'top',
'menu_id' => 'sidebar-menu',
) ); ?>
```

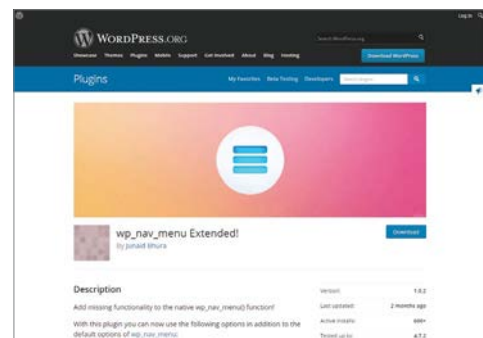
5. The default display

The menu should now appear wherever you included the template part in step 3. Depending on your theme's CSS, the menu may or may not be partially styled already. Currently it will show the whole navigation, but we can cut this down to the relevant pages later.



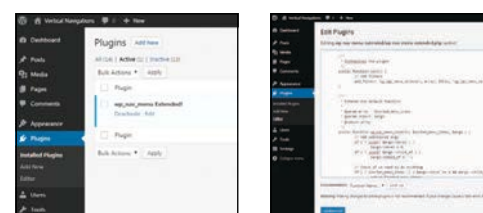
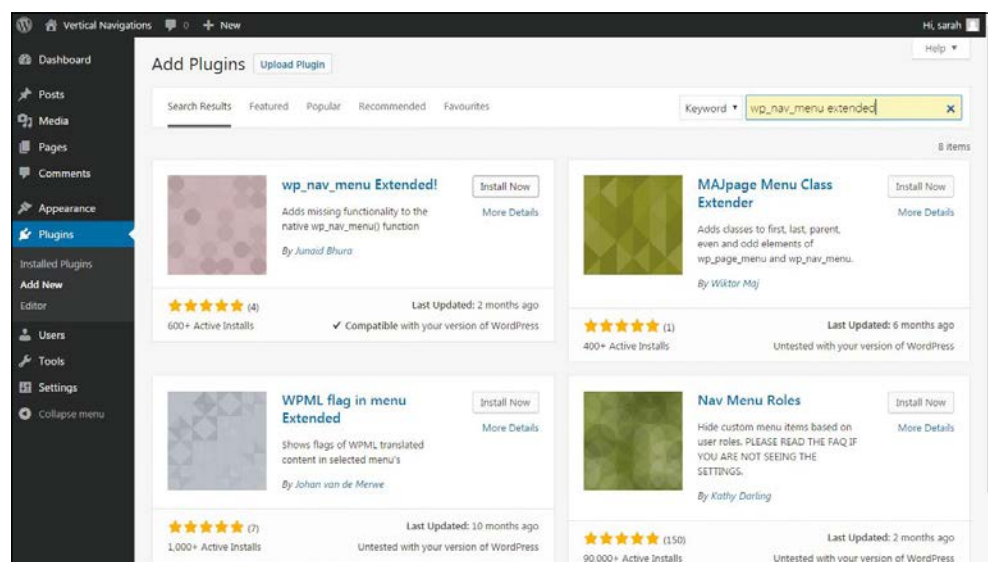
6. Install a menu filter or walker

A menu filter is a function that extends a default WordPress function. By extending the `wp_nav_menu` function, we can alter the default behaviour. This can be advanced, but for beginners there are many pre-written walkers and plugins. Install the 'wp_nav_menu Extended!' plugin by Junaid Bhura.



Additional parameters

Use other menu parameters to customise the menu as usual, adding before and after elements, wraps and classes. Check the `wp_nav_menu()` reference on [WordPress.org](https://developer.wordpress.org/reference/functions/wp_nav_menu/) for the full list.



Left

Install the 'wp_nav_menu Extended!' plugin from the Plugins Directory. Ensure the plugin is up to date and compatible with your WordPress version!

Top left

Activate the plugin. There are no other options to configure, and the alterations will be ready to use in step 7 without affecting any old code

Top right

The plugin includes a class that extends the functionality of `wp_nav_menu()`. It allows you to select a sub-level and optionally an alternate page ID

Tutorials

Code custom menus for WordPress

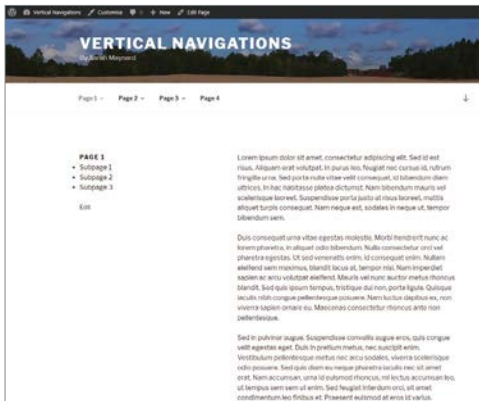
7. Update the menu arguments

With the plugin installed, new argument options can be added to the menu. Add the level parameter. A value of 2 will instruct the function to display the second level of the menu tree, and display only the subpages of the current page.

```
<?php wp_nav_menu( array(
    'theme_location'=> 'top',
    'menu_id'       => 'sidebar-menu',
    'level'         => 2
) ); ?>
```

8. The updated display

Now the display only shows the submenu for the current page. Pages with no submenu display no menu. However, this also means no menu appears on the subpages themselves as they have no children; ideally we want the siblings to be shown instead. This can be solved by updating the code in the following steps.



Menu ID vs object ID

Each page object has its own ID. Because an object may appear multiple times in a menu, each menu item also has its own ID. A menu ID references menu data; an object ID references page data.

9. Get menu ID

Firstly we need the ID of the menu to later query the list items in the next step. By looking at the ID of the menu currently assigned to the location, we will always get the correct ID even if the menu is later updated. Place this code at the top of `navigation.php`.

```
$menu_name = 'top';
$locations = get_nav_menu_locations();
$menu_id = $locations[ $menu_name ];
```

10. Find the current menu item ID

Next, we must find the current menu ID. The menu ID is separate to the page ID, and is required to query the navigation structure rather than the page parent/child structure. This allows the submenus to follow the menu order, not the page order, so you can edit the menu separately.

```
$menu = wp_get_nav_menu_items($menu_id, array(
    'posts_per_page' => -1,
    'meta_key' => '_menu_item_object_id',
    'meta_value' => $post->ID
));
$menu_item_id = $menu[0]->ID;
```

11. Find the parent menu item ID

Now that we have the current menu ID, we can use it to find the ID of the parent menu item. This is the variable we need in order to populate the subpages with the sibling menus. We look at the post meta information for the current menu ID and pull out the parent value.

```
$menu_item_parent = (int) get_post_meta(
    $menu_item_id, '_menu_item_menu_item_parent', true);
```

12. Find parent object ID

To add a link to the parent page with your submenu, utilise the `$menu_item_parent` variable to query the post `_menu_item_object_id` metadata value. This will retrieve the ID of the parent page object, which you can use to get the permalink and title in step 14.

```
$menu_object_id = (int) get_post_meta($menu_item_parent, '_menu_item_object_id', true);
```

13. Add a container

To more easily manipulate the menu, add a container that will surround both the menu and the title that will be added in the next step. This way the element as a whole can be hidden or affixed with CSS later. The element will be closed in step 15.

```
echo "<div id='sidebar-container'>";
```

14. Compare current and parent ID

If the parent menu ID returns 0 (ie, the menu item has no parent/is the top-level item), then we set the parent variable back to the current menu ID. This returns the submenu for the current page rather than the submenu for the nonexistent parent. If there is a parent, echo the parent link.

```
if ($menu_item_parent == 0) {
    $menu_item_parent = $menu_item_id;
} else {
    echo "<h3><a href='".get_the_permalink($menu_object_id)."'>".get_the_title($menu_object_id)."</h3>";
}
```

15. Update the menu arguments

Now we have the menu ID for parent pages and the parent menu ID for subpages, we can update the menu arguments using the parameters added by the 'wp_nav_menu Extended!' plugin. This instructs the menu to retrieve the correct submenu to display. Add the closing div tag to complete the container.

```
wp_nav_menu( array(
    'theme_location' => 'top',
    'menu_id'       => 'sidebar-menu',
    'level'         => 2,
    'child_of'      => $menu_item_parent
) );
```



Top left

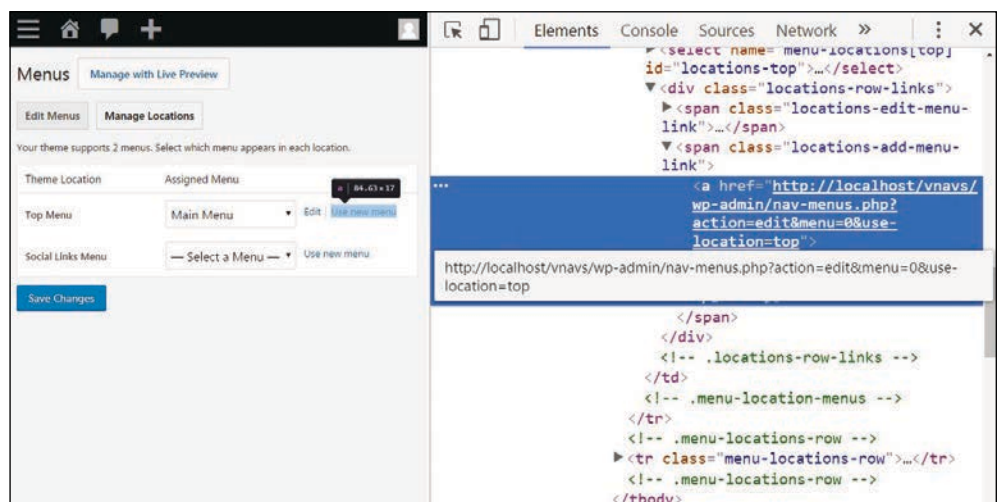
You can usually find the `theme_location` attribute of a menu by looking in the theme's `functions.php`. The key is the value to insert as `theme_location`

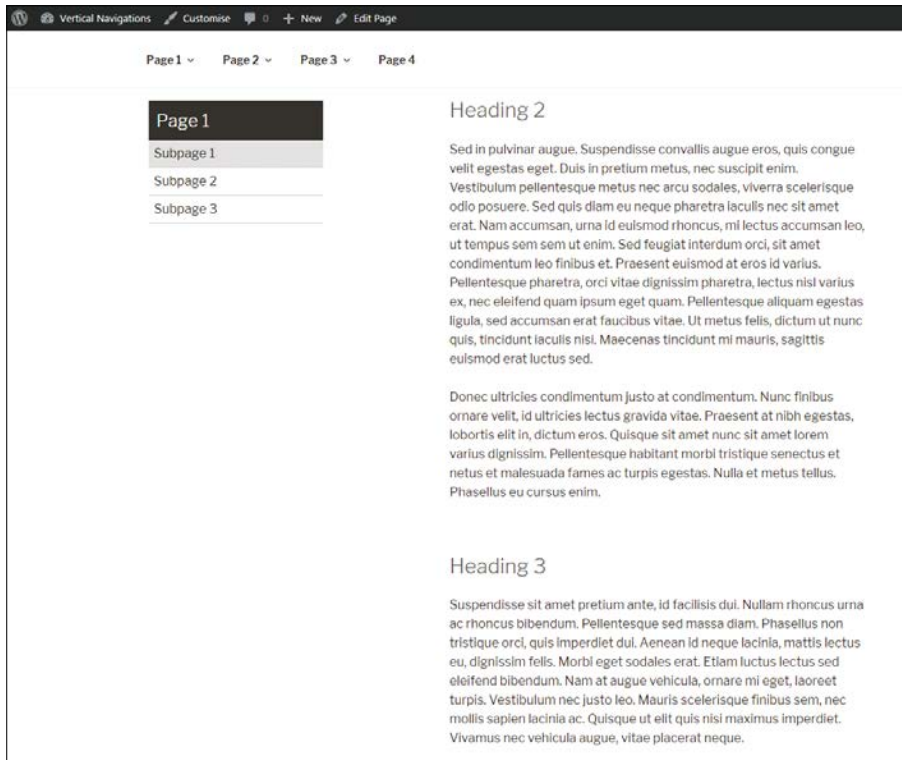
Top right

You can also find the `theme_location` if the theme already includes the menu in a similar way, by finding where the menu code was originally included

Right

Alternatively, find it by going to `Menus > Manage Locations` and inspecting the 'Use New Menu' link for the 'use-location=xxx' parameter





jQuery for sticky menus

Add this jQuery by enqueueing via the **functions.php** or pasting into **footer.php**. It'll add the stickymenu class to the container when it reaches the top.

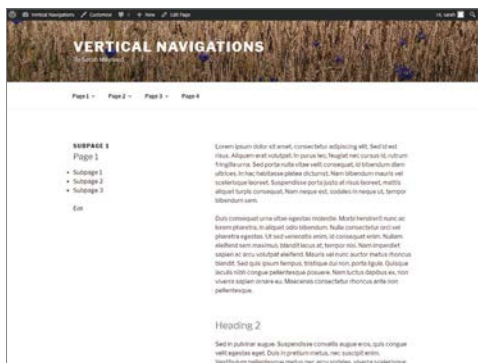
```
<script type="text/javascript">
(function($) {
var stickyHeaderTop = $('#sidebar-
container').offset().top;
$(window).scroll(function(){
if( $(window).scrollTop() >
(stickyHeaderTop - 100) ) {
$('#sidebar-container').
addClass("stickymenu");
} else {
$('#sidebar-container').
removeClass("stickymenu");
}
});
})( jQuery );
</script>
```

Change #sidebar-container to your container's ID. (stickyHeaderTop - 100) adds a buffer so the changes happen when the menu gets within 100px of the top. Use CSS in step 19 to style the menu while it's sticky.

echo "</div>";

16. The updated display

The modifications we just made will show the sibling menu while on child pages, and the subpage menu while on parent pages. For pages with no submenu, they will still have a blank sidebar. The parent link will also appear on subpages so the user can easily navigate back to the top.



17. Style the menu

In your child theme's CSS file, add styling to the new menu. The CSS below removes the bullet points, adds a border between items and increases the click area of the link to the whole element while adding some much needed spacing.

```
#sidebar-menu {
list-style: none;
margin: 10px 0; }
#sidebar-menu li {
```

```
border-bottom: 1px solid #ccc; }
#sidebar-menu a {
display: block;
width: 100%;
padding: 5px 7px; }
```

18. Hover and current page classes

Use link hover classes and WordPress's default current page classes to add active states. The CSS below highlights the page in the menu that you are currently on using .current_page_item, and underlines the link when you hover.

```
#sidebar-menu li.current_page_item a {
background: #ddd;
}
#sidebar-menu a:hover {
background: #eee;
text-decoration: underline;
}
```

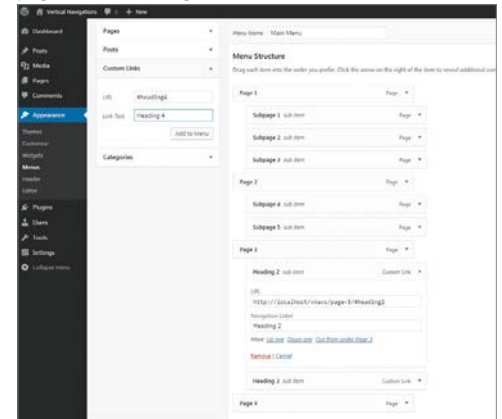
19. Add a sticky menu

Use JavaScript or jQuery to add a class to your menu that scrolls off the top of the screen, then use CSS to apply a 'position: fixed;' rule to it. This CSS will keep the side menu on the page as the user scrolls through the content and ensures it doesn't overlap with the top sticky menu.

```
#sidebar-container.stickymenu {
position: fixed;
top: 110px;
}
```

20. Use link anchors

If your page has long content instead of subpages, use the Custom Links in the menu options to add menu anchors to predefined heading IDs. The menu will then display the anchors to which you can jump, making long pages easier to navigate.



21. Hide on mobile

Use media queries to alter your sidebar menu on smaller screens. The breakpoint for small screens is 767px in the Twenty Seventeen theme, so the media query below will hide the submenu on smaller screens as its functionality is replaced by the burger menu.

```
@media (max-width: 767px) {
#sidebar-container {
display: none;
}
}
```


web workshop

How to create an animated transition

As seen on hlebdom.ru/en

Responding title

This title changes in response to how the user interacts with the content menu.

Presentation response

The hiding and revealing of content in response to user interactions can be defined through JavaScript or the use of the CSS target selector.

Menu interactions

An interactive menu allows additional content to be shown upon user selection. This allows the page to appear uncluttered.



How to create an animated transition

↓ **DOWNLOAD TUTORIAL FILES** www.filesilo.co.uk/webdesigner

Animated content

The animation effect is applied to this content area. The illustration is introduced through a 'puff of smoke' effect.



he use of animation to introduce content is an opportunity for you to enhance the usability experience for your website.

Whether it's part of a call to action, or just to influence the user's perception, this use of animation is a tool to guide the user's attention to content of interest.

Our workshop focuses on the use of animation to introduce the main page content. However, the approach makes use of a class name that can be applied to specify which content is to have the animation effect. This allows you to apply the effect to specific parts of the webpage.

Although we use this effect to introduce the page content, there are many other uses that the effect can be creatively applied to. For example, consider how additional content can be introduced in reaction to an event such as the user selecting a form option or clicking a button. Not only does this allow you to draw the user's attention to an event's associated content, but it also acts as a feature to present content to users 'on demand'. This avoids your content-laden webpages appearing to be cluttered and confusing, thereby leading to an improved user experience and outcomes.



A tool for UX designers too

"The use of animation to introduce content can be strategically applied to draw attention and limit visible content to what the user needs at a specific time.

Designing your website to present information in context to a user's interactions allows content to be more easily understood."

Leon Brown, Freelance web developer and trainer

<comment>
What our
experts think
of the site

Technique

1. Webpage

The first step is to create the webpage. This example is fairly simple, consisting of just the document template and the visible article container. The document head contains links to the external CSS file, while the body contains the visible content. The article container has two classes - intro and splash; the latter is used to apply the transition effect.

2. CSS initiation

The HTML is now complete, so the next stage is to create a file called **styles.css** to store the visual formatting. Start this file with rules to make the HTML container and its content body cover the full visible window, along with default background colour and font settings.

```
html, body {
  display: block;
  width: 100%;
  height: 100%;
  margin: 0;
  padding: 0;
  font-family: Helvetica, sans-serif;
  background: #000;
  color: #fff;
}
```

3. Content container

Our example HTML uses the 'intro' class to identify where the unique page content is placed. For simplicity, we will use this with a background image and set it to 100% width and height of its parent container - the main document body. These settings allow you to place this content container somewhere else, upon which it will resize itself to fit wherever it is placed within.

4. Splash class

Elements that are to have the splash transition effect applied to them will have the 'splash' class applied to

them. This class applies margin settings to ensure that the splash container is positioned in the centre of its parent during its animated changes. References to the animations to use are also defined; in this case, 'splash1' is applied for half a second, while 'splash2' is also applied for one whole second.

```
.splash {
  animation: splash1 .5s, splash2 1s;
  margin: 0 auto 0 auto;
}
```

5. First animation settings

The transition effect is made from a combination of two animations. The first animation defines the transitions for opacity transparency, width and border radius, to make the content appear to be fading and zooming in. The use of border radius from 100% to 0 makes the selected container appear to fill the space in which it has been placed.

```
@keyframes splash1 {
  from {
    opacity: 0;
    width: 0;
    border-radius: 100%;
  }
  to {
    opacity: 1;
    width: 100%;
    border-radius: 0%;
    border-radius: 0;
  }
}
```

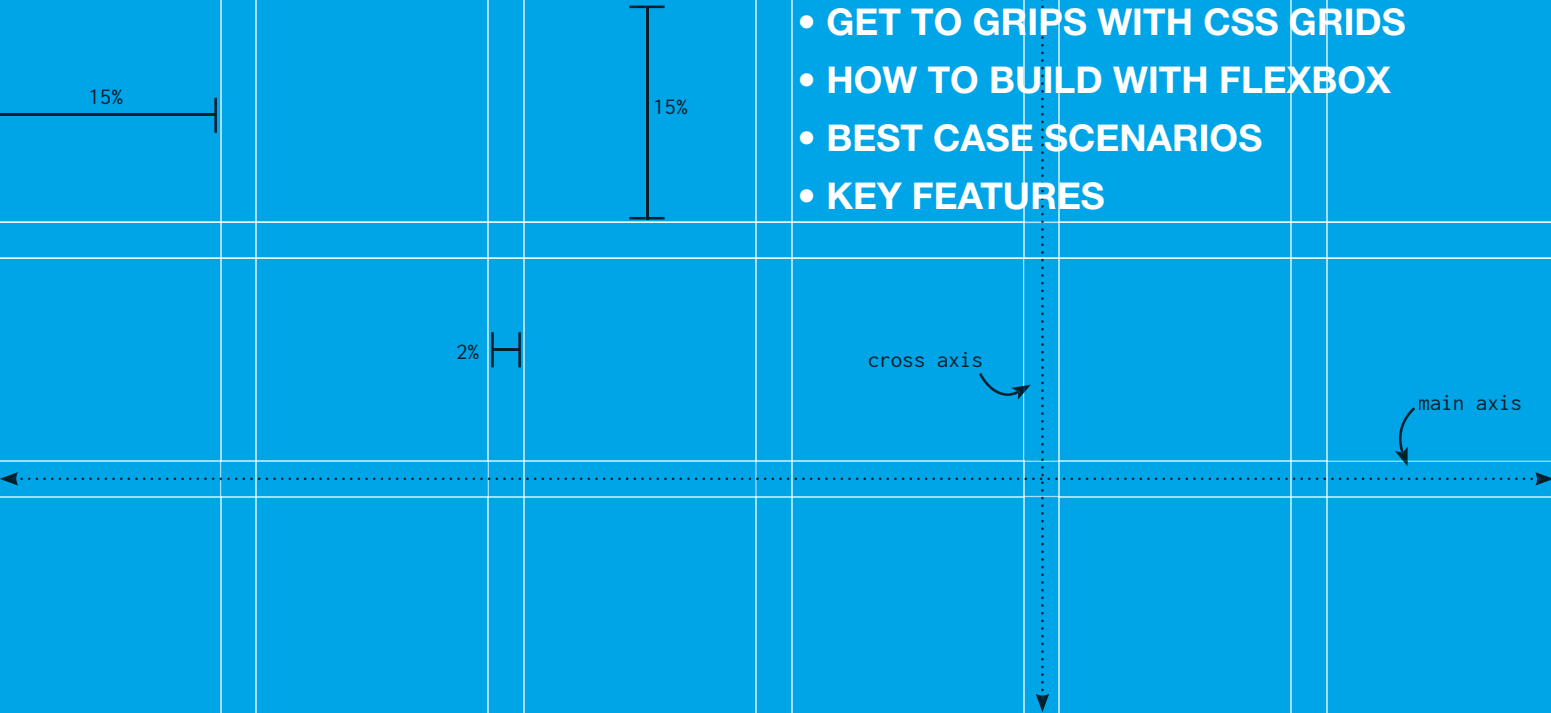
6. Second animation settings

The second animation settings are used to animate the blur filter to make the transition effect appear to make the content come into focus. Be aware that you should add browser vendor versions of the filter attribute before the official 'filter' to ensure compatibility with all browsers. For example, -moz-filter, -webkit-filter, etc.

CSS LAYOUTS

WE DO THE MATHS
YOU DO THE DESIGN

- GET TO GRIPS WITH CSS GRIDS
- HOW TO BUILD WITH FLEXBOX
- BEST CASE SCENARIOS
- KEY FEATURES



Website design has taken some pretty huge leaps in its short two-decade history, starting with the HTML 2.0 specification back in 1995. Along with it came the ability to create hierarchy and meta data in webpages, as well as tags used specifically for styling. A year later, CSS was created to try to solve the problem of complicated markup and table contorting that the community had begun using to create complex layouts and designs.

Fast-forward to the late 2000s and CSS3 is introduced, bringing along with it new layout modules among other improvements. These layout modules have been specifically designed for the state of the web as it is now and with the future in mind. These are the Flexible Box Layout (Flexbox) and the Grid Layout Module (CSS Grid) as well as Media Queries. Implementation of CSS3 into browsers has been different to previous versions, allowing for certain modules to be implemented one at a time instead of all at once. This is why, although CSS3 has been around for a decade now, the layout modules have had varying browser support and implementations.

Flexbox and CSS Grid are supported by an algorithm allowing the browser to calculate the width of every element inside the container and allocate the leftover space, either evenly or fractionally on a per element basis, something we've been dreaming of and trying to hack together for years!

These new tools allow for complicated layouts that can fit any and every screen size when used correctly. We can also create fairly complicated layouts which will fit the content, even when we don't know how much content or how many items we'll be placing into the page. This means writing far less code to create beautiful and interesting layouts and designs that can be reused and repurposed easily.



LAYOUT IN CSS HAS ALWAYS BEEN A TRICKY BEAST TO TAME, MUCH LIKE CSS ITSELF, IN THAT YOU CAN UNDERSTAND THE BASICS AND GET UP AND RUNNING IN HALF AN HOUR BUT MASTERING ITS INTRICACIES AND QUIRKS TAKES YEARS OF PRACTICE. FLEXBOX & GRID AIM TO MAKE THIS MUCH EASIER.

Steven Roberts

Lead Frontend Developer, Better Brand Agency
codepen.io/matchboxhero, @matchboxhero10

FLEXBOX VS CSS GRID

Flexbox is a CSS layout module which allows for flexible one-dimensional, single-axis layouts which automatically adjust to the browser's width and height. Flexbox also allows for the reordering of content, which is particularly useful when combined with media queries to order elements differently for mobile and desktop.

PROS

- Flexbox can be used to create responsive layout elements without the use of media queries.
- The flex-basis property combined with the ability to nest flex containers allows for complicated layouts without the need for changes to the HTML.

CONS

- The new way of thinking about positioning and layout is completely different from previous methods and can take some getting used to.
- Applying properties to both the parent and child elements is new and could prove confusing at first.

CSS Grid (the specification, not to be confused with grid systems such as those found in bootstrap and other frameworks) is another CSS layout module which allows for flexible or rigid two-dimensional, two-axis layouts. CSS Grid allows for absolute control with powerful positioning tools and functions built right into CSS.

PROS

- CSS Grid's new units, functions and templating properties are powerful and allow for elegantly written selectors as well as overlapping content if needed.
- CSS Grid template areas allow for named areas on the grid, reducing confusion when working with complex layouts.

CONS

- Browser support is minimal at the moment, but the specification became a W3C Candidate Recommendation on 29 September 2016.
- Using functions in CSS, if you're not accustomed to SASS or LESS, could prove confusing at first.

FLEXBOX

IN FLEXBOX YOU APPLY THE 'DISPLAY: FLEX;' PROPERTY TO AN ELEMENT AND ALL DIRECT CHILDREN BECOME FLEX-ITEMS OF THEIR PARENT, THE FLEX-CONTAINER

5 KEY FEATURES OF FLEXBOX

The flex-direction property

The flex-direction property allows you to change the direction of the axis by rotating the flex-container by 90 degrees.

Simple vertical and horizontal centring

This is something we've dreamed about for years: the ability to vertically centre elements with one line of code. Just one. Finally! `align-items: center;`

Flexbox sizing

Flexbox is supported by an algorithm, allowing the browser to calculate the space occupied by the flex-items as well as the space left over.

Flex-basis

Flex-basis is one of the properties you can apply to the flex-items which will allow you to base each item on a given width, using `flex-grow` and `flex-shrink`; you can also choose what happens with the leftover space, if there is any.

Order

Flexbox has a trick up its sleeve: the `order` property. Combining this with media-queries allows for the reordering of UI elements for different screen sizes.

THE FLEX-DIRECTION PROPERTY

When the display property is set to flex, the default value for flex-direction is row. This property has a number of different settings you can choose from.

By default, Flexbox will align all the flex-items horizontally, dynamically and equally using all of the available horizontal space. This is because of the default value of 1 set by flex-grow.

The value column will rotate the axis of the flex-container, 90 degrees, aligning the flex-items vertically. Other values include `row-reverse` and `column-reverse`, which are fairly self-explanatory.

FLEX-BASIS

Using the property flex-basis allows you to specify an ideal width. Imagine we have a flex-container with six items inside of it, each just a box with a number inside. If we set a flex-basis of 500px on each of the flex-items, but the viewport width is only 1000px wide, you would expect that either the viewport would scroll horizontally or the items would wrap onto the next line.

This isn't what happens - instead the items are spread evenly across the viewport. The reason for this is the default value for flex-grow being 0, meaning it cannot get any bigger, and the default

FLEXBOX SIZING

When the display: flex; property is applied to the parent, the children of this element will become flex-items. Each of these flex-items will be as wide as its contents, unless the contents are wider than the available space in the flex-container - then, the flex-items will be shrunk in ratio to their original size, calculated as if the flex-container width was infinite.

This is the default behaviour, but can be changed using the shorthand flex property or flex-grow, flex-shrink and flex-basis properties.

The default values would be written as:

```
flex-grow: 0; flex-shrink: 1; flex-basis: auto;
/* or */ flex: 0 1 auto;
```

value for flex-shrink being 1, meaning that it can shrink.

```
.flex-container {
  display: flex;
}
.flex-item {
  flex-grow: 0;
  flex-shrink: 1;
  flex-basis: 500px;
/* or */
  flex: 0 1 500px;
}
```

BEST CASE SCENARIOS

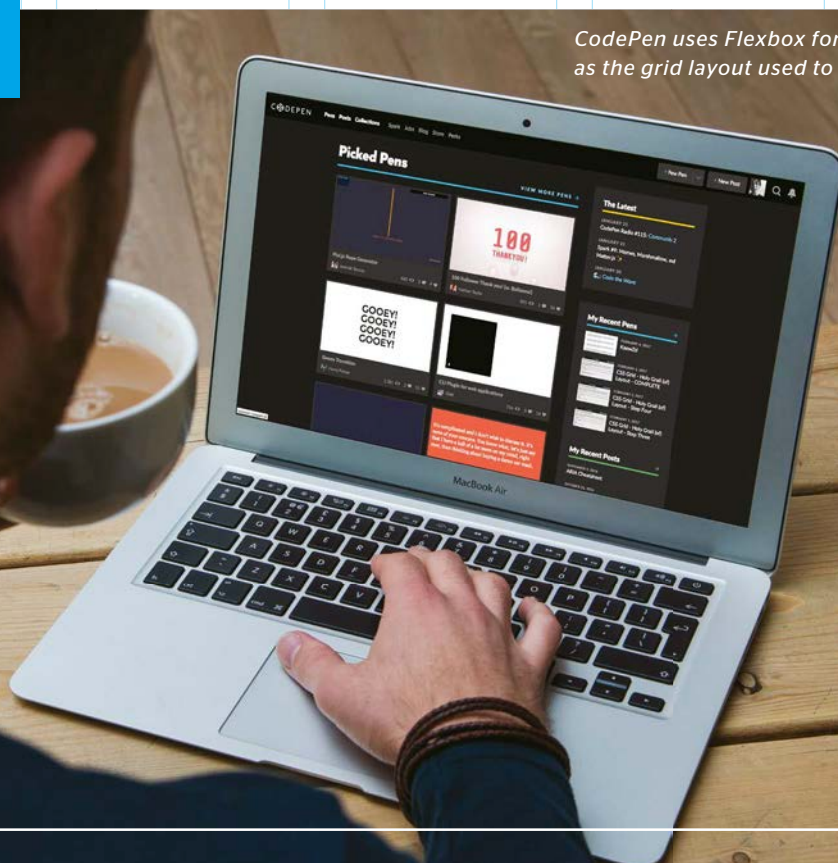
Website and app headers

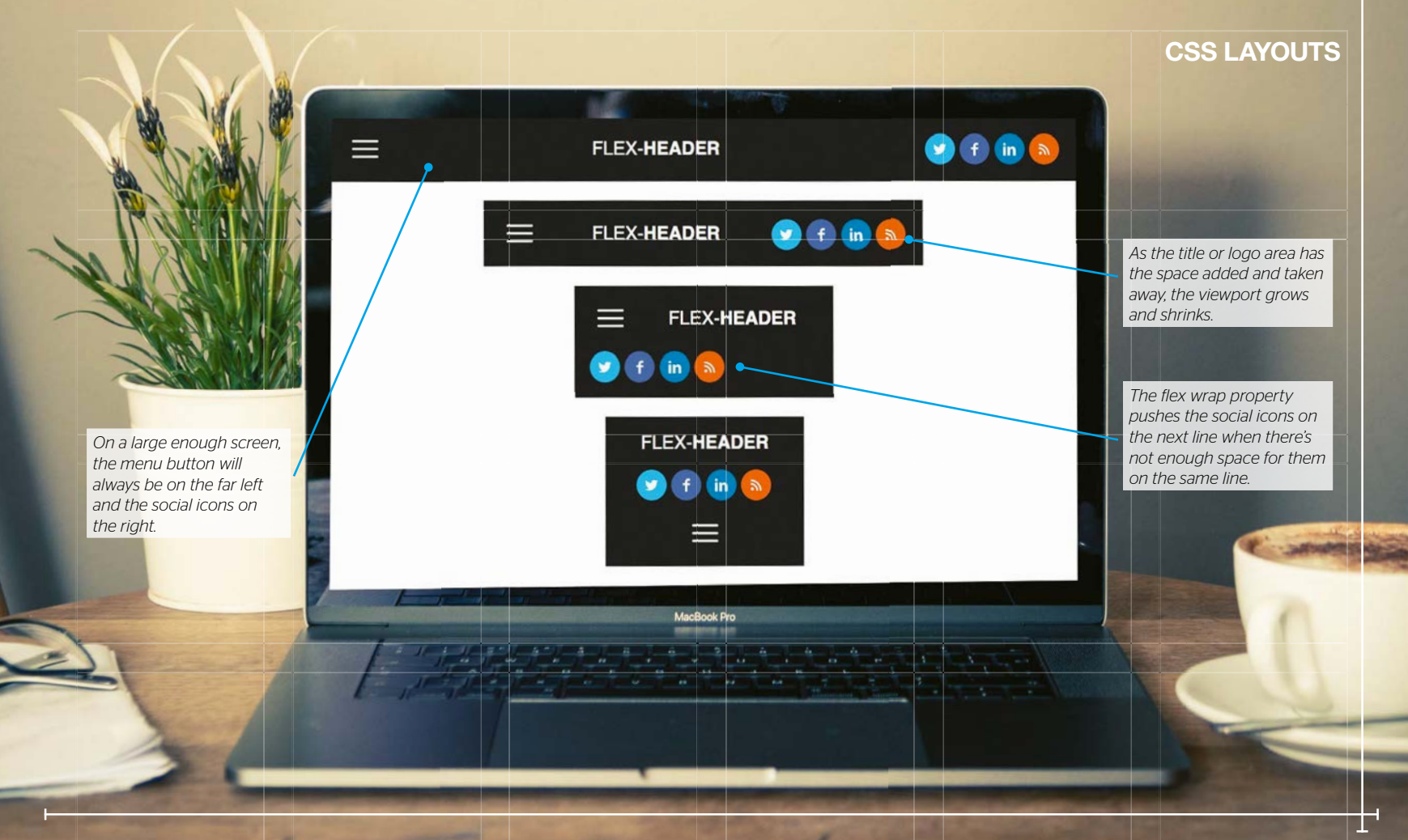
Website and app headers are one of the places Flexbox shines, giving us the ability to have all items just 'fit', as well as ordering the content differently for different screen sizes.

Dynamic content

When dealing with dynamic content, we don't always know the length of content before we insert it into the page. Flexbox allows us to apply a set of rules for the content to follow.

CodePen uses Flexbox for the header as well as the grid layout used to display the pens.





TUTORIAL: BUILD A HEADER WITH FLEX

This tutorial will show you the basics of Flexbox by utilising the `flex` property and how to use the other new properties to allow elements to grow, shrink and stack to fit the available space.

01. The HTML

The HTML is really simple. We have a header element (the flex-container) with three divs inside (the flex-items): one for the logo, one for the menu button and the other for social icons. See: bit.ly/2kc24D5.

```
<header class="flex-container">
  <div class="flex-item"> LOGO/TITLE </div>
  <div class="flex-item"> SOCIAL ICONS </div>
  <div class="flex-item"> MENU BUTTON </div>
</header>
```

02. Get flexy!

Here we apply the `display: flex;` property to the header, which initialises Flexbox, allowing us to apply other Flexbox properties. The `flex-wrap: wrap;` property is then applied, which will (when needed) wrap elements onto the next line. Next, the `justify-content: space-between;` line allocates the leftover horizontal space between all three elements. The final property we're applying is `align-items: baseline;` - this will vertically align all of the items text to the baseline. Scaling the browser

window up and down will reveal the magic. See: bit.ly/2kPA5LI.

03. Mobile first

Here we're adding `flex-direction: column;` to the flex-container to change the axis, which we will change later with a media-query, along with aligning the text to the centre. The flex-items now get some CSS, including the shorthand flex property which will set them all to grow, not shrink, and be as wide as they need to be.

```
.flex-item {
  flex: 1 0 auto;
  width: 100%;
  padding: 12px; }
```

04. Media queries

Now that the mobile layout is pretty solid, let's look to change the design as the viewport has more space to use; we'll do this using media-queries. First, we add a media query to the flex-container (the header), of around the 400px mark, to change the flex-direction back to default. Within that same media query, we set the text-alignment to right, as on larger screens we want two out of the three elements aligned to the right. Then, we apply the

same media-query to the children and reset the width back to auto - we were only using 100% for the purposes of applying the centring. See: bit.ly/2kWCzKJ.

05. nth-child

Using nth-child, we can select certain children of the flex-container and apply ordering and other rules to affect how the element behaves. Inside of the media-query, we're applying ordering and flex shorthand rules. You could do this using classes instead.

Applying the `flex: 0 0 auto` property on everything except the logo-container allows it to stretch while keeping the menu and social icons stuck to either side at their automatic sizes.

See: bit.ly/2IOHqtJ.

```
@media (min-width: 400px) {
  .flex-item:nth-child(1) {
    order: 2;
    text-align: center; }
  .flex-item:nth-child(2) {
    order: 3; }
  .flex-item:nth-child(2), .flex-item:nth-child(3) {flex: 0 0 auto; }
```

AUTOPREFIXER
Autoprefixer allows you to provide support for as many browsers as possible by simply pasting your code into the website. Check it out here: bit.ly/2ejNcjs

CSS GRID

THIS NEW LAYOUT MODULE ALLOWS US TO DESCRIBE PAGE STRUCTURE IN CSS INSTEAD OF HTML

5 KEY FEATURES OF CSS GRIDS

The fr unit

fr, short for fraction, allows grid templates to be measured in fractions, with or without a gap or margin between items. Something Flexbox is sadly missing.

The minmax() function

Functions in CSS! They're putting that algorithm to good use! The minmax() function does pretty much what you would expect.

Grid auto-flow

Auto-flow properties allow for complicated layouts to be created easily, especially when working with dynamic content.

Grid placement

CSS Grid allows for any element to be prioritised and positioned anywhere on the grid before all other items, which are then positioned automatically.

Grid areas and templates

Named areas on the grid are defined, which can then be referenced to define the layout in an almost ASCII fashion.

THE MINMAX() FUNCTION

The minmax() function is something we wish we'd had since the inception of responsive design. It allows us to apply a minimum and maximum width or height to either a column or a row. It works by setting two values inside the function: the first is the minimum value, followed by a comma and then the maximum value. You can also use the keywords min-content and max-content.

WORKING TOGETHER

These two properties work together, making it really simple to create complex layouts. You start by naming an element in the CSS with the grid-area property. This is customisable: you can choose any name, but it must not contain any spaces. Those named areas can then be used to

GRID AUTO-FLOW

Grid auto-flow properties will allow items to wrap onto the next row in the grid if the space is not available on the current row, leaving a gap in the grid to preserve the flow of the DOM.

The new grid-auto-flow allows the grid to remove elements from their position in normal flow to position them onto the grid in the spaces left over from previous overflows.

define templates for the grid to lay the content onto using the grid-template-areas property. It's worth noting that you can have the grid-template-areas span multiple lines in your CSS to represent the rows, allowing you to literally, ASCII style, describe the layout of your grid. See tutorial.

“YOU CAN HAVE THE GRID-TEMPLATE-AREAS SPAN MULTIPLE LINES IN YOUR CSS TO REPRESENT ROWS”

BEST CASE SCENARIOS

Dashboards

Admin dashboards and CMS backends would be perfect to build in CSS Grid! This enables you to name and fix areas on the grid, removing a lot of the hard work in creating these complex layouts.

Dynamic content

Grid's placement tools allow for two sets of dynamic content to be scattered across the grid without having to create complicated arrays with the data beforehand.

The WordPress admin could be built using CSS Grid, given that it has clearly defined grid areas.

TUTORIAL: CREATE A COMPLEX LAYOUT

The idea here is to build the holy grail layout in Grid, but obviously it needs to be responsive. The code to accomplish this complex layout is pretty short and really easy to follow and understand. We've also got a little bonus at the end, since Grid makes this layout so easy to accomplish!

01. The HTML

As always, you can't style elements that don't exist. So we have a containing `div` with the class of `grid` and inside that the five elements we need to create our holy grail layout: `<header>`, `<article>`, `<aside>` $\times 2$ and `<footer>`. We've then added the class of `grid-item` to each of the elements and added a class to the sidebars to differentiate them in the CSS. Drop in the header from the last tutorial if you like, too: Grid and Flexbox get along just fine.

codepen.io/matchboxhero/pen/pRLwJY

02. Named Areas

Jumping into the CSS, we're going to start naming our elements to create a grid template. Mobile first, we will make our grid template. It's worth noting that you can have the `grid-template-areas` property span multiple lines in your CSS to represent the rows, allowing you to literally (ASCII style) describe the layout of your grid.

```
.grid {
  display: grid;
  grid-template-areas:
    "header"
    "article"
    "left-sidebar"
    "right-sidebar"
    "footer"
  ;
  grid-template-columns: 1fr; }
```

As you can see, we have described - in our template - the layout we want the grid to use. We have also defined how many columns we want; by setting it to `1fr`, we're saying there is one column and it should span the whole available width.

codepen.io/matchboxhero/pen/ggeRPv

03. Responsive

The next step is to start adding media queries to redefine the grid template at a certain breakpoint. Since we want the layout to now have two columns in the middle, we need to define these. At this breakpoint, we want the header to span the entire first row and the footer the last;

```
@media (min-width: 540px) {
  .grid {
    grid-template-areas:
      "header header"
      "left-sidebar article"
      "right-sidebar right-sidebar"
      "footer footer"
  }
```

```
;
grid-template-columns: 1fr 2fr;
} }
```

By repeating the name of the area, Grid knows the element should span two columns; you can also do the same thing with rows. The left sidebar and article are being split into thirds, one third being applied to the sidebar and two thirds the article.

codepen.io/matchboxhero/pen/VPXWmW

04. It really is this easy

Now we repeat the previous step but change some of the values, so at a larger breakpoint the article and sidebars will all be on the same row.

```
@media (min-width: 1200px) {
  .grid {
    grid-template-areas:
      "header header header"
      "left-sidebar article right-
      sidebar"
      "footer footer footer"
    ;
    grid-template-columns: 1fr 3fr 1fr;
  } }
```

The `grid-template-areas` property can override the order of the DOM, allowing for any layout you can dream up and the ability to change and reorder it, every 50px if you want to. Please don't,

but you could.

codepen.io/matchboxhero/pen/KaoqGK

05. Bonus round

That's it, the holy grail layout built in Grid with two simple media-queries. Just as a bonus, let's make the footer stick to the bottom of the page when there is not enough content to fill the viewport. We

accomplish this by first applying a minimum height to the grid of the viewport height.

That works, but the header and footer are stretched vertically to fill the leftover space. We can use the `minmax()` function to stop the header and footer from expanding by defining the grid's rows.

```
.grid {
  min-height: 100vh;
  height: 100vh; /* required by
  chrome */
  grid-template-rows:
    minmax(min-content, max-content)
    auto
    minmax(min-content, max-content)
  ; }
```

Using the `min-content` and `max-content` keyword values inside the `minmax()` function, we've told the header and footer not to grow into the available space, allowing the middle row to use all of the extra space.

GRID BY EXAMPLE!

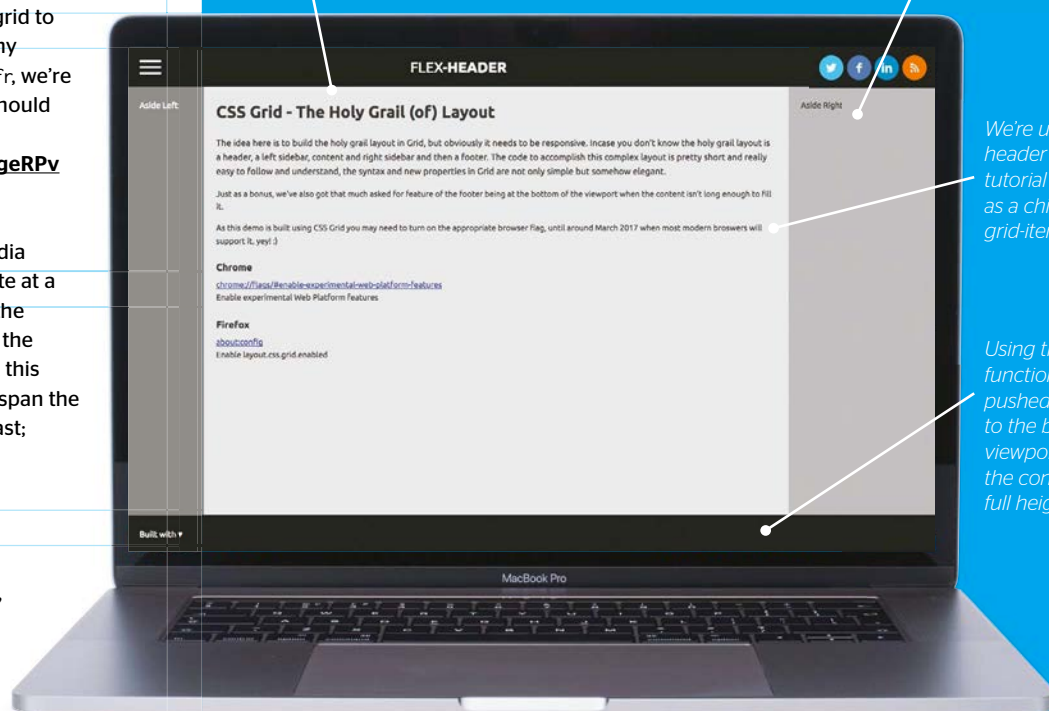
Rachel Andrew's Grid by Example site has some great videos teaching you everything you need to get started with CSS Grid <http://bit.ly/2kFBcfZ>

We have described the page layout using an ASCII art fashion: `grid-template-areas: "header header header" "left-sidebar article right-sidebar" "footer footer footer";`

Both of the sidebars will be as tall as the row they belong to, as we're not using any float or position hacks.

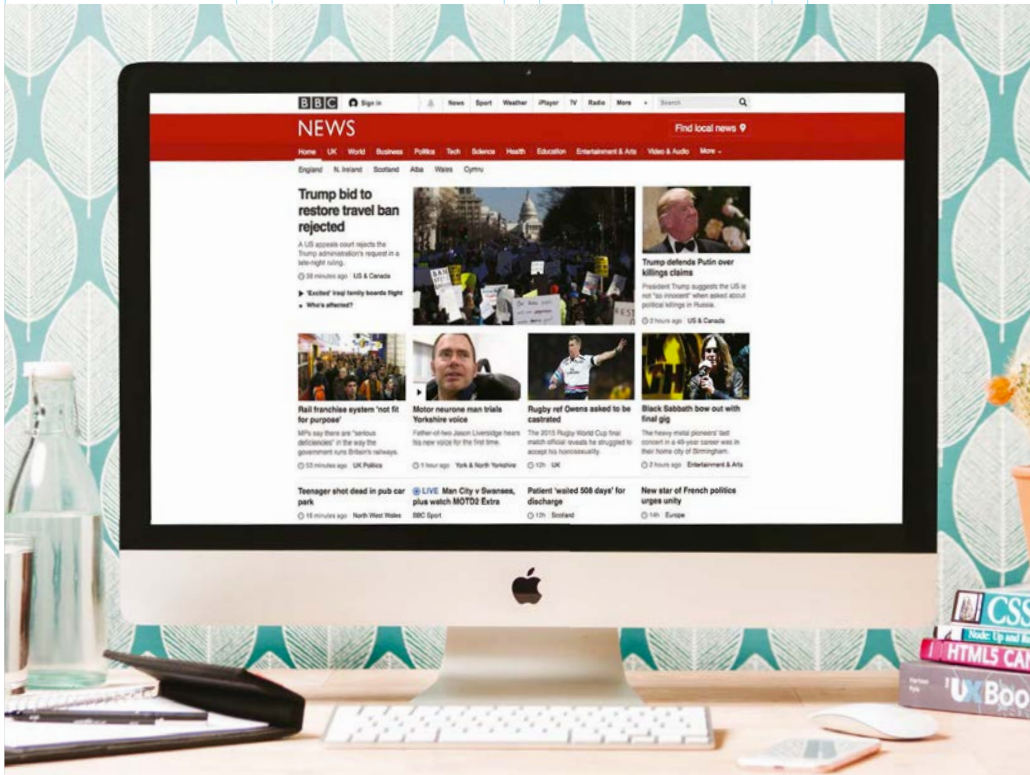
We're using the flex-header from the last tutorial by placing it as a child of the first grid-item.

Using the `minmax()` function, we've pushed the footer to the bottom of the viewport, even though the content isn't the full height.



LAYOUTS IN ACTION

HERE ARE A FEW EXAMPLES OF REAL-WORLD WEBSITES WHICH ARE USING THESE TECHNIQUES TO CREATE INTERESTING AND BEAUTIFUL DESIGNS



BBC NEWS Flexbox

The BBC uses Flexbox throughout its news and other sites to achieve the grid of stories. The BBC needs to provide support to as many users as possible, so it employs fallbacks allowing the site to

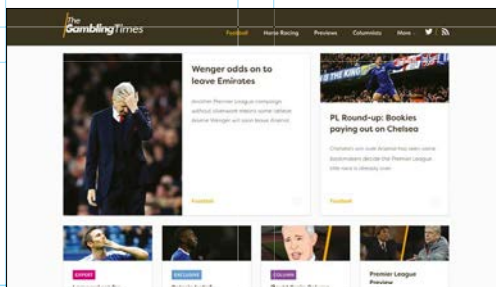
still appear functional in older browsers. It has open-sourced its code, which can be downloaded from GitHub (github.com/bbc/gel-grid) and used freely in your projects.



THE EXPERIMENTAL LAYOUT LAB OF JEN SIMMONS

Grid

labs.jensimmons.com utilises a really interesting and amazing use of Grid. The layout adapts and changes depending on the size of the viewport. Transforming the grid by rotating it -45 degrees gives the page a really interesting art direction. This is only the beginning of what's to come once designers and developers become familiar with this new layout module.



THE GAMBLING TIMES

Flexbox

TheGamblingTimes.com uses Flexbox for almost every aspect of layout. The news grid utilises Flexbox, relying on the `flex-basis` property, as well as nesting Flexbox containers to emphasise the first two stories without any changes to the markup. The `flex-grow` property also permits stories to spread themselves among the available space, allowing the same code to be used no matter how many stories are on any given page.

YOUR QUESTIONS ANSWERED

Q. Can I use Flexbox and CSS Grid together?

A. Since they are both values of the `display` property, they cannot be used on the same element. However, they can (and should) be nested inside each other, using Grid for page layouts and Flexbox for widgets and other flexible elements inside the grid.

Q. How good is Flexbox browser support?

A. Browser support for Flexbox is now in a good place. We can use it (along with autoprefixer), knowing we'll be supporting every modern browser, mobile included, and Internet Explorer (IE) from 10 onwards. Taking a little more care, like the BBC has, you can provide fallbacks all the way back to IE8.

Q. How can I find out about Grid browser support?

A. With the specification only being finalised in September of last year, browser support is minimal at the moment, but due to land in all of the modern desktop browsers in March of this year. The Can I Use website (caniuse.com) provides a great service which allows you to check support for any CSS property.



3 TO FOLLOW

Šime Vidas

twitter.com/simevidas

Curator of Web Platform Daily and always up to date with the latest features.

Chris Coyier

twitter.com/chriscoyier

This guy lives and breathes the web and is also behind css-tricks.com and CodePen. It's worth hearing what he has to say.

Jen Simmons

twitter.com/jensimmons

Host of The Web Ahead podcast and designer advocate at Mozilla, Jen covers a large amount on standards, CSS, HTML and the web in general.

FOR THE GNU GENERATION

www.linuxuser.co.uk

FREE 3 DISTROS ON THE DVD

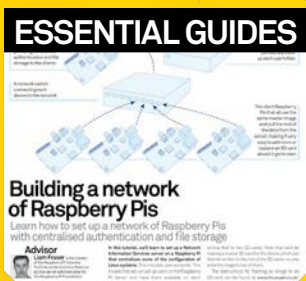


Linux User & Developer

Available
from all good
newsagents &
supermarkets
today

ON SALE NOW:

Build the perfect network » Exploit CMS vulnerabilities » Manage Ubuntu disks

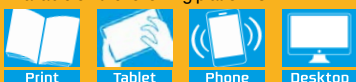


BUY YOUR ISSUE TODAY

Print edition available at www.imagineshop.co.uk

Digital edition available at www.greatdigitalmags.com

Available on the following platforms



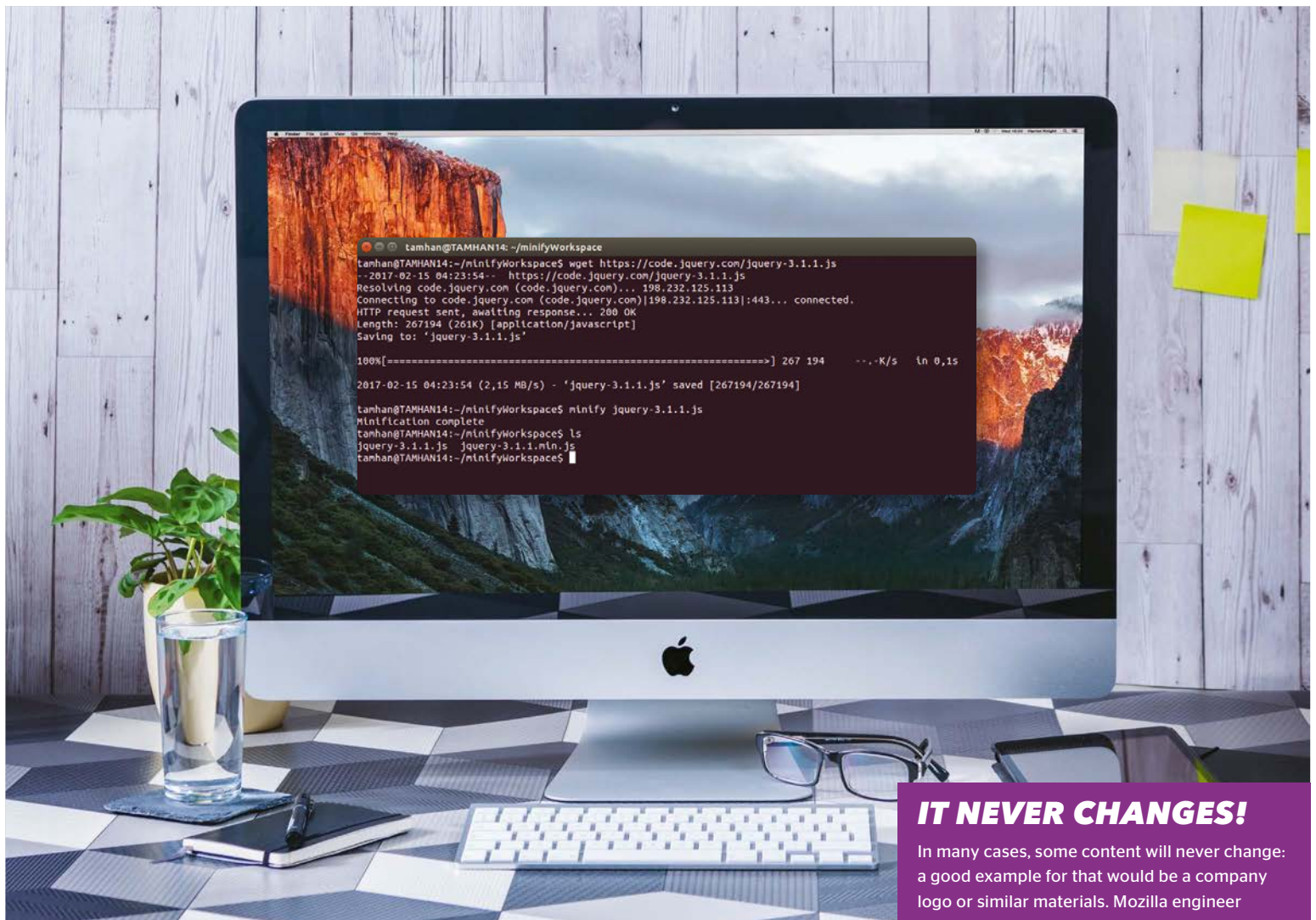
facebook.com/LinuxUserUK

twitter.com/LinuxUserMag

GET A SUPERFAST SITE

GET A SUPERFAST SITE

**20 tips, tools and techniques to speed
up your site and retain visitors**



COMPRESS YOUR JAVASCRIPT

Minify your code to get the most compact version

While comment times might not be a significant problem for C++ programmers, the developers of web applications suffer twice: they must be transferred to every client. Secondly, comments which are intended to help you and your team understand the code are clearly visible to everybody.

Fortunately, obfuscators solve these problems. Their basic modus operandi is simple: code is input, and is modified to create the most compact version which remains functionally compatible. Keep in mind that a large amount of minifiers exist: they differ in approach, capabilities and workflow integration. One simple product is the stock minifier, which can be downloaded via npm (<http://bit.ly/2IK45eY>). After the initial installation, your workstation's command line supports the minify command. A small example is shown in the image above. This shows the program being used to reduce a run-of-the-mill version of jQuery.

In addition to such "local" games, minify can also be run on an entire folder at once. This is accomplished via the following command sequence:

```
tamhan@TAMHAN14:~/minifyWorkspace$ minify ~/minifyWorkspace
```

Minification complete

One interesting feature involves the product's ability to remove already-minified files via the `-c` option: if it is not used, repeatedly minifying a folder leads to multiple minified files being minified over and over again:

```
tamhan@TAMHAN14:~/minifyWorkspace$ ls
jquery-3.1.1.js  jquery-3.1.1.min.js  jquery-3.1.1.min.min.js
```

An interesting aspect of minifiers involves their debugger support: not all are created equal. In some cases, mapper files are created: they provide debuggers with a deobfuscation guide, which allows you to debug processed code.

IT NEVER CHANGES!

In many cases, some content will never change: a good example for that would be a company logo or similar materials. Mozilla engineer Patrick McManus developed an extension to the testing protocol which allows the server to inform the visitor browser that a resource will be constant until the end of time.

Implementing immutability is accomplished by entering the following passage to the response:

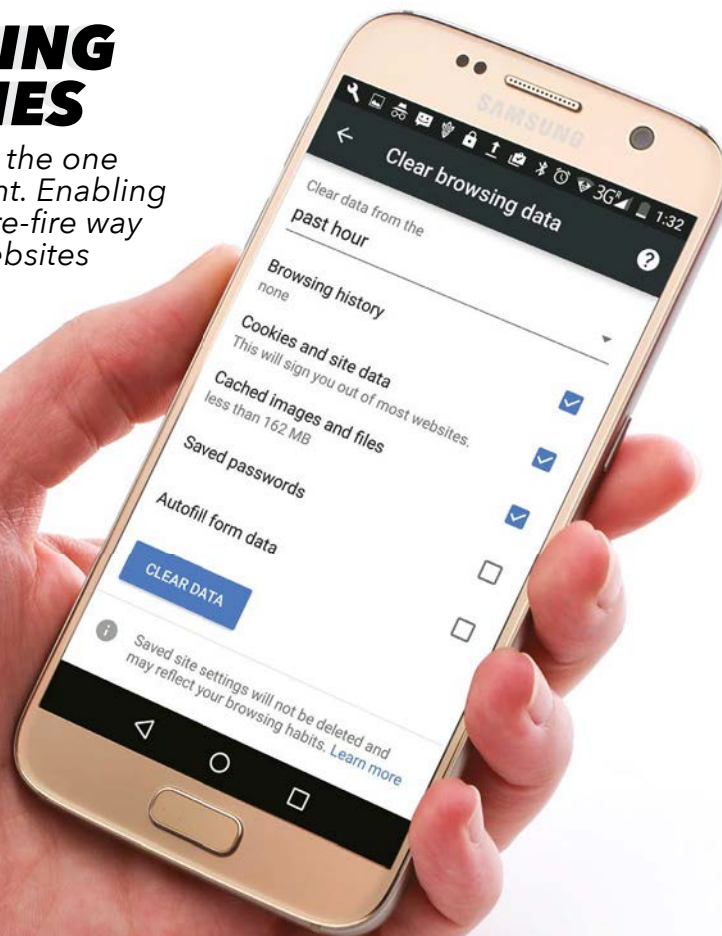
```
Cache-Control: max-age=365000000,
immutable
```

While discussing the exact approach to adding it is beyond the scope of this article - every server behaves differently - note that Facebook has reported significant profits by marking some resources immutable: on many browsers, the maximum expiry time is limited to about one year.

Be aware, however, that immutable elements are, indeed, immutable - even if the user presses the **F5** key, their content will not be updated. Changing any immutable resource requires you to change the filename: this can be sub-optimal from an SEO perspective.

SET CACHING PROPERTIES

The fastest request is the one which doesn't get sent. Enabling HTTP caching is a sure-fire way to accelerate your websites



When PalmOne introduced the first high-speed Treo, users were unhappy due to the browser: one of the most significant points of criticism concerned the missing local cache, which was unable to store resources efficiently.

Understanding the benefit of caching is easy: most websites change very rarely, if at all. Browsers use a local store to keep resources, which can then be fetched quickly, thereby accelerating the process as no network round trips are required.

Sadly, the browser has no efficient way to figure out what has changed. The only way to test this problem effectively involves the sending of caching records along with content: they inform the browser about the time when the material will expire.

Requests sent before can be fulfilled from the local cache, while requests coming in after have to be sent to the server once again.

The method required for enabling caching depends on the server type. If you use Apache, getting started involves finding the .htaccess file. It is a text file which informs that server about the various

parameters required - a simple caching configuration looks like this (see also the documentation on the Apache mod_expires module: bit.ly/2kMft8E):

```
# enable expirations
ExpiresActive On
# expire GIF images after a month in the
client's cache
ExpiresByType image/gif A2592000
# HTML documents are good for a week from
the
# time they were changed
ExpiresByType text/html M604800
```

By the way, caching is beneficial not only to websites which get visited frequently. If a customer visits your online store but once, they still benefit: resources such as logos and images shared between invocations will also be loaded faster.

During development, the cache can be cleared by pushing **F5**. Alternatively, you can also use the private browsing feature.

PLUG-IN BE DARNED!

A few years ago, technologies such as Flash and Java applets were cool; today, their functionality can be achieved with HTML5. Eliminating plug-ins is beneficial for many reasons: first, plug-in content usually can't be indexed by search engines. Second, users get slowed down significantly: if the plug-in is installed, they must wait until the (often slow) loading process is completed. Also, many users don't have plug-ins such as Flash installed any more: if your website demands their presence, your visitor will look somewhere else.

GET A SPEED BOOST

Server bandwidth is a traditional bottleneck. If CPU load is low, enabling GZIP makes for a great speed boost. This is usually handled in the .htaccess file, and does just what you expect: before resources get sent, the server applies the industry-standard zip compression to reduce its size.

This is beneficial when dealing with users connecting via a slow connection: the time spent compacting and decompressing is usually small compared to the amount of time saved during the transfer process.

IMAGE, ON A DIET!

Assuming that you have read the last issue of **Web Designer**, the content of this tip should not be a surprise. Using the right image compression format reduces the amount of data which needs to be transferred to your clients. This saves time and bandwidth which can be used to serve other clients. In addition to that, an efficient storage format reduces the amount of disk space needed for images: this is especially important when the website is provided by a low-cost shared hosting service.

MAKE IT LARGE!

Reducing the amount of missed clicks accelerates the user: if you have to tap an item five times, it really slows you down. Especially when dealing with mobile websites, making sure that page elements are large is significant. That official guidance from Google can be found at bit.ly/1Mht5aS - it provides minimum spaces which must not be exceeded. In general, making important elements as large as possible is a great idea: misclicks are a sure-fire way to annoy the living hell out of your users.

STARVE YOUR PAGE

Reducing the amount of data transferred is always sensible. In some cases, deferring loading is enough to improve speed

Website speed is a holistic problem: a variety of tools and adjustments lead to optimisation. In addition to that, pure speed is but part of the problem: if the above-the-fold content of a website shows up fast enough, the user is likely to ignore slower loading.

First of all, small JavaScript files can be inlined - this means that the client does not need to wait for an extra response to fulfil, but can instead start parsing and rendering immediately:

```
<html>
<head>
<script type="text/javascript">
/* contents of a small JavaScript file */
</script>
</head>
<body>
```

A more complex approach involves the declaration of deferred loading processes: this means that the web browser can complete a first rendering pass before

having downloaded the relevant files. The simplest version involves setting the async property:

```
<script async src="my.js">
```

When set, the browser will complete a first rendering run before having downloaded and rendered the file. Since this process is somewhat complex, we will not discuss it further here - lots of information on the topic is available on various places all over the internet.

Another small but nifty trick involves the passing of width and height attributes: this information allows the browser to render images 'finally', long before the actual image file has been downloaded:

```
<img . . . height="42" width="42">
```

Finally, be aware that CSS files can also be optimised in a process similar to the one discussed in other steps of this tutorial. This leads to faster parsing.

HTML-MINIFIER

Even though the amount of savings possible in HTML markup is significantly less than the ones found in JavaScript code, eliminating excess comments and redundant structures in your HTML will still lead to a slightly faster website.

Since going through files by hand is counterproductive, a group of developers created a program called html-minifier. Its behaviour is quite simple: unoptimised files go in, optimised files out. In addition to that, the product can also be integrated into various workflow systems.

The product can be used both as standalone and as programmatic extension - if you want to run it from the command line, install it as follows:

```
npm install html-minifier -g
```

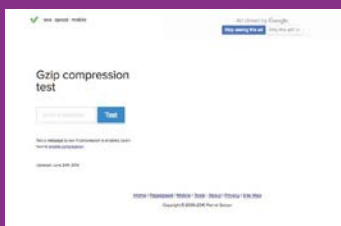
When this is done, simply run the product from your workstation's command line. Alternatively, the minifier can also be used as a Node.js module - this is especially helpful if your CMS takes in content from databases or similar resources dynamically.

Finally, a significantly detailed write-up of the various optimisations found in the product can be found at perfectionkills.com/optimizing-html. It makes for valuable reading for everyone - some of the code smells described are sure to give you a good chuckle.



"SMALL JAVASCRIPT FILES CAN BE INLINED - THIS MEANS THAT THE CLIENT DOES NOT NEED TO WAIT FOR AN EXTRA RESPONSE"

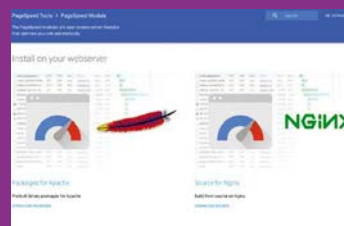
4 TOOLS FOR ANALYSIS AND IMPROVED PERFORMANCE



Compression tester

varvy.com/tools/gzip

Finding out if GZIP compression is enabled can be difficult. Fortunately, visiting VarVy solves the problem: simply enter the URL of your website and click the Test button. The website will then proceed to displaying a page telling you whether GZIP is enabled and if so, how much bandwidth has been saved by using it.



Magic Minify!

bit.ly/2kXyxBf

Google's PageSpeed Module integrates into Nginx or Apache, and minifies resources automatically. When running your own web server, this reduces the efforts involved in minification as the resources don't need to be compressed on your workstation. The product is not suitable for most shared webspaces.



PingDom

tools.pingdom.com

PingDom's analysis feature allows you to test your website from various servers located around the world. In addition to providing performance graphs and various bits of data about how your site compares to the competition, it offers advice about techniques which can be used to improve its performance.



Check your Cacheness

giftofspeed.com/cache-checker

Open the GiftOfSpeeds Checker and provide the URL of your choice: the website will proceed to provide a tabular overview of all resources and the caching levers which your server specified for them. This information can then be used to adjust other parameters of your web server.

SPEED UP THAT SERVER!

Google demands that web servers must respond to incoming requests within less than 200 milliseconds. Here are a few tricks!

Web server speed is a function of two factors: first of all the hardware available, and secondly the configuration of the server software used. The amount of control afforded over these parameters depends on the agreement between web host and client - in some shared hosting setups, no control is possible over the server hardware and software. In this case, web designers are advised to look at their side of the equation: many CMSs like WordPress come with one or more caching plug-ins which accelerate page accesses and thereby reduce CPU load.

Should a web property find itself hosted on a dedicated or semi-dedicated server, checking

resource allocations is the next order of business.

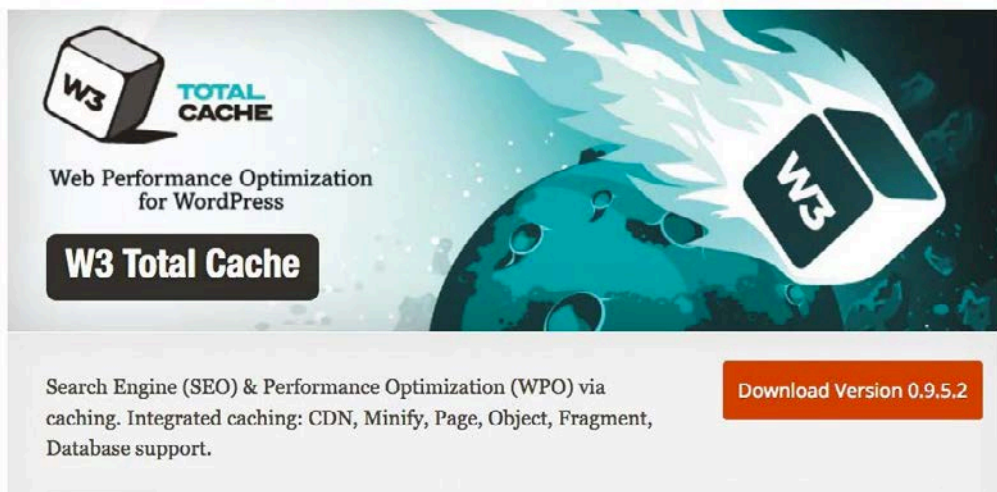
Virtual machine-based plans often suffer from significant overprovisioning: while profitable for the infrastructure owner, an

overprovisioned server tends to sag under heavy load. In such cases, shopping around is helpful - in many a scenario, the mere threat of changing your provider is enough to unlock significant savings on the published rate.

To cut a long story short, web masters must keep an eye on their server hardware. Picking up the necessary Linux knowledge might be a bore, but tends to pay off and look great on résumés.

LOOK AROUND!

Web hosting is a highly competitive market: switching hosts every two years is a sensible approach to problems!



W3 TOTAL CACHE

Web Performance Optimization for WordPress

W3 Total Cache

Search Engine (SEO) & Performance Optimization (WPO) via caching. Integrated caching: CDN, Minify, Page, Object, Fragment, Database support.

[Download Version 0.9.5.2](#)

LIBRARY STARVATION!

jQuery, for sure, is popular. But is it really needed in each and every web project?

An absolute classic of web design anti-patterns is a small website which, eventually, was upgraded with some extra visual effects via the inclusion of an (uncommon, and likely unminified) version of jQuery which provides but a single animation. We have actually seen 10k websites loading 500k worth of libraries.

The following step is not for the feeble-of-hand, but can make sense once a site becomes really large. Take a look at each and every library, and perform an analysis of the amount of code it contributes vs the functionality delivered. In many cases, a project uses but a small part of the library and so carries excess ballast.

Addressing this problem can be accomplished in three ways: first of all, many library vendors

have started to provide slimmed-down 'alternative' or lite versions of their products. Deploying those usually cuts the lard.

If this is not possible, an alternative library should be located - radio.js, for example, is a small and compact implementation of an event system. Finally, an own library can be created out of snippets of the

original library or from scratch. This seemingly nuclear option often works well, as the outer interface can be retained.

SEO CONTENT

CDNs can impact your SEO: consult with your Search Engine Optimisation professional before deploying a CDN.

MAKE THE COMPETITION WORK

The discussion about the strengths and weaknesses of library-provided content delivery networks is probably about as old as their existence. From the point of view of speed optimisation, using library-provided content delivery networks can have significant advantages.

Keep in mind that modern browser caches do not limit themselves to one website: if a resource needed by site B has been stored for website A, the browser will happily fulfil the request for B from its cache.

Especially when working in a network of websites which get visited in groups, using popular content delivery networks can save a lot of time: if a library was already downloaded for one of your competitors' websites, a visit to your website gets it fetched from the cache. That way, the loading time will occur on your competitor's tab - if that isn't great, we really don't know what is.

GO INTERNATIONAL!

Even though the performance of intercontinental network connections has improved significantly in the last few years, transferring a large amount of small records still takes a lot of time due to the increased latency (see the figure below).

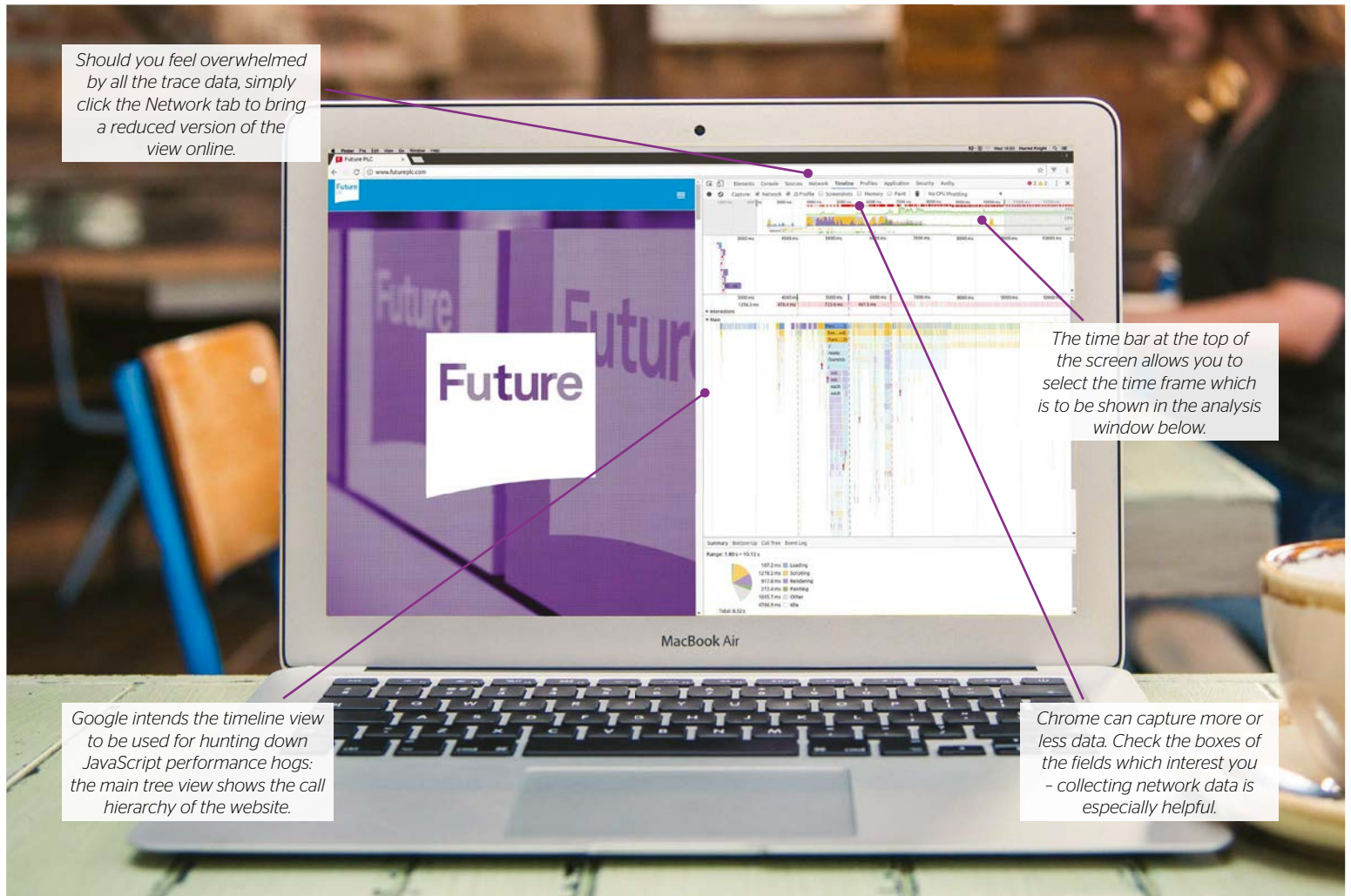
```
tamhan@TAMHAN14:~$ ping www.unne2.com
PING www.unne2.com (37.60.244.102) 56(84) bytes of data:
64 bytes from ip:37-60-244-102.siteground.com (37.60.244.102): icmp_seq=1 ttl=5
64 bytes from ip:37-60-244-102.siteground.com (37.60.244.102): icmp_seq=2 ttl=5
64 bytes from ip:37-60-244-102.siteground.com (37.60.244.102): icmp_seq=3 ttl=5
64 bytes from ip:37-60-244-102.siteground.com (37.60.244.102): icmp_seq=4 ttl=5
^C
--- www.unne2.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 300ms
rtt min/avg/max/mdev = 150.161/151.477/153.486/1.251 ms
tamhan@TAMHAN14:~$ ping www.google.com
PING www.google.com (216.58.209.36) 56(84) bytes of data:
64 bytes from waw02a05-in-f4.1e100.net (216.58.209.36): icmp_seq=1 ttl=52 time=
64 bytes from waw02a05-in-f4.1e100.net (216.58.209.36): icmp_seq=2 ttl=52 time=
64 bytes from waw02a05-in-f4.1e100.net (216.58.209.36): icmp_seq=3 ttl=52 time=
64 bytes from waw02a05-in-f4.1e100.net (216.58.209.36): icmp_seq=4 ttl=52 time=
^C
--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 300ms
rtt min/avg/max/mdev = 41.948/42.945/44.916/1.221 ms
tamhan@TAMHAN14:~$
```

Accessing a European web server from Slovakia is much faster than an American one

Content delivery networks are a nifty, if somewhat pricey way to work around the problem. Their behaviour is as simple as you might assume from the name: resources provided by content delivery networks are uploaded to the CDN's intake server, which then spreads them across the various servers provided by the company. In the next step, your website is updated to include a script which forwards requests to the CDN. Users then connect to the CDN infrastructure, which forwards their request to the physically closest source for the file in question.

LOAD TIME ANALYSER

Google Chrome contains a very powerful analysis toolkit which provides information about resource consumption during a page load



USE AN IMAGE MAP

When small files get loaded in a piecemeal fashion, the network round trip latency can be larger than the actual time needed for the transfer. One nice way to work around this problem has been used in the 3D programming space for a long time: programmers of games suffer from a similar problem.

The accepted workaround involves the use of image maps. An image map is but an image which contains lots of icons next to one another. This way, the browser can download the entire image map with a single HTTP request. In the next step, CSS can be used to select individual areas of the image map which are needed to populate the various elements of your website.

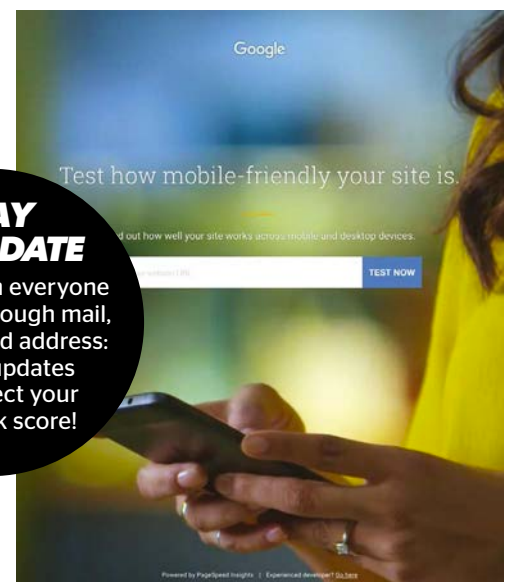
TEST WITH GOOGLE

If improving PageRank is high on the list of reasons for speed optimisations, why not collect some advice straight from the Big G?

Sadly, that usability of the tool could be a lot better: after entering the URL, you are provided with a summary report giving numerical scores in a total of three different categories. More information can only be accessed by clicking the orange button which points downwards: you then get three screens providing bullet-point overviews of issues found in your website. More technical information can be obtained by providing an email address, which is then graced with an email containing further recommendations on the optimisation of the website in question.

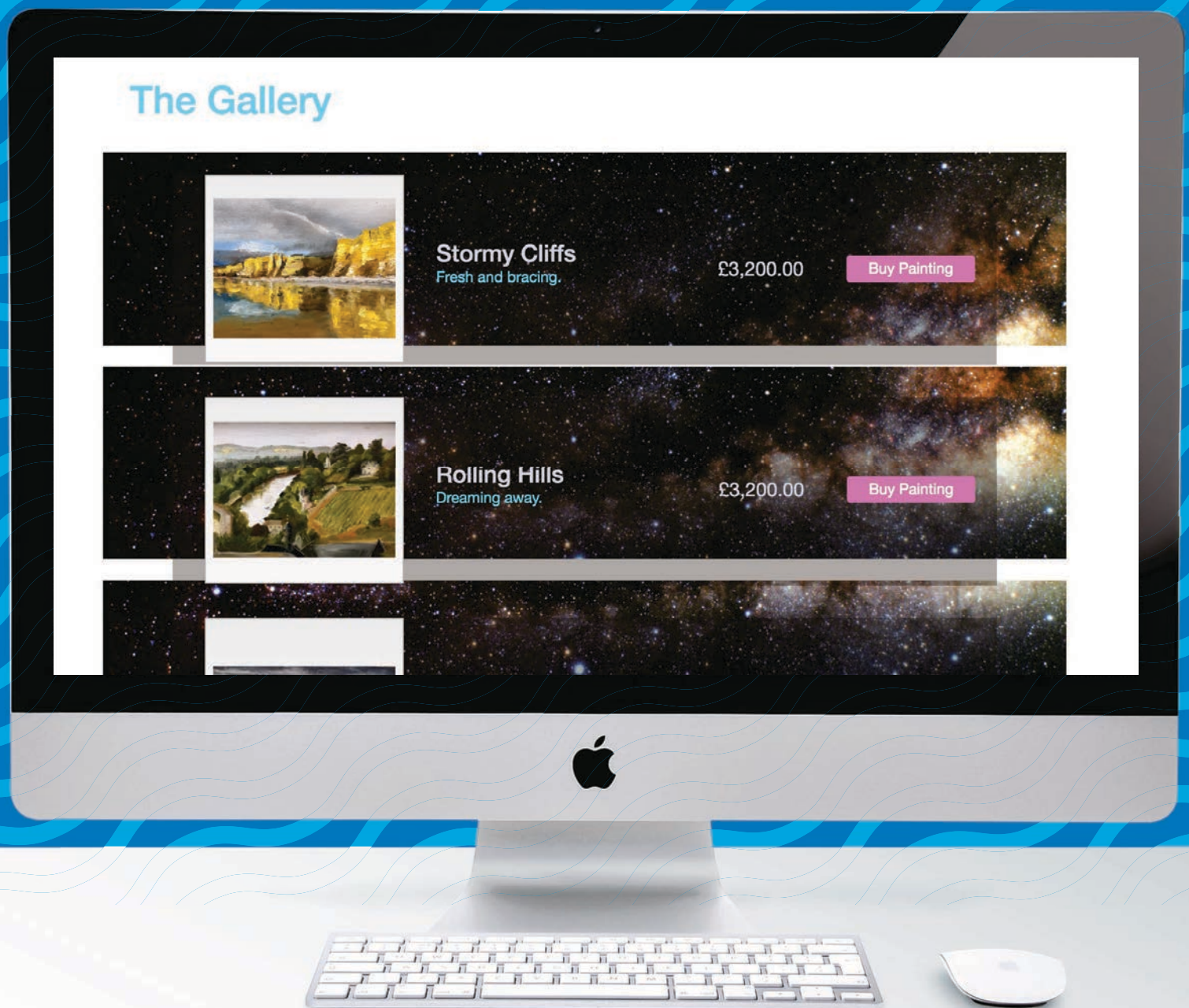
STAY UP TO DATE

Even though everyone of us gets enough mail, supply a valid address: missing updates could affect your PageRank score!



Create a parallax effect with Angular 2

How to implement a basic parallax scroll effect for a webpage built within the Angular 2 framework



For small scale or personal projects, you might feel that simple HTML and CSS, along with your standard JavaScript libraries, are all you need to build a great site. By following best practices, you can indeed avoid messy CSS and closely-coupled 'spaghetti' JavaScript code.

However, this requires a lot of governance on large code-bases that are being maintained by multiple people. If you're not on top of it, over time your application can get weighed down by technical debt and become increasingly unmaintainable. Angular 2 can really help enforce code structure that is readable and maintainable while also enabling some great UX with features like two-way data binding.

Even a subtle parallax effect can leave a great impression on visitors to a webpage. It's achieved by causing different parts of the page to move at different speeds in the frame as the user scrolls. The illusion of depth and perspective it creates is something difficult to achieve with static scrolling.

There are lots of ways to implement parallax scrolling, from very subtle background displacement, to multi-directional, multi-layer effects.

But how can I achieve a parallax effect and still get all the great features of Angular 2? This tutorial can get you started by showing how you can achieve a simple parallax effect on individual frames on your page!

1. Set up local development environment

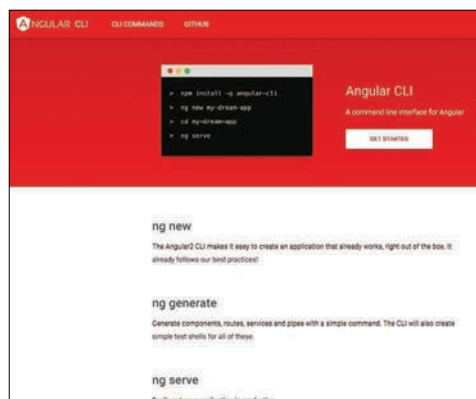
If you already have Node installed, skip straight to step two now. Node gives us the ability to run our Angular app on a local development environment. Because Angular runs on Typescript, we need to have our application running locally so we can transpile our Typescript to JavaScript and see changes straight away in the browser.



2. Install Typescript compiler and Angular CLI

If you already have these, skip to Step 3. Now we've got node, we can use the node package manager (npm) to install our Typescript compiler from the command line. The folks at Angular also recommend installing the Angular 2 Command Line Interface (CLI). We're going to do so for this tutorial as it's a lot easier and faster to generate Angular code and install dependency packages.

```
$ npm install -g typescript
$ npm install -g angular-cli
```



3. Open tutorial app

As this tutorial will focus mainly on the implementation of parallax and not fundamental features of Angular 2 and how to implement them, we will start with a pre-made Angular 2 app and add a parallax effect to it. Download the tutorial file 'the-gallery' and place it in your root directory.

4. Launch the server

Now that we have an Angular app, we can launch our local node server and see our changes updated live in the browser by navigating to `http://localhost:4200/`. Navigate to the directory containing 'the-gallery' tutorial folder. From inside the project folder run the command:

```
$ cd the-gallery
$ ng serve
```

5. Check in the browser

Now that we have our local server running we can check if our environment is set up correctly and our Typescript is transpiling correctly. Go to 'localhost:4200' in your browser to see the base of the page layout.

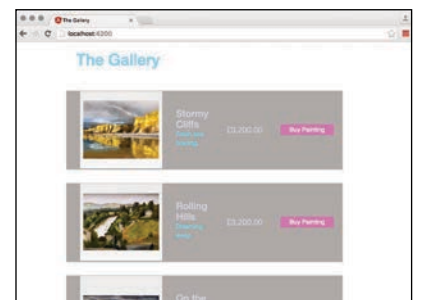
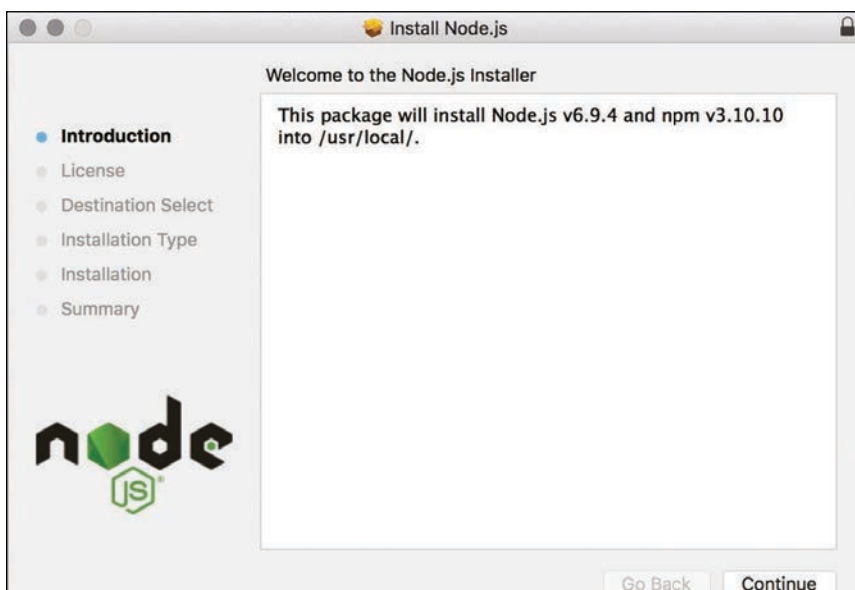
6. Install Parallax module

From the command line using npm, install the Ng2 Parallax Scroll module, which is made available under open source license by Allen Kim.

```
$ npm install ng2-parallax-scroll --save
```

Custom Angular 2 modules

There are many open-source, custom modules available via npm. The challenge is maintaining the right dependency versions to utilise them. You can manage this in the package.json file generated by CLI.



Left

Once you've downloaded node, install it using the wizard. This will include an npm install that you can use to install custom modules

Top left

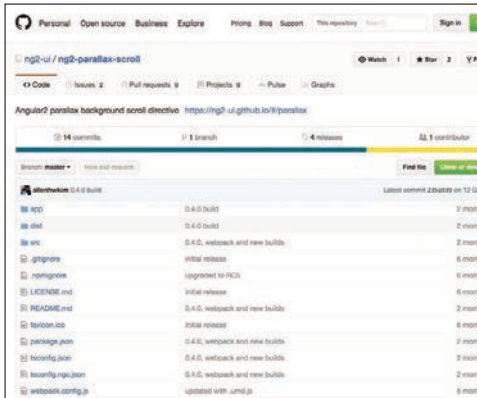
If the app fails to launch you will be able to view any errors in the console. A successful build looks all nice and green like this

Top right

This is what our simple Angular app should look like in the browser at this stage. Next we'll be adding a simple parallax effect to set off each listing

Developer tutorials

Create a parallax effect with Angular 2



7. Import parallax module

So far we've only installed the parallax module and its source files are now stored in our node_modules folder. We need to import it to AppModule so it is available to our root NgModule class decorator function. Add an import statement to app/app.module.ts.

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
```

Generating components

You often won't need to edit the generated component Typescript file. However, with some custom module imports you might like to set your own initialisation parameters.

```
import { FormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/http';
import { ModalModule } from 'ng2-modal';
import { Ng2ParallaxScrollModule } from 'ng2-parallax-scroll';
```

8. List module in NgModule decorator

We now need to list the parallax module as a metadata object within the imports array of the AppComponent NgModule decorator function. This will make the parallax module's features public to all our app's components.

```
@NgModule({
  imports: [
    BrowserModule,
    FormsModule,
    HttpClientModule,
    ModalModule,
    Ng2ParallaxScrollModule
  ]
})
```

9. Generate parallax component

Angular components are a great way to separate out distinct functionality in our application. This reduces code duplication and improves maintainability by enforcing good modularisation practice. We're going to use the CLI to generate a new component called parallax.

```
$ ng generate component parallax
```

10. Check CLI generated code

The CLI has created Typescript, HTML and CSS files for our parallax component. It has also made it available to our other application components via our AppModule.

Typescript code has been generated for us importing the Angular Component module and setting up our component class and decorator function.

```
import { Component, OnInit } from '@angular/core';
@Component({
  selector: 'app-parallax',
  templateUrl: './parallax.component.html',
  styleUrls: ['./parallax.component.css']
})
export class ParallaxComponent implements OnInit {
  constructor() { }
  ngOnInit() { }
}
```

11. Check the browser

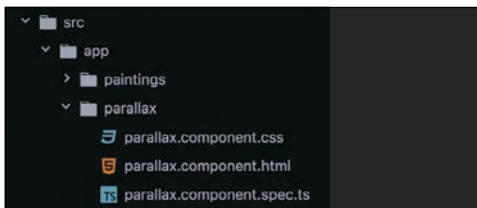
Our app should still load at this point but we're not expecting to see any change yet. We can see in DevTools that the parallax component is not rendering because we haven't included our custom parallax component element in our app component html page - only the painting component is there.

12. Add parallax component

Add a <div> element to the AppComponent HTML template containing our parallax app elements. Make sure it is before the painting container, otherwise it will be loaded over the top and interfere with painting container functionality.

```
<div>
<app-parallax>
```

```
$ ng generate component parallax
installing component
create src/app/parallax/parallax.component.css
create src/app/parallax/parallax.component.html
create src/app/parallax/parallax.component.spec.ts
create src/app/parallax/parallax.component.ts
update src/app/app.module.ts
```



Top

If your CLI command was successful, you will see 'create' and 'update' keywords against relevant paths in the console

Right

There is now a new folder in src/app named for our new component containing our component files!

Top/middle

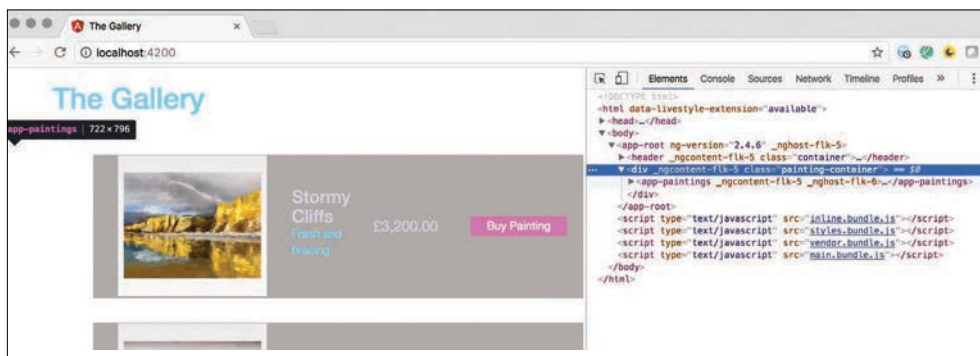
We will be using an image of space courtesy of NASA as our background. This amplifies the subtle parallax effect of depth and perspective.





Parallax responsive design

We want to make sure our page follows responsive design principles and is mobile first in its approach. In order to evaluate this, let's remove the painting component and assess the parallax component on its own for a minute. It's easy to do this in Angular 2 - all we need to do is go to our app.component.html template and comment out the 'painting-container' <div> element. Now only the heading and parallax containers remain. The two sides of this image show the parallax container in both wide and narrow viewports. You can see on the narrow viewport that the image has been re-sized so that its height matches the container. Now when you scroll there is no parallax effect. This could be overcome by using an a larger vertical image.



```
</app-parallax>
</div>
```

13. Get parallax background image

The image we will be using as a background to apply the parallax scrolling effect to is called 'space-08.jpg' and is stored in the tutorial folder. Add this to your application's images folder.

14. Add the image to the template

The parallax component has been included in the template for the AppComponent already. Now we need to add the image to the template for the parallax component itself. Add the image inside a <div> element to parallax.component.html.

```
<div ng2-parallax>
  
</div>
```

15. Create parallax-container CSS class

In app.component.css, create a CSS class called 'parallax-container' and add declarations for position and top.

```
.parallax-container {
  position: center;
  top: 100px;
}
```

16. Add parallax-container class

Add the CSS class to the div containing our parallax app in the AppComponent template.

```
<div class="parallax-container">
  <app-parallax>
</app-parallax>
</div>
```

17. Create parallax CSS class

We need to define a CSS class specifically for the <div>

element containing the image to which the parallax effect is being applied. Add this class to parallax.component.css.

```
.parallax {
  min-height: 180px;
  max-width: 900px;
  min-width: 300px;
  margin-top: 20px;
  margin-bottom: 10px;
  margin-left: 50px;
}
```

18. Add .parallax CSS class

Apply the .parallax CSS class to the <div> element in parallax.component template.

```
<div ng2-parallax class="parallax">
  
</div>
```

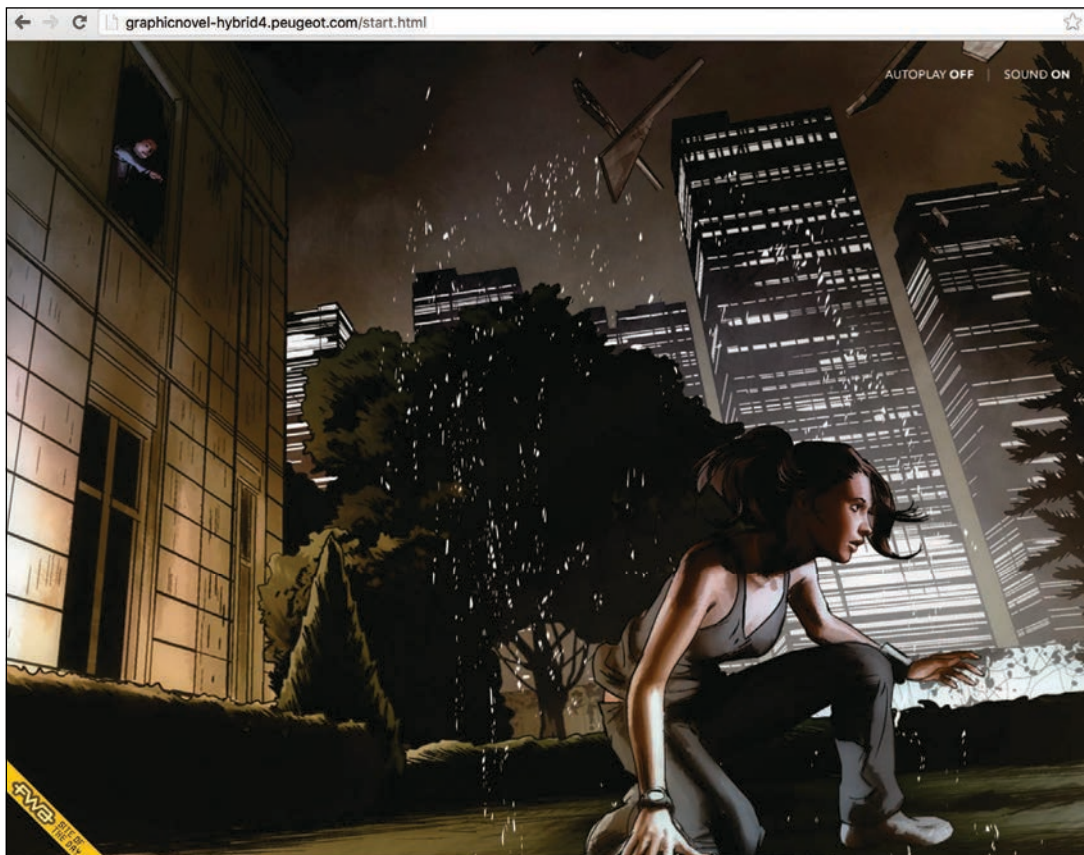
19. Boxing in

We get a better effect from the parallax scrolling if we include blank <div> elements above and below our parallax <div>. Add this to the parallax component template. Create a .border CSS class in parallax.component.css with only a height: 0px; declaration. This may need adjusting if you choose a different background image.

```
<div class="border">
</div>
<div ng2-parallax class="parallax">
  
</div>
<div class="border">
</div>
```


Developer tutorials

Create a parallax effect with Angular 2

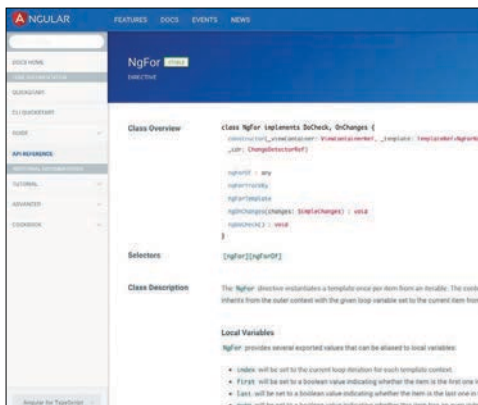


Customising parallax

If you're feeling ambitious, you could have a go at creating your own parallax functions to allow endless flexibility. The basic mechanics are actually very simple: we catch an event, and move objects on the page at differing speeds in response to it. The most common event to tie to parallax is scrolling, but you're not limited to this and there are many sites out there that implement parallax in response to mouse movement. The difficult part is implementing movement that feels natural. A good parallax site will move background objects slowly and foreground objects faster, in a manner which mimics real perspective. There are also many nuances to getting the effect to work reliably, responsively and perform smoothly. Badly implemented parallax can not only jar visually, but kill user experience by triggering constant JavaScript execution and repaints. Fortunately, there are already many libraries out there that have done the hard part for you!

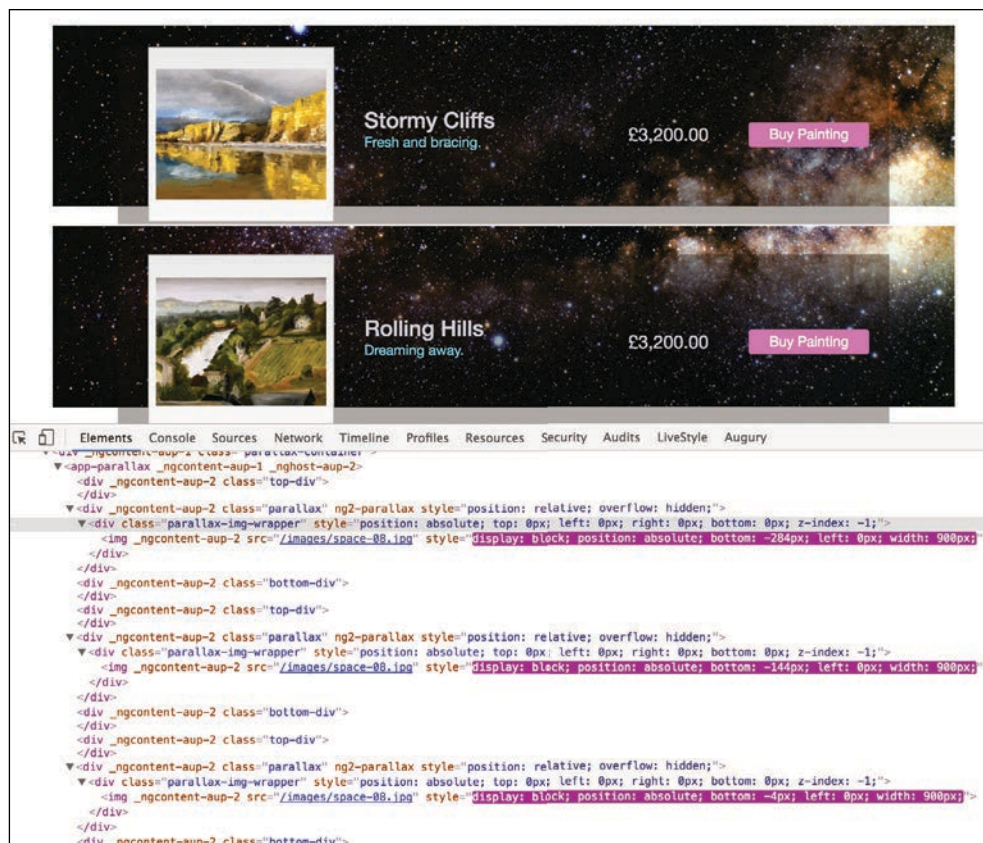
20. Multiple parallax images

Let's put a separate instance of the parallax scrolling image behind each of the paintings we have for sale. This will amplify the effect and give us the ability to choose different images for each section if we want. Duplicate the code from step 19 twice. An improvement to this would be to use an Angular ngFor loop here.

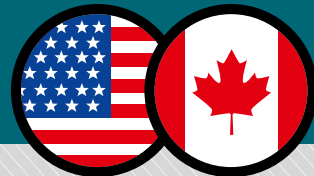


21. Check the browser

'The Gallery' Angular 2 app should now have a parallax scrolling background effect! Scroll up and down to test it out! Why not try some more open-source parallax scrolling custom modules to get more advanced customisation such as parallax scroll speed? Happy scrolling!



Special offer for readers in **North America**



6 issues FREE

When you subscribe*

FREE
resource
downloads
every issue

**“The only magazine
you need to design
and develop
stunning websites”**



Order hotline **+44 (0)1795 592862**

Online at **www.imaginesubs.co.uk/wed**

*Terms and conditions This is a US subscription offer. You will actually be charged £80 sterling for an annual subscription. This is equivalent to \$105 at the time of writing, exchange rate may vary. 6 free issues refers to the USA newsstand price of \$14.99 for 13 issues being \$194.87, compared with \$105 for a subscription. Your subscription starts from the next available issue and will run for 13 issues. This offer expires 31 May 2017.

Quote
USA3
for this
exclusive
offer!

Build a speedy universal React site

Produce a lightweight React-powered site that is server-rendered by default with the help of the Next.js framework





Web development – particularly front-end – has shifted dramatically in the past few years. While

previously sites were largely static and server driven, frameworks like React came along and gave us the single-page app. The server dealt with API calls and not much else.

With universal JavaScript, we can write one set of components that can run either on the server or the client. This helps improve performance as well as productivity from a developer's point of view.

Next.js is a framework that helps create React applications in a universal style. It comes with Webpack and Babel already set up, so it is easy to start a project and get going.

The framework is deliberately small, which gives developers the opportunity to expand it further for the need of each project. Modules can be imported where necessary without any conflicts with the underlying code.

That is not to say it is light on features. Common developer tools such as code splitting and hot reloading are there from the get-go. It also includes support for newer concepts such as CSS scoped within each React component to keep things easily transferable.

In this tutorial we will be using all these features to help build a small and speedy travel website that's fed data from a server through JSON.

1. Install and run Next.js

Firstly we need to download the framework itself and all of its dependencies. We are also going to fetch axios, which is a tool that will enable us to easily perform network requests.

Open up the command line, navigate to the tutorial files and run the following commands. This downloads the files and then starts the Next.js server.

```
> npm install
> npm run dev
```

2. Create an index page

If we open a browser to the supplied location (usually `http://localhost:3000`), an error appears. We need to make an index page.

All components served in the `pages` directory will get server rendered automatically. Open up `pages/index.js` and export a basic React component. Refresh the page and check that you see it.

```
export default class
  extends React.Component {
    render() {
      return (<div>
        This is the index page
      </div>)
    }
  }
```

3. Add a layout component

We can create a component that wraps around our pages to provide a persistent layout.

Open up `components/Layout.js` and add this to the bottom. It uses ES2015 and React shorthand to export a simple presentational component which will display its children. Inside `index.js`, wrap the page in the new Layout component.

```
/* Layout.js */
export default ({ children,
  title = "Discovery" }) => (
  <div>
    <header>Header</header>
    { children }
    <footer>Footer</footer>
  </div>
)

/* index.js */
<Layout>
  <div>[...]</div>
</Layout>
```

4. Adjust the <head> element

Currently we are not able to add anything in the head of the page, such as the title. Next.js provides a `<Head>` component that will add its contents to the head where necessary.

Inside `Layout.js`, add a Head component just inside our container `<div>` to set the page title.

```
<Head>
  <title>{ title }</title>
  <meta charSet="utf-8" />
  <meta name="viewport" content="initial
    -scale=1.0, width=device-width" />
</Head>
```

5. Add basic styling to layout

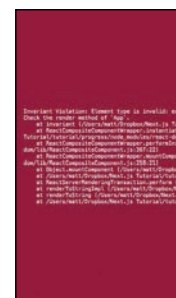
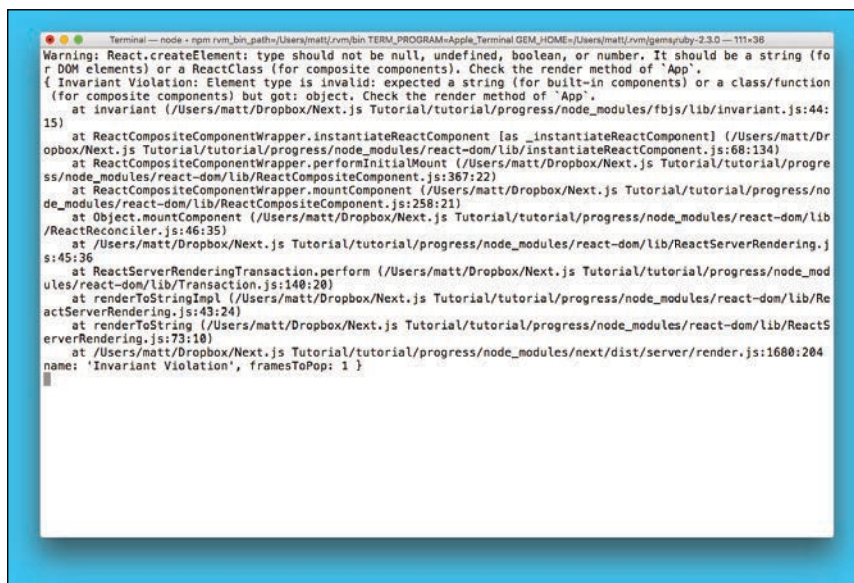
At the moment, our page is looking rather drab. We should begin by applying styling to the body, which would need to go inside the head.

These styles are only going to be minimal, so we can add them inline in the head of the layout to avoid an extra request. These should be placed in curly brackets and backticks, else the server will read them as part of the component and throw an error.

```
<style>{`
* {
  box-sizing: border-box;
}
html, body {
  min-height: 100%;
}
```

Use of prefetch

Next.js supplies a `prefetch` module that can register a service worker in supported browsers and preload links for instant loading. It can be used as a drop-in replacement for the `Link` module.



Left

Keep an eye on the console when Next.js is running. If any errors occur, they will be logged out to here to help track down later

Top left

Server-side errors will display this bright pink error screen instead of the content. Client-side issues get logged in the browser console as usual

Top right

When a module with errors hot-reloads, it may cause display issues that persist even when the module is fixed. A manual page reload is required

Developer tutorials

Build a speedy universal React site

```
font-size: 20px; }  
'}  
</style>
```

6. Apply CSS variables

We can keep CSS variables separate and import them to keep settings like colour and sizing together. As Next.js uses Babel, we can use ES2015 imports and template literals to compose styling in the head.

The styles are located in the `styles/variables.js` file. With that imported, we can add their values to the styles.

```
html, body {  
  [...]  
  color: ${cssVars.fontColor};  
  background-color:  
    ${cssVars.backgroundColor};  
  font-family: ${cssVars.fontFamily};  
}
```

7. Link back to the homepage

As with most websites, our heading should include a link back to the homepage. While we can use anchor tags as usual, Next.js provides a `<Link>` component which will fetch the page as JSON and load inline, to avoid a full page reload.

styled-jsx support

Next.js v2 will include support for the `styled-jsx` module, which will allow for scoped styles to be written in a more familiar syntax and behaves more like the encapsulation found in Web Components.

Replace the contents of the header in `Layouts.js` with the following.

```
<Link href="/" />  
  
</Link>
```

8. Apply styling to the layout

Now the basic styles are in the head, we can add layout-specific styles using JavaScript. With Next.js, this is done using a package called Glamour, which defines CSS rules as objects.

Pass some styles to the imported `css()` function and add the returned value as the class name of the layout container. Note that camel caps are used instead of hyphens in style properties.

```
const applicationStyle = css({  
  "width": "80%",  
  "maxWidth": "1000px",  
  "margin": "0 auto"  
});  
[...]  
<div className={applicationStyle}>
```

9. Styling component children

Where children aren't their own components, we can style them in their parent by using an ampersand in the selector. We can then nest objects to define their styles.

Add the following to the CSS object.

```
"& header": {  
  "margin": "2rem",  
  "textAlign": "center"
```

```
},  
"& header img": {  
  "width": "100%",  
  "maxWidth": "400px"  
},  
"& footer": {  
  "marginTop": "2rem",  
  "fontSize": "0.5rem",  
  "textAlign": "center"  
}  
}
```

10. Fetch the locations

When pages are server-rendered, the `getInitialProps` method will only run on the server. It runs on the client only when navigated to with the `<Link>` component. This makes it a great place to fetch data.

Back inside `index.js`, create the method to fetch the data and add it to props. In this example, it picks up a JSON file, but this could be a call to an external API.

```
static async getInitialProps() {  
  const locations = await axios.get(  
    "http://localhost:3000/static  
    /locations.json");  
  return {  
    locations: locations.data  
  }  
}
```

11. Display a location preview

With the information fetched, we need to display it somehow. As Next.js is based on React, ordinary React components work with it as well. A ready-made `LocationPreview` component is included and will display



Top

If data is appearing incorrectly, it may be the session storage keeping a broken copy of it. Locate it in the browser developer tools, clear it and refresh

Right

Should the styling of the facts list need to change, we can confidently update its individual scoped styles without affecting the rest of the site

Top/middle

Next.js will automatically redirect to the error page for 404 and 500 errors. For any other cases, this will require a custom server setup



London is the capital of the United Kingdom located towards the south east of Great Britain. It is home to a wide range of cultures and traditions going back centuries, with tourist spots and architecture spanning the centuries.

Facts

Second largest city in Europe

It has a population of over 8.5 million people - roughly 13% of the population of the UK.

It's not called Big Ben

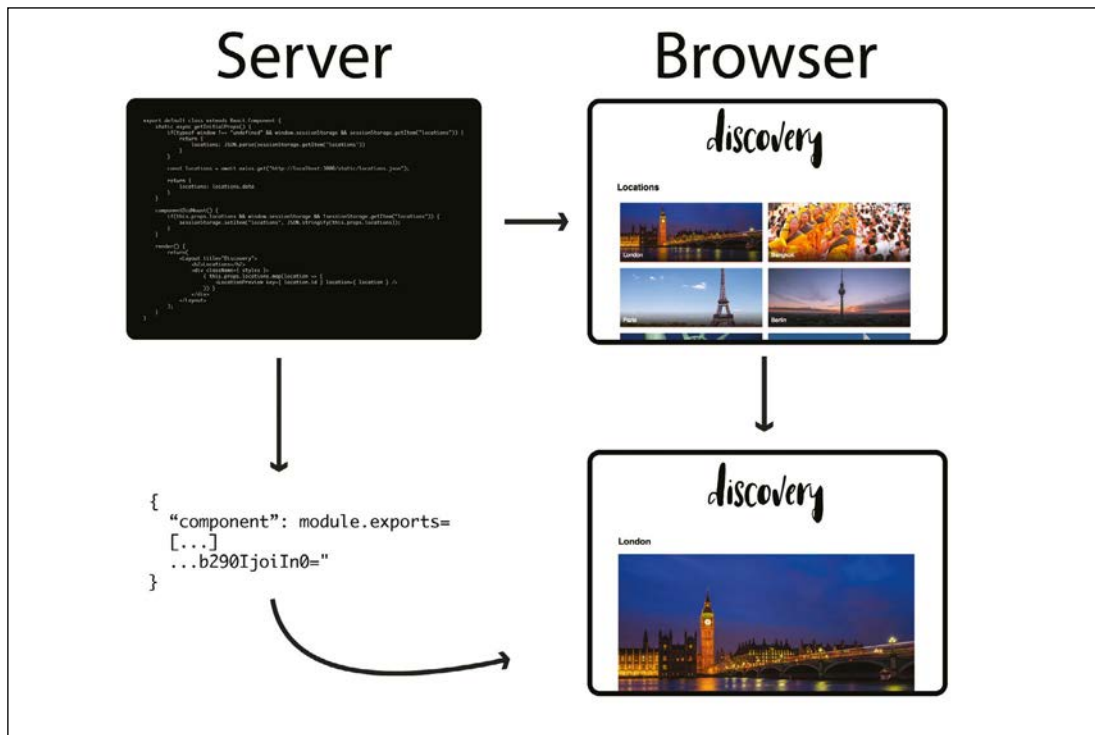
'Big Ben' is the name of the bell that is housed inside Elizabeth Tower.

The tube is mostly overground

Despite being called the London Underground, 60% runs above ground level.

Fake houses hide railway tracks

23 and 24 Leinster Gardens in Paddington are only building facades used to disguise the Metropolitan line behind it.



The benefit of universal JS

Frameworks like React gave us the ability to have app-like experiences on the web without stopping to make server requests on each interaction.

The problem with this approach is that it limits usable experiences to the fastest browsers. Initial page loads are often only an empty container that waits to be populated by JavaScript. Search engines can also struggle to index these sites.

Universal JavaScript applications can pre-render a page on the server for the client-side JavaScript to pick up and carry on with. This means the user gets meaningful content quicker while everything else loads in, search engines can index these pages with greater success and the all-round experience is improved.

locations based on the JSON data. Import that component and display them within the Layout component in `index.js`.

```
<h2>Locations</h2>
<div className={styles}>
  {this.props.locations.map(
    location => (
      <LocationPreview key={location.id}
        location={location} />
    )
  )}
</div>
```

12. Style on hover and focus

Similar to the Layout component, we can add scoped styles to only apply to the contents of the index page. While a lot of the styling is dealt with inside `LocationPreview`, we need to add hover and focus states on the parent, as we cannot target the Link component with CSS.

Also add an overflow property to make sure focus rings display when tabbing through the links.

```
const styles = css({
  "overflow": "hidden",
  "& a:hover div, & a:focus div": {
    "borderBottomWidth": "6px"
  }
});
[...]
```

13. Store locations in browser

As our locations data will not change all that often, we can cache it using browser storage in order to speed up navigation.

By using the `componentDidMount` method, we can perform this action in the browser regardless of whether it was server rendered or not. When it is added to the page, the data gets stored.

Add the following lines just before the render method in `index.js`.

```
componentDidMount() {
  if(this.props.locations &&
    window.sessionStorage &&
    !sessionStorage.getItem("locations")) {
    sessionStorage.setItem("locations",
      JSON.stringify(this.props.locations));
  }
}
```

14. Use cache if available

Similarly, if we hit the homepage from another page, we can load the locations from session storage instead of performing another request. Switching pages will be near instantaneous.

At the top of `getInitialProps`, check for session storage support in the case it's rendering server-side. If it exists, load from there.

```
if(typeof window !== "undefined" &&
  window.sessionStorage &&
  sessionStorage.getItem("locations")) {
  return {
    locations: JSON.parse(sessionStorage
      .getItem("locations"))
  }
```

```
}
}
```

15. Create the location page

There needs to be a way to view the extra information. The `LocationPreview` component links to `/location`, so open up `pages/location.js` and repeat the process from the homepage.

Add the following to the bottom of `location.js`. Using the 'title' prop on layout, we can change the title of the page. Be sure to copy over `componentDidMount()` from `index.js`.

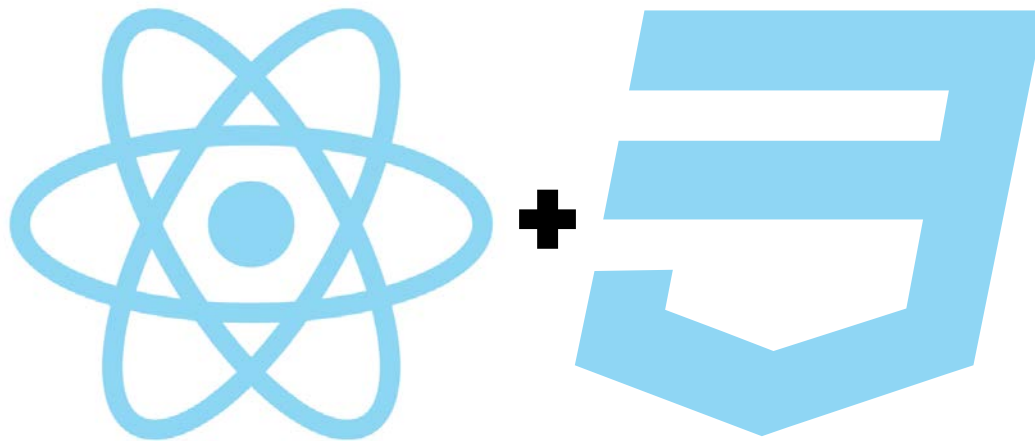
```
export default class extends
  React.Component {
  componentDidMount() {
    [...]
  }
  render() { return(
    <Layout title={'Discovery'}>
    </Layout>
  )}
}
```

16. Fetch location from server

We will fetch from the same data source as the index page and filter to match the ID of the selected location.

Set up `getInitialProps`, but this time get the 'query' object Next.js passes in and use the supplied ID to filter the locations.

```
static async getInitialProps({query}) {
  const locations = await axios.get(
    "http://localhost:3000/static
    /locations.json");
}
```

Why write CSS in JavaScript?

For years it was best practice to separate HTML, CSS and JavaScript to keep things maintainable. Frameworks like React and its JSX syntax already combine HTML with JavaScript within components. By adding CSS at that point, we keep everything together. Components will have a set style wherever in the project they may appear.

The CSS can also update to different changes in state. Depending on the library used, it's possible to dynamically change its styling of the component on the fly instead of using other techniques like adding a class.

A combination of old and new approaches may avoid overcomplicating applications with numerous similar components. Common markup like body and heading styles may be better being globally applied.

```
let location = locations.data.filter(
  location => location.id==query.id);
return {
  locations: locations.data,
  location: location[0]
}
```

17. Use cache where possible

In a similar process to the homepage, we can pick the locations in session storage and present them when available. The difference here is that we need to parse the JSON and filter the values.

Add the following at the start of `getInitialProps` in order to avoid making an unnecessary request to the server.

```
if(typeof window !== "undefined" &&
  window.sessionStorage &&
  sessionStorage.getItem("locations")) {
  let locations = JSON.parse(
    sessionStorage.getItem("locations"));
  let location = locations.filter(
    location => location.id==query.id);
  return {
    location: location[0]
  }
}
```

18. Display location data

With the location data in props, we can display it like any regular React component.

The data includes an image, a description and some facts. Loop through the facts using the 'map' method to

output them as part of an unordered list.

```
<ul>
  { this.props.location.facts.map((fact, index)
=> (
  <li key={ index }>
    <div className="title">
      {fact.title}</div>
    <div className="description">
      {fact.description}</div>
    </li>
  ))}
</ul>
```

19. Style the fact list

Finally, we can scope styles for an individual list. By keeping styles small and focused on an individual section, it keeps things clear and organised for developers.

Create a new list style constant and apply it to the `` tag. Styles listed here will not affect the rest of the page.

```
const factStyles = css({
  "listStyle": "none",
  "padding": 0,
  "& > li": {
    "margin": "1rem 0",
    "& .title": {
      "fontWeight": "bold"
    }
  }
});
```

20. Create an error page

Occasionally things can go wrong. Fortunately, the Next.js

server environment will automatically detect missing pages or server errors and redirect users to a special page located at `pages/_error.js`.

For our error page, supply a message to the user in that case and link back to the homepage.

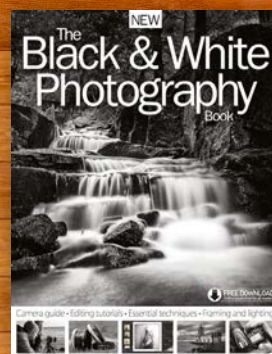
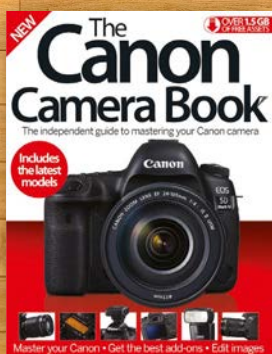
```
export default class extends
  React.Component {
  render() { return(
    <Layout title="Discovery | Error">
      <h2>There was an error</h2>
      <Link href="/">Go Home</Link>
    </Layout>
  )
}
```

21. Detect and supply error code

Next.js will supply details on the request as parameters in `getInitialProps`. It is possible to perform different actions such as alert-monitoring services based on the type of request that came in.

Pick up the 'res' (the response) or 'xhr' (the client-side request) and supply the status code as a prop to show to the user should this page be displayed.

```
static async getInitialProps({res,xhr})
{
  const statusCode = res
  ? res.statusCode :
  (xhr ? xhr.status : null);
  return { statusCode };
}
```

Discover one of our great bookazines

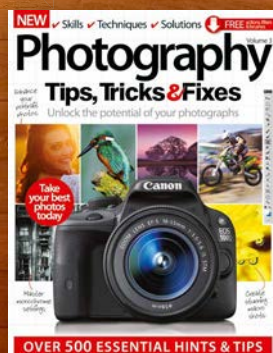
And save 20% on your first order when you buy direct from us



BRAND
NEW!

Photoshop Creative Annual

Discover the most creative projects for Photoshop CC, CS and Elements users, as this collection of practical guides is bursting with essential tricks and step-by-step tutorials.



Get great savings when you buy direct from us



1000s of great titles, many not available anywhere else



World-wide delivery and super-safe ordering

Future

www.imagineshop.co.uk

Magazines, back issues & bookazines.

Use the code "WELCOME20" when you make your first order to receive your discount



Get your listing in our directory

To advertise here contact Chris

chris.mitchell@futurenet.com

+44 (0)1225 687832

COURSE LISTINGS



Featured: Makers Academy

www.makersacademy.com

Twitter: @makersacademy

Facebook: MakersAcademy

About us

Makers Academy is a fully immersive, 12-week computer programming boot camp. With their help, you will learn the principles of software craftsmanship and they'll also help you get your first job. They're Europe's number-one developer boot camp, running highly selective classes of the offline course every six weeks, and a

remote course every 12 weeks. They take a 'learn by doing' approach, through project-based work. Students are encouraged to work in pairs on coding challenges, with weekly tests, culminating in a final project. They help set up job interviews via their network of hiring partners including ThoughtWorks and Deloitte Digital.

What we offer

- **On-site:**
12 week full-time coding course from the on-site campus in London
- **Remote:**
12 week full-time coding course remotely from home

5 tips from the pros

1. Research all your options

Do your research into lots of different boot camps, read the reviews, read the student blogs and reach out to previous graduates and speak to them.

2. Dabble in code

Although the course is for beginners, it's important that you've started to at least try to learn to code on your own.

3. Prepare for the interview

We send you everything to prepare for the interview. Make

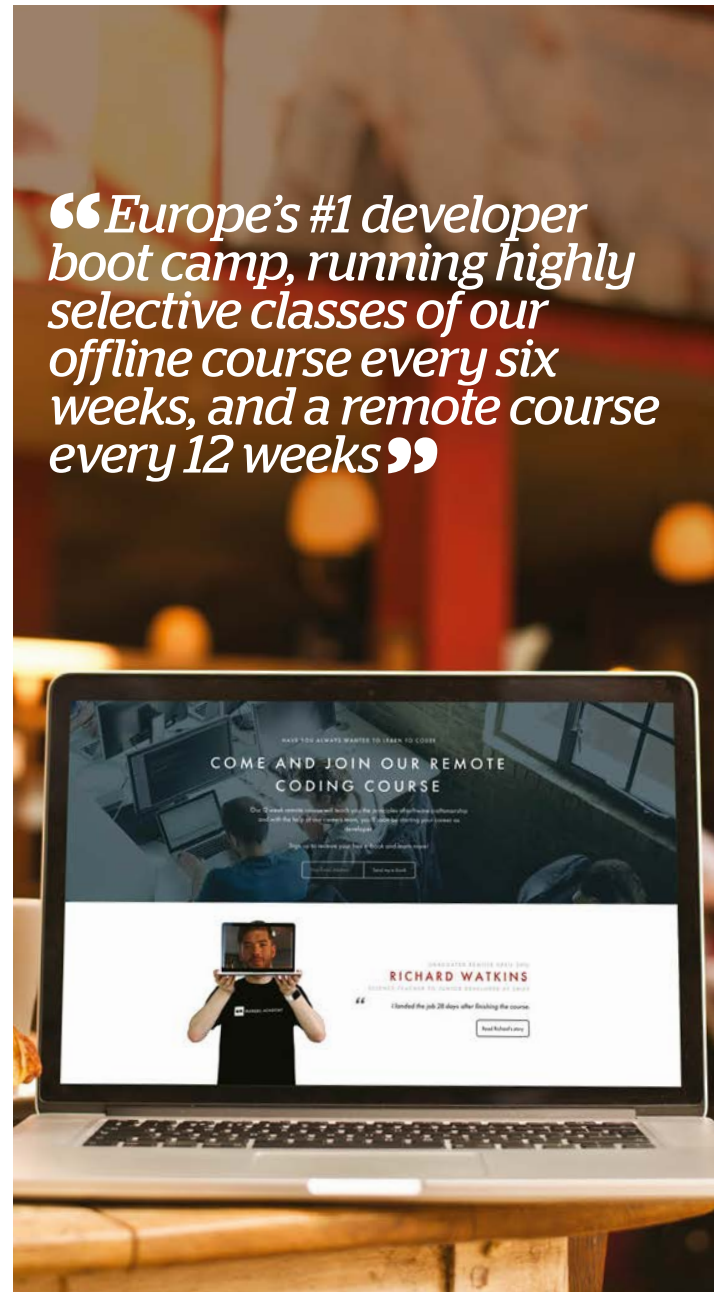
sure you go through all the resources and give yourself two weeks to prepare.

4. Budget

The course is full-time for three months and it can take up to three months after to secure a job. It's important to financially plan for the period you won't be working.

5. Visit us!

Book a visit and come visit us! Come see the Makers Academy HQ in person and learn more.



“Europe’s #1 developer boot camp, running highly selective classes of our offline course every six weeks, and a remote course every 12 weeks”



Richard Watkins

Science teacher to junior developer at Shift

Makers Academy was frustrating and daunting but amazing and I wouldn't change any of it. I landed the job 28 days after finishing the course.



Ina Tsetsova:

Email campaign manager to graduate software developer at ThoughtWorks

I found a really nice community and I've met really cool people. I got a job quicker than I expected.



Hannah Carney

3D designer to junior developer at Play Consulting

Makers Academy not only focuses on your learning for code, but they also focus on your well-being. Work feels like fun and I've finally found a job I love.



Get your listing in our directory

To advertise here contact Chris

chris.mitchell@futurenet.com

+44 (0)1225 687832



UDEMY

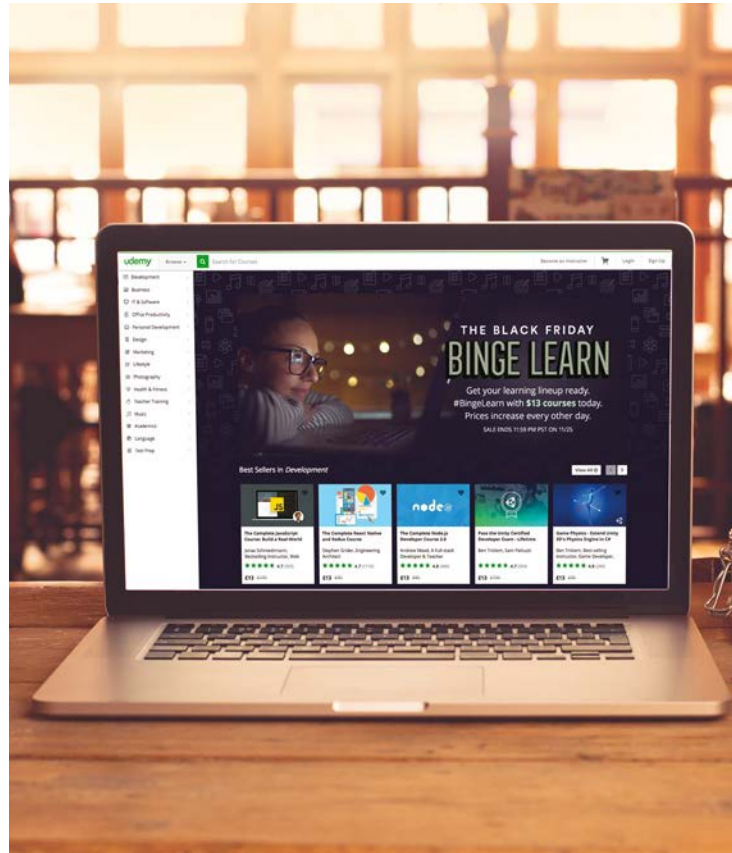
www.udemy.com

Twitter: @udemy

Facebook: udemy

The inspiration for Udemy began in a small village in Turkey, where founder Eren Bali grew up frustrated by the limitations of being taught in a one-room school house. Realising the potential of learning on the internet, he set out to make quality education more accessible. Udemy is now a global marketplace for learning and teaching online. Students can master new skills by choosing from an extensive library of over 40,000 courses including HTML, CSS, UX, JavaScript and web development.

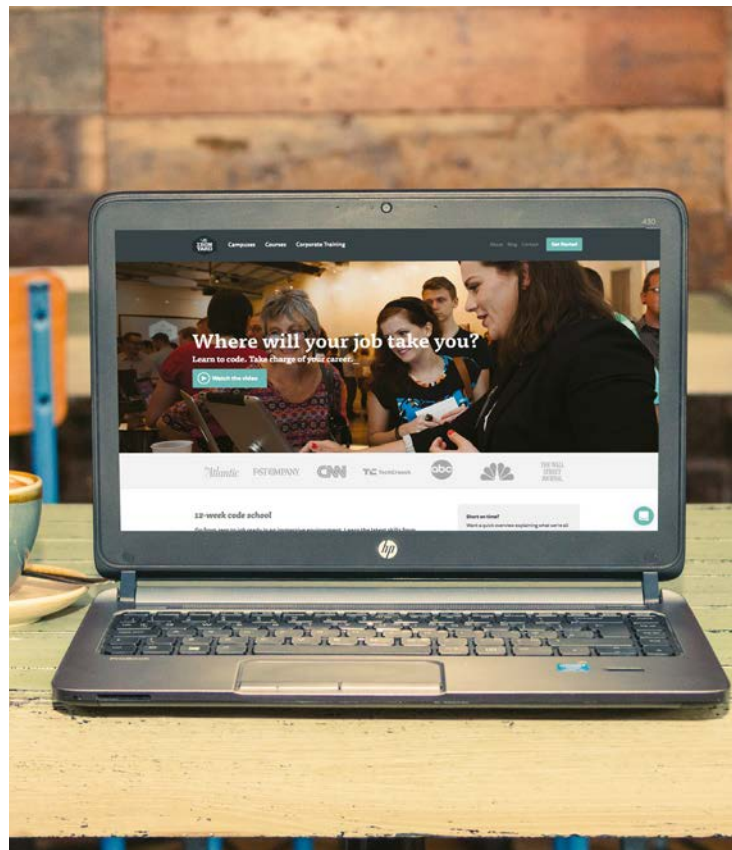
40,000+ courses: There is a course for every designer and dev
Self-paced learning: Learn how to code at your own pace



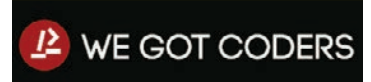
Twitter: @TheIronYard
Facebook: TheIronYard

The Iron Yard is one of the world's largest and fastest-growing in-person code schools. It offers full-time and part-time programs in Back-End Engineering, Front-End Engineering, Mobile Engineering and Design. The Iron Yard exists to create real, lasting change for people, their companies and communities through technology education. The in-person, immersive format of The Iron Yard's 12-week courses helps people learn to code and be prepared with the skills needed to start a career as junior-level software developers.

12-week code school: Learn the latest skills from industry pros
Free crash courses: One-night courses, the perfect way to learn



WEGOTCODERS



www.wegotcoders.com

hello@wegotcoders.com

We Got Coders is a consultancy that provides experts in agile web development, working with startups, agencies and government. Take one of their 12-week training course that covers all that is required to become a web developer, with highly marketable full-stack web development skills.

- Classroom-based training
- Real-world work experience
- Employment opportunities

FUTURELEARN



www.futurelearn.com

feedback@futurelearn.com

Choose from hundreds of free online courses: from Language & Culture to Business & Management; Science & Technology to Health & Psychology. Learn from the experts. Meet educators from top universities who'll share their experience through videos, articles, quizzes and discussions.

- Learn from experts
- Free courses
- All-device access

GYMNASIUM



www.thegymnasium.com

help@thegymnasium.com

Gymnasium offers free online courses designed to teach creative professionals in-demand skills. Courses are all self-paced and taught by experienced practitioners with a passion for sharing practical lessons from the design trenches.

- Gain real-world skills
- Get expert instruction
- Career opportunities



Get your listing in our directory

To advertise here contact Chris

chris.mitchell@futurenet.com

+44 (0)1225 687832

HOSTING LISTINGS



Featured host: Netcetera

www.netcetera.co.uk
0800 808 5450

About us

Formed in 1996, Netcetera is one of Europe's leading web hosting service providers, with customers in over 75 countries worldwide.

As the premier provider of data centre colocation, cloud hosting, dedicated servers and managed web hosting services in the UK, Netcetera offers an array of services designed to more effectively manage IT

infrastructures. A state-of-the-art data centre environment enables Netcetera to offer your business enterprise-level colocation and hosted solutions. Providing an unmatched value for your budget is the driving force behind our customer and managed infrastructure services. From single server to fully customised data centre suites, we focus on the IT solutions you need.

What we offer

- **Managed hosting** - A full range of solutions for a cost-effective, reliable, secure host.
- **Cloud hosting** - Linux, Windows, Hybrid and Private Cloud Solutions with support and scalability features.
- **Data centre colocation** - Single server through to full racks with FREE setup and a generous bandwidth.
- **Dedicated servers** - From QuadCore up to Smart Servers with quick setup and fully customisable.

5 tips from the pros

1. Reliability, trust, support

Reliability is a major factor when it comes to choosing a hosting partner. Netcetera guarantees 100% uptime, multiple internet routes with the ability to handle DDOS attacks, ensuring your site doesn't go down when you need it.

2. Secure & dependable

Netcetera prides itself on offering its clients a secure environment. It is accredited with ISO 27001 for Security along with the options of configurable secure rackspace available in various configurations.

3. 24/7 Technical support

Netcetera has a committed team of knowledgeable staff available 24/7 to

provide you with assistance when you need it most. Our people make sure you are happy and your problems are resolved as quickly as possible.

4. Value for money

We do not claim to be the cheapest service available, but we do claim to offer excellent value for money. We also provide a price match on a like-for-like basis, as well as a price guarantee for your length of service.

5. Ecofriendly

Netcetera's environmental commitment is backed by use of ecocooling and hydroelectric power. This makes Netcetera one of the greenest data centres in Europe.



Testimonials

Roy T

"I have always had great service from Netcetera. Their technical support is second to none. My issues have always been resolved very quickly."

Suzu B

"We have several servers from Netcetera and their network connectivity is top-notch, with great uptime and speed is never an issue. Tech support is knowledgeable and quick in replying. We would highly recommend Netcetera."

Steve B

"We put several racks into Netcetera, basically a complete corporate backend. They could not have been more professional, helpful, responsive or friendly. All the team were an absolute pleasure to deal with, and nothing was too much trouble, so they matched our requirements 100%."



Get your listing in our directory

To advertise here contact Chris

chris.mitchell@futurenet.com

+44 (0)1225 687832

Supreme hosting



www.cwcs.co.uk
08001777 000

CWCS Managed Hosting is the UK's leading hosting specialist. They offer a fully comprehensive range of hosting products, services and support. Their highly trained staff are not only hosting experts, they're also committed to delivering a great customer experience and are passionate about what they do.

- Colocation hosting
- VPS
- 100% Network uptime

UK-based hosting



www.cyberhostpro.com
0845 5279 345

Cyber Host Pro are committed to providing the best cloud server hosting in the UK; they are obsessed with automation. If you're looking for a hosting provider who will provide you with the quality you need to help your business grow then look no further than Cyber Host Pro.

- Cloud VPS servers
- Reseller hosting
- Dedicated servers

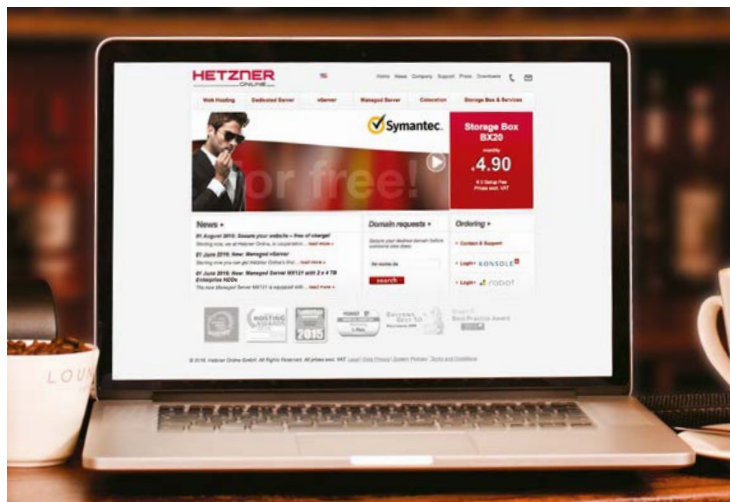
Cluster web hosting



www.fasthosts.co.uk
0808 1686 777

UK based and operating 24/7 from dedicated UK data centres. Fasthosts keep over one million domains running smoothly and safely each day. Services can be self-managed through the Fasthosts Control Panel.

- Dedicated servers
- Cloud servers
- Hosted email



Budget hosting

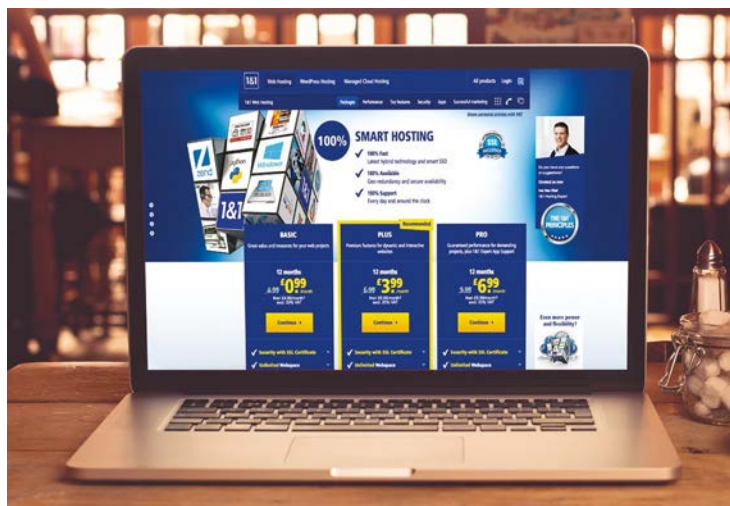


www.hetzner.com
+49 (0)9831 505-0

Hetzner Online is a professional web hosting provider and experienced data centre operator. Since 1997, the company has provided private and business clients

with high-performance hosting products as well as the infrastructure for the efficient operation of sites. A combination of stable technology, attractive pricing, flexible support and services has enabled Hetzner Online to strengthen its market position both nationally & internationally.

- Dedicated/shared hosting
- Colocation racks
- SSL certificates



All-inclusive hosting



www.1and1.co.uk
0333 336 5509

1&1 Internet is a leading hosting provider that enables businesses, developers and IT pros to succeed online. Established in 1988, 1&1 now

operates across ten countries. With a comprehensive range of high-performance and affordable products, 1&1 offers everything from simple domain registration to award-winning website building tools, eCommerce packages and powerful cloud servers.

- Easy domain registration
- Professional eShops
- High-performance servers

SSD web hosting



www.bargainhost.co.uk
0843 289 2681

Since 2001, Bargain Host have campaigned to offer the lowest possible priced hosting in the UK. They have achieved this goal successfully and built up a large client database which includes many repeat customers. They have also won several awards for providing an outstanding hosting service.

- Shared hosting
- Cloud servers
- Domain names

Value Linux hosting



patchman-hosting.co.uk
01642 424 237

Linux hosting is a great solution for home users, business users and web designers looking for cost-effective and powerful hosting. Whether you are building a single-page portfolio, or you are running a database-driven eCommerce website, there is a Linux hosting solution for you.

- Student hosting deals
- Site designer
- Domain names

Flexible cloud servers



elastichosts.co.uk
020 7183 8250

ElasticHosts offer simple, flexible and cost-effective cloud services with high performance, availability and scalability for businesses worldwide. Their team of engineers provide excellent support 24/7 over the phone, email and ticketing system.

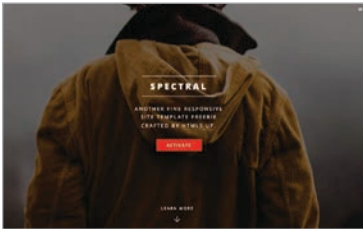
- Cloud servers with any OS
- Linux OS containers
- 24/7 expert support

Free with your magazine

Instant access to these creative resources...

Essential assets and templates

Get textures, fonts, backgrounds and more



Exclusive video tutorials

Learn to code/create with HTML, CSS, JS & PHP



Tutorial project files

All the assets you'll need to follow our tutorials



Plus, all of this is yours too...

- All-new tutorial files to help you master this issue's HTML, CSS and JavaScript techniques
- 115 minutes of expert Polymer.js videos from Pluralsight (www.pluralsight.com)
- 200 Futuristic light leaks for Photoshop from SparkleStock (www.sparklestock.com)
- 38 Vintage noise textures from Slogan Wanted

Log in to www.filesilo.co.uk/webdesigner

Register to get **instant access** to this pack of must-have creative resources, how-to videos and tutorial assets

**Free
for digital
readers too!**
Read on your tablet,
download on your
computer



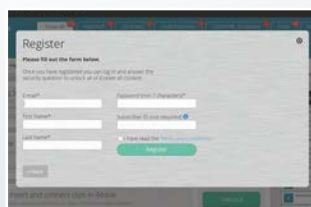


The home of great downloads – exclusive to your favourite magazines from Future!

- Secure and safe online access, from anywhere
- Free access for every reader, print and digital
- Download only the files you want, when you want
- All your gifts, from all your issues, in one place

Get started

Everything you need to know about accessing your FileSilo account



01 Follow the instructions on screen to create an account with our secure FileSilo system. Log in and unlock the issue by answering a simple question about the magazine.



02 You can access FileSilo on any computer, tablet or smartphone device using any popular browser. However, we recommend that you use a computer to download content, as you may not be able to download files to other devices.



03 If you have any problems with accessing content on FileSilo, take a look at the FAQs online or email our team at the address below.

filesilohelp@imagine-publishing.co.uk

An incredible gift for subscribers



Subscribe today & unlock the free gifts from more than 45 issues

Access our entire library of resources with a money-saving subscription to the magazine – that's more than 900 free resources

Over 45 hours of video guides

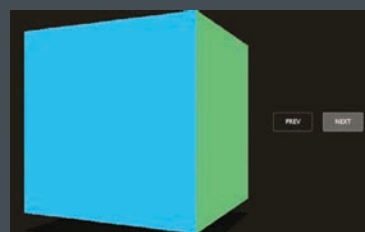
Let the experts teach you to create and code

More than 400 tutorials

Get the code you need to get creative

Over 210 creative assets

Templates, fonts, textures and backgrounds



Head to page 32 to subscribe now



Already a print subscriber?
Here's how to unlock FileSilo today...

Unlock the entire Web Designer FileSilo library with your unique Web ID – the eight-digit alphanumeric code printed above your address details on the mailing label of your subscription copies – also found on any renewal letters.

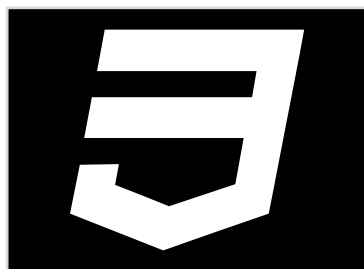
More than 900 reasons to subscribe

+ More added every issue



WEB DESIGNER'S GUIDE TO PHOTOSHOP

Learn the essential tricks, tips and techniques that Photoshop and Adobe products can bring to your projects



BUILD PERFECT LAYOUTS

Get down and dirty with the intricacies of Flexbox



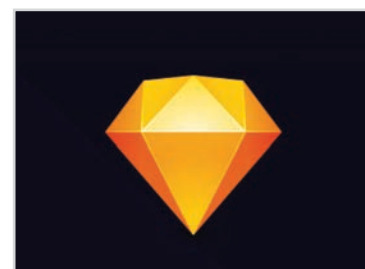
CREATE DYNAMIC 3D HEADERS

Employ the power of the PixiJS library to add imagination



10 BEST NODE.JS FRAMEWORKS

Discover the best frameworks and libraries to enhance your projects



DESIGN UI WITH SKETCH

Learn how to use Sketch to create engaging interface elements

Visit the **WEB DESIGNER** online shop at
www.imagineshop.co.uk
for back issues, books and merchandise

ALL IN YOUR NEXT
WEB DESIGNER
Issue 260 on sale
Thursday 30th March 2017



Ultra Secure Public Cloud

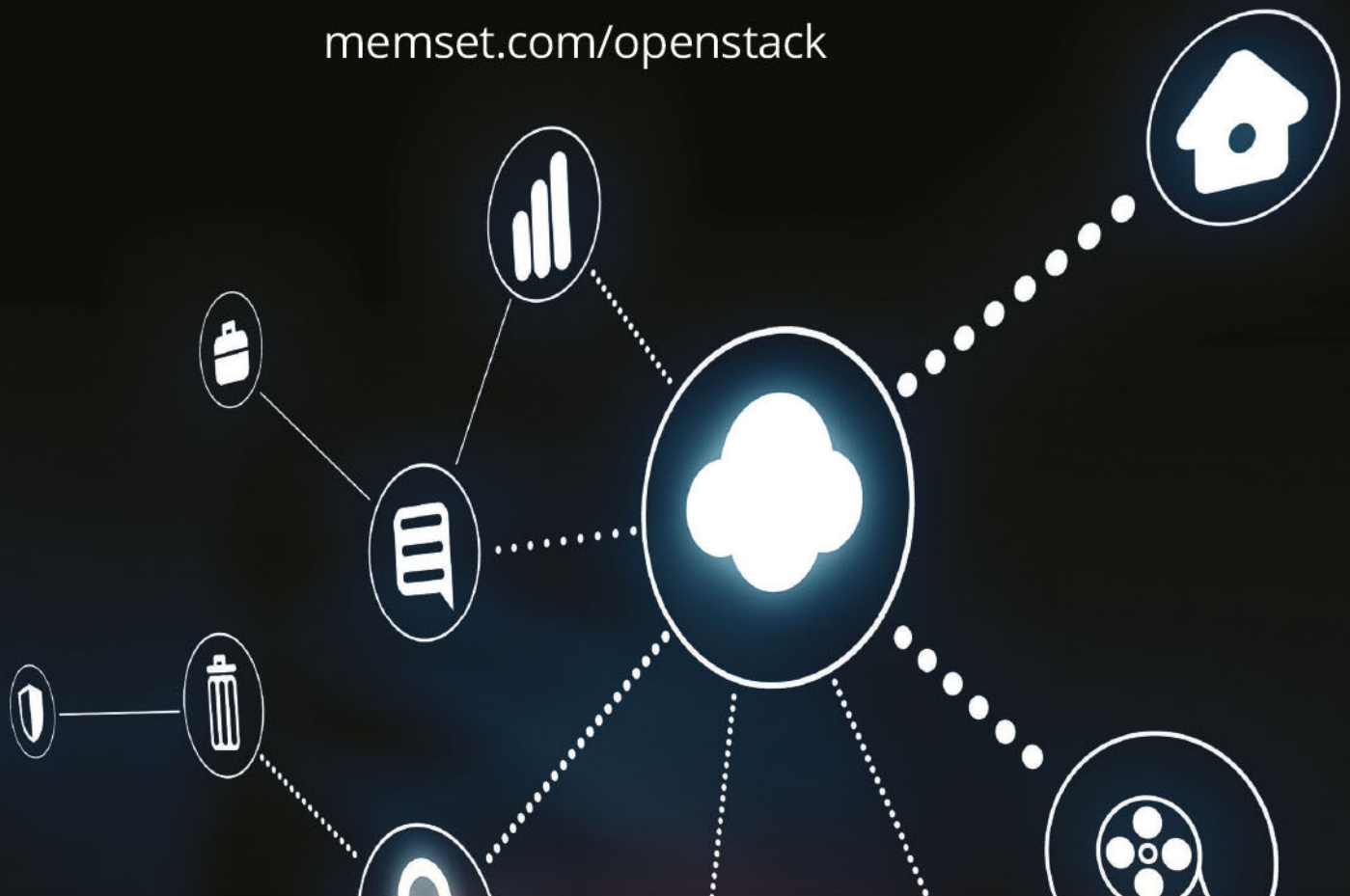
POWERED BY



Take a test drive now

FREE TRIAL

memset.com/openstack



Find out more at www.memset.com
or email sales@memset.com or call free on 0800 634 9270

Network security

Firewall.

Rules
(incoming)

	Name	Source IP	Destination IP	Source port	Destination port	Protocol	TCP flags	Action	
#1	icmp	0.0.0.0/0	0.0.0.0/0	0-65535	0-65535	icmp	ack fin rst psh urg	accept	↓ ↑ ×
#2	ssh	85.10.212.62/32	0.0.0.0/0				fin rst psh urg	accept	↓ ↑ ×
#3	smtp	0.0.0.0/0					ack fin rst psh urg	accept	↓ ↑ ×
#4	http	0.0.0.0/0					fin rst psh urg	accept	↓ ↑ ×
#5	pop3	0.0.0.0/0					ack fin rst psh urg	accept	↓ ↑ ×
#6	imap	0.0.0.0/0					fin rst psh urg	accept	↓ ↑ ×
#7	tcp established	0.0.0.0/0				tcp	rst	accept	↓ ↑ ×

➕ Add rule



e.g. Dedicated Root Server PX61-NVMe

Intel® Xeon® E3-1275 v5
Quad-Core Skylake Processor
64 GB DDR4 ECC RAM
2 x 512 GB NVMe Gen3 x4 SSD
Guaranteed 1 Gbit/s bandwidth
100 GB Backup Space
30 TB traffic inclusive*
No minimum contract
Setup Fee £101.00

monthly £ **50**

Free Firewall for Your Dedicated Root Servers!

Hetzner Online's stateless firewall is a free security solution for your dedicated root server. Starting now on the customer interface Robot, you can use the firewall feature to define your own filtering settings for traffic, such as the originating IPv4 address or TCP/UDP sender port. With this feature, Hetzner Online helps you protect your dedicated root server from Internet dangers. **And it is naturally free of cost.**

www.hetzner.de/gb

* There are no charges for overage. We will permanently restrict the connection speed if more than 30 TB/month are used. Optionally, the limit can be permanently cancelled by committing to pay £1 per additional TB used.

All prices exclude VAT and are subject to the terms and conditions of Hetzner Online GmbH. Prices are subject to change. All rights reserved by the respective manufacturers. Intel, Intel Logo, Intel Xeon and Xeon Inside are brands of the Intel Corporation in the USA or other countries.