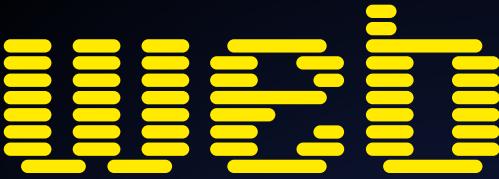


FREE 340 CREATIVE DESIGN RESOURCES



Expert tutorials, techniques and inspiration

designer
TM

JAVASCRIPT

2019

25 FRAMEWORKS, TOOLS & APIs YOU NEED FOR THE NEW YEAR

**NEW MEDIA
QUERY**

Implement 'prefers-color-scheme' with CSS custom properties

WORDPRESS 5

GET A CLOSER LOOK AT GUTENBERG

**CSS LIQUID
EFFECTS**

Apply SVG filters to add fluidity

REACT AND REDUX

How to manage asynchronous actions with Thunk



BUILD A WEBVR SCENE

Use the power of three.js to add another dimension

STORM

FASTER EVERYTHING

Introducing STORM, a Hosting Platform that helps your agency and websites run smoother. 82% of our customers have seen a 25% improvement in site speeds.

OVER 130



GOOGLE
REVIEWS

SPEND LESS TIME ON SERVER MANAGEMENT.

- ⚡ Improve your page load times
- ⚡ Streamline your workflow
- ⚡ More time on web development

⚡ *STORM, what a game changer! It's made everything quicker in our agency and made our sites faster too. ⚡*

INSITES DIGITAL

*Terms and conditions apply, please visit the website for more information



STORM
FASTER EVERYTHING

14 DAY

★ FREE TRIAL ★
nim.host/STORMtrial

Call, email or visit us at

0203 126 6740
sales@nimbushosting.co.uk
nim.host/clientstories

 **Nimbus Hosting**

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
Editor

Celebrate good times



The web is always moving forward. There isn't a day that goes by without some code genius somewhere tweaking, creating or simply making a codebase better, and JavaScript is no different from the rest.

There is always an update to a library, a new tool to make your coding life better, a new API ready to change the way you code your projects. With 2019 here, we thought it would be a good time to check out what JavaScript has to offer for the year ahead. In our lead feature (page 40) Matt Crouch looks at 25 of the best APIs, frameworks and tools that you should be thinking of using in the year ahead.

WordPress has had a major facelift with the release of version 5, so we see what Gutenberg has to offer and take a look at some of the other new features that have been overshadowed a little. We discuss what will happen to TinyMCE and take a quick peek at keyboard shortcuts, and how to convert older content to work better.

Elsewhere we have a host of top-quality tutorials to whet the appetite. Learn how to apply stunning SVG effects to text and images with the help of CSS, find out how to use the new media query 'prefers-color-scheme' to implement light or dark themes in CSS, and discover how to create a WebVR scene using the ever-popular JavaScript 3D library three.js. Plus, in the latest tutorial in our React series we show you how to manage actions with Redux Thunk. Enjoy.

While the concept of a web component has been around since 2011, the specifications have constantly changed. In 2018 things settled down

Highlight

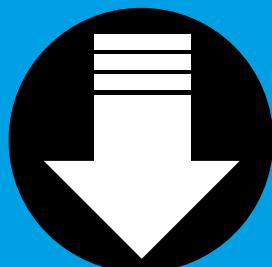


Solutions tend to span across many channels. However, there is not a 'one size fits all' solution for designing across multiple user interfaces

Web Designer talks to This Place, an agency where agility, strategy and design collide.
Page 30

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

Visit our blog for opinion, freebies & more www.creativebloq.com



FREE - exclusive with this issue
340 Designer resources

Assets - 200 Vibrant Overlays, 16 minimalist business card mockups, 61 actions, hi-res textures and double exposure gradients, 50 glossy blood splatter brushes from Sparklestock
- Tutorial files and assets



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...



Matt Crouch

Matt is a software engineer at Vidsy in London and regular contributor to **Web Designer**. JavaScript is constantly growing and getting more powerful. In this issue, Matt takes a look at what the language holds for the year ahead and picks the best APIs, frameworks and tools, and how they can benefit your next project. **Page 40**

Web apps are now first-class citizens. It's great to see browsers gain the tools they need for developers to make quality, native-like experiences accessible to all.

Sarah Maynard



Sarah is a freelance WordPress developer and digital illustrator based in Wiltshire. She loves to encourage others to get started with WordPress. In this article she covers the brand new features coming in WordPress 5.0. **Page 84**

Mark Shufflebottom



Mark is a professor of Interaction Design at Sheridan College near Toronto. In this issue, Mark takes a look at creating SVG filters and applying these through CSS to elements in order to create unique and original styling. **Page 50**

Richard Mattka



Richard is an award-winning interactive director, designer and developer. In this issue Richard demonstrates how you can create a WebVR experience, using the popular JavaScript 3D library three.js. **Page 66**

Steven Roberts



Steven is a designer and developer from Teesside, currently heavily invested in Red Dead Redemption 2. In this issue he'll be teaching you how to implement light or dark mode using a new media query. **Page 62**

Leon Brown



Leon is a freelance web developer and trainer who assists web developers in creating efficient code for projects. This issue he recreates a host of techniques as inspired by the top-class sites seen in Lightbox. **Page 19**

David Howell



David is a journalist with over 20 years' experience in publishing and runs his own business, Nexus Publishing. In this issue he takes time out to talk to the talented crew at global agency This Place and finds out how they operate. **Page 30**

Tam Hanna



Tam has a healthy dislike of all things sharing. Letting others harness the fruits of his hard development work is kryptonite to him. This issue he shows how JavaScript obfuscation scratches that itch. **Page 78**

Mark Billen



Mark is a freelancer writer who has been writing about web design and technology for over 15 years. In this issue he takes a peek at some of the best-looking sites on the web and gives you a little insight into how they were made. **Page 16**

Follow us!

Facebook: [www.facebook.com/
WebDesignerUK](http://www.facebook.com/WebDesignerUK)
Twitter: [twitter.com/
webdesignermag](http://twitter.com/webdesignermag)

Future PLC Richmond House, 33 Richmond Hill, Bournemouth, Dorset, BH2 6EZ

Editorial

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

Art Editor Andy Salter
Acting Art Editor in Chief Claire Howlett
Senior Art Editor Will Shum
Production Editor James Price

Contributors

Damian Rees, Mark Billen, Leon Brown, David Howell, Matt Crouch, Mark Shufflebottom, Richard Mattka, Steven Roberts, Tam Hanna, Sarah Maynard

Photography

James Sheppard
All copyrights and trademarks are recognised and respected

Advertising

Media packs are available on request
Commercial Director Clare Dove
clare.dove@futurenet.com
Senior Advertising Manager Mike Pyatt
michael.pyatt@futurenet.com
01225 687538

Account Director George Lucas
george.lucas@futurenet.com

International Licensing

Web Designer is available for licensing. Contact the Licensing team to discuss partnership opportunities.
Head of Print Licensing Rachel Shaw
licensing@futurenet.com

Subscriptions

Email enquiries contact@myfavouritemagazines.co.uk
UK orderline & enquiries 0344 848 2852
Overseas order line and enquiries +44 (0) 344 848 2852
Online orders & enquiries www.myfavouritemagazines.co.uk
Head of subscriptions Sharon Todd

Circulation

Head of Newtrade Tim Mathers

Production

Head of Production Mark Constance
Production Project Manager Clare Scott
Advertising Production Manager Joanne Crosby
Digital Editions Controller Jason Hudson
Production Manager Nola Cokely

Management

Brand Director Matthew Pierce
Chief Content Officer Aaron Asadi
Commercial Finance Director Dan Jotcham
Head of Art & Design Greg Whittaker

Printed by William Gibbons, 28 Planetary Road, Willenhall, WV13, 3XT

Distributed by Marketforce, 5 Churchill Place, Canary Wharf, London, E14 5HU www.marketforce.co.uk Tel: 0203 787 9001

ISSN 1745-3534

We are committed to only using magazine paper which is derived from responsibly managed, certified forestry and chlorine-free manufacture. The paper in this magazine was sourced and produced from sustainable managed forests, conforming to strict environmental and socioeconomic standards. The manufacturing mill holds full FSC (Forest Stewardship Council) certification and accreditation.

All contents © 2018 Future Publishing Limited or published under licence. All rights reserved. No part of this magazine may be used, stored, transmitted or reproduced in any way without the prior written permission of the publisher. Future Publishing Limited (company number 2008865) is registered in England and Wales. Registered office: Quay House, The Ambury, Bath BA1 1UA. All information contained in this publication is for information only and is, as far as we are aware, correct at the time of going to press. Future cannot accept any responsibility for errors or inaccuracies in such information. You are advised to contact manufacturers and retailers directly with regard to the price of products/services referred to in this publication. Apps and websites mentioned in this publication are not under our control. We are not responsible for their contents or any other changes or updates to them. This magazine is fully independent and not affiliated in any way with the companies mentioned herein.

If you submit material to us, you warrant that you own the material and/or have the necessary rights/permissions to supply the material and you automatically grant Future and its licensees a licence to publish your submission in whole or in part in any/all issues and/or editions of publications, in any format published worldwide and on associated websites, social media channels and associated products. Any material you submit is sent at your own risk and, although every care is taken, neither Future nor its employees, agents, subcontractors or licensees shall be liable for loss or damage. We assume all unsolicited material is for publication unless otherwise stated, and reserve the right to edit, amend, adapt all submissions.

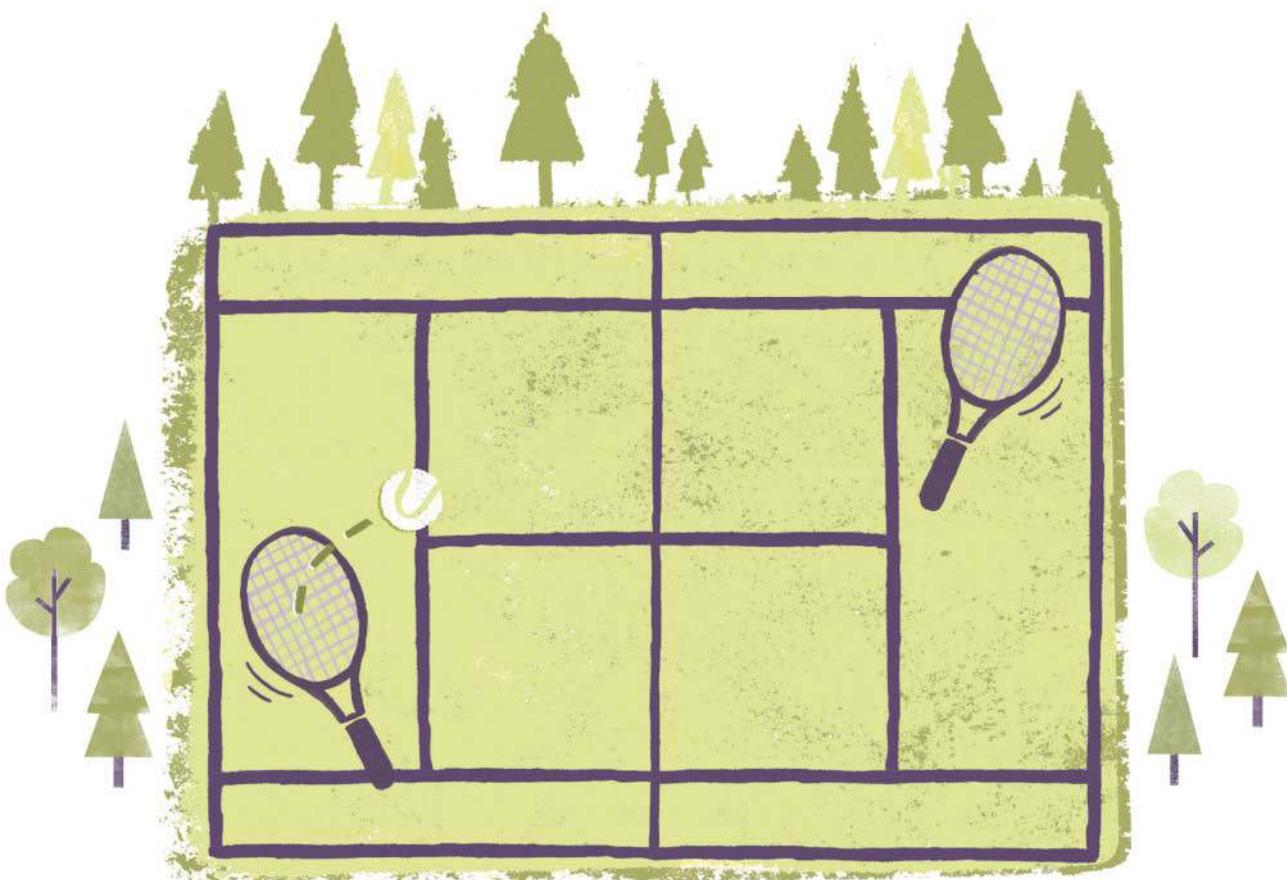


Future plc is a public company quoted on the London Stock Exchange (symbol: FUTR)
www.futureplc.com

Chief executive Zillah Byng-Thorne
Non-executive chairman Richard Huntingford
Chief financial officer Penny Ladkin-Brand
Tel +44 (0)1225 442 244

45 THOUSAND

TENNIS COURTS EVERY DAY!



Did you know that European forests, which provide wood for making paper and many other products, have been growing by over 45 thousand tennis courts 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@futurenet.com • [@WebDesignerMag](https://www.twitter.com/@WebDesignerMag) • www.creativebloq.com

08 Web browser unity?

Edge is set to switch to Chromium. Is this a good or bad thing? **Web Designer** takes a closer look

11 Putting humans first

UX Director Damian Rees talks about the risks of placing technology before users

12 Webkit: The best must-try resources out there

Discover the libraries and frameworks you need

16 Lightbox

A showcase of inspirational sites and the techniques used to create them

30 A place for design

Agility, strategy and design collide at global agency This Place. **Web Designer** says hello

38 Back issues

Catch up on any issues of **Web Designer** that you've missed by downloading a digital edition

40 JavaScript 2019

There are a host of great new features heading to the browser. We list 25 APIs, frameworks and tools that you need to try in the year ahead

84 What's new in WordPress 5.0?

WordPress is getting a complete facelift. See what's changed in the new version of the CMS

92 Hosting listings

An extensive list of web hosting companies. Pick the perfect host for your needs

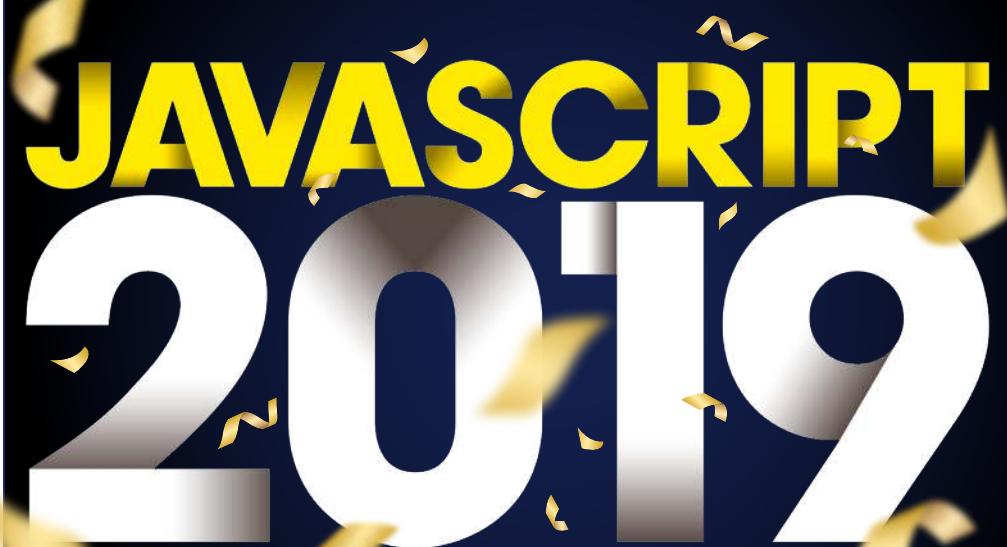
94 Course listings

Want to start learning? Check out what courses are out there with this list

98 Next month

See what's in the next issue of **Web Designer**

Cover focus



40



16



20

TUTORIALS

50 Apply SVG filters with CSS

Add special effects to text and images

56 Get started with Babel 7

Discover how to use the latest version

62 Light or dark mode with CSS

Introduce different themes with the help of new media query 'prefers-color-scheme'

66 Create WebVR using three.js

Build a WebVR scene with the three.js library

70 Message board text effect

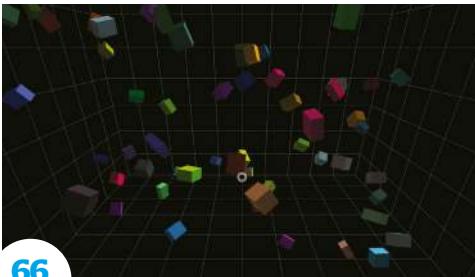
Animate text like an airport message board

72 Manage with Redux Thunk

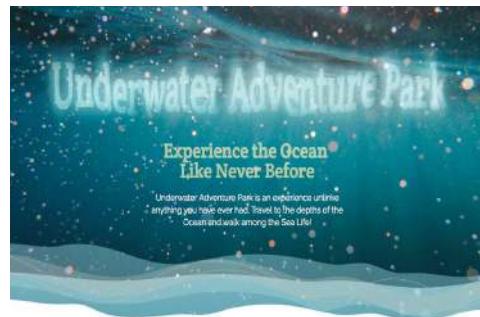
Communicate with the server and keep up to date without getting tangled in state

78 Protect your JavaScript code

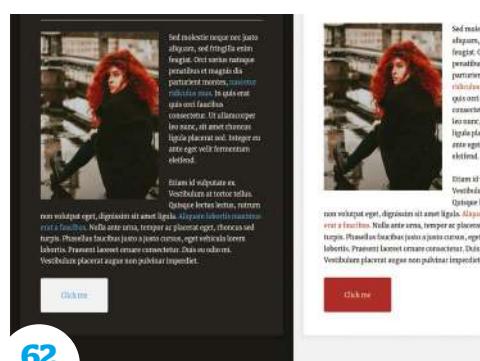
Everyone can view your source code. Introduce obfuscation to keep it from prying eyes



66



50



62



70



30

FILESILO

96 Get the latest must-have resources and videos

- 200 Vibrant Overlays, 16 minimalist business card mockups, 61 actions, textures and gradients, and 50 blood splatter brushes from Sparklestock



myfavouritemagazines

Visit the **Web Designer** online shop at
myfavouritemagazines.co.uk

for the latest issue, back issues and specials

DIGITAL EDITION

Do you want to get your hands on a digital edition? Head to your preferred app store, download, install and purchase the issue of choice

Google Play – bit.ly/2wetvGp
iTunes – apple.co/2igtBYq



Subscribe today and save

bit.ly/2sGwB3h

Header

The tools, trends and news to inspire your web projects

Microsoft Edge and Chromium unite

What does Edge's move to Chromium mean for designers and developers? **Web Designer** takes a closer look



rowers – everyone is using them, but as we all know some are more popular than others. Chrome is way out in front in the popularity stakes. According to stats from StatCounter, Chrome has a market share of around 60 per cent worldwide. Microsoft's reign in such a position happened a very long time ago (in browser years). As Internet Explorer's popularity waned, Microsoft looked to its new Edge option, but it was all too late. Google had conquered the browser market, and there was very little hope of the software giant ever getting back to the glory days. A quick glance at the stats to the right will show you exactly how popular the IE/Edge browser is in today's market.

Has the popularity of Chrome finally persuaded Microsoft to join its browser party? The software giant recently announced in a post on its Windows Blog that, "We intend to adopt the Chromium

open source project in the development of Microsoft Edge on the desktop to create better web compatibility for our customers and less fragmentation of the web for all web developers." It went on to say, "We intend to become a significant contributor to the Chromium project, in a way that can make not just Microsoft Edge – but other browsers as well – better on both PCs and other devices." You can read the whole post at bit.ly/2QeZFwm.

Sounds like good news for web designers and developers: another browser using the same engine as the world's favourite can only mean less work for those building websites and apps. Better compatibility, fewer hacks and improved support are all on the cards. But does having two of the web's big hitters working together raise any issues? Admittedly, Microsoft is onto a loser either way. Stay with what it has and there is very little hope that its browser will ever

be adopted by the masses. Move to the Chromium camp and it seems to be getting into bed with another tech giant. For Microsoft the move means that it can stop trying to play catch-up and is free to reassign all the good people working on Edge elsewhere.

But, unsurprisingly, the move hasn't met with universal approval. In a post titled 'Goodbye, EdgeHTML' from Mozilla (mzl.la/2QLqfws) they are quick to get to the point: "By adopting Chromium, Microsoft hands over control of even more of online life to Google." It's hard to argue. Web developer, consultant and trainer Tim Kadlec also had something to say in a recent post called 'Risking a Homogeneous Web'. You can read his thoughts at bit.ly/2SWJP6x.

Will the change make much difference? Will Edge users abandon it and go to Chrome? Will Microsoft's contribution make a better more harmonious web? What do you think?

STAT ATTACK

EDGE AND IE

How popular are Microsoft's browsers across the globe?

Worldwide

4.96%



Edge/IE combined is very similar to Firefox

Europe

6.65%



Edge/IE combined is fourth most popular

North America

8.61%



Edge/IE combined sits in third place

Africa

2.76%



Edge/IE combined is seventh most popular

Asia

2.74%



Edge/IE combined is seventh most popular



Subscribe today and save

40%

bit.ly/2qLxVi4

Source: gs.statcounter.com
(correct as of November 2018)

Sites of the month

CLIMATE INFO CONTACT

ENDPAIN / CHARLOTTE'S WEB

EXPRESS DENIM PERFECT

bit.ly/2E4hKpA

Daniel Castrillón brings together different elements to create his unusual characters.



Graphics Mixtape 2: Mutatis

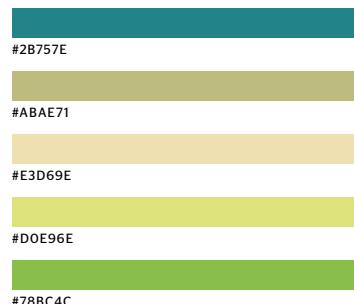
bit.ly/2E4hKpA

Daniel Castrillón brings together different elements to create his unusual characters.



Colour picker Classical Grass

bit.ly/2Uflfhs



Typesetter Flomic

bit.ly/2SyTv76

Flomic is a geometric sans built from folded lines. Use it for headlines to add a unique twist and impact.



WordPress Olvi

bit.ly/2QzVsTw

Big-image slider, neat animations and multiple options makes this ideal for those looking for something creative.

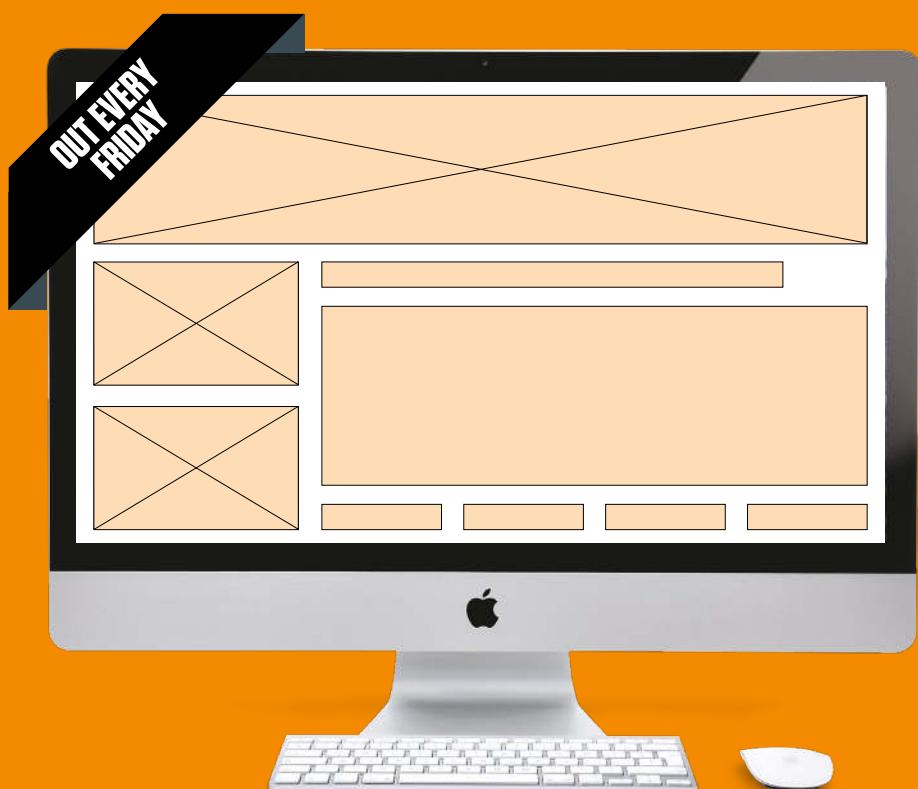


HEAR FROM THE PROS

SIGN UP TO THE

designerTM

NEWSLETTER TODAY!



Get weekly news, tips & inspiration

SIGN UP NOW!

bit.ly/2KI5b7Y

Putting humans first

Place users before technology or risk everything



Damian Rees
Founder & UX Director
experienceux.co.uk

“Human problems are studied first, and then a way to solve those issues with technology follows. Technology is considered last”

In every boardroom across the world, feverish discussions about artificial intelligence, virtual/augmented reality, blockchain, Internet of Things, wearable technology or some other form of ‘disruptive tech’ are taking place right now. Organisations are clamouring to find new and innovative ways to engage with customers and sell more stuff. It’s the natural order of things – to attempt to keep up with the competitors and avoid getting left behind.

The pressure to be first to market and remain a step ahead of their competitors is huge and inevitably leads many to follow the rest of the market. This filters down through the ranks, and several projects kick off with a goal to implement something new and shiny to appease the board. A minute percentage of these projects will be successful, but most will be half-baked, technical vanity projects. When they launch to their intended audience, they are summarily rejected. Why? Because they don’t meet their needs and don’t offer anything different or better than what they already use.

These organisations focus on the technology first, and users are an afterthought. Very few organisations will be considering how the technology impacts their customers, or indeed if their customers want or need it.

The big industry disruptors instead place the focus on humans first. They start with Mary, who lost her bank card on the bus, only to find it down the back of the sofa. She gets to pause her card, instead of permanently cancelling it (thanks, Monzo). They find out the pain Joe goes through when he’s wading through pages of claims forms, and then they work out how to cut it down to three minutes in an app (great job, Lemonade). They look for the problems and negative experiences people are having with companies, then they solve these problems for them in clever ways. It’s a complete reversal of how most businesses work, and we’ve seen this approach bring success over and over.

In Monzo’s case above, it has disrupted the banking industry, not by focusing on technology but by offering customers a solution to problems they were having with traditional banks. High charges for payments abroad, slow updates on pending transactions outside of banking hours and little clarity on spending are all areas that Monzo has worked on. Their solution, while it utilises technology, doesn’t do anything technologically groundbreaking. They simply offer a phone-based app that allows instant updates, categorised spending and free purchases abroad. They are constantly innovating their service, and they are doing extremely well by taking customers away from traditional banks. At the time of writing, Monzo is valued at \$336m and has over 750,000 users.

Lemonade is disrupting the insurance sector in the US at a similar rate. When you think about getting insurance I would guess that most people’s concerns are around the hoops they have to jump through to sign up, and then the concern about actually getting paid when they make a claim. And that’s exactly what Lemonade has focused their offering on. The Lemonade site states that it takes “90 seconds to get insured” and “3 minutes to get paid” when you make a claim. For a human being, the benefits of switching to this type of service are a no-brainer. And it’s no wonder these major disruptors are wiping the floor with their competition.

Lemonade’s niche in the marketplace comes from using technology in innovative ways, but that’s not what they lead with. Behind the scenes, Lemonade relies on AI and a series of anti-fraud algorithms. What makes them different is that human problems are studied first, and then a way to solve those issues with technology follows. Technology is considered last and only as an enabler.

The heart of change within every industry is undeniably coming from technology. I predict the major success stories will be from organisations who use their understanding of technology to solve problems customers have been experiencing in traditional businesses. They will put their users firmly in first place. And those that don’t? I think many of those will face almost certain failure. What do you think?

webkit

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



Photopea

photopea.com

Photoshop is a top-quality image editor, but if you don't need its full range of tools a cheaper alternative is available in the shape of Photopea. This is a powerful and free online image editor

that supports PSD, XCF, Sketch, XD and CDR formats. Create files, adjust images, add layers, introduce filters and much more, and export as PNG, JPEG, SVG, PDF and many more.

TOP 5 CODEPENS

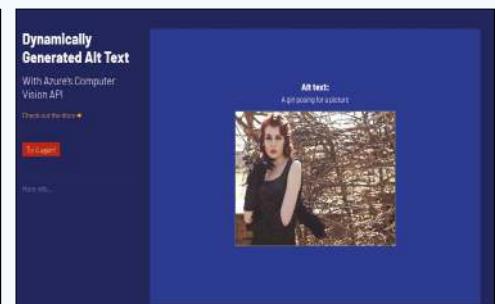
Let this selection of interesting codebases inspire you



Variable font

bit.ly/2ElxCEe

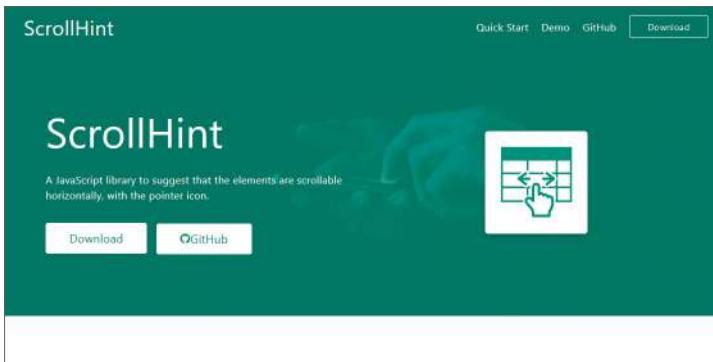
A living, breathing font that is in a constant state of motion. Check out the CSS and JavaScript to see the finer points.



Dynamically Generated Alt Text

bit.ly/2PHcsTh

Instantly create alt text for your images. This codepen uses Azure's Computer Vision API to recognise image elements and create text.



ScrollHint

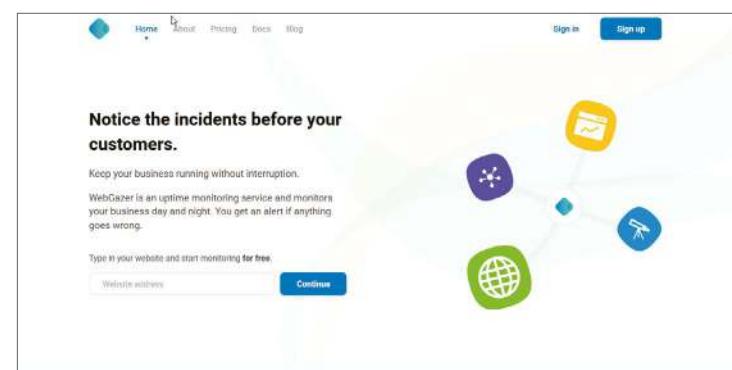
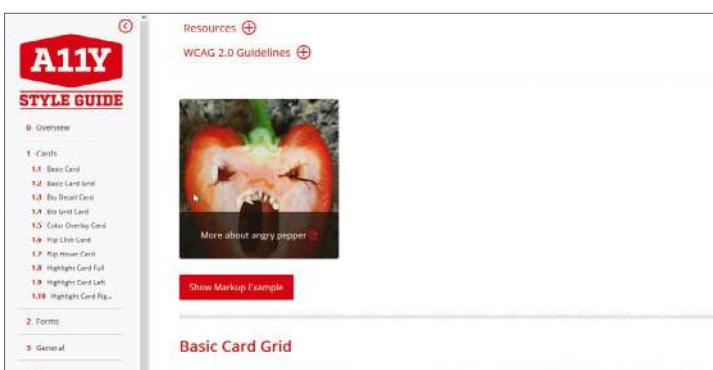
appleple.github.io/scroll-hint

This is a straightforward JavaScript library that hints to the user, using a pointer, that onscreen elements are scrollable horizontally.

Contraste

contrasteapp.com

An app that enables users to quickly know if a combination of colours, for text and background, passes accessibility thresholds as defined by the W3C.



A11Y Style Guide

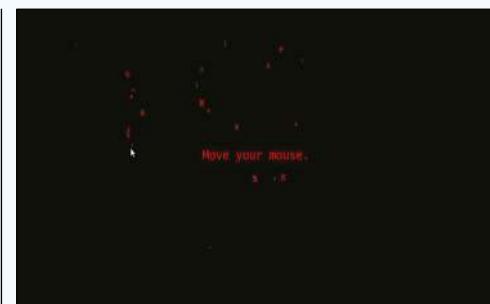
a11y-style-guide.com/style-guide

This is a style guide with an accessibility twist. It offers a host of components, i.e. cards, forms, media, with code examples that include accessibility elements.

WebGazer

webgazer.io

Every website needs to be up and running. This is an uptime monitoring service that will send an alert when something needs sorting.



Rainy Day

bit.ly/2SSZ0xz

This recreates the weather outside, with moving grey clouds coupled with constant rain. Uses canvas and SVG to create.

Virus in the machine

bit.ly/2SPbZjO

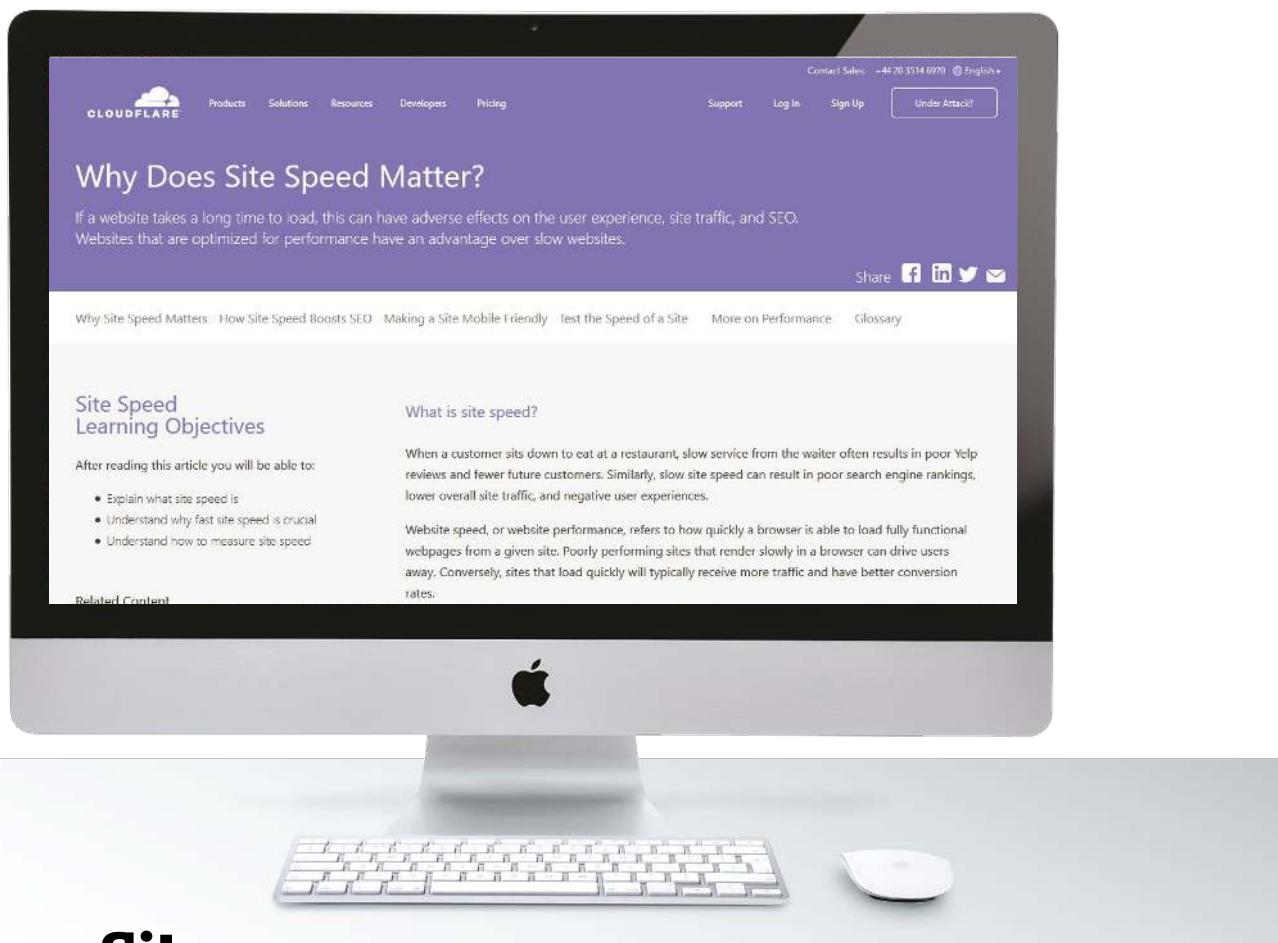
Move the mouse cursor and see a trail of letters and numbers pour onto the screen and rise to the top.

Drone

bit.ly/2Ewh39H

A demo using the three.js unreal bloom effect. Watch the virtual drone fly through the lit arches from all angles.

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



Why Does Site Speed Matter?

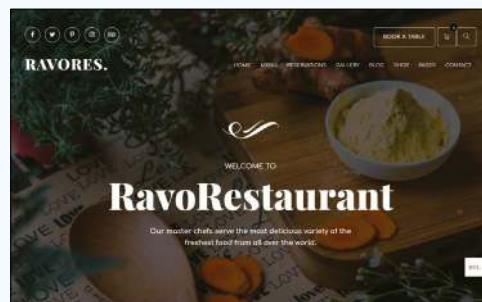
bit.ly/2Evs6iJ

The title gives a big clue to what you can expect from this resource. This is a selection of short articles that informs the user of what they should

be doing. Find out why it's important to SEO, how to improve mobile performance, how to test, and there is also a section on performance.

TOP 5 WORDPRESS THEMES

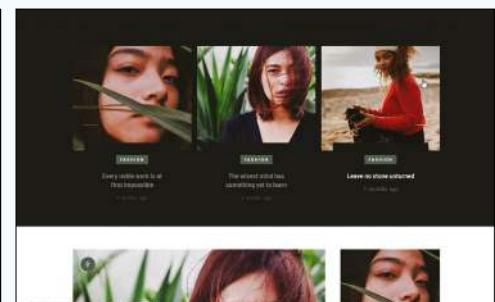
Need a new look for your site? Check out this collection of themes



RavoRes

bit.ly/2SPmeVI

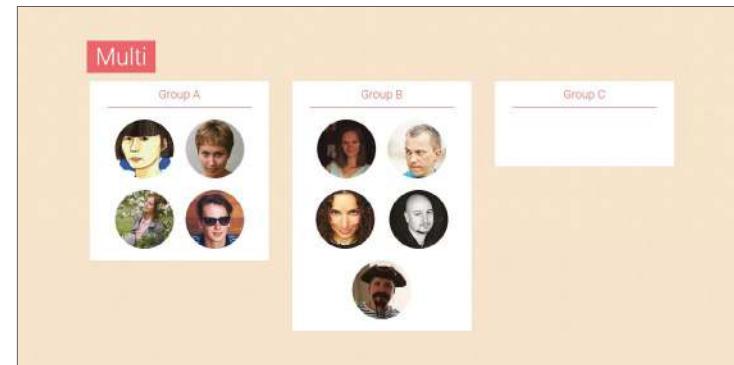
Big, bold images accompanied by big, bold typography are key components of this restaurant-style theme.



Hannah

bit.ly/2LkSJln

A clean and simple theme that makes great use of space, imagery and typography to have an impact.



Slang

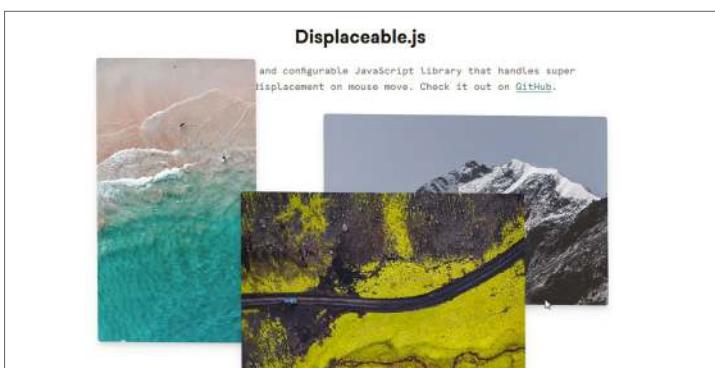
slang.kylestetz.com

Slang is a simple audio programming language implemented in JavaScript. The best part of it is that you can play and create tunes very quickly.

Sortable

sortablejs.github.io/Sortable

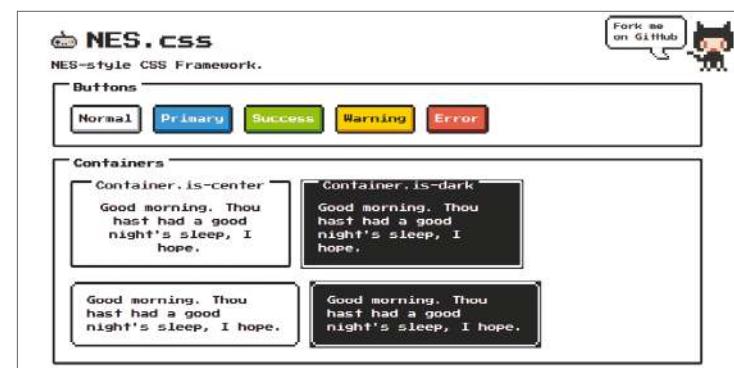
A JavaScript library that sorts items. Check the site to see exactly what it can do, like the Multi option for dragging from box to box.



Displaceable.js

subtlebits.com/projects/displaceable

This is a lightweight configurable JavaScript library that offers very smooth element displacement on mouse movement.



NES.css

nostalgic-css.github.io/NES.css

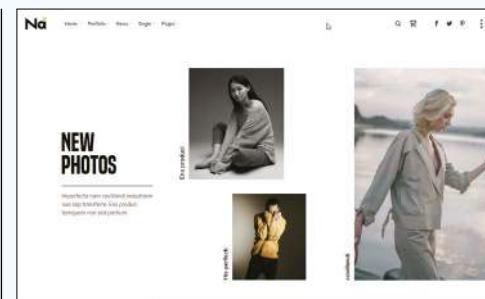
Who doesn't love a bit of retro 8-bit action? We do, and this framework features a collection of brilliant retro-styled components.



BoostUp

bit.ly/2QwzktT

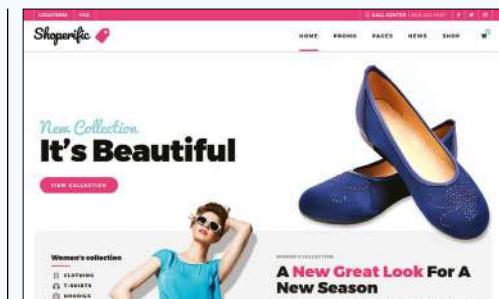
Multiple theme styles aimed at agencies, social media, marketing, advertising and SEO. All are colourful and illustration-led.



TheNa

bit.ly/2R176GY

A clean, image-led photography-style theme, with multiple options for the homepage offering horizontal scrolling.



Shoperific

bit.ly/2ChwO24

A flexible e-commerce site with a host of bold homepage designs that mix a clean style with great colour palettes and imagery.

Fishfinger Creative Agency

fishfinger.me



fishfinger

Welcome
FISHFIN



Designer:

Fishfinger (In-house) fishfinger.me



come to
FINGER

“An award-winning creative studio based in London, Fishfinger specialises in design, motion, web and social projects”

Colours



#100828



#EF2945



#B7DEF7



#F2AF46

Tools

jQuery, Barba.js, skrollr

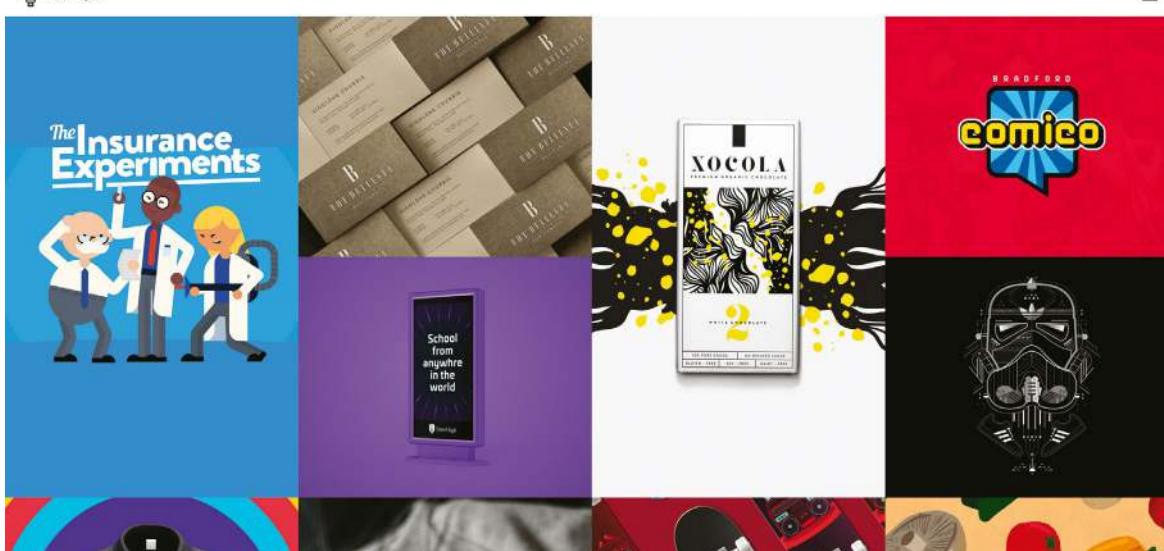
Fonts

abcABC
1234567890
abcABC
1234567890
abcABC
1234567890

TT Norms font, designed by Pavel Emelyanov and Ivan Gladkih for TypeType, is used exclusively in Regular, Medium and Bold.



fishfinger

**Top**

This colourful portfolio site opens on a random selection of intro headers, featuring charming Barba.js animations

Middle

The 'Work' section places project case studies into a gridded layout featuring tiled, rollover previews to click through

Bottom

Selected projects are beautifully presented within long, image-rich and colourful pages that tell a visual story of the delivered assets

Create a wave underline effect for a navigation list

Use this animated underline effect to indicate interactivity of text items

1. Document initiation

The first step is to initiate the HTML document. This consists of the HTML container representing the webpage, containing a head and body section. While the head section is used to load the external CSS stylesheet, the body section is used to store the content elements created in step 2.

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

2. HTML content

The page content consists of a list of links that the wave effect will be applied to. Take note of how the 'waves' class is applied to the list container. This will be used by the CSS stylesheet to find the links the wave effect will be applied to.

```
<ul class="waves">
    <li><a href="#">item</a></li>
    <li><a href="#">item</a></li>
    <li><a href="#">item</a></li>
</ul>
```



**ABOUT
SERVICES
WORK**

CONTACT

3. Default link presentation

Create a new file called 'styles.css'. This step sets all of the links inside the waves container to have spacing above them. Each link item is set to use inline-block as their display mode in order to enable a margin to be applied above them. Relative positioning is also applied to each link so that the wave line can be placed relative to the link location.

```
.waves a{
    display: inline-block;
    position: relative;
    margin-top: 1em;
}
```

4. Default line

The wave line is made using the 'before' and 'after' virtual elements of the link. These only appear when the link is hovered by the user, upon which these virtual elements are set to appear across the bottom of the parent link. Positioning is relative to the parent link due to position mode used in step 3.

```
.waves a:hover::before,
.waves a:hover::after{
    content: "";
    position: absolute;
    display: block;
    width: 100%;
    height: 20px;
    bottom: -5px;
}
```

5. Default line: more

The wave line is made using a background radial gradient that is repeated across the space created by the virtual elements. This background is set to a specific size that shows only half of each circle, with an animation called 'move' applied to infinitely repeat over a duration of one second. This animation is also set to play as linear, as part of the illusion of the wave being infinite.

```
.waves a:hover::before,
.waves a:hover::after{
    background: radial-gradient(16px,
        transparent, transparent 4px, black 4px,
        black 10px, transparent 11px);
    background-size: 30px 40px;
    background-position: 0 0;
    animation: move 1s infinite linear;
}
```

6. After line

The 'after' virtual element requires some modification from the previously defined default settings. Positions for the element and also the background pattern are changed so that the bottom half of the circles are placed to form a wave pattern with the 'before' element. This element is also changed to use the 'move2' animation.

```
.waves a:hover::after{
    height: 20px;
    bottom: -25px;
    left: 0;
    background-position: 15px -20px;
    animation: move2 1s infinite linear;
}
```

7. Animation settings

The final step is to define the keyframe animations that were referenced in steps 5 and 6. Only one frame is required for each animation - the 'to' final frame. This results in the animation moving the horizontal background position from the original position to the positions defined in each animation. By having no first frame to start from, the animations can be reused for elements that have different starting points.

```
@keyframes move{
    to{ background-position-x: 60px; }
}
@keyframes move2{
    to{ background-position-x: 75px; }
}
```

ESPN - The NBA's Best Signature Moves

ESPN

SHOOTING

[SEE EVOLUTION OF SHOOTING](#)



espn.com/espn/feature/story/_/id/24504198

Designer:

Kasper Laigaard kasperlaigaardstudio.com

“Sports network ESPN serves up a series of basketball ‘Game Changers’ with a video-heavy homage to NBA’s hottest shots”

ING

Colours



#FF8838



#52B0DC



#FFB07B



#008ED3

Tools

jQuery, GSAP / TweenMax,
AngularJS

Fonts

abcABC

1234567890

Tungsten font by Hoefler & Co. provides the recognisable bold styling applied to the site's names and headings throughout.

abcABC

1234567890

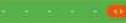
Benton Sans designed by Cyrus Highsmith and David Berlow, from Font Bureau, available via Adobe Fonts, is used on all subsequent text.

GAME CHANGERS

In honor of "Basketball: A Love Story", our experts ranked the signature moves they loved most. Join us as we journey through the decades, showing the evolution of basketball's best go-to dribbles, shots and slams.



DUNKING



MICHAEL JORDAN

MICHAEL JORDAN
The first time Jordan played the Cavaliers – the team he formulated more than any other in his NBA career – he delivered his signature one-handed layup for two points in the 40-point surge. The image of Jordan cracking the ball with his right hand

© 2018 Michael Jordan Foundation. All rights reserved.



1980s

Above

An initial 'Home' splash screen provides some editorial context, social sharing links and a simple slider control for navigation

Middle

Five important shot types are featured, with each adopting its own colour and plasticised player models suspended in the browser

Bottom

Each shot has its own vertically scrolling timeline, highlighting top exponents from every decade and showcasing looping video highlight reels



Create a snap section slider control

A draggable slider control for horizontal navigation of an article's section content

1. Initiate HTML page

First, define the HTML document. This consists of the HTML container, which in turn stores sections for the head and body. While the head section is used to load the external JavaScript and CSS stylesheet, the body is used to store the on-page content created in step 2.

2. HTML content

The on-page content consists of an article container that stores several sections and an input slider. Each section is used to store the content that can be scrolled to using the slider input. For simplicity, this example uses a collection of h1 headings to present the number of each section.

```
<article>
  <section>
    <h1>1</h1>
  </section>
  <section>
    <h1>2</h1>
  </section>
  <section>
    <h1>3</h1>
  </section>
  <input type="range" value="0" />
</article>
```

3. CSS article

Create a new file called 'styles.css'. This step in styling is to define the settings required by the content container. Relative positioning is used to enable all children to be positioned in relation to where the article is. Whitespace set to 'nowrap' enables child articles to appear horizontally next to each other.

```
article{
  position: relative;
  white-space: nowrap;
  width: 100vw;
  overflow: hidden;
}
```

4. Article elements

Sections are set to the same width as their parent article, with a minimum height set to 25% of the screen. A transition is applied to the 'left' attribute so that any positioning change appears animated. Sections are required to use inline block display for horizontal positioning. Meanwhile, the input absolute

positioning results in it being placed over the sections at the bottom left corner of the article.

```
article > section{
  position: relative;
  display: inline-block;
  width: 100%;
  min-height: 25vh;
  transition: left .25s linear;
}
article > input{
  position: absolute;
  bottom: 0;
  left: 0;
}
```

5. JavaScript: scroll function

Create another file called 'code.js'. This step defines the 'scrollTo' function, to be used for moving the horizontal 'left' position of the sections. The function accepts a parent parameter for the article to search, and an xPos parameter to set each section's position. The 'for' loop identifies the sections as each child of the provided parent - except for the last child, which is the input element.

```
var scrollTo = function(parent, xPos){
  for(var n=0; n<parent.children.length-1; n++){
    parent.children[n].style.left =
      (0-xPos) + 'px';
  }
}
```

6. On load

Page content is only available after the page has fully loaded, so an event listener is used to execute JavaScript to manage page element interactions. This step searches for all of the range inputs placed as first-level children within an article container. A 'for' loop is used to reference each item found so that code in steps 7 and 8 can be applied to them.

```
window.addEventListener("load", function()
{
  var controls = document.querySelectorAll
    ('article > input[type="range"]');
  for(var i=0; i<controls.length; i++){
    *** STEP 7 HERE ***
  }
});
```

7. Slider settings

The input slider is set to have a maximum value reflecting the number of pixels that are covered by each container. This can be achieved by counting the number of sections within the parent of the control, as identified by the 'sectionCount' variable. This is then multiplied by the width of the slider's parent article.

```
var sectionCount = controls[i].parentNode.
querySelectorAll("section").length-1;
controls[i].max = sectionCount *
  controls[i].parentNode.offsetWidth;
```

8. Slider movement

This step applies the first of two events listeners to the slider. The 'input' event is triggered in real time, responding to any movement of the slider. This step sets the listener to trigger a call to the 'scrollTo' function, providing the slider's parent article and the slider's value as the parameters. As a result of the call, the article's sections are moved in relation to the value of the slider.

```
controls[i].addEventListener("input",
function(){
  scrollTo(this.parentNode, this.value);
});
```

9. Slider release

A second event listener is applied to the slider for when the slider is released, so that the position can be snapped to the closest section. Firstly, the section number 'n' is identified by rounding the amount of article widths that fit into the slider value. The slider is then updated with 'n' multiplied by the parent article width - i.e. the horizontal scroll co-ordinate. Finally, scrollTo is called with the parent article and new slider value.

```
controls[i].addEventListener("change",
function(){
  var sectionCount = this.parentNode.
  querySelectorAll("section").length;
  var n = Math.round(parseInt(this.value)
    /this.parentNode.offsetWidth);
  this.value = n * this.parentNode.
  offsetWidth;
  scrollTo(this.parentNode, this.value);
});
```

HuffPost - Existing While Black



HUFFPOST

EXISTING
WHILE
BLACK

WHAT DOE
IT FEEL LIK
WITHIN

huffingtonpost.com/interactives/existing-while-black

Designer:
Gladeye gladeye.com

ES
E



EDITION

Jo Etta M. Harris was nursing her child in her car before a family outing. Gil Perkins was talking on the phone outside of his home. Kelly Shepard's boys were shopping for video games. And in each instance, someone saw them as suspicious or a threat and called the police.

This isn't new. It happens every day. The experiences of Harris, Perkins, Shepard – and so many others – are reminders that black people don't have the privilege to simply exist in peace.

Whether we've been pulled over and mistaken for a suspect, followed in stores or disrupted in our own homes and neighborhoods, many of us have had to go above and beyond to prove that we have the right to occupy common spaces. Many of us have had to prove that we are not criminal, knowing that failing to do so could be fatal. The families of Tamir Rice and John Crawford III know that firsthand.

These aren't exaggerated situations, but real-life injustices that black people in the United States and beyond have to deal with. Stories of white people using 911 as their personal concierge hotline against black people have dominated the mainstream news this year. These stories are by no means new. And despite the danger these situations put us in, the white people making these calls rarely face repercussions.

A HuffPost/YouGov poll found that 15 percent of black Americans surveyed said they "have had the police called on them because of their race" more than once, compared to five percent of white Americans. Fifty-four percent of black Americans expressed that they've felt that "others have been suspicious of them based on the color of their skin," while only 6 percent of white people said they have felt this way. These results aren't a surprise considering that black people made up 24 percent of all police killings in 2015, despite comprising only 13 percent of the population, according to figures from [The Washington Post](#).

HuffPost asked black readers to share their stories of being subjected to racial profiling and discrimination. They described moments when someone called the police on them for no apparent reason aside from their race. They recalled scenarios of cops stopping and searching them because their skin color made them look "suspicious." They also said how maddening it is to live with the constant anxiety of possibly having their presence — and innocence — questioned.

Below, you will find their stories. Due to how deeply racism is woven into society's DNA, this list is by no means comprehensive. Yet it is a small collection of real anecdotes that underscores the unjust policing of black bodies, according to readers. HuffPost will continue to update this list and highlight the constant burden we face. If you have a story of living while black, tell us. This issue deserves more attention than a few headlines in the

“Huffington Post’s electronic feature editorialises story submissions on how aspects of life are policed for those of black ethnicity”

Colours



#000000



#E1DFD9



#0DBE98



#89A3FC

Tools

HTML5, JavaScript,
Facebook for Websites

Fonts

...ABC
1234567890

MFRED typeface by Matt Willey, assisted by Henrik Kubel, is a condensed caps-only headline face in one weight used on the main heading.

abcABC
1234567890
abcABC
1234567890

Times New Roman Pro also features styling the paragraph with italic and regular text running down the right side.

EXISTING
WHITE
BLACK

WHAT DOES IT FEEL LIKE WHEN EVERY MOVE YOU MAKE IS POLICED?

EXPLORE THE STORIES OF EXISTING
WHITE BLACK

[READ THEIR STORIES →](#)

HUFFPOST

ABOUT US
ADVERTISE
ISSUE OUR ADDS

Contact Us
888-442-1402

Comments
Author
User Agreement

Privacy Policy
Community Guidelines
HuffPost Terms & Conditions

©2018 Huff Post Inc. All rights reserved. Huff Post News

In Etta M. Harris was nursing her child in her car before a family outing. Gil Perkins was talking on the phone outside his home. Kelly Shepard's boys were shopping for video games. And in each instance, someone saw them as suspicious or a threat and called the police.

That isn't new. It happens every day. The experiences of Harris, Perkins, Shepard... and so many others are reminders that black people don't have the privilege of being seen as just another person.

Whether we've been pulled over and mistaken for a suspect, followed in stores or disrupted in our own homes and neighborhoods, many of us have had to go above and beyond to prove that we have the right to occupy common spaces. Many of us have had to prove that we are not criminal, knowing that failing to do so could be fatal. The families of Tamir Rice and John Crawford III know that firsthand.

These aren't exaggerated situations, but real-life injustices that black people in the United States and beyond have to deal with. Stories of white people being afraid of black people and holding biases against black people have dominated the mainstream news this year. These stories are by no means new. And despite the danger these situations put us in, the white people making these calls rarely face repercussions.

A HuffPost YouGov poll found that 15 percent of black Americans surveyed said they "have had the police called on them because of their race" more than once, compared to five percent of white Americans. Fifty-four percent of black Americans expressed that they've felt that "others have been suspicious of them based on the color of their skin," while only 6 percent of white people said they have felt this way. These results aren't a surprise, given that black people make up 12 percent of the U.S. population, while police make up only 3 percent of the population, according to figures from *The Washington Post*.

HuffPost asked black readers to share their stories of being subjected to racial profiling and discrimination. They described moments when someone called the police on them for no apparent reason aside from their race. They recalled scenarios of cops stopping and searching them because their skin color made them look "suspicious." They also said how maddening it is to live with the constant anxiety of possibly having their profiled.

Below, you will find their stories. Due to how deeply racism is woven into society's DNA, this list is by no means comprehensive. Yet it is a small collection of real anecdotes that underscores the unjust policing of black bodies, according to readers. HuffPost will continue to update this list and highlight the constant burden we face. If you have a story of living while black, tell us. This issue deserves more attention than a few headlines in the news cycle.

Left

The opening page in its entirety spans multiple screens and illustrates the oversized, poster-like approach to the typography

Below

Thirteen articles are browsed by big headings and polaroid-style snaps, with a custom cursor prompting visitors to read more

Bottom

On click, full story text comes out of the right side, letting readers digest each article and follow email links for submitting or correcting stories

Create animated content introductions for article areas

Use animation to introduce components of content sections in response to user interactions

1. Document setup

The first step is to initiate the webpage document. This is the description of the webpage structure, consisting of the HTML container, which stores the head and body sections. While the head section is used to load the external CSS stylesheet, the body section is used to store the webpage content created in steps 2 and 3.

2. HTML: navigation

This example will use a simple navigation that enables users to switch between the different content. The navigation links are stored within a nav container, providing advantages for SEO to indicate that the links are for navigation purposes. Each link references an article container that is to be created in step 3.

```
<nav>
  <a href="#p1">Part 1</a>
  <a href="#p2">Part 2</a>
</nav>
*** STEP 3 HERE ***
```

3. Article containers

The available content areas are managed using article containers that can be called from the URI using their unique ID value. Each section has a main heading, followed by two elements - a text and picture item. Elements use the 'data-animate' attribute to identify which animation from the CSS should be applied.

```
<article id="p1">
  <h1>Part 1</h1>
  <span data-animate="text">Text item.</span>
  <span data-animate="picture"></span>
</article>
<article id="p2">
  <h1>Part 2</h1>
  <span data-animate="text">Another text
item.</span>
  <span data-animate="picture"></span>
</article>
```

4. Navigation layout

Create a new file called 'styles.css'. This step defines the navigation width and position to be fixed across the top of the screen. Fixed positioning enables the navigation to be visible regardless of the page scroll position. This avoids the requirement for JavaScript to reposition the navigation in response to page scrolling.

```
nav{
  position: fixed;
  top: 0;
  width: 100%;
  z-index: 9999;
}
```

5. Article containers

The articles are set to use relative positioning so that their children can be placed in relation to their location. Articles are set with a width and height matching the size of the screen, with a top margin to avoid conflict with the navigation. Overflow is also set to enable scrolling for any content that doesn't initially fit onto the display area.

```
article{
  position: relative;
  height: 100vh;
  width: 100vw;
  padding-top: 1em;
  overflow: auto;
}
```

6. Default definitions

The animated elements inside each article are all set to use absolute positioning, with a transition set applied to all attributes lasting for a duration of one second. This enables the selected elements to be easily animated from within CSS by setting or overwriting their attributes, without needing support from any JavaScript.

```
article > *:first-child,
article [data-animate]{
  position: absolute;
  display: block;
  transition: all 1s;
}
```

7. Article heading

The heading for each article is identified as the article's first child. This enables any item to be used within the HTML, rather than be restricted to a specific element type such as h1. This element is set with a specific font size and hidden overflow with no height - i.e. invisible by default.

```
article > *:first-child{
  height: 0;
  overflow: hidden;
```

```
  font-size: 4em;
}
article:target > *:first-child{
  height: 1em;
  transform: translate(0%, -100%);
}
```

8. Picture definition

The element using the 'data-animate' attribute set to 'picture' is set to be a specific size and placed out of view using vertical positioning and opacity. Its vertical position and opacity are altered when the article is targeted via the URI, triggering the picture to animate into view. The transition-delay attribute is used to create a one-second delay to the animation effect.

```
article [data-animate="picture"]{
  right: 10vw;
  top: 100vw;
  width: 20vw;
  height: 20vw;
  background: blue;
  opacity: 0;
  transition-delay: 1s
}
article:target [data-animate="picture"]{
  top: 0;
  opacity: 1;
}
```

9. Text definition

The data-animate attribute set to 'text' is used to reference the text containers. Placed out of view by default using a negative left co-ordinate, these are animated into view by using a positive left co-ordinate when their parent article is targeted via the URI. The transition-delay attribute is used to provide a unique activation time within the overall article opening.

```
article [data-animate="text"]
{
  left: -100vw;
  top: 40vh;
  transition-delay: .5s;
}
article:target [data-animate="text"]{
  left: 20vw;
  color: green;
}
```

Subscribe from just £24.99*

PLUS



FREE KITSOUND RACE WIRELESS EARPHONES

GIFT
WORTH
£14.99

Order hotline 0344 848 2852

Please quote **jan19** when ordering by phone

Lines open Monday to Friday 8am-7pm and Saturday 10am-2pm (GMT)



myfavouritemagazines
Official Magazine Subscription Store



Great reasons to subscribe

- Pay just £24.99 every 6 months
- Spread the cost with Direct Debit
- Never miss an issue
- Delivered direct to your door



Enjoy your favourite magazine anywhere, anytime!

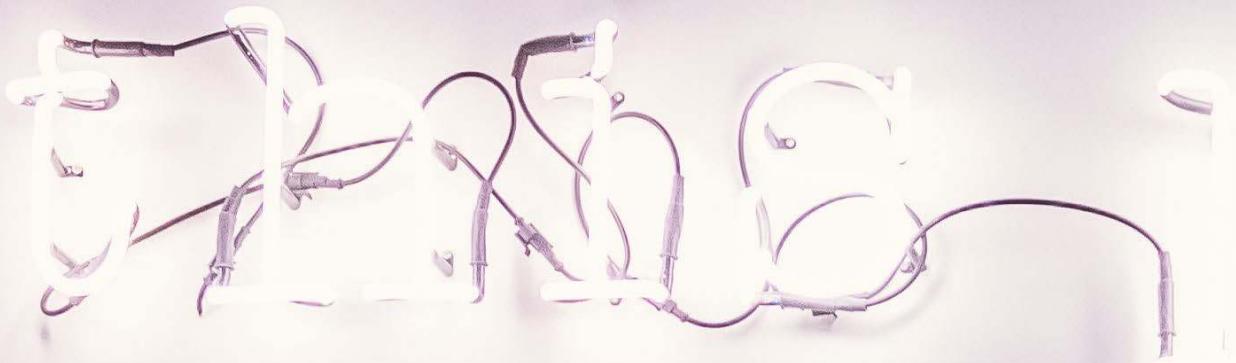
- Upgrade to include digital for just an extra £5

SEE THE FULL RANGE AND ORDER ONLINE AT

www.myfavouritemagazines.co.uk/PIN/JAN19

Call UK **0344 848 2852** or overseas +44(0) 344 848 2852 & quote jan19

***Terms and conditions:** Savings calculated against the full RRP (single issue price x frequency). You can write to us or call us to cancel your subscription within 14 days of purchase. Gift is only available for new UK subscribers. Gift is subject to availability. Please allow up to 60 days for the delivery of your gift. In the event of stocks being exhausted we reserve the right to replace with items of similar value. Payment is non-refundable after the 14 day cancellation period unless exceptional circumstances apply. Your statutory rights are not affected. Prices correct at point of print and subject to change. Full details of the Direct Debit guarantee are available on request. For full term and conditions please visit: bit.ly/magtandc. Offer ends 28th February 2019.



A PLACE FOR DESIGN

Agility, strategy and design collide at This Place. As a global agency, their partnerships with leading brands shape their digital channels and communications, resulting in rich user experiences. Talented, driven and insightful designers, This Place defines today's design trajectory



WHO

THIS PLACE

WHAT

User Interface, User Experience,
UI UX, Mobile, Digital Innovations,
Design, Management Consulting,
Digital Strategy, Digital
Transformation, E-commerce, App
Development, Responsive Web
Development, Vendor Consulting
and Programme Management

WHERE

London, Seattle, Tokyo,
Hong Kong

WEB

thisplace.com

KEY CLIENTS

- [1] Ahold Delhaize
- [2] T-Mobile
- [3] Woolworths
- [4] UBS
- [5] Big W

A place for design

AHOLD DELHAIZE GLOBAL RETAIL PLATFORM

aholddelhaize.com



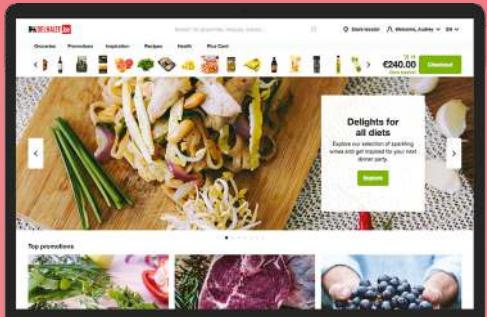
AHOLD DELHAIZE is one of the world's largest retail store groups, with over 3,500 grocery store operations across Europe, Asia and North America. This Place was commissioned to lead user experience and design for creating a best-in-class responsive web platform; one that at its core is built to 'white label' principles, providing a consistent shopping experience while catering for each supermarket brand's individual functional and stylistic needs.

The design process started with a discovery phase that included competitor analysis as well as a creative exploration of what an outstanding experience could look like. This Place created a variety of designs that were tested with stakeholders as well as customers, before converging on the approach that would inform the rest of the design process.

Due to the enormity of the task, and with multiple stakeholders and suppliers located in different parts of Europe, they went down the route of using Axure to create thorough wireframes for each component and every breakpoint. This enabled them to easily share designs and have teams comment on the high-fidelity designs. The final set of documents delivered contains wireframes for over 90 components used across over 1,300 key pages.

The primary requirement was to design an experience that would cater to a wide and varied global customer demographic while also being flexible enough to accommodate the specific branding elements, marketing materials and content from each supermarket chain. This required continuous stress testing as well as developing the lowest common denominator of building blocks that would work across any brand property.

The end result was a responsive e-commerce platform catering to web, tablet and mobile devices for Ahond Delhaize's 19 supermarket brands across seven countries, built to best-in-class usability standards.



A bold, clean approach ensures each product is easily accessible via a design approach that places the user experience at its centre



Adrian Wong (Senior Designer)
coding his latest project



Hanny Kamal (Designer)
taking part in one of This
Place's meet and greet events



THIS PLACE WAS FOUNDED IN 2012 WHEN DUSAN HAMLIN (CEO) AND BEN ALDRED (GLOBAL CREATIVE DIRECTOR) SAW A NEED TO CREATE A FORTUNE 500-FOCUSED DIGITAL PRODUCT STRATEGY, DESIGN AND INNOVATION STUDIO. Big brands often have the most significant issues when innovating digital product experiences quickly and cost-effectively. Dusan and Ben wanted to change that, and it continues to be their mission.

"We like to refer to ourselves as a product design and innovation studio," Dusan explained to **Web Designer**. "We are a collective of proven strategists, digital architects, digital designers and front-end developers based out of London, Seattle, Hong Kong and Tokyo.

"While we love to build end-to-end products for our clients, we often do our best work when in a multi-stakeholder partnership with our clients' in-house team or with a small number of partner agencies, where together we create a market-beating delivery team. These are often complex, multimillion-dollar re-platforming projects or entirely new digital propositions, delivered faster and better when the right group of stakeholders are pulled together to deliver."

Dusan is a proven digital business leader who has founded and led two hugely successful international companies from a standing start, first founding Inside Mobile, which became M&C Saatchi Mobile / M&C Saatchi Performance. In addition to building innovative mobile and digital strategies for key brands such as Adidas, Vodafone and Philips CE, Dusan has also held leadership positions within the global top five digital networks, such as Isobar and WPP. He studied for a Bachelor of Business

Administration degree in Economics at Anglia Ruskin University but dropped out early to begin his career in digital in the late 1990s.

Ben, meanwhile, has a vast amount of experience in running design studios. As the founder of Mylo Design and This Place, Ben built up a client roster of international brands, including luxury hotel group the Mandarin Oriental and infrastructure conglomerate the YTL Corporation. Ben's thorough understanding of consumer behaviour and UX design in the digital sphere, combined with many years of running digital agencies, means he

has been able to build and scale This Place's service proposition effectively and develop a team from inception to the growing consultancy it is today. Ben holds a Bachelor of Design degree in Product Design from the University of Leeds.

One of the first tasks Dusan and Ben undertook was the pressing question of what to call their new agency. "This Place's name came about when the Talking Heads song *This Must Be The Place* was played on the radio while working at Dusan's house, as we had yet to find a studio for our company," Ben explained. "We liked the fact that This Place had some intimation of 'home', our 'place to create', but most importantly, it was a short URL that would translate reasonably well for overseas markets and 'Google' well! Sourcing the domain name was not easy, but we felt strongly that This Place was the right name for us, and eventually the URL owner took pity on us and sold us the domain at a reasonable cost."

Dusan and Ben set out to create a different working environment for their business, placing collaboration at its heart. The work they have produced has resulted in a steady stream of clients, as Dusan explained: "Our experience is that if you hire best in class people, nurture a collaborative team, get stuff done and nurture a no-blame culture, you can build incredibly dedicated teams that solve problems fast. With that in mind, we have been lucky enough to be able to grow via client referrals and by actively seeking engagements from industries that most need our team of doers!"

As the client base for This Place has predominantly been large corporations, this has influenced how each project is approached. "Historically, we've always worked with big companies - predominantly Fortune 500 - and the vast majority of our clients have been with us for quite a long

time," said Dusan. "We are obsessed with making them number one in their industry and keeping them there."

Dusan continued, "Technology is advancing at an ever-faster pace, so the research, data and design techniques at our disposal keep us constantly busy ensuring customer experiences are the best they can possibly be. We like to be challenged by the scope and scale of big challenger brand projects, and that's what we've been doing for the most part. That said, while we like tackling big complex projects, we are always looking for interesting companies



"**ABOVE ALL, DESIGN SHOULD SOLVE A BUSINESS OR CUSTOMER PROBLEM. AND OFTEN, THE SOLUTION DOES TEND TO SPAN ACROSS MANY CHANNELS. HOWEVER, THERE IS NOT A 'ONE SIZE FITS ALL' SOLUTION FOR DESIGNING ACROSS MULTIPLE USER INTERFACES**

DUSAN HAMLIN

CEO

open to digital product change. We avoid working with companies that do not fit our ethics or culture. For example, adult content, tobacco brands, etc. Money is not our god - our projects are usually hard work and complex, so we need to be motivated for the outcome."

With such a diverse portfolio of clients, is there a design ethos within This Place? Dawn McKerracher (Strategy Director) said, "It's challenging to select one project in particular, but a good example of how we work and the standards we strive to is our award-winning work for Ahold Delhaize, one of the largest supermarket groups in Europe and the US. We've worked on many projects, and ultimately designed a digital device-responsive frontend for the Delhaize supermarket e-commerce site that's now being used in multiple countries for their multimillion-dollar business. It has seen a huge increase in customer usage and revenue directly linked to the work we and the other partners on the projects have done."

As no two projects are the same, the approach and timeframe to complete them also vary wildly. Breaking down each large project into stages with clearly defined deliverables and deadlines is how This Place is able to create world-class experiences for their clients. Each project has a custom-built team that is selected to meet the bespoke needs of each client.

"Creating connected experiences across all digital channels is absolutely a priority for our clients and for us, as this is increasingly an expectation from customers and users," Dusan explained. "Above all, design should solve a business or customer problem. And often, the solution does tend to span across many channels. However, there is not a 'one size fits all' solution for designing across multiple user interfaces. Not every business needs to be across all interfaces if this doesn't make business sense for the customer or the company. What we aim to do with our strategy work is to look at the big picture and determine how we could help the end customer and best serve our clients, by creating a digital experience that makes sense for the optimal user experience and commercial realities."



"THE MAIN DESIGN TREND IS THE USE OF TEXTURED ILLUSTRATIONS. SITES SUCH AS WETRANSFER, DROPBOX AND INTERCOM ARE GOOD EXAMPLES OF THIS. THE USE OF TEXTURE IS A GREAT WAY TO GIVE FLAT DESIGNS A MORE TACTILE AND LESS OVERLY-POLISHED FEEL. THIS STYLE HAS A GREAT PHYSICACY TO IT BUT CAN AT TIMES BE TOO UBIQUITOUS"

DAWN MCKERRACHER
STRATEGY DIRECTOR

The significant variation in the work This Place has created has required using a toolset that both utilises established applications and also embraces the constant evolution of the digital tools they have available.

"We use a huge variety of management tools, including a bespoke tool (GLUE) that we have built in-house specifically for managing multiple product design initiatives for some of our clients," said Dawn. "We have also learned to be flexible and use our clients' or other agency project tools if that is more effective for the project's end goal."

Dusan also explained, "HTML, JavaScript and CSS will most probably continue to provide the combination of structure, style and interactivity on the internet that they deliver now. On the other hand, jQuery - while it has enough of a footprint to still be around for some time - will likely slowly be replaced by new JavaScript frameworks such as React, Angular and Vue. They are paving the way to a more stable and defined future of developing applications for the web. HTML, CSS and JavaScript will always be at the core of a website. The most significant shift will be in choosing the new frameworks that provide the technology to build scalable apps and make them more interactive.

"Our team has been trying out the following tools recently: Figma, a collaborative interface design tool; Framer X, a tool entirely focused on interaction design; and Principle, a tool for designing animated and interactive user interfaces.

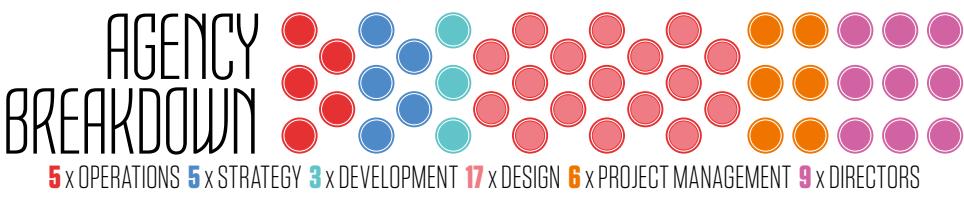
"We started using Figma to figure out how it's different from Sketch, and we quickly found that for our organisation and clients it's a better solution as it allows for greater collaboration. When it comes to Framer X, we've seen that it's more geared towards the needs of design as opposed to development and coding. We use Principle to mock up animations and various types of

interactions. While Principle is not as sophisticated as Framer X, it allows for quick prototyping. It's important to adapt, stay flexible and use what's best for our team and our clients.

"The design tool market is evolving and changing very rapidly, so for us it's important to test out new tools and stay on top of



James Stump (Designer) and Dawn McKerracher (Strategy Director) consider where This Place will expand to next



A place for design

T-MOBILE GLUE (GLOBAL LANGUAGE FOR UNIFIED EXPERIENCES)



T-MOBILE WORKS with a lot of external agencies and suppliers to deliver digital products and services across multiple platforms. Keeping the brand's vision, goals and identity consistent is a challenge, especially when brand governance is conducted via distributing static, non-editable PDFs and printed materials.

This Place solved this by creating a living brand ecosystem and code repository, keeping multiple disparate partner agencies all aligned and continually updated so T-Mobile's evolving brand position is always maximised.

The agency initially conducted interviews and workshops with stakeholders aimed to highlight the pain points of their legacy approach. The result was an outline of the features and requirements for the GLUE minimum viable product.

The next step was to develop a robust information architecture - a sitemap and the primary flows depicting how the users will perform everyday tasks within the platform. Having the processes and interactions thoroughly documented helped the development team to work on and create a first draft of the delivery schedule.

Rather than going for a complete set of wireframes first, This Place proceeded just to sketch out the main layouts and templates. They conducted user testing of the sketches with T-Mobile staff and suppliers rapidly iterating at this low-fidelity level. Once they had solid proof from the users that the design approach was working, they moved to high-fidelity wireframes and creative designs.

In less than six months from project kick-off, they had conceived, designed and delivered the first iteration of GLUE within the T-Mobile IT infrastructure as a stand-alone tool. This single living document platform for all brand and design guidelines is always up to date, tracking the evolution of the brand and all its applications. Designers as well as developers started to populate the product with This Place's help and are now using it productively.



Integrating great design to deliver the information architecture T-Mobile needed meant understanding how the brand is communicated

WOOLWORTHS STRATEGIC ROADMAP



WOOLWORTHS, AUSTRALIA'S largest supermarket group, is facing the same issues as grocers across the globe; an increasingly competitive marketplace and the absolute necessity to innovate the customer purchasing experience in order to continue in a leadership position.

The central digital team, Woolies X, had their work cut out with competitor threats growing within this shifting market. Coupled with constantly evolving consumer behaviours and demands, it was tough for the team to move beyond the incremental optimisation of existing journeys.

WooliesX asked This Place to undertake an end-to-end evaluation of their digital customer experience, measure it against local-market competitors, established global e-tailers and newer disruptive brands, and then devise a strategy for overhauling the entire experience.

This Place mapped existing WooliesX workstreams into a customer journey framework and developed an overarching vision and supporting strategic pillars. They addressed the gaps and customer pain points in the existing experience by identifying relevant, achievable and prioritised opportunities. This process was informed and validated by data analysis, market and audience insight as well as observations from the experience review.



Mapping how customers use the stores, coupled with data and market analysis, enabled This Place to support Woolworths as it innovated

Alastair Holmes
(Associate Creative Director) is a happy chappy after seeing the latest feedback from a client



2012

Dusan Hamlin and Ben Aldred found This Place with ambitions to change the world!
Employees: 2

2014

This Place begins working with Ahold Delhaize, one of the world's largest retail store groups.
Employees: 8

2015

This Place's MindRDR project continues to fascinate the technology industry.
Employees: 14

2016

They open an office in Seattle and start their relationship with T-Mobile, launch GLUE
Employees: 31

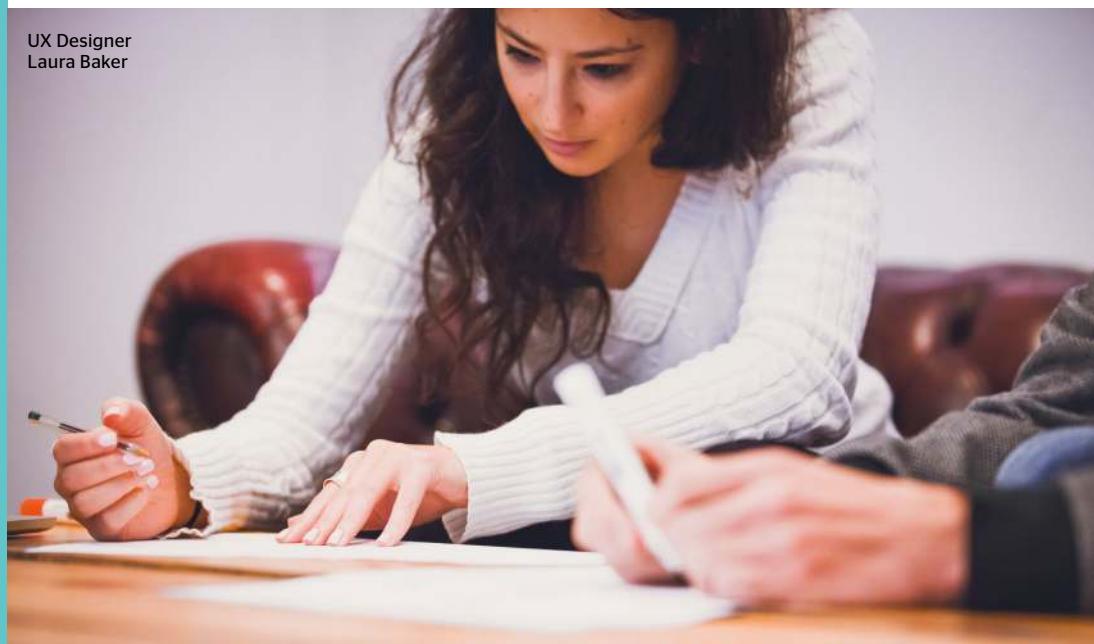
2017

They join forces with Asteria and ascend to the top tier of Tokyo Stock Exchange.
Employees: 35

2018

This Place develops and designs an 'endless aisle' for Ahold Delhaize's concept store.
Employees: 45

UX Designer
Laura Baker



new developments in the industry. We're constantly trying out new features to make sure we're working with the software that's best for our clients and our projects."

In addition, the client base that This Place attracts has required them to increasingly move into other areas of design, as Dawn outlined to **Web Designer**. "We are finding that in recent months and years the relationship between product design for our clients and the advertising, general and social media worlds are becoming closer and closer. More of the e-commerce process is shifting from brand-owned sites and into social sites. This shift in where the purchase or customer interaction takes place means that we are now building closer design relationships for our clients with the advertising agencies or the social sites directly."

We asked Dawn to identify any current design trends. "The main design trend is the use of textured illustrations. Sites such as WeTransfer, Dropbox and Intercom are good examples of this. The use of texture is a great way to give flat designs a more tactile and less overly polished feel. This style has a great physicality to it but can at times be too ubiquitous."

Dawn continued, "Another significant trend is the breaking of traditional style rules: A lot of the concepts that were considered a design faux-pas in the past are becoming quite relevant now, as we consumer test them and see behavioural advantages. Horizontal scrolling is a good example. Conventionally a big 'no-no', it is now being used by Apple for their new iPad, and as we know, Apple's designs always cause an echo around the design community.

"And we're also seeing a lot of nostalgia. A lot of designs at the moment have a 90s feel to them - from bright blue links that look like they were taken from an early Wikipedia page to fixed elements that overlap with each other, there's a certain homage to the era that birthed the internet. It's not necessarily about pure performance and more focused on overall design aesthetic. There are also a lot of 'soft' designs - think websites that look like they melt on your screen, 16-pixel curves on corners, no drop shadow and so on. These designs have an utterly soft feel to them. And lastly, we're seeing a lot of college-esque

illustrations, inspired by Matisse's art: bright colours and collage-type shapes, put together to create abstract imagery."

Dusan also commented on what the future of digital design could look like: "I think in the future, there will be more designing for non-owned platforms. For example, designing an experience that exists as part of Facebook or Instagram. Instagram, in particular, has been rolling out many features, enabling users to shop from Instagram stories, story videos and regular feed posts. So there is

now more pressure on retailers to explore these capabilities in order to compete - more agile businesses that, quite often, drive most of their revenue from social and not a destination website or app. It's also where the customers are, so it makes perfect sense to explore these platforms more, even for larger companies."

The determination to deliver a diverse range of services to their clients means having an equally eclectic mix of designers and developers. Dusan explained what it takes to work at This Place. "At This Place, we look for those who are unique, consumed with improving, curious, humble and fun.

"We have in common a strong empathy for the human condition, critical thinking, entrepreneurial spirits and a love for solving the most challenging problems through design. We work best with people who can listen as well as lead, to see value in others' opinions and who enjoy constant collaboration.

"Our main advice for those who are looking to get into the industry is to keep up to date with the latest news, launches and tools. Also, don't be afraid to get your point of view heard, remember that design can often be subjective and not proven, so your

experience combined with user research will determine if you are on the right track."

With their fingers on the pulse of digital design today, This Place is able to define the design environment for their clients, and also craft engaging experiences that deliver new channels of content for some of the world's largest brands. With their ability to understand what drives consumers today, the team at This Place have become creators who define what digital design means today and tomorrow.

This Place
Web
thisplace.com

Founders
Dusan Hamlin, Ben Aldred

Year Founded
2012

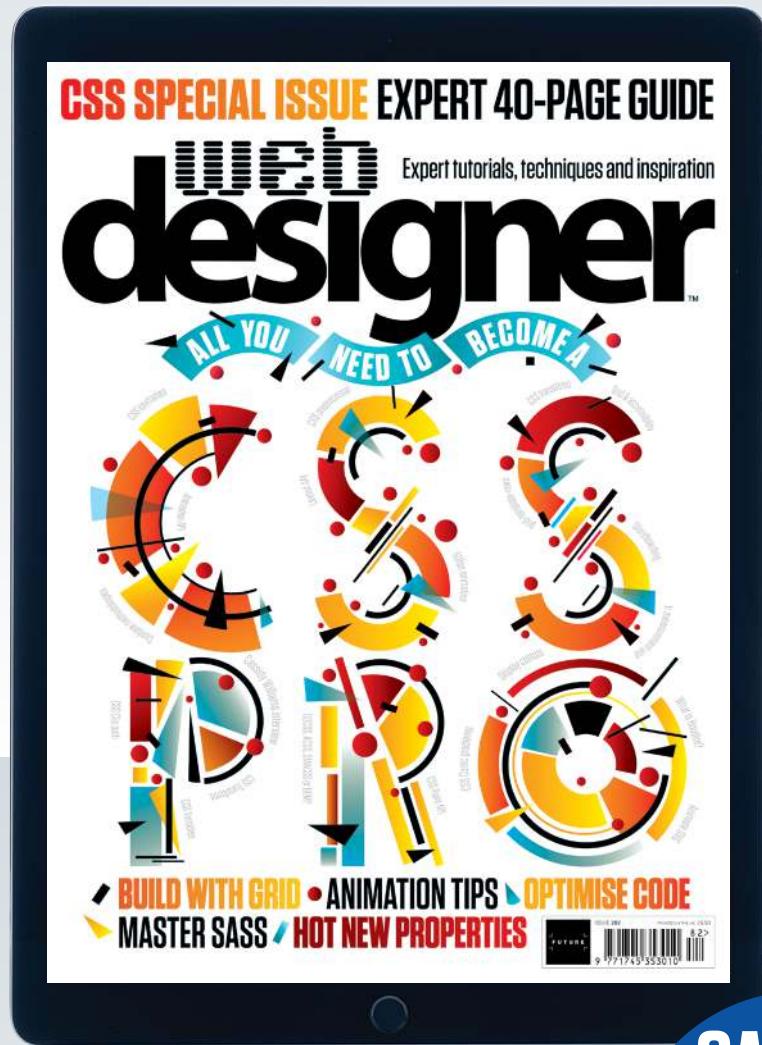
Current Employees
45

Location
London, Seattle, Tokyo, Hong Kong

Services

- [1] Design - We help maximise revenue and reduce cost through better design.
- [2] Strategy - We craft strategies to address today's business challenges and define tomorrow's vision.
- [3] Development - We deliver production-ready software across platforms and devices.
- [4] Experiments - We build intellectual property by exploring how new ideas and technologies create value.
- [5] Studios - We offer a fully managed digital product studio for clients.

NEVER MISS AN ISSUE!



Issue 282

What you need to know to become a CSS pro: animation, Grid, Sass, optimisation, methodologies, hot new properties and APIs

SAVE UP
TO 43%
WITH A DIGITAL
SUBSCRIPTION
[SEE PAGE 28](#)



Issue 281

Design, develop and create with top Google tools, add glitch effects with CSS, animate with GSAP and code UX patterns



Issue 276

Build your own augmented reality app, get started with Three.js, power up with Sass and Stylus and generate web components

GOT AN APPLE DEVICE?

Download **Web Designer**'s digital edition for your iPad, iPhone or iPod touch

<https://apple.co/1hsGYgl>

CATCH UP ON ANY ISSUES YOU'VE MISSED BY DOWNLOADING OUR DIGITAL EDITIONS <https://bit.ly/2rh73Xk>



Issue 280

30 best HTML, CSS and JS APIs and elements, research your keywords, interactive JS charts and What is the Web of Things?



Issue 279

Pro tips to build animation for the web, 23 ways to make your app shine, work with variable fonts and what's new in Node 10



Issue 278

What's new in JavaScript, say hello to Google Flutter, build with CSS Grid, work with WebRTC and create patterns with CSS



Issue 277

Hot new CSS properties for today, build interactive 3D, streamline your design workflow with Sketch and code a real-time React app



Issue 275

50 must-try tools for designers and developers, how to add awesome audio, Progressive Web Apps and interactive images



Issue 274

The new rules and top tools for UX, 5 hot new CSS properties, use the CSS Paint API, add particle effects and discover JSX



Issue 273

SEO Today, 20 hot new libraries and frameworks, ten pages of ReactJS tutorials, get started with Rust and build a WebGL racing game



Issue 272

Build with Web Components, 8 WordPress security secrets, Web Accessibility, convert visitors to customers and data visualisations

PREFER TO READ ON ANDROID, PC OR MAC?

Web Designer magazine is also available on Google Play and Zinio

<https://tinyurl.com/yalm3wul>



<https://bit.ly/2xPbv4p>

JAVASC



THERE ARE GREAT NEW FEATURES THAT
ARE HEADING TO THE BROWSER. WE LIST
25 FRAMEWORKS, TOOLS AND APIs
YOU SHOULD TRY OUT IN THE YEAR AHEAD

CRYPT-IN



“

Web apps are now first-class citizens. It's great to see browsers gain the tools they need for developers to make quality, native-like experiences accessible to all

Matt Crouch, Software Engineer at Vidsy
@mattcrouchuk • mattcrouch.net

”



Featured elements

[Browse elements](#)**time-field-polymer2**

A time field Web Component based on Polymer 2.x.



gatanaso

polymer-midi-player

MIDI Player Web Component



jazz-soft

date

A Web Component collection that helps you format your dates.



aide-de-camp

Featured collections

[Browse collections](#)**smarthtmlelements-core**

7 items

High Quality Custom Elements Web UI Framework

**vaadin-core**

46 items

An evolving set of free, open source web components for building mobile and

**granite-elements**

27 items

A collection of web-components made on the granite coast of Brittany

WEB COMPONENTS

IThe concept of a component is used heavily in JavaScript frameworks. They are used to break apart a project into several reusable chunks that can be moved about independently of one another. But what happens when it's time to change framework? Previously written components will no longer work. Either there is added workload to convert the components, or the project stays tied to that framework.

Web components are a solution to that issue. They are written using native web technologies, are self-contained and expose values and behaviours through attributes like any other HTML element.

They are made up of three separate specifications - HTML templates, shadow DOM and custom elements.

HTML templates define the visual appearance of a component. They contain the structure of content, which can then be copied and reused for each component on the page.

Shadow DOM is a concept of keeping the internal markup of the component away from the rest of the document. This keeps styles and other logic from breaking out and affecting other components.

Finally, custom elements are the glue that holds everything together. They are responsible for the lifecycle of the component, which can include extra logic for when a component is added or removed from the page.

Web components can be used alongside their framework counterparts. As they are supposed to act like native elements, most frameworks will work with them without issue. Some frameworks, like Angular, may need to be told of their existence ahead of time to function properly.

As they can be readily composed together, components can be imported from many sources. Websites like [webcomponents.org](#) provide several ready-made elements that can be dropped in and used straight away.

While the concept of a web component has been around since 2011, the specifications have constantly changed. In 2018 things settled down, with Firefox implementing the latest versions of shadow DOM and custom elements specifications in October. With Edge due to ship their implementations in 2019, all major browsers will natively support them.

• [webcomponents.org](#)

WEBASSEMBLY

2 One complaint about using the browser to host applications is that JavaScript runs too slowly when it comes to processor-intensive tasks, such as game engines or video editing.

WebAssembly - or 'WASM' - is a compile target for code written in other languages more suited to heavy lifting. This enables existing programs written in languages like C++, Rust or Go to make it to the web. By compiling these down into a binary format, it enables them to be split into chunks and downloaded efficiently.

These binary files are then compiled in memory and instantiated using the new WebAssembly object. There is a push to make these importable like any other JavaScript modules, but work in this area is still ongoing.

While WASM is a new language to work with in the browser, it does not make it a competitor to JavaScript. Both languages have their benefits, with JavaScript still lending itself better to day-to-day operations in the browser. They can work together and have autonomy over separate parts of an application.

WASM has cross-browser support, now including mobile. This opens the door to many opportunities that were previously limited to native applications. Over the next year we may see more games and application developers embrace this technology.

• [webassembly.org](#)

PAYMENT REQUEST API

3 When creating an e-commerce website, lots of thought goes into improving the checkout flow to make sure the user's experience is as smooth and pain-free as possible. Entering payment details remains a repetitive process. If any field is slightly wrong, the whole process breaks and it needs to be started again. On the developer's side, making sure the right validation checks are in place for each payment method is a time-consuming process.

The Payment Request API does away with this part of the checkout experience. The page can request the information it needs and the browser provides the user with the fields to enter.

The user only needs to enter their details once, and they can be used across different websites and transactions. Depending on the device, they can even use the payment methods linked to the user account, such as Google and Apple Pay.

TIP WORKING WITH PAYMENT PROVIDERS

Lots of websites use third parties to take payment details. Check to see if your provider supports this API to help your users. These include Stripe, WePay and Worldpay

The first details the payment methods accepted. These can be different types of payment cards, as well as details for integrating with other third-party payment solutions. There is even scope to include another web application as a payment method by using the Payment Handler API.

The second is a breakdown of what is being purchased. This can be in any format that makes sense to the transaction. This also includes options for shipping costs, which can be calculated based on other factors, such as delivery address.

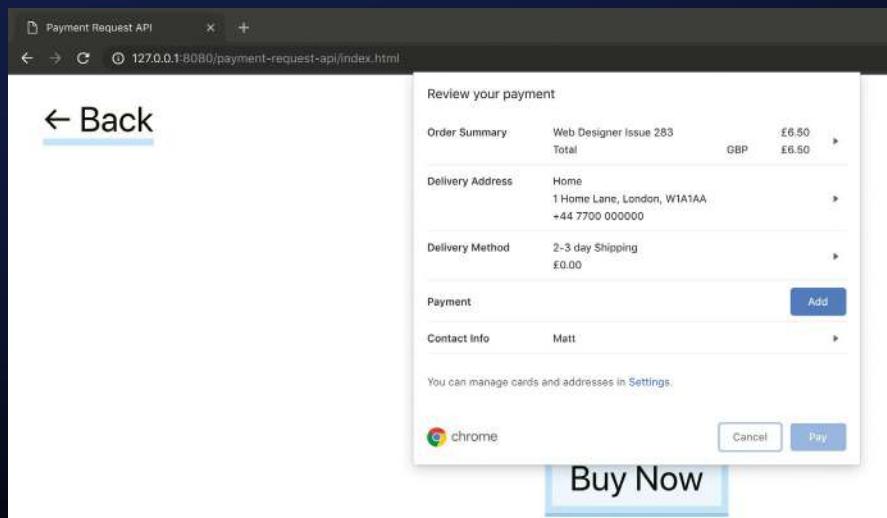
There is also a third, optional argument to specify any extra information that needs to be collected, such as an email address or contact number.

The Payment Request API is available in the latest versions of most browsers. As some users may opt not to use it, however, it's important to have a fallback payment method in place.

• w3.org/TR/payment-request

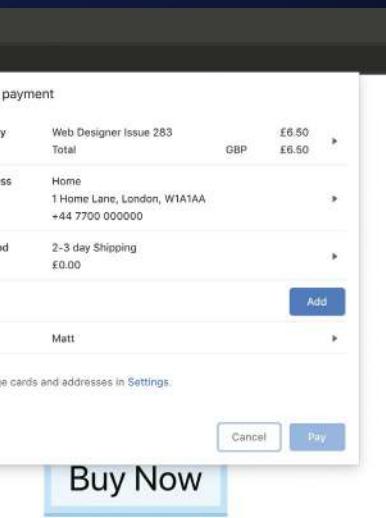
```
const request = new PaymentRequest([
  { supportedMethods: "basic-card" },
  {
    total: {
      label: "Total",
      amount: { currency: "GBP", value: 6.5 }
    }
  }
]);
```

The object takes a few parameters that describe the payment that should take place.



The user only needs to enter their details once, and they can be used across different websites and transactions

“ ”



The first details the payment methods accepted. These can be different types of payment cards, as well as details for integrating with other third-party payment solutions. There is even scope to include another web application as a payment method by using the Payment Handler API.

The second is a breakdown of what is being purchased. This can be in any format that makes sense to the transaction. This also includes options for shipping costs, which can be calculated based on other factors, such as delivery address.

There is also a third, optional argument to specify any extra information that needs to be collected, such as an email address or contact number.

The Payment Request API is available in the latest versions of most browsers. As some users may opt not to use it, however, it's important to have a fallback payment method in place.

• w3.org/TR/payment-request

VUE

4 Vue has been steadily on the increase in the past year. Recent updates have had more of a focus on developer experience and quality improvements. Vue 2.5 was released back in 2017 and included updates to error handling, server rendering and TypeScript definitions. Single functional components – one of Vue's more popular features – were updated to support scoped CSS and also improved template compilation.

Since then, updates to the core Vue experience have been mostly focused on bug fixes and performance improvements. The focus has now shifted to complementary development products.

Version 3 of Vue's CLI tool moves to a more configurable approach. Features such as routing, linting and data management are selected during initialisation. These features can be added and removed as needed without picking apart the build process.

Along with the CLI, there is a web interface available to visualise the different parts of an application. This acts as a dashboard to help analyse load times, sizes and pain points within the build as it runs.

Looking forward, 2019 will see the release of Vue 3.0. While breaking changes will be kept to a minimum, this version will focus on making use of ES2015 class-based components. It will also support fragments and portals, much like the features in React.

• vuejs.org



The screenshot shows the React.js homepage. At the top, there's a navigation bar with links for 'React', 'Docs', 'Tutorial', 'Community', 'Blog', a search bar, and GitHub links. Below the header, the word 'React' is prominently displayed in blue. A sub-header below it reads 'A JavaScript library for building user interfaces'. There are two buttons: 'Get Started' and 'Take the Tutorial >'. The main content area is divided into three columns: 'Declarative', 'Component-Based', and 'Learn Once, Write Everywhere'. Each column contains a brief description of its benefits. Below this, a section titled 'A Simple Component' shows a code snippet for a React component. The code defines a component with a render method that returns a single paragraph. It also includes a note about JSX being optional and a link to the Babel REPL.

REACT

5

React 16.6 brought the 'suspense' feature, which made dynamic content easier to work with.

Combined with `React.lazy()`, it creates a point at which to split code into smaller chunks. Suspense can render a fallback component, such as a loading spinner, while the component loads.

In 2019, suspense will grow into a more flexible tool. By triggering data fetching as part of the flow, users will get a seamless, native-like experience.

One of the easiest ways to get up and running with React is by using the 'Create React App' tool to stamp out a project. A recent update upgraded its dependencies and improved its features.

Create React App 2.0 uses Babel 7 and Webpack 4 to build projects, which makes it faster to build and opens the door to newer features like the shorthand fragment syntax.

Projects made with the previous version can be upgraded by updating the 'react-scripts' dependency. Updating ejected applications is a more manual process, but they can be upgraded piece by piece.

React shows no sign of slowing down in 2019. Hooks enable functional components to state and lifecycle behaviours. Concurrent mode improves the performance of slow-rendering components. Both features are slated for a release in the first half of the year.

• reactjs.org

CSS LAYOUT API

6

As a general rule, the layout of a page should be left down to CSS.

With JavaScript, handling changes to layout is prone to poor performance and lots of manual upkeep.

But what happens if a site requires a non-standard layout? The traditional grid-like placement that CSS provides may not work for more creative design ideas.

The Layout API enables developers to create their own layout modules, much like the existing grid or flex modules in use today. By defining the rules it should work to, the API can place items anywhere within its containing element. It is the second Houdini API to make it to browsers.

Houdini introduced the concept of 'worklets', which work similar to web workers but are highly specialised in one activity. The Layout API works through a specific layout worklet. These get registered with the browser, which enables them to be picked up in CSS.

```
class MasonryWorklet {
  *layout(children, edges, constraints,
  styleMap) {}
}
registerLayout("masonry", MasonryWorklet);
```

A layout worklet has one method called 'layout', which deals with the positioning of elements. It runs as a generator function, enabling it to run in parallel to the rest of the page to keep everything running smoothly.



The first parameter is a reference to the immediate child elements of the one the layout is applied to.

Size data is available from the second and third parameters. One provides position and dimension information about the element being styled, with the other defining the constraints the element has to work within.

The final parameter describes the rest of the styles of the element, including any custom properties it may be interested in.

```
.el {
  display: layout(masonry);
}
```

A custom module is applied by using the usual 'display' property on an element. It uses the 'layout' function, along with the name of the worklet.

The Layout API is in Chrome Canary to try out, with Firefox working on its implementation. By the end of the year, the specification is likely to stabilise and more browser implementations can be possible.

• w3.org/TR/css-layout-api-1

CREDENTIAL MANAGEMENT API

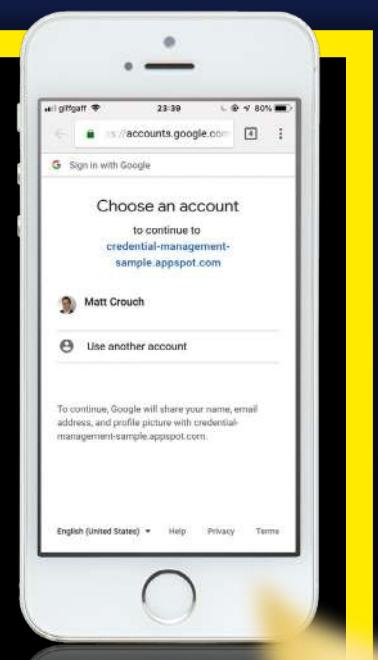
7

Instead of having to remember lots of individual login details, users can log in with the credentials stored within the browser. That way, users on multiple devices can have their details follow them around without needing to log in each time.

While a browser's autocomplete feature may reduce the friction with entering the password itself, the Credential Management API can automatically log a user in as soon as they hit the page. It can also log in using other forms of authentication, such as a public key or through an authentication provider like OpenID.

This API is currently only supported by Chrome and Samsung Internet browsers, with Edge currently in development.

• w3.org/TR/credential-management-1



CSS PAINT API

CSS PAINT API

8 CSS and JavaScript are often seen as separate parts of a website – only ever coming together in small ways, such as adding a class name. Developers have few opportunities to change how the browser decides to render a page other than directly updating the DOM.

That is all set to change in the next year as a new set of specifications start to land in browsers. They provide access to the render pipeline in the browser through JavaScript, to help developers finely tune the browser's actions to their needs.

The Houdini task force is comprised of a collection of people from browser and technology companies. They have the aim of opening up the way everything renders in the browser in a way that makes it readily approachable for developers. This includes layouts, fonts and animations.

The first specification already hitting browsers is the Paint API, which provides access to the way browsers generate images for use as backgrounds, borders or image masks.

```
class MyPaintWorklet {
  static get inputProperties() {
    return [];
  }
  paint(ctx, size, properties) {
  }
}
registerPaint('my-paint',
  MyPaintWorklet);
```

The Paint API uses a paint worklet to define what the browser should do each time it needs to update the screen. Each paint can use a new version of the worklet, which means that any persistent data should be stored elsewhere.

The 'paint' callback method gets passed three values. The first is the context, which works much like a canvas context. It can be drawn upon in the same way and has most of the same capabilities, but as this can be called multiple times a second it should be kept simple.

PROPERTIES & VALUES API

w3.org/TR/css-properties-values-api-1
CSS custom properties are strings by default. This API enables them to be any specified type, such as a number or colour. Browsers can then apply effects and transitions correctly

The second value provides information about the element being painted, such as its height and width. This can be used to make sure the end result fits perfectly each time it gets used.

The final value contains all the style properties on the object. This includes any input custom properties defined within the getter. These

use the Typed Object Model – another new Houdini feature that makes CSS values easier to digest in JavaScript.

The last call registers the worklet with the browser and opens it up to the CSS side.

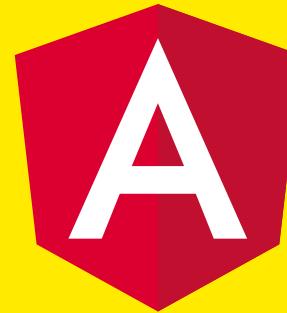
```
.el {
  background-image: paint(my-paint);
}
```

To use within CSS, it involves a one-line function. The passed value is the name of the effect provided when registering the worklet.

The API and its associated worklet is available in Chrome and Opera, while it remains a key focus for Safari, Firefox and Samsung Internet browsers. While the specification is nearing completion, expect to see it hit those browsers soon.

Later iterations on the specification provide a mechanism for supplying arguments against the CSS function, similar to how gradients are currently defined. This enables easier customisation without having to apply custom properties. Other options, such as defining an alpha channel for layering, are also being considered.

w3.org/TR/css-paint-api-1



ANGULAR

10

Angular is a framework involved in all parts of an application, including data handling and interface updates. While it can inflate bundle sizes, all parts of the application will work together seamlessly.

Version 7 of Angular adds improvements to every part of the framework. While there are few changes that developers will get to play around with, there are lots of changes to benefit speed and reliability behind the scenes.

It is now possible to add performance budgets for bundle sizes. This will make sure that a build never gets excessively large without being made aware.

As part of an overall update to Material Design, the respective Angular CDK components have also been updated. For example, the scrolling module enables support for virtual scrolling, which helps to keep long lists scrolling smoothly.

Upgrading to version 7 will, for most, involve running one line using the CLI.

```
ng update @angular/cli @angular/core
```

Future updates will focus on the new Ivy rendering engine. This will improve the ability to remove unused code from the bundle, resulting in dramatically reduced file sizes. Due to the structure of Angular applications, the rendering engine can be replaced without requiring any changes to the internal logic.

angular.io

One framework.
Mobile & desktop.

GET STARTED

DEVELOP ACROSS ALL PLATFORMS

Learn one way to build applications with Angular and reuse your code and abilities to build apps for any deployment target. For web, mobile web, native mobile and native desktop.

POLYMER

11 In its first iteration, the Polymer library was the gateway to a future of custom elements. By embracing the then-upcoming web component specifications, developers were able to piece together websites from ready-made building blocks.

Over time, these specifications have evolved, and one of the building blocks - HTML imports - was no longer part of the plan. The recent release of Polymer 3.0 addresses this issue and focuses more on using more approachable ES modules to achieve a similar goal.

The original aim of the Polymer library was to become lighter as the support for those specifications grew, but it ended up growing instead. To refocus on that aim, 3.0 is the last version of the library to be released. Moving forward, it is recommended components are made with LitElement and lit-html.

```
class MyElement extends LitElement {
  render(){
    return html`

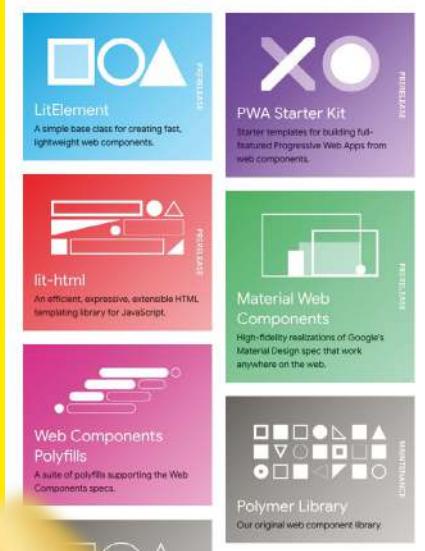
This is a LitElement

`;
  }
}
```

LitElement is a lightweight wrapper around web components to make them easier to work with. The 'html' tagged template literal is lit-html, which is a templating library with DOM diffing to keep page updates as small as possible.

Both LitElement and lit-html are in a pre-release state at the moment, but the Polymer team aims to get them released at some point early this year.

• polymer-project.org



WEB ANIMATIONS API

12 Animation with JavaScript has historically had a bad rap. Moving elements by changing position values causes lots of repaints and results in poor performance. CSS animations are the better choice, but they are hard to manipulate on the fly.

The Web Animations API provides access to the same animation engine that powers the CSS side.

```
const animation = el.animate(
  [
    { transform: "rotate(0)" },
    { transform: "rotate(360deg)" }
  ],
  1000
);
```

Every element has an 'animate' method, which takes a series of keyframes plus either a settings object or a duration. This returns an animation object, which can then be updated to play, pause and adjust the playback speed.

There is added benefit coming from Houdini, with animation worklets. These work on a similar syntax to the API, but allow for finer control. They can be controlled by inputs other than time, such as scroll or touch position. This is still in development, but Chrome is currently running an origin trial until March in order to gather more feedback. To find out more, you can head over to the Chrome developer blog at bit.ly/2QqRBsm.

Support for the Web Animations API itself is small but is quickly growing. Chrome,

Firefox and Opera have support for most of the specification, while Safari is due to be joining them in its next release.

By using feature detection, those who have support for it can feel the benefits straight away.

• w3.org/TR/web-animations-1

SERVICE WORKERS

13 Service workers enable an application to run a background process that manages some lower-level features. This can include push notifications, syncing data in the background and even providing an experience for when the device is offline. The level of control is up to the developer.

Service workers have been around for some time on Android, but they have also recently made the jump over to iOS.

While the focus is on mobile devices, they also work on desktop. Now that the majority of visitors will have access to the functionality, there is no better time to start improving the experience for users.

• w3.org/TR/service-workers-1

WRITABLE FILES API

14 The long-term goal of web applications is to get them to be on the same level as their native counterparts. Some functionality, however, remains lacking – particularly around managing local files.

Writing files currently involves creating a download and saving to a specific directory. Saving iterations over a specific file will create multiple copies, which is not always a good experience.

The Writable Files API will provide developers with more flexibility and enable users to decide where files should be stored.

This opens the door to more fully featured projects on the web, such as an online code editor or complex graphics applications.

This API is currently part of Google's capabilities project. It's currently in the process of gathering feedback on potential features before it hits the browser in an origin trial. Issues around security, and what kind of files can be written where, need to be ironed out before the API can be made official.

Find out more about the capabilities project over at the Chrome developer blog:

bit.ly/2QQoqyE

• wicg.github.io/writable-files

GENERIC SENSOR API

15 The sensors on the devices we carry around every day are a great way of providing extra benefit to an application. Using the physical environment, such as with an accelerometer, can provide added immersion for creative experience.

If we already have events for these kinds of things, why is there a new API? Devices are constantly changing and adding new and varied sensors. To save having to create new APIs for each new sensor that comes out, the Generic Sensor API creates a simplified interface for them all.

While the API is still in development, Chrome and Firefox both include it behind flags to play around with. Other browsers have it under consideration for the future.

• w3.org/TR/generic-sensor

OFFSCREEN CANVAS

16 There are times when a canvas element is the best tool for the job. For example, online games can use them to generate sprites, or videos can have processing effects applied to them.

The problem is that any interactions with the element will always work on the main thread, which slows things down for the user. For repeated animations or heavy processing, this can become a problem.

OffscreenCanvas decouples the canvas logic from the element. Without the link to the DOM, it can be used in a worker to free up the main thread.

Currently only Chrome supports this feature, with Firefox including it behind a flag.

• <https://mzl.la/2L9EHjt>

17 While Svelte is a relatively unknown framework, it has been steadily gaining traction since its release a couple of years ago. The recent State of JavaScript survey ranked it more popular last year than the historic titans of Backbone and jQuery.

The aim of Svelte is to keep the file size down by not shipping a framework at all. As part of the build process, it analyses the components and optimises them before compiling to vanilla JavaScript. The result is an application with the smallest possible size.

With version 2, there are improvements to syntax to make components as readable and predictable as possible. Updates to lifecycle hooks and computed values make it easier for the compiler to know where to make optimisations.

Features are deliberately light to keep each application fast. Many updates since version 2 have focused on avoiding situations that result in poor performance in the compiled output.

Discussions are being held about a rewrite of component logic in Svelte to bring it in line with other frameworks. As a compiler, it is able to make its own modifications to help function better, while remaining a clean environment for developers.

• svelte.technology

RESIZE OBSERVER

18 A truly component-based approach to web development is closer than ever. While CSS and JavaScript can be bundled in one package, it's difficult to then reuse that component reliably across pages.

When adding elements to a resize observer, it gets notified when the bounds of the element change. The visuals can then be updated based on the space available. As an example, the same card component can look different in either a sidebar or the main body.

This observer is currently only available in Chrome, while it is in development in Firefox. Other browsers can fall back to monitoring browser resize events, but this will hit performance.

• wicg.github.io/ResizeObserver

PAINT TIMING API

19

Performance will be a big focus in 2019. Getting file sizes down can help, but there is more to performance than what gets downloaded to the browser. The browser ‘paints’ its contents onscreen every time something changes. Until that first paint happens, users will be left staring at a blank screen. When it gets hung up in other details, such as downloading assets or processing JavaScript, the time to that first paint can take longer.

The Paint Timing API keeps detailed information on two events – the first paint and the first

‘contentful’ paint, which refers to something more than just styling, such as text or an image.

The PerformanceObserver interface describes what to do when these events occur. This can then be used to send the data to a server for analysis later on.

Right now, this is only available in Chrome and Opera, but other browsers such as Edge and Safari have both shown an interest and will possibly include it in the future.

• w3.org/TR/paint-timing



WEB SPEECH API

20

This API is made of two parts – synthesis and recognition.

Speech synthesis will take text supplied to it and read it out in the voice chosen by the user. The SpeechSynthesis object provides access to all the different options, including what it should say.

Recognition involves taking a microphone input and understanding the words being said. The SpeechRecognition object is used to detect the speech, which will fire a ‘result’ with what it thinks the user said and how confident it is.

The synthesis side of the API is supported in all major browsers. Speech recognition support is currently limited to Chrome with a prefix, or Firefox with a flag enabled.

There are discussions to expand the recognition side to include other inputs, including WebRTC to provide real-time transcripts of video conferencing. While these are still in the early stages, it could help make webinars and online presentations more accessible and easier to reference in the future.

• w3c.github.io/speech-api

Web Speech API Demonstration

Click on the microphone icon and begin speaking for as long as you like.

Welcome to Web Designer

English United States

TOOLS

ELECTRON

21

Use HTML, CSS and JavaScript to build native applications for Windows, MacOS and Linux. Electron powers many popular applications, such as Visual Studio Code, Slack and Skype for desktop.

• electronjs.org

CORDOVA

22

Build multiple different mobile and desktop applications using web technologies and one codebase. PhoneGap – Adobe’s distribution of Cordova – provides additional tools, such as iOS development on Windows.

• cordova.apache.org

STORYBOOK

23

Create a gallery of UI elements in a customisable environment separate from any application. It works with popular frameworks such as React and Vue, along with support for HTML snippets.

• storybook.js.org

TYPESCRIPT

24

Bring type safety to JavaScript. Define expected types for variables and have build tools and IDEs warn of any issues. Version 3 delivers better support for newer concepts, such as rest and spread operators.

• typescriptlang.org

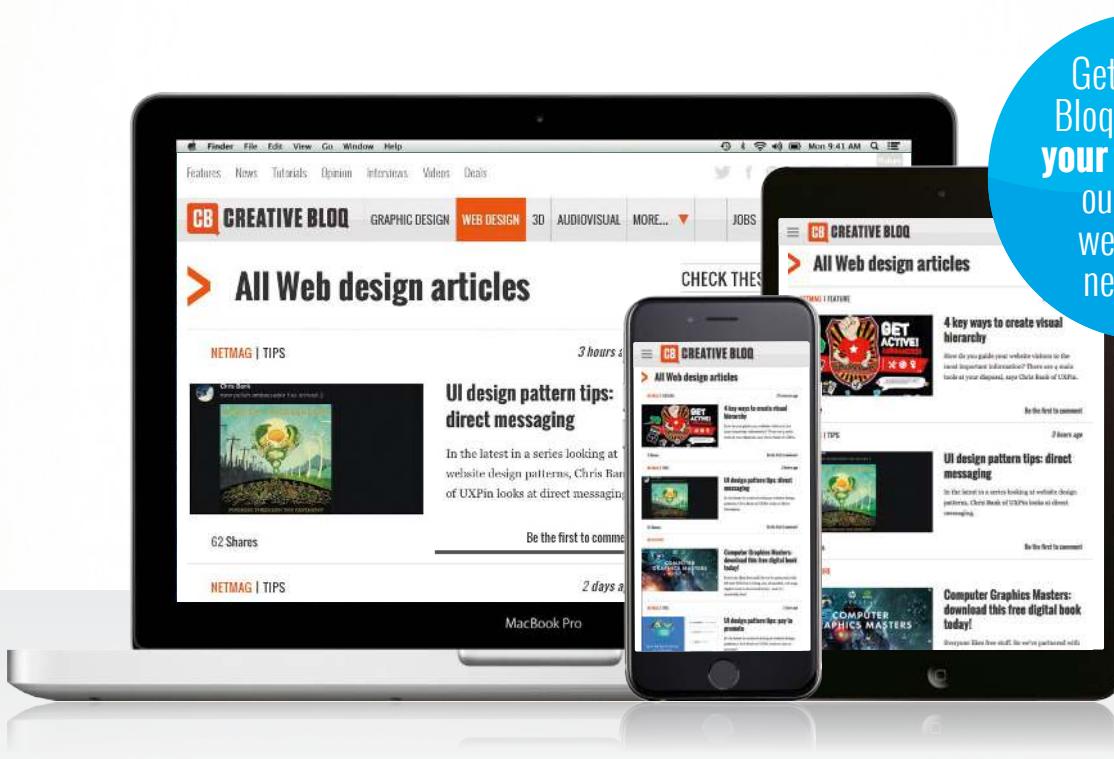
PRETTYER

25

Make sure code formatting stays consistent across a project. Have Prettier comb over files and update the formatting automatically. Some editors also support formatting on every save.

• prettier.io

The number one destination for **web design** news, views and how-tos.



Get Creative Bloq **direct to your inbox** with our weekly web design newsletter



Graphic design

Web design

3D

Digital art

www.creativebloq.com

Apply stunning SVG filters with CSS

Add special effects to text and images by using filters from SVG, which can be applied through CSS to anything on your page



**Subscribe
today**

Go to page 28
to learn more



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



SVG has been around since the early 2000s, and yet there are still interesting ways that designers are finding to use it.

In this tutorial the focus will be on the filters that are applied through SVG, but instead of applying them to an SVG image they will instead be applied to any regular page content. The way the filter is applied to the SVG is actually through CSS, by telling it what ID the filter has. Using that same idea, the filter can be applied to regular text, for example. The good part about this is that you can add some great graphical looks to your text, which would have only been previously possible by applying a number of Photoshop filters and saving as an image. Using the SVG filter, the text remains accessible and selectable, as it is still just a regular text element on your page.

The code here will create a displacement map to text that also contains an alpha map to make it appear watery and fit the theme of the page. Then another filter will be created that makes a menu appear as water blobs, which stick slightly together but blob apart as they move further away. Again this keeps with the theme of this particular page and shows two creative ways to apply SVG filters to other content.

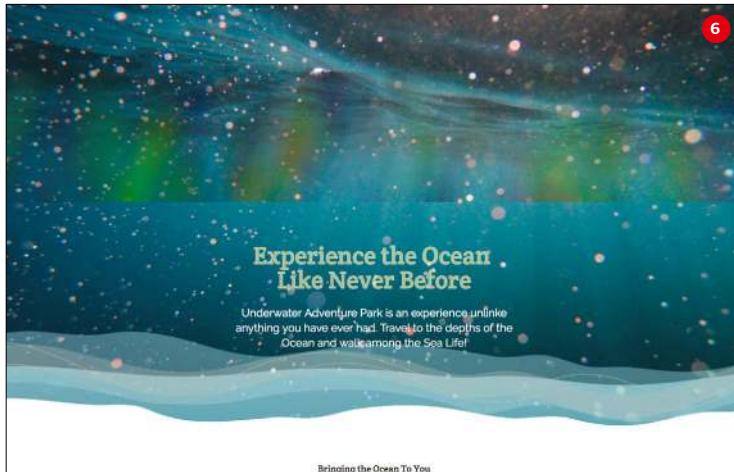
1. Starting out

Drag the 'start' project folder onto your code IDE and open the 'index.html' page. You will see there is some page content already written. The header section needs to be created, and this will contain the headline that will be affected by an SVG filter. Add the code here, just inside the 'body' tag.

```
<div class="bg">
  <div class="middle">
    <h2 class="headline">Underwater
      Adventure Park</h2>
    <div class="intro_block">
```

2. Close the header

Now the header is finished, with all the text for it in place. If you view your page in the browser at the moment, you will see a water image with some text on it. The headline, which is still currently unstyled, is



going to be styled up and have the SVG filter applied to it.

```
<h3 class="subhead">Experience the Ocean
<br>Like Never Before</h3>
  <p class="intro">Underwater
    Adventure Park is an experience unlike
    anything you have ever had. Travel to the depths of
    the Ocean and walk among the
    Sea Life!</p>
  </div>
</div>
</div>
```

3. Create an SVG filter

The SVG code can be added anywhere on the page, but as it won't be seen, it can be a good idea to place it at the bottom, before the closing body tag. The SVG filter creates turbulence noise. Notice the filter has an ID - this is what enables the CSS to apply this to another element on the page.

```
<svg xmlns="http://www.w3.org/2000/svg">
  <filter id="displacementFilter">
    <feTurbulence type="turbulence"
      baseFrequency="0.004" numOctaves=
      "2" result="turbulence" />
  </filter>
</svg>
```

4. Hide the SVG

Move over to the 'page.css' file now, and above all the CSS code for the rest of the page will be where our new CSS will go. Here the SVG is set to not display on the page at all. The heading two tag is set to have the right typeface applied to it.

```
svg {
  display: none;
}
h2 {
  font-size: 5.5vw;
  font-family: 'Crete Round', serif;
}
```

Getting more info

The Mozilla Developer Network (MDN) has a great series of documents about SVG filters ([mzl.la/2E77hK2](#)). There's also W3Schools, although it isn't quite as detailed.

5. Adding to the headline

The line-height is set to zero because later the headline will be animated, so having control over its scaling on the page is important. It's also given some padding so that it sits with the right amount of space around it and the colour is changed.

```
.headline {
  line-height: 0;
  display: inline-block;
  padding: 70px;
  color: #ccffff;
```

6. Finish the headline

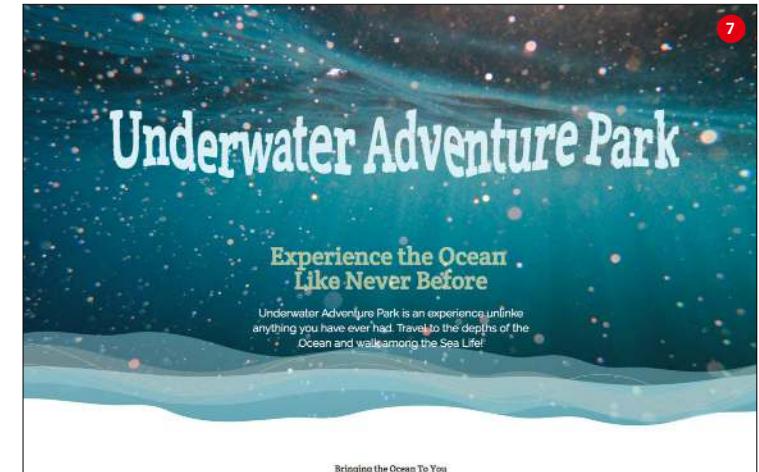
In finishing off the headline class, the next line applies the 'displacementFilter' ID in the SVG to the text. The 'translate3d' ensures that the text becomes hardware accelerated. The scale is changed slightly to ensure that when the displacement is applied it looks right.

```
filter: url(#displacementFilter);
transform: translate3d(0, 0, 0);
transform: scaleY(1.8) rotateY(-2deg);
}
```

7. Make it displace

If you test the filter at this stage the turbulence completely replaces the text. That's easy to fix. Go back to the filter code in the index.html page. This takes the turbulence and the source graphic, which is the text, and applies it as a displacement filter. Try changing the base frequency and the number of octaves in the turbulence.

```
<feDisplacementMap in2="turbulence" in="SourceGraphic" scale="30" xChannelSelector="R"
yChannelSelector="G" result="disp" />
```



Tutorials

Apply stunning SVG filters with CSS

8. Soften the edges

The edges look a little harsh for a watery effect. That can be cured with a Gaussian blur. Add the code after the displacement map. When you refresh the page, it has definitely blurred the text but the displacement is gone. Again these are elements that can be fixed on the way to creating the effect.

```
<feGaussianBlur in="SourceGraphic"  
stdDeviation="15" result="blr" />
```

9. Compositing the two

Add the composite line here below the previous Gaussian blur. You will see that this combines the blur and the displacement together, and also creates a watery translucent effect to the text. It has gone some way to softening the edges, but not enough. It would be good if the original blur could be added into this.

```
<feComposite in="blr" in2="disp" operator="in"  
result="comp" />
```

Visual builder

There is an online visual editor for making your own SVG filters, which is invaluable as a resource, at yoksel.github.io/svg-filters

10. Merging the blur

A merge operation enables the final result of the composite to be merged with the blur effect. This now looks like it fits with the background image and seems to fit with the lines of light coming through the water. The best part about the text is that it is still selectable and part of the page, unlike if you created this in Photoshop.

```
<feMerge result="final">  
    <feMergeNode in="blr" />  
    <feMergeNode in="comp" />  
</feMerge>
```

11. Create an animation

Go back to the 'page.css' file and add in the keyframes as shown here. This will just scale up the font size from a zero vertical width to a 5.5 vertical width. At the start this will be applied to the headline, so that the text scales up and into place on the screen. As the text moves, the displacement will also change over the length, giving a watery ripple.

```
@keyframes scaler {  
    from {  
        font-size: 0vw;  
    }  
    to {
```

```
        font-size: 5.5vw;  
    }  
}
```

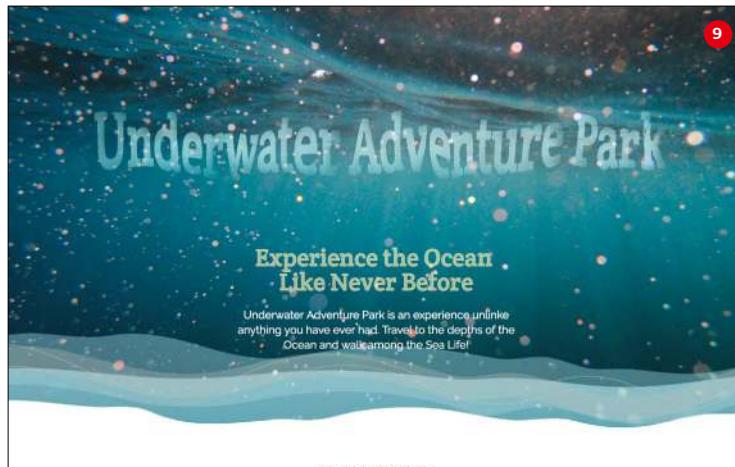
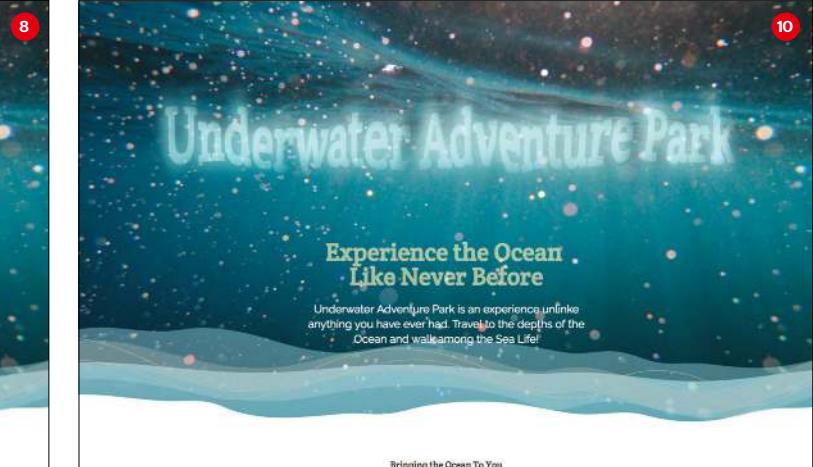
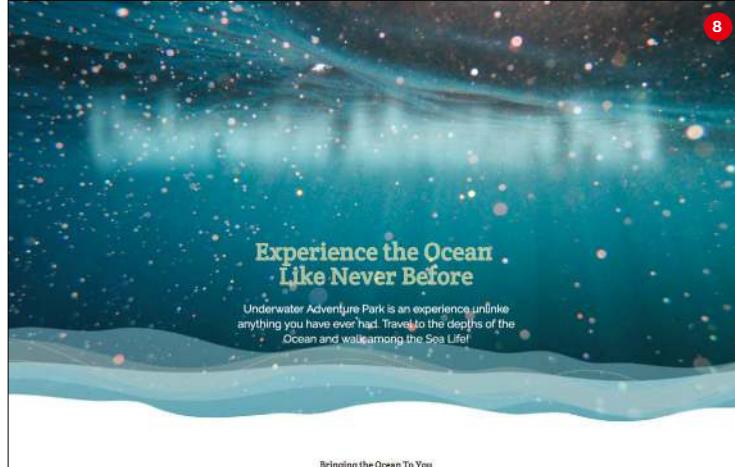
12. Change the h2 style

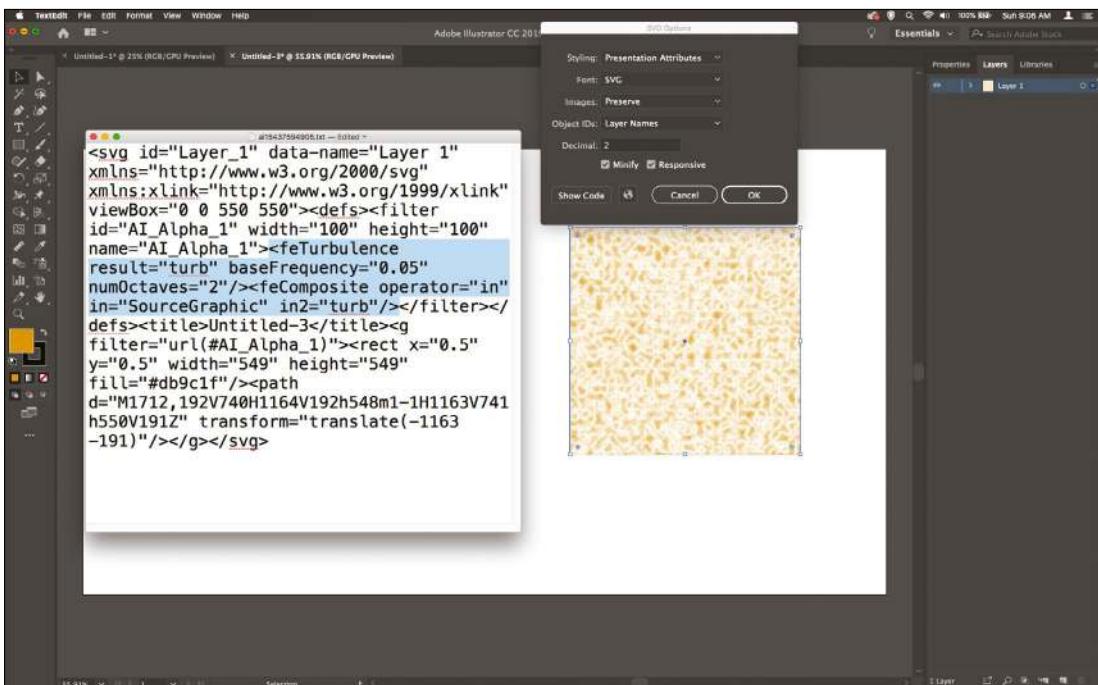
The h2 style was previously added in step 4. Replace that code with this new code, which adds the animation over four seconds to the heading. The animation stops and holds on the last keyframe. Save this and test it in the browser to see the text rippling into place.

```
h2 {  
    line-height: 0;  
    font-size: 0vw;  
    animation-name: scaler;  
    animation-duration: 4s;  
    animation-fill-mode: forwards;  
    font-family: 'Crete Round', serif;  
}
```

13. Adding navigation

Now let's create a suitable water-blob-inspired animation using another SVG filter. Add the following navigation content to the very top of the body code, before the heading that was started in the first step of the tutorial. This will create the basics of a hamburger menu in a circle.





SVG filters through Illustrator

If you feel that you can't write your own SVG filter, there is a way to bypass the process. In Adobe Illustrator is a number of prebuilt SVG filters that you can apply in the same way as we have in this tutorial. Draw a simple shape such as a rectangle on the screen and then go to Effect > SVG Filters and choose a filter. Now choose File > Export > Export As and choose SVG as the output. Click the export button and a dialogue box opens. Click the 'Show Code' button and the code will open in a text editor. You can see inside the 'filter' tags are the code that you can just cut and paste into your SVG filter right inside your code editor, and there it is, ready to use.

```
<nav class="menu">
  <input type="checkbox" href="#" class="menu-open" name="menu-open" id="menu-open" />
  <label class="menu-open-button" for="menu-open">
    <span class="hamburger hamburger-1"></span>
    <span class="hamburger hamburger-2"></span>
    <span class="hamburger hamburger-3"></span>
  </label>
```

14. Finish the navigation

Now the remaining navigation elements can be added. This also uses the Font Awesome open source icon library, which has been added to the head section to link from the CDN of this library. Each menu circular element will have an icon inside of it.

```
<a href="#" class="menu-item"> <i class="fa fa-car"></i> </a>
  <a href="#" class="menu-item"> <i class="fa fa-ship"></i> </a>
  <a href="#" class="menu-item"> <i class="fa fa-map"></i> </a>
  <a href="#" class="menu-item"> <i class="fa fa-suitcase"></i> </a>
</nav>
```

15. Add the new filter

Another filter is going to be added for this effect. In the SVG, add this code after the closing filter tag of the code added previously. Here the effects are built up in a very similar way to previously. This will allow for the menu to look like sticky blobs of liquid moving apart.

```
<filter id="shadowed-blob">
  <feGaussianBlur in="SourceGraphic" result="blur" stdDeviation="20" />
  <feColorMatrix in="blur" mode="matrix" values="1 0 0 0 0 0 1 0 0
  0 0 1 0 0 0 0 0 18 -7" result="blob" />
  <feGaussianBlur in="blob" stdDeviation="3" result="shadow" />
  <feColorMatrix in="shadow" mode="matrix" values="0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 1 -0.2" result="shadow" />
```

16. Finish the filter

The remainder of the filter is added here, which completes the effect that will be placed on each circle of the menu items. This will cause the elements to have the liquid blob effect added. Save this page and then switch over to the 'design.css' file.

```
<feOffset in="shadow" dx="0" dy="2" result="shadow" />
<feComposite in2="shadow" in="blob" result="blob" />
<feComposite in2="blob" in="SourceGraphic" result="mix" />
</filter>
```

17. Apply the filter

The CSS code can now be added to the different 'design.css', just to keep all of the navigation CSS together in the same place. Some code has been added, but here the filter is applied to the menu, which will be a fixed menu so that it is present on the screen at all times.

```
.menu {
  filter: url("#shadowed-blob");
  position: fixed;
  padding-top: 20px;
  padding-left: 80px;
  width: 650px;
  height: 150px;
  box-sizing: border-box;
  font-size: 20px;
  text-align: left;
}
```

18. Making the menu work

The menu is set to turn invisible when the menu is open. The hover element of each of the menu items is created so that there is a change when the user hovers over this. Each child of the menu is given a 0.4-second transition when the menu items are returning to their original position.

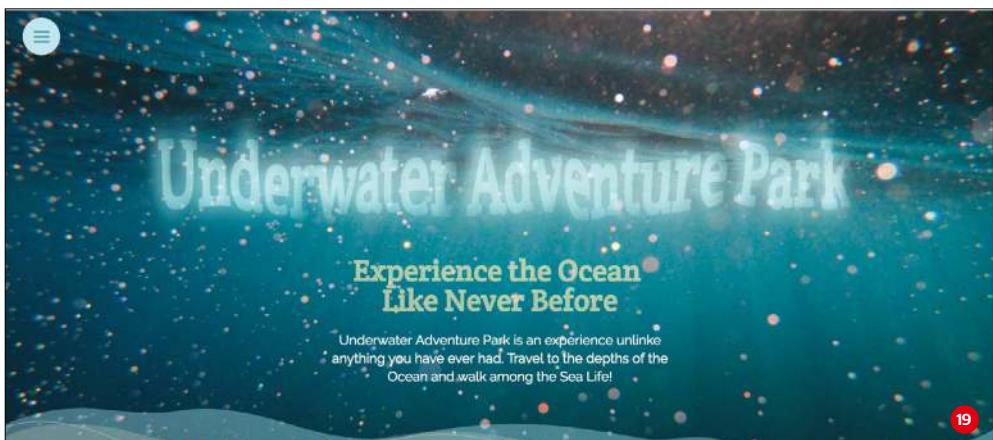
```
.menu-open {
  display: none;
}
.menu-item:hover {
  background: #47959f;
  color: #b2f0f8;
}
.menu-item:nth-child(3), .menu-item:nth-child(4), .menu-item:nth-child(5), .menu-item:nth-child(6) {
  transition-duration: 400ms;
}
```

19. Burger icon to the top

The burger icon is elevated above the other elements by changing its z-index. The menu grows when the user hovers over this, and clicking the

Tutorials

Apply stunning SVG filters with CSS



menu will now get the burger to animate from three lines of a burger to an 'X', indicating the option to remove the menu.

```
.menu-open-button {  
    z-index: 2;  
    transition-timing-function: cubic-bezier(0.175, 0.885, 0.32, 1.275);  
    transition-duration: 400ms;  
    transform: scale(1.1, 1.1) translate3d(0, 0, 0);  
    cursor: pointer;  
}  
.menu-open-button:hover {  
    transform: scale(1.2, 1.2) translate3d(0, 0, 0);  
}
```

20. Moving the elements

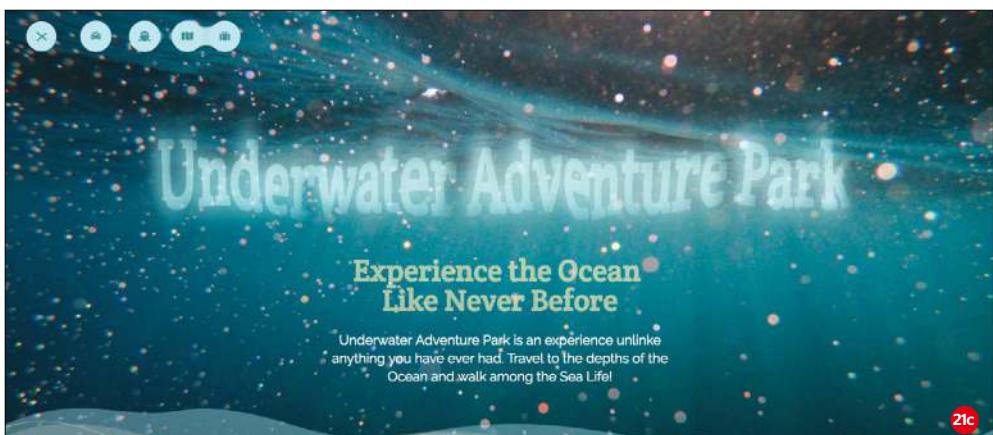
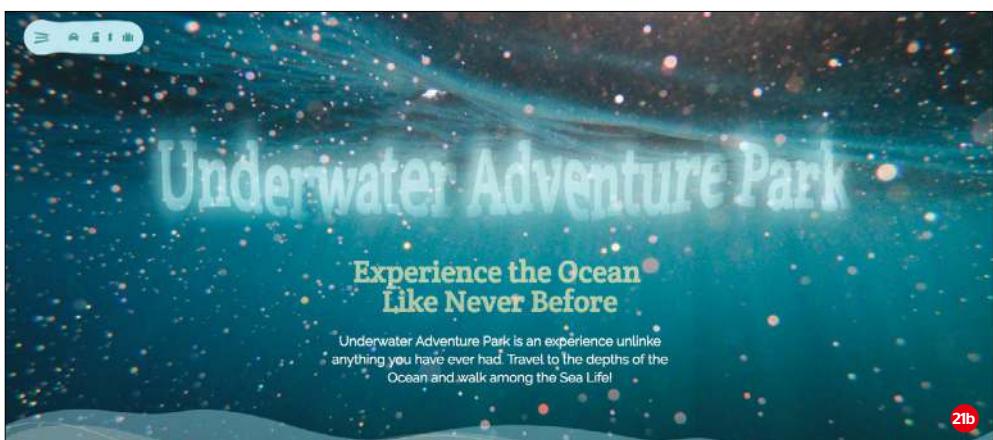
The first menu item is actually the third child of the menu, as there is a checkbox and the hamburger before it. Adding this enables the first menu element to move into position once the menu is clicked by the user. Each menu element will move out with a slightly longer time.

```
.menu-open:checked + .menu-open-button {  
    transition-timing-function: linear;  
    transition-duration: 400ms;  
    transform: scale(0.8, 0.8) translate3d(0, 0, 0);  
}  
.menu-open:checked ~ .menu-item {  
    transition-timing-function: cubic-bezier(0.165, 0.84, 0.44, 1);  
}  
.menu-open:checked ~ .menu-item:nth-child(3) {  
    transition-duration: 390ms;  
    transform: translate3d(110px, 0, 0);  
}
```

21. Remaining movement

The remaining menu elements are moved out with different speeds. This enables the elements to stick together in the early stages of the animation, which will give the blobby liquid look, using the SVG filter. Save the documents and see the finished results in your browser.

```
.menu-open:checked ~ .menu-item:nth-child(4) {  
    transition-duration: 490ms;  
    transform: translate3d(220px, 0, 0);  
}  
.menu-open:checked ~ .menu-item:nth-child(5) {  
    transition-duration: 590ms;  
    transform: translate3d(330px, 0, 0);  
}  
.menu-open:checked ~ .menu-item:nth-child(6) {  
    transition-duration: 690ms;  
    transform: translate3d(440px, 0, 0);  
}
```





iPad is a trademark of Apple Inc., registered in the U.S. and other countries. App Store is a service mark of Apple Inc.

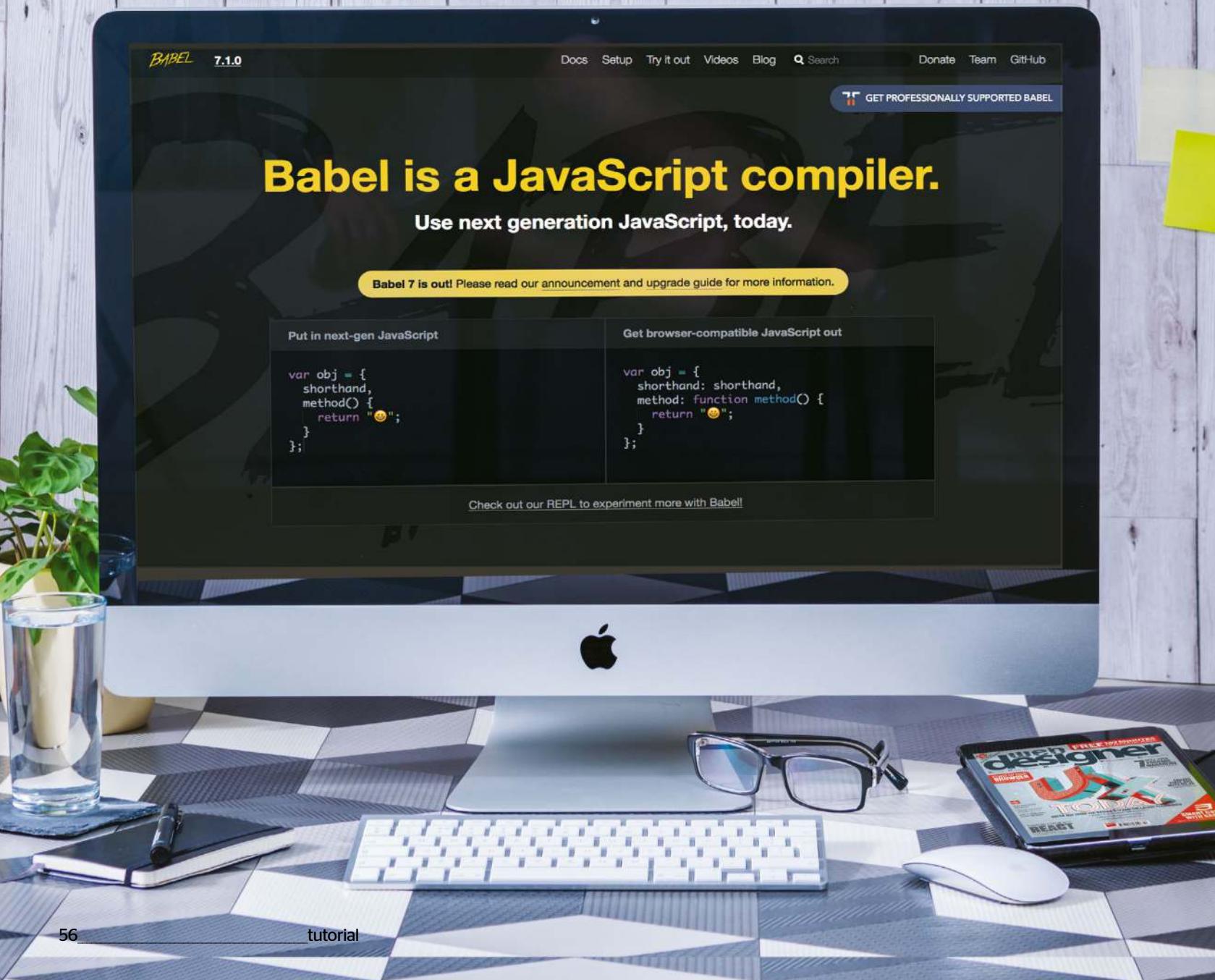
Try out a digital edition for **FREE** today!

Just search for 'ImagineFX' on these selected platforms...



Get started with the JS compiler Babel 7

Discover how to use the latest version of the JavaScript compiler and get your code to work everywhere



JavaScript is unique due to the breadth of the ecosystem. While new standards add syntactic sugar, getting them supported on browsers takes time. Babel works around this problem via automatic transpilation. The idea behind the product is simple: Babel takes in ES6 or ES7 code and replaces new syntactical elements with emulation code. Its output confirms to classic JavaScript syntax and runs on older browsers like Internet Explorer.

Babel's initial release took the world by storm. Soon after it first appeared, various frameworks like React, Vue and Ember embraced it. Developers often use the product without knowing that it works in the background – more than one popular npm project has a dependency on Babel.

These dependencies transformed the release process of the predecessor into a conflict-fraught affair. Version 7, still managed by a small maintainer team, thus tried to be as compatible as possible. Breaking changes are few and far between, while code generation quality remains high.

If you haven't worked with Babel before, let this be your guide. Being able to use advanced JavaScript features without compatibility worries makes life much easier.

1. Version check

Babel usually lives in the Node runtime environment. Let's start out by checking the versions used. The output provides the version state found on the Ubuntu 14.04 workstation used to create the following article. This isn't pedantry – the figure accompanying this step shows that the Babel team dropped support for quite a few Node.js versions.

```
tamhan@tamhan-thinkpad:~$ node --version
v8.14.0
tamhan@tamhan-thinkpad:~$ npm --version
6.4.1
```

2. Change of package names

One breaking change in version 7 has involved moving the Babel packages into their own

```
tamhan@tamhan-thinkpad:~/workspaceB7$ ls
index.js  node_modules  package.json  package-lock.json
tamhan@tamhan-thinkpad:~/workspaceB7$ npm run build

> workspaceb7@1.0.0 build /home/tamhan/workspaceB7
> babel index.js -d lib

Successfully compiled 1 file with Babel.
tamhan@tamhan-thinkpad:~/workspaceB7$ ls
index.js  lib  node_modules  package.json  package-lock.json
tamhan@tamhan-thinkpad:~/workspaceB7$ █
```

4

namespace. Older packages were not removed from the various repositories. This is important, as the use of legacy package names leads to the situation shown in the figure accompanying this step.

```
tamhan@tamhan-thinkpad:~/workspaceB7$ npm
install --save-dev @babel/core @babel/cli @
babel/preset-env @babel/node
...
+ @babel/core@7.2.0
+ @babel/node@7.2.0
+ @babel/cli@7.2.0
+ @babel/preset-env@7.2.0
```

3. Add a build action

The step above assumes that you work inside of an npm project. In that case, running Babel via the build action is easy. Open package.json and modify it as demonstrated in the code below:

```
{
  ...
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test
specified\\\" && exit 1",
    "build": "babel index.js -d lib"
  },
}
```

4. Transpile code by hand

Putting Babel to work involves firing off the build action. This is best accomplished via the npm run command. The -d value informs Babel that the results must be placed in the 'lib' folder – the figure accompanying this step shows that the folder gets created on the fly.

```
tamhan@tamhan-thinkpad:~/workspaceB7$ npm
run build
> workspaceb7@1.0.0 build /home/tamhan/
workspaceB7
> babel index.js -d lib
Successfully compiled 1 file with Babel.
```

5. A question of configuration

Invoking Babel without further configuration options does not enable transpilation. Code can be transpiled only if the framework receives further

Don't install Babel globally

Babel's maintainer team explicitly advises against installing Babel into your workstation's global binary store. The product gets used in various other frameworks, some of which depend on specific versions and might break otherwise.

All of Babel

Support for Node.js 0.10, 0.12, 4 and 5 has been dropped #5025, #5041, #7755, #5186

level of awesomeness? high

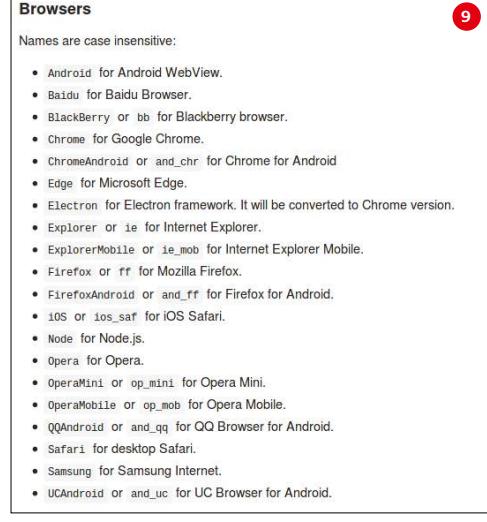
1

```
tamhan@tamhan-thinkpad:~/workspaceB6$ npm install --save-dev babel-cli
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN workspaceb6@1.0.0 No description
npm WARN workspaceb6@1.0.0 No repository field.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.4 (node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.4: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
+ babel-cli@6.26.0
added 237 packages from 120 contributors and audited 1529 packages in 9.597s
found 0 vulnerabilities
```

2

Tutorials

Get started with the JS compiler Babel 7



The screenshot shows a terminal window titled "index.js (~/workspaceB7/lib) - gedit". The code editor displays the following JavaScript code:

```
index.js
"use strict";

function tamstest() {
  [1, 2, 3].map(function (n) {
    return n + 1;
  });
}
```

To the right of the terminal, a list of browser names is shown, each preceded by a small red circular icon with the number 9.

Browsers
Names are case insensitive:
• Android for Android WebView.
• Baidu for Baidu Browser.
• BlackBerry or bb for BlackBerry browser.
• Chrome for Google Chrome.
• ChromeAndroid or and_chr for Chrome for Android
• Edge for Microsoft Edge.
• Electron for Electron framework. It will be converted to Chrome version.
• Explorer or ie for Internet Explorer.
• ExplorerMobile or ie_mob for Internet Explorer Mobile.
• Firefox or ff for Mozilla Firefox.
• FirefoxAndroid or and_ff for Firefox for Android.
• iOS or ios_saf for iOS Safari.
• Node for Node.js.
• Opera for Opera.
• OperaMini or op_mini for Opera Mini.
• OperaMobile or op_mob for Opera Mobile.
• QQAndroid or and_qq for QQ Browser for Android.
• Safari for desktop Safari.
• Samsung for Samsung Internet.
• UCAndroid or and_uc for UC Browser for Android.

information about the target environment. This can be done via a command line parameter, or by creating a file called '.babelrc' in the project root.

6. Configure the babelrc

Babel configures itself via a set of plugins, each of which applies transpilation transforms to the code base. We use the preset-env package – it comes with a pre-configured set of transformations intended to cover most bases.

```
{
  "presets": ["@babel/preset-env"]
}
```

7. Time for a test drive

Add a bit of new-age JavaScript to index.js to test the program against some live code. The code accompanying this step would not work on legacy browsers – when done, the implicit function gets replaced with a ‘normal’ declaration, as shown in the figure.

```
function tamstest(){
  [1, 2, 3].map((n) => n + 1);
}
```

8. Adjust targeting

preset-env applies most transpilations by default: the product’s goal is to create universally compatible JavaScript without regard to bandwidth and performance costs. You can change its configuration by passing in a targets object – the example accompanying this step targets specific versions of Chrome and IE.

```
{
  "presets": [
    [
      ...
    ]
  ]
}
```

Don’t over-transpile

Limit transpilation if your target system is well-defined. Directly using JavaScript language constructs leads to shorter and faster code – wrappers need to be parsed and interpreted at runtime.

```

  "@babel/preset-env",
  {
    "targets": {
      "chrome": "58",
      "ie": "11"
    }
  }
}
```

9. Advanced targeting, the second

Babel’s browser targeting isn’t limited to Chrome and Internet Explorer. Thanks to cooperation with browserslist, developers can mix and match from more than a dozen targets, as shown in the figure accompanying this step.

10. Advanced targeting, the final

Browserlist can also take advanced queries. Its homepage (github.com/browserslist/browserslist)

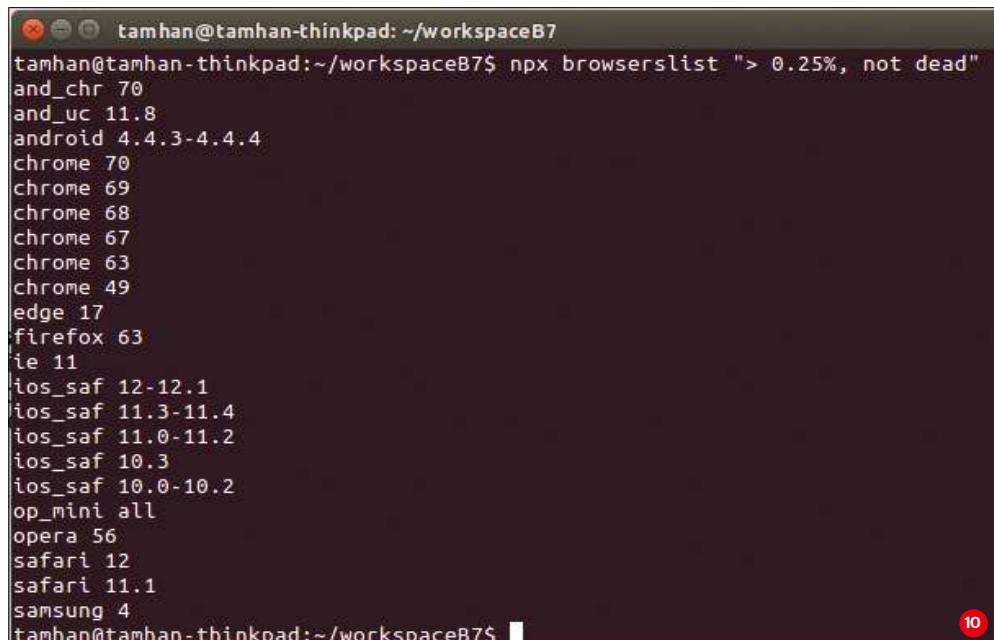
lists the configuration options, almost all of which can also be used inside Babel by modifying babelrc. Queries can be evaluated locally if your workstation has npx installed.

```
{
  "targets": "> 0.25%, not dead"
}
```

11. Automatic transpilation

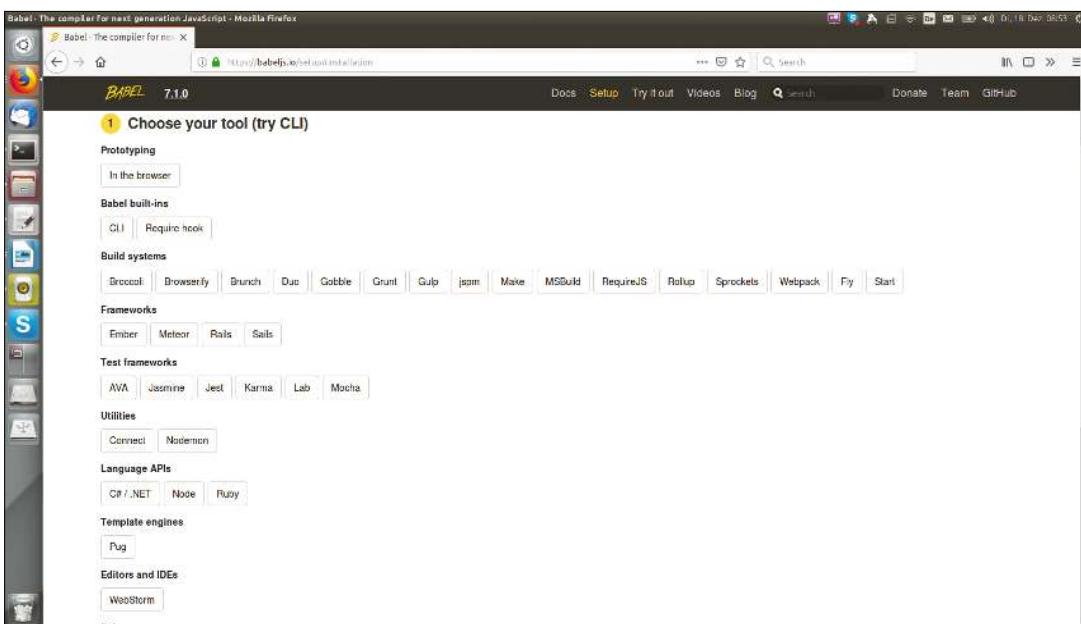
Having to invoke Babel by hand gets tedious quickly. The nodemon utility monitors filesystem resources and fires off commands as changes get detected. In theory, adding nodemon support is handled via a small change to package.json.

```
{
  "name": "workspaceb7",
  ...
  "main": "index.js",
  "scripts": {
    ...
  }
}
```



The screenshot shows a terminal window with the following output:

```
tamhan@tamhan-thinkpad: ~/workspaceB7
tamhan@tamhan-thinkpad:~/workspaceB7$ npx browserslist "> 0.25%, not dead"
and_chr 70
and_uc 11.8
android 4.4.3-4.4.4
chrome 70
chrome 69
chrome 68
chrome 67
chrome 63
chrome 49
edge 17
firefox 63
ie 11
ios_saf 12-12.1
ios_saf 11.3-11.4
ios_saf 11.0-11.2
ios_saf 10.3
ios_saf 10.0-10.2
op_mini all
opera 56
safari 12
safari 11.1
samsung 4
tamhan@tamhan-thinkpad:~/workspaceB7$
```



The art of setup

Babel integrates itself into dozens of build systems and development frameworks - the product can even work together with Microsoft's .net framework if programmatically controlled transpilation is needed. Visit babeljs.io/setup#installation to find an assistant providing step-by-step instructions for almost every configuration known to man. Simply click the button best describing your situation and follow the text.

Keep in mind that the instructions found in the tool can sometimes be outdated. At the time of writing, the command line setup routine deploys an outdated version of Babel, and it does not use the updated package names for the latest version.

```
{
  "start": "nodemon --exec babel-node index.js",
```

12. Check for presence

Some workstations have nodemon installed globally. If this is not the case, invoking the program will yield an error message similar to the one shown accompanying this step. Fortunately, deploying nodemon is easily accomplished via the npm install command.

```
tamhan@tamhan-thinkpad:~/workspaceB7$ npm
install --save-dev nodemon
```

13. Check functionality

Fire off npm start in a terminal window and proceed to change the content of index.js with an editor like gedit or Visual Studio Code. After saving, nodemon will output status information.

```
[nodemon] restarting due to changes...
[nodemon] starting `babel-node index.js'
[nodemon] clean exit - waiting for changes
before restart
```

14. Fix transpilation

While nodemon's detection should work flawlessly at this point, the contents of the index.js file that are

found in 'lib' do not update. This is caused by a nicety of babel-node - it does not commit the transpiled files to the disk. It instead fires off a modified version of the Node CLI, which works with the transpiled files.

15. Transpile code programmatically

Babel isn't limited to working on the command line. If the correct packages are installed, code can also be transpiled from another program. The snippet accompanying this step applies a set of basic transformations to an input string. Keep in mind that the configuration settings, usually, are obtained from a babelrc file.

```
tamhan@tamhan-thinkpad:~/workspaceB7$ npm start
> workspaceb7@1.0.0 start /home/tamhan/workspaceB7
> nodemon --exec babel-node index.js

sh: 1: nodemon: not found
npm: ERR! file sh
npm: ERR! code ELIFECYCLE
npm: ERR! errno ENOENT
npm: ERR! syscall spawn
npm: ERR! workspaceb7@1.0.0 start: `nodemon --exec babel-node index.js`
npm: ERR! spawn ENOENT
npm: ERR!
Failed at the workspaceb7@1.0.0 start script.
npm: ERR! This is probably not a problem with npm. There is likely additional log
ging output above.

npm: ERR! A complete log of this run can be found in:
npm: ERR!     /home/tamhan/.npm/_logs/2018-12-18T09_04_19_688Z-debug.log
```

Tutorials

Get started with the JS compiler Babel 7

```
var babel = require("@babel/core");
import { transform } from "@babel/core";
import * as babel from "@babel/core";
babel.transform("code()", options,
function(err, result) {
  result.code;
  result.map;
  result.ast;
});
```

16. Transpile entire files

Source code usually does not get stored in string variables. The Babel API accounts for this via a set of file-related functions, which forgo the input string for a variable with a filename. The results, however, get returned as a normal JavaScript variable.

```
babel.transformFile("filename.js", options,
function (err, result) {
  result; // => { code, map, ast }
}
);
```

17. Sync and async

Babel 7 introduced synchronous and asynchronous versions of most API calls. Make sure to pick the right one for your needs - while transpiling small examples can be done on the fly, setting Babel loose on more complex files can easily lead to delays running into dozens of seconds.

18. Learn about individual plugins

Should you ever find yourself wondering about what happens in the background, simply visit babeljs.io/docs/en/plugins. It provides a list of all plugins currently contained in the Babel

distribution, and also contains a few hints for all those seeking to create a plugin of their own.

19. Strip out TypeScript specifics

Babel isn't limited to transpiling new-age JavaScript elements. The product contains a feature-constrained TypeScript engine. It strips out typing information and replaces 'advanced' elements. Sadly, Babel does not perform type-checking - this eliminates one of the most significant benefits of the TypeScript language.

```
{
  "presets": ["@babel/preset-typescript"]
}
```

The screenshot shows the Babel REPL interface. On the left, there's a sidebar with 'SETTINGS' (Evaluate, Line Wrap, Minify, Prettify, File Size, Time Travel) and 'PRESETS' (es2015, es2015-loose, es2016, es2017, stage-0, stage-1, stage-2). The main area has a code editor with the line '1 "use strict";' and a status bar at the bottom right showing the number 20.

The screenshot shows the 'Upgrade to Babel 7' page. The left sidebar includes links for Guides (What is Babel?, Usage Guide, Configure Babel, Learn ES2015, Upgrade to Babel 7, Upgrade to Babel 7 (API)), General (Editors, Plugins, Presets, Caveats, FAQ, Roadmap), and Usage (Options, Config Files, cli, polyfill). The main content area has sections for 'Upgrade to Babel 7' (with a note about Node.js support), 'All of Babel' (mentioning support for Node.js 0.10, 0.12, 4 and 5), 'Config Lookup Changes' (mentioning issues with node_modules, symlinks, and monorepos), and a sidebar with links for All of Babel, Config Lookup Changes, Yearly Preset Deprecations, Stage Preset Deprecations, Remove proposal polyfills in @babel/polyfill, Versioning Dependencies, Package Renames, Scoped Packages, Switch to -proposal- for TC39 Proposals, Remove the year from package names, "use strict" and this in CommonJS, Separation of the React and Flow presets, Option parsing, Plugin/Preset Exports, Resolving string-based config values, Path-based only and ignore patterns, Babel's CLI commands (@babel/node, @babel/runtime, @babel/plugin-transform-runtime), and Spec Compliancy (@babel/plugin-proposal-object-rest-).

CREATE THE ART OF THE FUTURE

www.3dartistonline.com



3DArtist[®]

Available from all good newsagents and supermarkets

ON SALE NOW:

> Ralph Breaks The Internet > Animation Secrets > UE4 Holograms



BUY YOUR ISSUE TODAY

Print edition available at www.myfavouritemagazines.co.uk

Digital edition available for iOS and Android

Available on the following platforms



facebook.com/3DArtistMagazine twitter.com/3DArtist

Implement light or dark mode in CSS

Learn how to implement light and dark themes in CSS using the new media query ‘prefers-color-scheme’

The image shows a computer setup with a BenQ monitor. On the monitor, there are two side-by-side web browser windows. Both windows have the same title: "Implementing light or dark mode in CSS". In the left window, the text "Using prefers-color-scheme and custom properties" is displayed in red. In the right window, the same text is displayed in blue. Below this text, both windows show the media query code: "@media (prefers-color-scheme: light | dark) { ... }". At the bottom of each window, the text "Anything written inside of the media query will only be shown when" is followed by a small icon of a computer monitor.

Implementing light or dark mode in CSS

Using **prefers-color-scheme** and **custom properties**

The media query code

```
@media (prefers-color-scheme: light | dark) { ... }
```

Anything written inside of the media query will only be shown when

Implementing light or dark mode in CSS

Using **prefers-color-scheme** and **custom properties**

The media query code

```
@media (prefers-color-scheme: light | dark) { ... }
```

Anything written inside of the media query will only be shown when



The CSS specification is ever-evolving. The process for implementing new features in CSS is complicated, but the simplified version is that the CSS

Working Group decides on the new elements to be added to the specification. It is then up to the browsers to implement these new elements, and the browsers choose in which order to implement them, which is why we now have a mismatch of support for new features. While this may be annoying at times, it's a much better way to do it than the browsers implementing the full specification, as we saw in the early days of the web.

The CSS Working Group is made up of members from all major browser vendors and other technology companies like Apple and Adobe. Apple, having recently launched its new version of MacOS, wanted a way to detect its snazzy new dark mode in the browser. In order to do this, Apple pushed a recommendation to the specification for a new level 5 media query.

```
@media (prefers-color-scheme: light | dark)
{ ... }
```

Using this media query, we can detect if the user is currently using light or dark mode in the OS. At the moment this is only supported by Safari Technology Preview 69 and above, but the other browsers shouldn't be far behind.

In order to test this you will need to be upgraded to Mojave 10.14 (MacOS) and have selected Dark appearance in System Preferences. There's a few ways we can use this new media query to implement different themes. We'll explore a few of them now in this tutorial.

1. Page setup

To begin, we need to create some HTML elements to style, so we'll start by creating a new pen on CodePen and adding some elements. We'll add a container for our content, in order to centre it, and some headings and text. We'll set the CSS to use Sass in order to use nesting in CSS.

```
<div class="content-container">
  <h1>Heading One</h1>
  <h2>Heading Two</h2>
  <hr>
  <p>...</p>
  <p>...</p>
</div>
```

2. Basic styles

Next we'll add some basic styles and include some fonts from Google in order to make our page look a bit nicer. We'll style all of our basic elements, applying new font sizes, colours and fonts.

```
body {
  font-family: 'Merriweather', serif;
  background-color: #ebeded;
```

```
color: #212121;
padding: 1.618rem;
line-height: 1.618;
font-size: 16px;
}
```

3. Container styles

Next we'll style our container to make the content a comfortable line length for reading. We'll also add a background colour and drop shadow. In order to centre the content box in the page, we'll use the keyword 'auto' on the margin properties' left and right values.

```
.content-container {
  padding: 1.618rem 3.236rem;
  max-width: 48.54rem;
  margin: 3.236rem auto;
  background-color: #fff;
  box-shadow: 0 0 12px 6px rgba(0,0,0,0.05);
  border-radius: .26966667rem;
}
```

4. Highlight styles

Most websites make use of colour somewhere, and at the moment we only have whites and greys, so now let's choose a highlight colour and create a style for applying this colour. We will apply the colour using a span tag, and will use it to highlight something in our content.

```
<span class="text--alpha">Lorem ipsum</span>
.text--alpha {
  color: #c3423f;
}
```

5. Implement the media query

Now we have a page with some basic styles, let's look at ways we can implement the media query. Let's include it and start overriding some of our styles. We'll start with the body styles.

```
@media (prefers-color-scheme: dark)
{
  body {
    background-color: #111;
  }
}
```

6. Override the remaining styles

Now that we can see the media query is working and our body background colour has changed, we need to override all of our remaining styles.

```
.content-container
{
  color: white;
  background-color: #212121;
}
.text--alpha {
  color: #50a8d8;
```

Heading One

Heading Two

Sed molestie neque nec justo aliquam, sed fringilla enim feugiat. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In quis erat quis orci faucibus consetetur. Ut ullamcorper leo nunc, sit amet rhoncus ligula placerat sed. Integer eu ante eget velit fermentum eleifend.

Etiam id vulputate ex. Vestibulum at tortor tellus. Quisque lectus lectus, rutrum non vulputate egel, dignissim sit amet ligula. Aliquam lobortis maximus erat a faucibus. Nulla ante urna, tempor ac plateat eget, rhoncus sed turpis. Phasellus faucibus justo a justo cursus, eget vehicula lorem lobortis. Praesent laoreet ornare consetetur. Duis eu odio mi. Vestibulum placerat augue non pulvinar imperdiet.

3

Heading One

Heading Two

Sed molestie neque nec justo aliquam, sed fringilla enim feugiat. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In quis erat quis orci faucibus consetetur. Ut ullamcorper leo nunc, sit amet rhoncus ligula placerat sed. Integer eu ante eget velit fermentum eleifend.

Etiam id vulputate ex. Vestibulum at tortor tellus. Quisque lectus lectus, rutrum non vulputate egel, dignissim sit amet ligula. Aliquam lobortis maximus erat a faucibus. Nulla ante urna, tempor ac plateat eget, rhoncus sed turpis. Phasellus faucibus justo a justo cursus, eget vehicula lorem lobortis. Praesent laoreet ornare consetetur. Duis eu odio mi. Vestibulum placerat augue non pulvinar imperdiet.

4

Heading One

Heading Two

Sed molestie neque nec justo aliquam, sed fringilla enim feugiat. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In quis erat quis orci faucibus consetetur. Ut ullamcorper leo nunc, sit amet rhoncus ligula placerat sed. Integer eu ante eget velit fermentum eleifend.

Etiam id vulputate ex. Vestibulum at tortor tellus. Quisque lectus lectus, rutrum non vulputate egel, dignissim sit amet ligula. Aliquam lobortis maximus erat a faucibus. Nulla ante urna, tempor ac plateat eget, rhoncus sed turpis. Phasellus faucibus justo a justo cursus, eget vehicula lorem lobortis. Praesent laoreet ornare consetetur. Duis eu odio mi. Vestibulum placerat augue non pulvinar imperdiet.

5

Heading One

Heading Two

Sed molestie neque nec justo aliquam, sed fringilla enim feugiat. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In quis erat quis orci faucibus consetetur. Ut ullamcorper leo nunc, sit amet rhoncus ligula placerat sed. Integer eu ante eget velit fermentum eleifend.

Etiam id vulputate ex. Vestibulum at tortor tellus. Quisque lectus lectus, rutrum non vulputate egel, dignissim sit amet ligula. Aliquam lobortis maximus erat a faucibus. Nulla ante urna, tempor ac plateat eget, rhoncus sed turpis. Phasellus faucibus justo a justo cursus, eget vehicula lorem lobortis. Praesent laoreet ornare consetetur. Duis eu odio mi. Vestibulum placerat augue non pulvinar imperdiet.

6

Creating light and dark themes

When providing a light and dark version of your website, it is recommended to make one of the themes the default and then overriding the default styles using the media query, rather than writing each theme out in full.

Tutorials

Implement light or dark mode in CSS

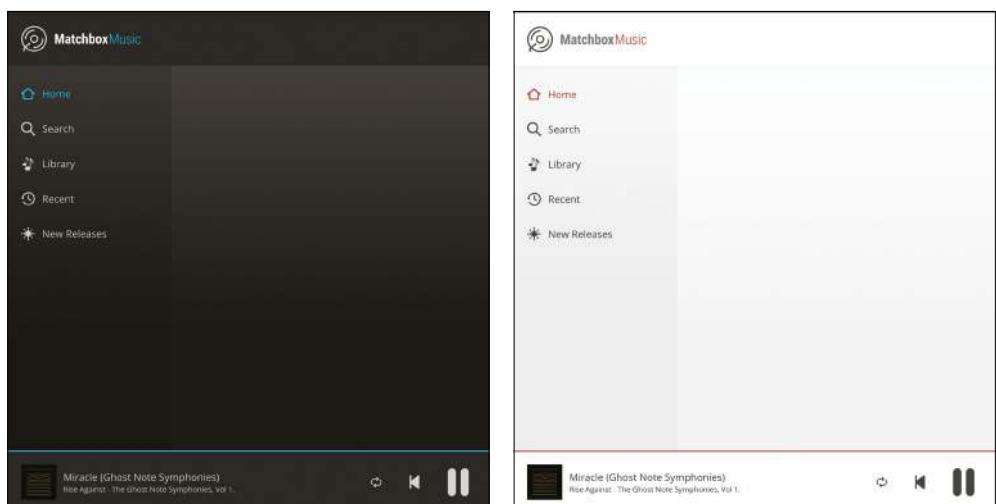
Build themes with custom properties

Custom properties have the same scope and specificity rules as everything else in CSS, enabling you to override any custom property by simply using a more specific selector.

Using custom properties to implement our theme enables us to keep adding variables, and to change those values under different circumstances.

We could use this technique to change the font, for example, if the font we're using doesn't work well on a dark background. We can change the colours of icons using SVG, enabling us to restyle an entire interface by simply changing a few custom property values.

We can also use this paired with JavaScript to read and write the values of our variables, and use this to enable users to create their own themes.



7. Maintainability

While what we've just done works perfectly well for our demo and could be maintained on smaller websites, this method would be a nightmare to manage on a larger project, with lots of different elements that all need overriding. We're also making heavy use of the cascade in our example above, whereas a large system may require more specificity in order to target all elements.



8. Another approach

So how else can we tackle the problem? Let's look at CSS filters. One of the values we can use on CSS filters is 'invert', so we could just apply this to the HTML and invert all of the colours, giving us a 'dark mode'.

```
@media (prefers-color-scheme: dark) {  
    html {  
        filter: invert(100%);  
    }  
}
```

9. Images

While the filter method works with the content we have in our document it still doesn't look great – our box shadow, for example, has also inverted, which looks quite strange. We have lost control over the styles, which becomes an even bigger problem when you have coloured backgrounds. We also have a whole new problem to consider when images are involved. Let's see what happens when we add an image to our page.

10. Custom properties

The methods we've explored so far either cause us to lose control over the styles or require a lot of maintenance in order to make sure everything is updated in dark mode. There's another way we can approach this: we can use custom properties in order to define our colours and then override them using the media query.

11. Create custom properties

In order to use custom properties, we define them at the top of our CSS inside the ':root' element. The root element has the same scope as HTML so will be available globally. We need to decide on the variable names and define their values.

```
:root {  
    --background-color: #eddeded;  
    --page-background: #fff;  
    --text-color: #212121;  
}
```

```
--color-alpha: #c3423f;  
}
```

12. Apply our custom properties

Now we have some custom properties defined we can use them in our CSS. We'll start with the body and apply the background and text colours. In order to use a custom property we use the 'var(--custom-property-name)' syntax.

```
body {  
    background-color: var(--background-color);  
    color: var(--text-color);  
}
```

13. Apply remaining properties

Using the same method, we can also update the 'background-color' of our container and the 'color' of our 'text-alpha' class to use our custom properties. All of the colours in our page are now controlled using custom properties.

```
.content-container {  
    background-color: var(--page-background);  
}  
.text--alpha {  
    color: var(--color-alpha);  
}
```

14. Re-add the media query

Now we can re-add the media query, but this time we can override the custom property values that are inside of it. We will place this right after the original root definition, and inside the media query we can now simply choose new values for all of our colour custom properties.

```
@media (prefers-color-scheme: dark) {  
    :root {  
        --background-color: #111;  
        --page-background: #212121;  
        --text-color: #eddeded;  
        --color-alpha: #50a8d8;  
    }  
}
```

```
    }
```

15. Full control

Custom properties give us full control to choose what colours and other properties we change and use. Let's update the box shadow on our page container to make it less transparent when using dark mode. To do this we need to create a new custom property for the page shadow.

```
:root {
  ...
  --page-shadow: 0 0 12px 6px rgba(0,0,0,0.05);
}
```

16. Apply the shadow

Now we've created another custom property we need to apply it to the correct element on the page. We can then override the value inside our root element in order to reduce the transparency.

```
@media (prefers-color-scheme: dark) {
  :root {
    ...
    --page-shadow: 0 0 12px 6px rgba(0,0,0,0.33);
  }
}
.content-container {
  ...
  box-shadow: var(--page-shadow);
}
```

17. Add an image

Now let's add an image back into our content, and then we can add some basic styles to float the image next to the content.

```
img {
  width: 100%;
  height: auto;
  float: left;
  max-width: 300px;
  margin-right: 1.618rem;
  margin-bottom: 1.618rem;
}
```

As we can see, since we're not using any filters the image is not altered between the two themes.

18. Add more components

Now we've got our custom properties we can keep adding elements to the page and styling them with our variables. Let's create a button class and add a button to our page.

```
.button {
  display: inline-flex;
  font-family: inherit;
  background-color: var(--color-alpha);
  color: var(--text-color);
  padding: 1.618rem 3.236rem;
  border: 0 none;
  border-radius: 0.25rem;
  text-decoration: none;
}
```

19. Button hover styles

Using the same variables, we can also create a hover style that can be used for both themes. In order to achieve this, we will invert the colours when the user hovers over the button and transition those properties in order to make the experience less jarring.

```
.button {
  ...
  transition: background-color 150ms,
  color 150ms;
}
```

```
&:hover {
  background-color: var(--text-color);
  color: var(--color-alpha);
}
```

20. Button custom properties

Custom properties have the same scope as regular CSS elements; this means we can override them using a more specific selector. We can take advantage of this and create some variables that are scoped to our button.

```
.button {
  --button-background: var(--color-alpha);
  --button-text: var(--background-color);
  background-color: var(--button-background);
  color: var(--button-text);
  ...
}
```

21. Utilising scope

We can utilise this scope in order to create different styles and hover interactions for our button in dark and light themes. We can change the value of our variables based on the media query or the state of the element, instead of repeating the property with a new value as we normally would.

```
.button {
  ...
  &:hover {
    --button-background: #ae3937;
    @media (prefers-color-scheme: dark) {
      --button-background: #2e98d1;
      --button-text: var(--background-color);
    }
  }
}
```

Heading One

Heading Two



Sed molestie neque nec justo aliquam, sed fringilla enim feugiat. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In quis erat quis orci faucibus consectetur. Ut ullamcorper leo nunc, sit amet rhoncus ligula placerat sed. Integer eu ante eget velit fermentum eleifend.

Etiam id vulputate ex. Vestibulum at tortor tellus. Quisque lectus lectus, rutrum non volutpat eget, dignissim sit amet ligula. Aliquam lobortis maximus erat a faucibus. Nulla ante urna, tempor ac placerat eget, rhoncus sed turpis. Phasellus faucibus justo a justo cursus, eget vehicula lorem lobortis. Praesent laoreet ornare consectetur. Duis eu odio mi. Vestibulum placerat augue non pulvinar imperdiet.

Heading One

Heading Two



Sed molestie neque nec justo aliquam, sed fringilla enim feugiat. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In quis erat quis orci faucibus consectetur. Ut ullamcorper leo nunc, sit amet rhoncus ligula placerat sed. Integer eu ante eget velit fermentum eleifend.

Etiam id vulputate ex. Vestibulum at tortor tellus. Quisque lectus lectus, rutrum non volutpat eget, dignissim sit amet ligula. Aliquam lobortis maximus erat a faucibus. Nulla ante urna, tempor ac placerat eget, rhoncus sed turpis. Phasellus faucibus justo a justo cursus, eget vehicula lorem lobortis. Praesent laoreet ornare consectetur. Duis eu odio mi. Vestibulum placerat augue non pulvinar imperdiet.



Aliquam lobortis maximus erat a faucibus. Nulla ante urna, tempor ac placerat eget, rhoncus sed turpis. Phasellus faucibus justo a justo cursus, eget vehicula lorem lobortis. Praesent laoreet ornare consectetur. Duis eu odio mi. Vestibulum placerat augue non pulvinar imperdiet.



Aliquam lobortis maximus erat a faucibus. Nulla ante urna, tempor ac placerat eget, rhoncus sed turpis. Phasellus faucibus justo a justo cursus, eget vehicula lorem lobortis. Praesent laoreet ornare consectetur. Duis eu odio mi. Vestibulum placerat augue non pulvinar imperdiet.

Custom property scope

This tutorial has covered how we implement light and dark themes, but it's just as important that your themes are designed well. When designing themes we need to make sure that we have enough contrast between our background and text colours and that our content is readable when using different themes.

Get started with WebVR and three.js

Learn how to create a WebVR scene using three.js



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



Virtual reality is a complex and rapidly evolving technology. It has come in and out of focus over many years, each time with varied levels of success, but it never gains enough momentum to remain.

There is a renewed effort to address this. A goal for developers is to make it easier for everyone to get into VR experiences – to remove as many barriers to entry as possible.

Enter WebVR, a JavaScript API for realising 3D virtual reality experiences in the browser. This requires low latency, high frame rate and excellent performance.

WebGL enables developers to create rich, console-quality experiences that render in real time on mobile devices and desktop browsers. Nearly universal browser and device support makes it a perfect approach for web developers wanting to create incredible VR experiences.

A go-to resource for 3D and VR on the web is the highly popular three.js. You'll be using it in this tutorial. It's free and open source, lightweight and boasts countless award-winning websites that have used it.

Other than having a JavaScript background, you can dive into this tutorial with no prior knowledge and create your first WebVR experience. The goal of this tutorial is to get you started and to get you inspired to continue to explore and utilise this very exciting technology.

1. Enable WebVR flag in Chrome

WebVR is still an experimental technology and requires HTTPS to run on your server, and won't run on mobile devices without a polyfill. github.com/immersive-web/webvr-polyfill. However, you can run content locally in Chrome for testing and building. Be sure to enable the Chrome WebVR flag. Go to <chrome://flags/#enable-webvr>, and then click enable to activate it. You may need to restart Chrome as well.

2. Install the WebVR API Emulation plugin

In order to test locally on your desktop and avoid the need for a device, there is a great plugin that you can use on Chrome by clicktorelease. The plugin will emulate a headset for you and allow you to move and rotate the headset virtually. You can get it here: bit.ly/2L8IM05.

1. WebVR
Enabled

Override software rendering list
Disabled

Canvas 2D Image Chromium
Default

2. WebVR API Emulation
Offered by www.clicktorelease.com

3. Open device toolbar in developer tools

To emulate a mobile device or headset, it works best to use the device emulation in Chrome tools. Open the developer tools in Chrome, and toggle on the device toolbar to see mobile view, rotate to landscape and select favourite phone emulation.



4. Create a basic HTML file

Next, you need to set up a basic HTML file. You can set up external CSS and JavaScript files or include inline for simplicity. Threejs's renderer class will create a <canvas> element for you. Add the following code to your index.html file:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>WebVR Demo</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
    <style>
      html, body { margin: 0; padding:0; overflow: hidden; }
    </style>
  </head>
  <body>
    <script>
      // Code will go here
    </script>
  </body>
</html>
```

5. Include three.js classes

Include a link to the three.js library in the head of your file – either hosted externally, or download it from the three.js repository. You'll also need the new WebVR class and BoxLineGeometry class for this tutorial. You can find the library and supporting classes at github.com/mrdoob/Three.js. Note: the code in this tutorial has been tested on the latest release of three.js v99.

```
<script src="libs/three.min.js"></script>
<script src="libs/WebVR.js"></script>
<script src="libs/BoxLineGeometry.js">
</script>
```

6. Add global variables

Between the script tags for the code, add the following global variables to globally access the camera, scene, renders, objects and raycaster. Also add a 'HIT' variable to keep track of objects that are intersected by the gaze of the camera. This will demonstrate how to know what a user is looking at in VR.

```
var clock = new THREE.Clock();
var container, camera, scene, renderer, room,
crosshair, HIT;
var objects=[]; // collection of objects
var num=100; // number of objects
var raycaster = new THREE.Raycaster();
```

7. Create a 3D scene

You're going to add a basic 3D scene, which will be the container for your objects. The scene is the stage that will render with the camera. All 3D presentations will have a scene or stage of some form. What is in that stage and in view of the camera is what the user will see. Add the following code to add a scene:

```
// create a scene object
var scene = new THREE.Scene();
```

Smooth frame rates

WebVR is a complex and technically challenging new medium. Three.js offers up several examples of how to use the new WebVR class and some good default settings to use. Consider smooth frame rates a higher priority than quality to avoid nausea. Also be mindful of things like distances, sizes and intuitive interaction to avoid discomfort for users.

Tutorials

Get started with WebVR and three.js

8. Add a perspective camera

Next, you need to add a camera. You'll use the perspective camera, meant for 3D scenes. The first attribute is the field of view of the camera. The second is the aspect ratio (width / height). Then you can indicate the near clipping plane and the far clipping plane distances, which define what is to be visible to the camera.

```
// create camera
camera = new THREE.PerspectiveCamera
( 70, window.innerWidth/window.innerHeight,
0.1, 1000 );
scene.add( camera );
```

9. Add a renderer and canvas element

The renderer handles the drawing of the objects in your scene that are visible to the camera. Set the antialias property to true to get smooth edges on the object. The renderer creates a domElement, which is actually an HTML <canvas> element. You can then append to the body. Note the use of the new VR-enabled flag of the renderer.

```
renderer = new THREE.WebGLRenderer({
antialias:true);
renderer.setPixelRatio( window.
devicePixelRatio );
renderer.setSize( window.innerWidth, window.
innerHeight );
renderer.vr.enabled = true;
document.body.appendChild( renderer.
domElement );
```

10. Add camera crosshair

To help users orient to the camera's point of focus, it is good practice to add a crosshair or targeting icon in front of the camera. You can add it directly to the camera object so it's always where it should be.

```
crosshair = new THREE.Mesh(
new THREE.RingBufferGeometry( 0.02, 0.04, 32 ),
new THREE.MeshBasicMaterial( {
color: 0xffffffff,
opacity: 0.5,
transparent: true
} )
);
crosshair.position.z = - 2;
camera.add( crosshair );
```

Keep up with requirements

Since WebVR is an emerging technology it is a good idea to keep up with the latest developments. You need to know which devices are supported and under what conditions. Bookmark these sources to keep up to date:

- **WebVR:** webvr.info
- **WebVR examples:** webvr.info/samples

11. Create a VR room object (optional)

Next, create a simple room object. This is nice to give the user a sense of orientation in the VR world. It creates a simple room box with lines to indicate the walls, floor and ceiling.

```
room = new THREE.LineSegments(
new THREE.BoxLineGeometry( 6, 6, 6, 10, 10, 10 ),
new THREE.LineBasicMaterial( { color:
0x080808 } ));
room.position.y = 2;
scene.add( room );
```

12. Add lights to the scene

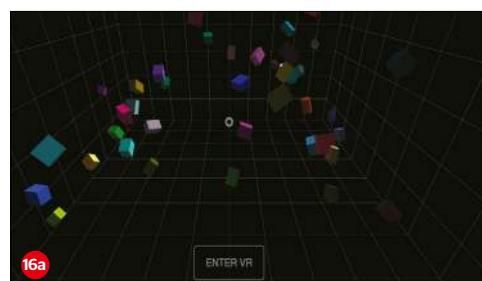
To light the scene, we'll use a simple hemisphere light and a directional light. It'll give a nice ambient visibility and some realistic shading from a uniform light source as well.

```
scene.add( new THREE.HemisphereLight
( 0x080606, 0x404040 ) );
var light = new THREE.DirectionalLight
( 0xffffff );
light.position.set( 1, 1, 1 ).normalize();
scene.add( light );
```

13. Create some objects

You're going to fill the room with objects next. Apply them randomly around the room. You will also set the rotation and scale randomly for variety. You can add a little bit more code in the next step, where it says 'create orbit attributes' to enable some custom orbit paths.

```
var geometry = new THREE.BoxBufferGeometry
( 0.15, 0.15, 0.15 );
for ( i=0;i<=num;i++ ){
    var material = new THREE.MeshLambertMaterial
    ( { color: Math.random() * 0xffffff } );
    var object = new THREE.Mesh
    ( geometry, material );
    object.position.set( Math.random()*4.0
    - 2.0, Math.random()*4.0 - 2.0, Math.
    random()*4.0 - 2.0 );
    object.scale.set( Math.random() + .5, Math.
    random() + .5, Math.random() + .5 );
    object.rotation.set( Math.random() * 2 *
    Math.PI, Math.random() * 2 * Math.PI, Math.
    random() * 2 * Math.PI );
    // create orbit attributes
}
```



14. Add orbit attributes to objects

To enable some nice random orbiting motion, and to keep the objects from 'escaping the room' we'll assign some initial angle data (in radians) and a distance. It enables a simple way to animate the objects in the render loop after.

```
// create orbit attributes
// calc distance as constant and assign to
object
var a = new THREE.Vector3( 0, 0, 0 );
var b = object.position;
var d = a.distanceTo( b );
object.distance = d;
object.radians = Math.random()*360 * Math.
PI/180; // initial angle
object.radians2 = Math.random()*360 * Math.
PI/180; // initial angle
object.radians3 = Math.random()*360 * Math.
PI/180; // initial angle
room.add( object );
objects.push( object );
```

15. Add a window resize handler

As we test our WebVR app, we'll be resizing the screen, moving it around, etc. It's a good idea to have a handler that adjusts the dimensions of the render area and updates things to keep it filling the screen properly and looking nice.

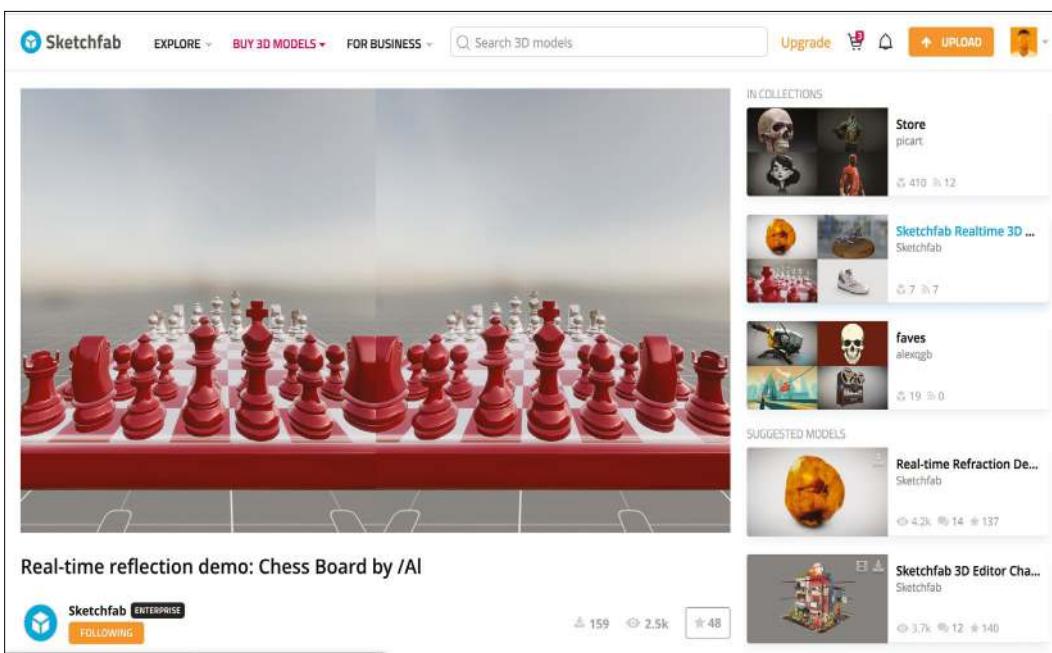
```
window.addEventListener( 'resize',
onWindowResize, false );
function onWindowResize() {
    camera.aspect = window.innerWidth / window.
innerHeight;
    camera.updateProjectionMatrix();
    renderer.setSize( window.innerWidth,
window.innerHeight );
}
```

16. Create the WebVR button

Three.js's new WebVR class includes a WebVR button, which handles toggling in and out of VR mode for us. It also handles if the device doesn't support VR mode. You can include it with this simple code:

```
// three.js webvr button to enter/
exit vr mode
document.body.appendChild( WEBVR.createButton
( renderer ) );
```





WebVR resources

Getting started in WebVR can be overwhelming. We've assembled some examples of sites using WebVR and resources to help you get rolling.

SketchFab: you probably already know this site because of its amazing gallery of assets, but it also has a WebVR mode that enables you to navigate in VR. [sketchfab.com]

A-Frame is a rock-solid framework for AR and VR. It handles the pains of fallbacks and device support for you, including enabling you to render in your browser for testing. It even abstracts away creation of common 3D objects. [aframe.io]

Three.js, meanwhile, has numerous examples with source code to help you to get started in WebVR. Utilise it as a perfect starting point.

three.js examples can be found here: bit.ly/2Qo27Rg

17. Start the VR animation loop

Typically, you'd be using the `requestAnimationFrame` to handle the render loop, but in VR you have to use a different loop handler to ensure everything is ready to render and that you avoid latency and render issues. Instead, use the new `setAnimationLoop` and pass in your render function.

```
// start the VR animation loop
renderer.setAnimationLoop( render );
```

18. Create the render function

Next, create a render function. If you didn't want to animate your objects or test for the camera/crosshair intersecting with objects, you could just use the following code:

```
function render() {
  // find intersections
  // animate the objects
  // render the scene
  renderer.render( scene, camera );
}
```

19. Test for intersections

To enable testing for objects intersecting the ray traced from the camera into Z space, add the following code to your render loop where you commented it in the last step:

```
raycaster.setFromCamera( { x: 0, y: 0 },
  camera );

var intersects = raycaster.intersectObjects(
  room.children );
if ( intersects.length > 0 ) {
  if ( HIT != intersects[ 0 ].object ) {
    if ( HIT ) { HIT.material.emissive.
      setHex( HIT.currentHex ); }
    HIT = intersects[ 0 ].object;
  }
}
```

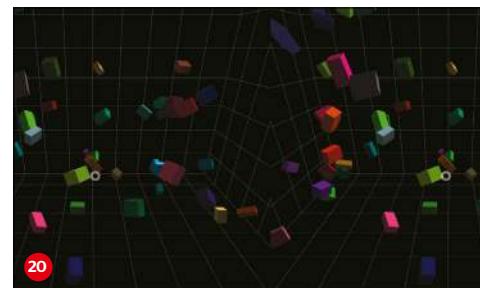
```
HIT.currentHex = HIT.material.emissive.
getHex();
HIT.material.emissive.setHex( 0x00ff00 );
}
} else {
  if ( HIT ){ HIT.material.emissive.setHex
  ( HIT.currentHex ); }
  HIT = undefined;
}
```

```
o.distance);
o.position.z = (Math.sin(o.radians3) *
o.distance);
o.position.y = (Math.sin(o.radians2) *
o.distance*5);
```

20. Animate objects along orbits

Next, you can animate your objects along their orbit paths using this code:

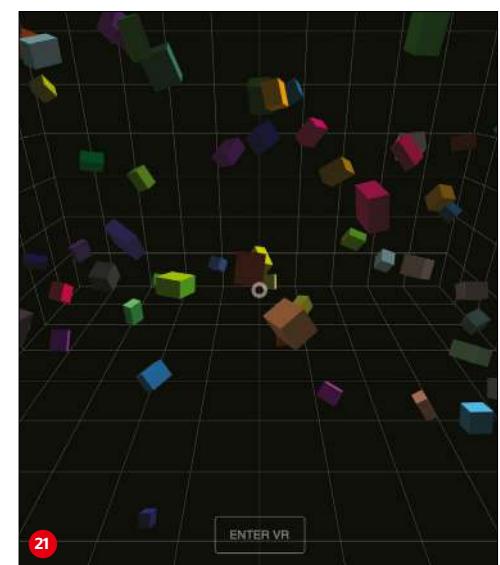
```
for (i=0;i<num;i++){
  var o = objects[i];
  o.rotation.y+=.01;
  if( i % 2 == 0 ) {
    o.radians+=.004;
    o.radians2+=.005;
    o.radians3+=.008;
  } else {
    o.radians=-.006;
    o.radians2=-.005;
    o.radians3=-.003;
  }
  o.position.x = (Math.cos(o.radians) *
```



21. Render the WebVR scene

Finally, you can render out your scene using the usably render functions. If you haven't added this line already, do it now. Once you've added this, you can test it all out and should see a WebVR scene rendering in your browser. You can also check it out on your mobile device or VR headset.

```
// render the scene
renderer.render( scene, camera );
```



web workshop

Airport message board text effect

Inspired by cocaineroutes.com

EXPERT ADVICE

Bring the map to life

There are many great uses of the Google Maps API on the web. What's interesting about the Cocaine Routes site is that a regular map is used as the backdrop to animate clouds and airline routes around the globe. Tapping into the journey to the user's home really elevates the creative use of the maps.

Airport message board

The effect seen on airport message boards, when the text changes and updates, is a similar effect that is found here on the homepage with text on the screen.

Call to action

The main call to action button on the screen is made to start the experience. This also has a rollover airline message board effect to the text.



Background journeys

The red lines in the background show different airline routes that are used to transport cocaine. These constantly update, with a white dot moving along the path.

Link to watch

The *Narcos* Netflix show is linked using the logo for the show at the bottom of the screen, to get users to the show.

Cloud effects

Cloud images are set to move across the screen and give the map a sense of being a live view of the journeys being made.



<comment>

What our experts think
of the site

Bringing the journey to life

The site takes static content and brings it to life in the journey of the drugs being smuggled around the world. The static Google map is animated with airlines and clouds. When the drug is transferred the Google Street View images are copied before being animated back, to show the content arriving at its location.

Mark Shufflebottom, Professor of interaction design

centre, both horizontally and vertically, in the browser viewport.

```
<div class="container">
  <div class="text"></div>
</div>
```

5. Add the messages

Now an array holds any messages that you need to display inside the text. The document is set to take the 'text' class as the class to update with new text. Then this is set to be the text scramble object.

```
<script>
var phrases = [
  'FROM THE GUADALAJARA CARTEL TO THE WORLD',
  'FROM THE GUADALAJARA CARTEL TO THE WORLD',
];
var el = document.querySelector('.text');
var fx = new TextScramble(el);
var counter = 0;
```

6. Bringing it to life

The next function is recalled over and over with the 'setTimeout' function, and this calls the text to be updated with the next phrase. Save the page and run this to see the text update in the browser and give the airport message board text effect.

```
var next = function next() {
  fx.setText(phrases[counter]);
  then(function() {
    setTimeout(next, 1400);
  });
  counter = (counter + 1) % phrases.length;
};
next();
```

```
height: 100%;
width: 100%;
align-items: center;
display: grid;
}
```

3. Right in the centre

The following code styles the text to sit right in the centre of the screen. The final 'dud' is for the in-between text as the text changes from one message to another. This just makes that colour a little less bright.

```
.text {
  margin: auto auto;
  font-weight: 100;
  font-size: 28px;
  color: #fafafa;
}
.dud {
  color: #e6e6e6;
}
```

4. On the page

The following code places the content onto the page and enables the 'container' to be a grid box that the text sits inside. The grid makes the text very easy to

Technique

1. The airport board letter effect

To create the text effect that looks like an airport message board, the following JavaScript library needs to be hooked up, so that the desired effect can be passed onto the letters in order to update the screen.

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

2. Styling up the content

Now, the following CSS needs to be added to either a CSS file or style tags on the page. This just puts the background map image to cover the page, and sets the typeface and then the container to put the text inside.

```
@import 'https://fonts.googleapis.com/
css?family=Roboto+Mono:100';
html,
body {
  background: url(bg.jpg) no-repeat center
  center;
  background-size: cover;
  font-family: 'Roboto Mono', monospace;
  height: 100vh;
}
.container {
```



Manage actions with Redux Thunk

Communicate with the server and keep the interface up to date without getting tangled up in state



Subscribe today
Go to page 28
to learn more



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



tate is a big part of a React application, which is why Redux is commonly paired with it. That data often comes from a database, which requires a request and a response. For some applications, this communication can be constant. It can be tricky trying to manage solely within React components.

This also introduces some new problems – what happens if it loads slowly, or doesn't load at all? Every component dealing with asynchronous data would have to contain logic to handle these scenarios.

A 'thunk' is a concept that can help with this situation. Each thunk is a function that returns another function. That function can then be called at a later point, much like a callback. If we could dispatch a thunk instead of an action object, we can add in some extra logic in reaction to another event.

Redux Thunk is a library that sits in between dispatched actions and the reducer. When it sees a thunk is dispatched, it passes some methods into the returned function that can be used to dispatch further actions, like a success or error event.

This is the second in a series of tutorials all about Redux. In the last tutorial we set up 'PhotoShare' – a photo-commenting application powered by Redux. In this one, we will make use of thunks to help pull in the data from a server rather than from a JSON file.

1. Install dependencies

There are two parts to this project – the front-end site and the back-end server it runs on. While the tutorial focuses on the frontend, we need a server running in order to fetch the photos. Download the project files and install the dependencies for both the site and the server. Run the servers for both and leave them running in the background.

```
/* one terminal inside /site */
> yarn
> yarn start
/* one terminal inside /server */
> yarn
> yarn start
```

2. Set up middleware

Redux Thunk is a 'middleware' – functionality that sits between actions and reducers that can change how those actions behave. Redux supports multiple sets of middleware that cover the entire application. They get added when the store is created using the 'compose' method. Add middleware to the 'createStore' method within index.js.

```
import { applyMiddleware, compose } from "redux";
import thunk from "redux-thunk";
[...]
const store = createStore(
  rootReducer,
  compose(
    applyMiddleware(thunk),
    devtools
```

1

```
server — node • node /usr/local/Cellar/yarn/1.12.1/libexec/bin/yarn.js start --slow — 51x17
matts-mbp-2:server matt$ yarn start --slow
yarn run v1.12.1
$ node index.js --slow
Adding artificial latency...
Example app listening on port 3001!
Delaying request to /photos by 2234ms...
Delaying request to /photos by 2430ms...
Delaying request to /photos by 1822ms...
Delaying request to /photos by 1207ms...
Delaying request to /photos by 2542ms...
Delaying request to /photos by 1576ms...
Delaying request to /photos by 1469ms...
```

```
)  
);  
dispatch(loadGalleryStart());  
};
```

3. Set up action creators

The first thing we need to do now is to load the photos into the gallery. Like regular actions, we need action creators for the various states that an asynchronous call will take. Most will have 'start', 'success' and 'error' actions. These let Redux know what the JavaScript is busy doing. Within actions/photos/photos.js, set up three action creators for these different states.

```
export const loadGalleryStart =() => ({
  type: LOAD_GALLERY_START });
export const loadGallerySuccess =
photos => ({
  type: LOAD_GALLERY_SUCCESS,
  photos
});
export const loadGalleryError =() => ({
  type: LOAD_GALLERY_ERROR });
```

4. Create a thunk for loading

Thunks work exactly the same as action creators. We still dispatch the return value, but this time it returns a function instead of an object. The middleware that we set up earlier will pass a dispatch method into the returned function. This enables us to send more actions to Redux after the initial dispatch. Create a 'loadGallery' method that returns a function. For now, have it dispatch an action to show that the gallery is still loading.

```
export const loadGallery = () =>
dispatch => {
```

5. Load data from the server

We are now ready to start fetching from the server we set up at the beginning. We can do this by using 'axios' – a package designed to work with promises across different browsers. Import axios and make a request for the photos within loadGallery. If the promise resolves, dispatch the success action, and if not dispatch the error action. With that, the structure of the thunk is complete.

```
import axios from "axios";
[...]
return axios
.get("http://localhost:3001/photos")
.then(response => dispatch(
  loadGallerySuccess(response.data)))
.catch(() => dispatch(
  loadGalleryError()));
```

6. Dispatch the thunk

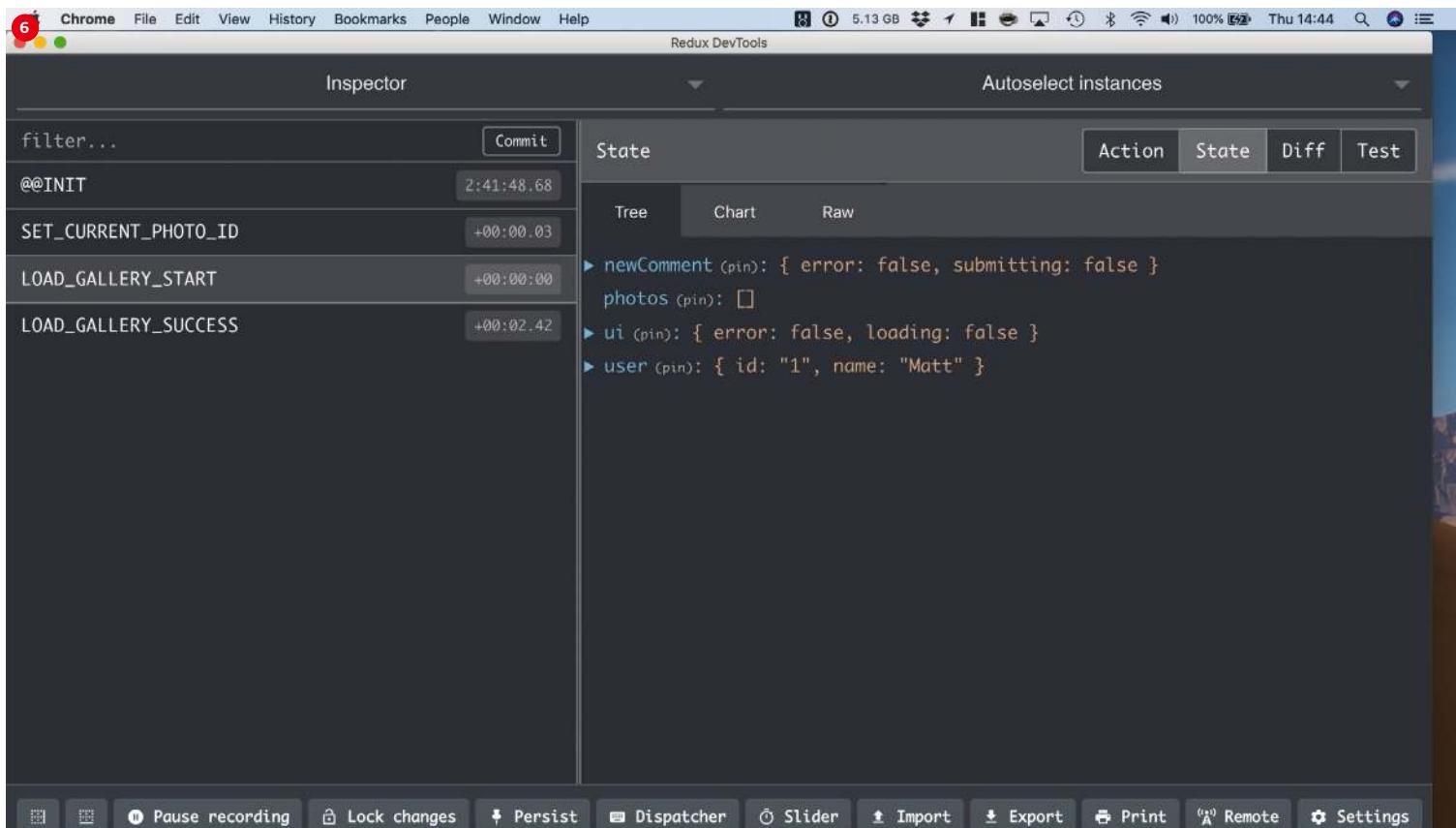
The thunk will not do anything until it's been dispatched. We can do that within a React component like any other action. A good time to start loading the photos is when the user views the main gallery. We can use React's componentDidMount lifecycle

Sending back errors

Axios gives lots of information about any unsuccessful requests, such as the status code and a stack trace. Send these back to the server or another third-party logging tool to keep track of issues.

Tutorials

Manage actions with Redux Thunk



method as a trigger, after checking the gallery is not already loaded. Within components/container/Gallery/Gallery.js dispatch a loadGallery action by adding it to mapDispatchToProps and calling it within componentDidMount.

```
componentDidMount() {
  if (!this.props.photosLoaded) {
    this.props.loadGallery();
  }
}

export const mapDispatchToProps = dispatch => ({
  loadGallery: () =>
    dispatch(loadGallery()),
});
```

7. Add photos on success

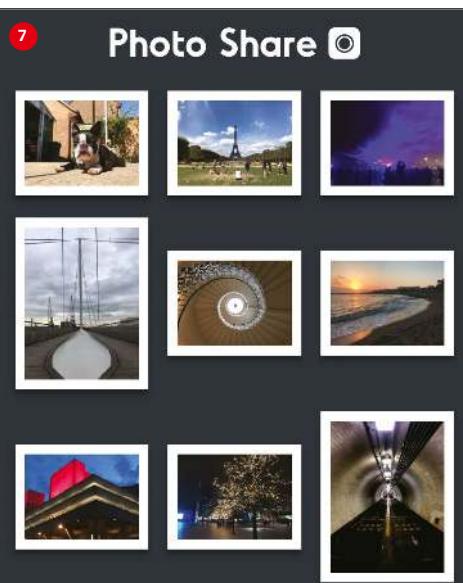
When the photos come back from the server, we dispatch a 'LOAD_GALLERY_SUCCESS' action with the photos. We need to get this into the state through the 'photos' reducer. Head to reducers/photos/photos.js and add a case for the success action. The payload

contains all the photos as an array. Once the state is updated, the photo selector passes the photos through to the gallery component to be displayed.

```
case LOAD_GALLERY_SUCCESS:
  return action.photos;
```

8. Set up the UI

Currently, the photos suddenly appear after they are loaded. On a slower connection, the user will be looking at a blank screen until the request finishes, if it ever does.



To wait or not?

It can sometimes be a better experience to update the UI immediately without waiting for a response from the server. Any action that requires server validation should show a loading state instead.

The actions we send to load photos can also be picked up in the UI reducer in order to keep the interface up to date with what's happening. Update the loading and error flags within the UI reducer at reducers/ui/ui.js.

```
case LOAD_GALLERY_ERROR:
  return { ...state,
    loading: false, error: true };
case LOAD_GALLERY_START:
  return { ...state,
    loading: true, error: false };
case LOAD_GALLERY_SUCCESS:
  return { ...state,
    loading: false };
```

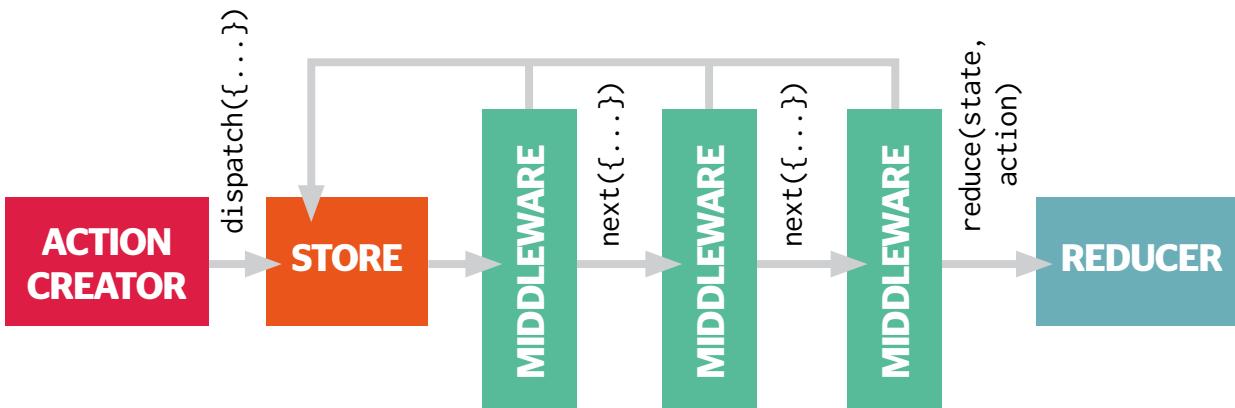
9. Add loading and error selector

As with the gallery photos themselves, we need selectors to get the various UI states values out of Redux. We can pass these to the gallery, which will then render different elements if either one is true. In selectors/ui/ui.js, add a couple of functions to get the values out.

```
export const isGalleryErrored =
  state => state.ui.error;
export const isGalleryLoading =
  state => state.ui.loading;
```

10. Add data to GalleryContainer

With the selectors ready, they can now be added to the Gallery container component. Adding them here



What is Redux middleware?

Without middleware, Redux enables us to dispatch actions as objects to then be sorted by the reducer. Some situations, such as server communication, require a bit more control over what gets sent and when.

Middleware sits in between the dispatched action and the reducer, enabling us to determine

if any action needs to be taken before continuing. In the case of Redux Thunk, the middleware knows that if it receives a function it should run it rather than carry on. It can then dispatch its own actions as needed.

Once it's finished with the action, it can pass it along to the reducer by calling 'next'. This makes

it possible to combine different middleware together without worrying about losing actions.

Middleware is readily available to be downloaded and used as part of a third-party package, but it can also be useful to keep other logic, such as analytics and error handling, away from the main application.

means that the component responsible for displaying the gallery does not need to know about how the data has arrived. Head to container/Gallery/Gallery.js and add the selectors to mapStateToProps. Make constants for the values to help display the state in the next step.

```

const { error, loading,
photos } = this.props;
[...]
export const mapStateToProps =
state => ({
error: isGalleryErrored(state),
loading: isGalleryLoading(state),
});
  
```

11. Show loading and error state

While we have the error and loading props, there is currently no UI to indicate when they are active. These props are Boolean values, which means we can toggle the display of components when they are true. Update the render method to make sure the <Error> and <Loading> components render instead of the gallery when needed.

```

if (error) {
return <Error />;
}
  
```



```

}
if (loading) {
return <Loading />;
}
  
```

12. Fetch the gallery again

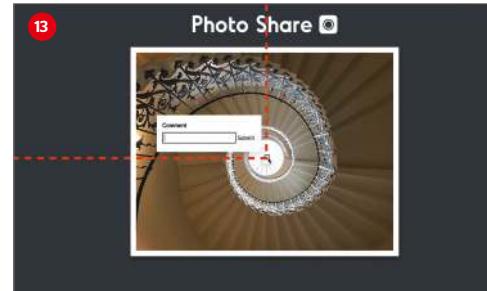
With the gallery loaded, we can move on to an individual photo. Clicking into any of the photos and refreshing the page does not load the photo back up, as there is no instruction on this page yet to load the gallery. Open container/Photo/Photo.js and load the gallery in componentDidMount like in the Gallery component. The photosLoaded check will not try to load the photos again if they were already loaded within the gallery.

```

if (!this.props.photosLoaded) {
this.props.loadGallery();
}
  
```

13. Add a new comment

The user can click on the photo where they want to leave a comment. The Photo presentational component will run the 'addNewComment' prop function when this happens. Inside the addNewComment function, calculate the point where the user has clicked within the photo. The server



requires a round integer percentage value when it gets saved.

```

const photo = e.target
.getBoundingClientRect();
const top = e.clientX - photo.left;
const left = e.clientY - photo.top;
const topPc = Math.round((top /
photo.width) * 100);
const leftPc = Math.round((left /
photo.height) * 100);
  
```

14. Tell Redux about the comment

With the position calculated, we then need to tell Redux about the comment so it can display the comment form. There is already an action set up to add the new comment on screen. Add 'addNewComment' into mapDispatchToProps and call it after we calculated the position of the click.

```

this.props.addNewComment(
topPc, leftPc);
[...]
export const mapDispatchToProps =
dispatch => ({
addNewComment: (top, left) =>
dispatch(addNewComment(top, left)),
});
  
```

15. Tell Photo about new comment

When new comment information is passed to Redux, we need to pass it into the Photo presentational component. This enables it to show the form at that position. Find the 'getNewComment' selector, add it to mapStateToProps and pass the prop into <Photo>.

```

export const mapStateToProps =
(state, props) => ({
  
```

Tutorials

Manage actions with Redux Thunk

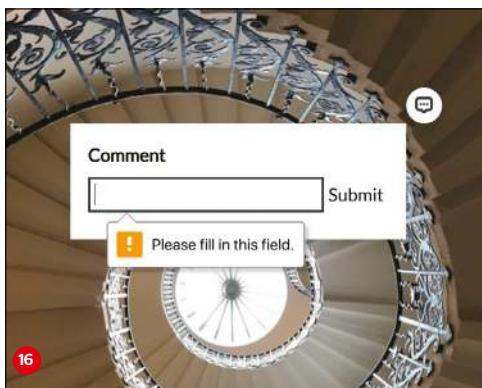
The screenshot shows the official Redux-Saga documentation site. On the left, there's a sidebar with navigation links for 'Read Me', '1. Introduction', '1.1 Beginner Tutorial', '1.2 Saga Background', '2. Basic Concepts', '2.1 Using Saga Helpers', '2.2 Declarative Effects', '2.3 Dispatching actions', '2.4 Error handling', '2.5 A common abstraction: Effect', '3. Advanced Concepts', '3.1 Pulling future actions', '3.2 Non-blocking calls', '3.3 Running tasks in parallel', '3.4 Starting a race between multiple...', and '3.5 Sequencing Sagas using yield'. The main content area features the 'Redux-Saga' logo and the title 'redux-saga'. Below the title is a brief description: 'redux-saga is a library that aims to make application side effects (i.e. asynchronous things like data fetching and impure things like accessing the browser cache) easier to manage, more efficient to execute, simple to test, and better at handling failures.' At the bottom of the main content area, there are several small status indicators: 'new v0.10.2', 'edit v0.10.2', 'download 700kms', 'edit settings', 'chat', 'iterations', 'blocks', and 'issues'.

Other asynchronous approaches

While Redux Thunk is a common approach to handling asynchronous actions, there are different options to consider that may suit other applications better.

Redux-Saga is a middleware that can help respond to different actions. It works through generator functions, introduced as part of ES2015, which can be paused while waiting for data to arrive. It can also watch for actions within the store and trigger behaviour based on them, such as updating the server when the user adds an item to a cart.

Another popular alternative is 'redux-observable', which can take actions and create other actions in response. For example, we can dispatch a 'FETCH_PRODUCTS' action and redux-observable knows to fetch products from the server and dispatch its own success/failure action instead.



```
newComment: getNewComment(state),  
});  
<Photo [...] newComment={  
  this.props.newComment} />
```

16. Call thunk in comment

Clicking on the photo now will bring up the new comment form. This is its own connected component. When the form is submitted, it calls a 'submitComment' prop function and gets passed. This is a thunk that we will make. Open up container/NewComment/NewComment.js and add the thunk to mapDispatchToProps. Pass that prop into the rendered presentational component.

```
<NewComment [...]  
  submitComment={submitComment} />  
export const mapDispatchToProps =  
  dispatch => ({  
    submitComment: comment => dispatch(  
      submitComment(comment))  
  });
```

17. Gather content for thunk

The thunk to add a new comment has a similar structure to the fetching of the gallery, including a start, success and error action. There is an extra argument passed into this thunk - the 'getState' function. This enables direct access to the current state in order to grab data from it. Create the 'submitComment' thunk in actions/newComment/newComment.js. Each comment is associated with a photo and a user. For this tutorial, the user ID is hard-coded into the 'user' reducer.

```
export const submitComment = comment  
  => (dispatch, getState) => {  
    dispatch(submitCommentStart());  
    const currentPhotoId =  
      getCurrentPhotoId(getState());  
    const user =  
      getCurrentUser(getState());  
    const { left, top } =  
      getNewComment(getState());  
  };
```

18. Post the request

With all the necessary data in place, we can submit the comment. Axios has a 'post' method to deal with POST requests, with the second argument being the data to send in that request. Add the request to the thunk, passing in data in snake case to match what the server expects.

```
return axios  
  .post(  
    "http://localhost:3001/comments", {  
      user_id: user.id,  
      photo_id: currentPhotoId,  
      comment,  
      left,  
      top  
    })
```

19. Handle success and error

If the promise from axios resolves or rejects, we need to tell the application about it. If it resolves successfully, the server will pass back the content of the comment. We should pass that in with the success action. If it gets rejected, fire an error action. Update the promise with 'then' and 'catch' blocks.

```
.then(({ data: {  
  id, comment, left, top } }) =>  
  dispatch(  
    submitCommentSuccess(  
      id, comment, left, top,  
      user, currentPhotoId)  
  )  
  .catch(() => dispatch(  
    submitCommentError()));
```



20. Add comment to photo

Right now, once the comment is added successfully it gets cleared from the screen but is not visible until the page refreshes. We can update the photos reducer to pick up on the new comment and add it to its comments array, to display like the rest of them. Open up reducer/photos/photos.js and add a case to handle the action. Create a copy of the state to make sure we don't accidentally mutate the existing state.

```
case SUBMIT_COMMENT_SUCCESS:  
  const { id, comment, top, left,  
    user, photoId } = action.payload;  
  const newState = JSON.parse(  
    JSON.stringify(state));  
  const photo = newState.find(  
    photo => photo.id === photoId);  
  photo.comments.push({  
    id, comment, left, top, user  
  });  
  return newState;
```

21. Hide other comments

Lastly, if another comment is open and the user wants to add a new comment, the UI gets too cluttered. We should hide the comment box if a new comment is being composed. We can hook into the existing 'ADD_NEW_COMMENT' action to clear the 'commentOpen' value. Head to reducer/ui/ui.js and add a case for that.

```
case ADD_NEW_COMMENT:  
  return {  
    ...state,  
    commentOpen: undefined  
  };
```

UP YOUR REAL-TIME REALISM

Discover how game engines can help your projects



FREE
3GB
OF RESOURCES
& VIDEO

SAVE UP TO
44%

When you sign up to
a print subscription. Visit
bit.ly/3DWsubs

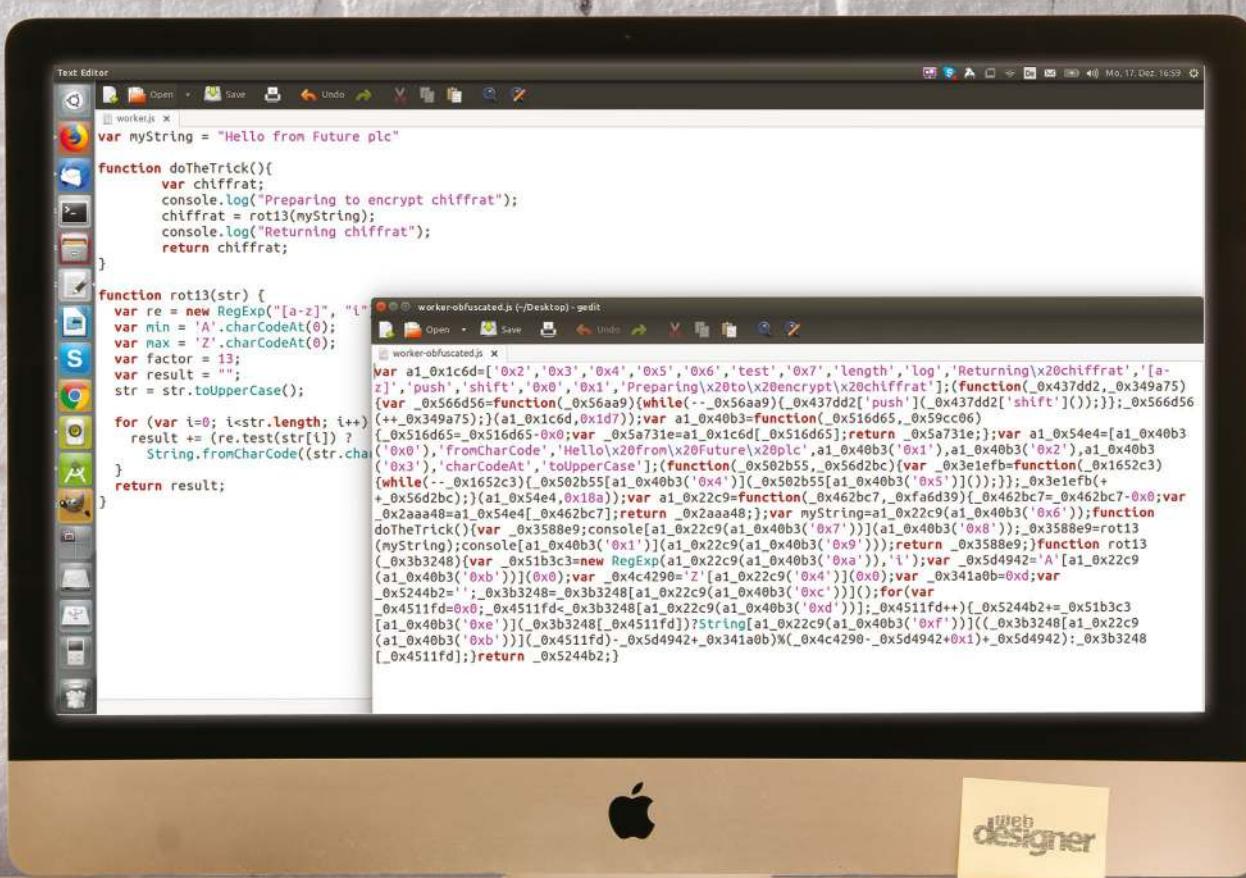
NEWSSTAND FOR IOS www.bit.ly/3dworld-app
PRINT www.myfavouritemagazines.co.uk/3dwsubs

Issue 243 on sale now



Protect your JavaScript code

One of JavaScript's defining features goes by the name 'View Source Code'. Introducing obfuscation helps keep code from prying eyes





hen Java hit the market, decompilers caused a moral panic. Alarmist newspaper reports claimed coding to be at the neigh of the apocalypse, with IP theft posed to become popular sport number one.

While the world obviously did not end, protecting algorithms and cryptographic secrets is a sensible approach. If your programming processes are more than one click away, attackers will prefer to clone a competitor's software. Bots and other miscreants combing the internet for AWS or Azure credentials usually do not bother with obfuscated code - the next website's login information can also be used to mine bitcoin and can be harvested with less effort.

Obfuscating JavaScript used to be complex. However, the process has become significantly easier during the last few years. Today, even small companies can and should protect their code from prying eyes. In some cases, getting secure is just an invocation of a Node.js package away.

The following steps introduce a commonly used JavaScript obfuscator, and also look at a few other topics related to the problem at hand.

1. Version check

Our JavaScript obfuscator lives in the Node runtime environment. Let us start out by checking the versions used. The output shown next to this step provides the version state found on yours truly's workstation used for the following steps:

```
tamhan@tamhan-thinkpad:~$ node --version
v8.12.0
tamhan@tamhan-thinkpad:~$ npm --version
6.4.1
```

2. Install the program

Javascript-obfuscator should be installed into the global assembly cache of your workstation. Invoke npm with the -g parameter and don't forget to provide superuser rights - the actual deployment process should be done in a few seconds.

```
> javascript-obfuscator@0.18.1 postinstall /usr/lib/node_modules/javascript-obfuscator
> opencollective postinstall
```

Thanks for installing javascript-obfuscator
Please consider donating to our open collective
to help us maintain this package.

Number of contributors: 13
Number of backers: 10
Annual budget: US\$ 170
Current balance: US\$ 0

Donate: <https://opencollective.com/javascript-obfuscator/donate>

```
tamhan@tamhan-thinkpad:~$ sudo npm install -g
javascript-obfuscator
[sudo] password for tamhan:
.

+ javascript-obfuscator@0.18.1
added 103 packages from 162 contributors
in 4.4s
```

```
return chiffra;
```

5. Implement the encryption

As this is not intended as an encryption 101, we should settle on a comparatively simple substitution cipher. ROT13 is not difficult, but it can be programmed quite verbosely. Dsoares implementation, found at bit.ly/2S19R8o comes with a set of 'speaking' variables and provides lots of food for our obfuscator.

```
function rot13(str) {
  var re = new RegExp("[a-z]", "i");
  var min = 'A'.charCodeAt(0);
  var max = 'Z'.charCodeAt(0);
  var factor = 13;
  var result = "";
  str = str.toUpperCase();

  for (var i=0; i<str.length; i++) {
    result += (re.test(str[i]) ?
      String.fromCharCode((str.charCodeAt(i) - min + factor) % (max-min+1)
        + min) : str[i]);
  }
  return result;
}
```

6. First obfuscation

Performing an obfuscation run of the code is simple. Invoke javascript-obfuscator, and pass in a dot to tell the program to work on all files found in the current working directory. The figure accompanying this step shows the result on the author's IBM workstation:

Eliminates CDNs

Obfuscated code changes its shape with each invocation. This renders content delivery networks inefficient - if you obfuscate your version of a library like jQuery, the browser must re-download it every time.

Tutorials

Protect your JavaScript code

```
tamhan@tamhan-thinkpad:~/Desktop/DeadStuff/2018Nov/FutureObfuscate/code$ tree
.
└── index.htm
    └── worker.js

0 directories, 2 files
tamhan@tamhan-thinkpad:~/Desktop/DeadStuff/2018Nov/FutureObfuscate/code$ javascript-obfuscator .

[javascript-obfuscator-cli] Obfuscating file: worker.js...
tamhan@tamhan-thinkpad:~/Desktop/DeadStuff/2018Nov/FutureObfuscate/code$ tree
.
└── index.htm
    └── worker.js
        └── worker-obfuscated.js

0 directories, 3 files
```

6

```
tamhan@tamhan-thinkpad:~/Desktop/DeadStuff/2018Nov/FutureObfuscate/code$ tree
.
└── index.htm
    └── obfusca
        └── worker.js
    └── worker.js

1 directory, 3 files
```

7

```
tamhan@tamhan-thinkpad:~/FutureObfuscate/
code$ javascript-obfuscator .
```

7. Redirect output

Firing off the files directly into the container folder is inefficient, as names must be changed before they can be used. A better way involves the use of the output parameter. If javascript-obfuscator finds it, the

It's not minification

Minification also changes the shape of JavaScript. Its goals, however, are completely different. While an obfuscator wants to make code as difficult to read as possible, a minifier tries to optimise it for faster delivery and better parsing.

program generates a subfolder in the current working directory and dumps the results of its labours there.

```
tamhan@tamhan-thinkpad:~/FutureObfuscate/code$ javascript-obfuscator . --output ./obfusca
```

8. Analyse the results

Dive into the 'obfusca' folder and open the new version of worker.js to feast your eyes on the abomination that is shown accompanying this step. The code's formatting was mangled badly. Method names, however, remained the same, as they are needed for external invocations. Furthermore, strings now sit in an array where an attacker can harvest them conveniently.

```
worker.js x
var a0_0xb9e2=['Hello\x20from\x20Future\x20plc','[a-
z]','test','charCodeAt'];(function(_0x58e063,_0x1b231a){var
_0x43c4c5=function(_0x39e842){while(--_0x39e842){_0x58e063['push']
(_0x58e063['shift'])();}};_0x43c4c5(++_0x1b231a);}
(a0_0xb9e2,_0x11c));var a0_0x395d=function(_0xa31e6d,_0x26d65d)
{_0xa31e6d=_0xa31e6d-_0x0;var _0x662f8d=a0_0xb9e2[_0xa31e6d];return
_0x662f8d;};var myString=a0_0x395d('0x0');function doTheTrick(){var
_0x28d05c;_0x28d05c=rot13(myString);return _0x28d05c;}function rot13
(_0x55414e){var _0x332d03=new RegExp(a0_0x395d('0x1'), 'i');var
_0x142336='A'['charCodeAt'](0x0);var _0x2a92da='Z'['charCodeAt']
(0x0);var _0x4c1303=_0xd;var _0x517c9a='';_0x55414e=_0x55414e
['toUpperCase']();for(var _0x4057bf=_0x0;_0x4057bf<_0x55414e
['length'];_0x4057bf++){_0x517c9a+=_0x332d03[a0_0x395d('0x2')]
(_0x55414e[_0x4057bf])?String['fromCharCode'](( _0x55414e[a0_0x395d
('0x3')]( _0x4057bf)-_0x142336+_0x4c1303)%(_0x2a92da-_0x142336+0x1)
+_0x142336):_0x55414e[_0x4057bf];}return _0x517c9a;}
```

8

```
var a0_0xb9e2=['Hello\x20from\x20Future
\x20plc','[a-z]','test','charCodeAt'];
```

9. Prevent string harvesting

Javascript-obfuscator comes with a selection of string-mangling algorithms, which can be configured using --string-array-encoding. Keep in mind that the output directory must be emptied out before each invocation, because forgetting to do so leads to 'recursive' obfuscation of the output files from the previous run.

```
tamhan@tamhan-thinkpad:~/FutureObfuscate/code$ javascript-obfuscator . --output ./obfusca
--string-array-encoding base64
```

10. Look at the results again

At this point, our obfuscator's output looks different - the array at the top of the file now is much less readable. This, however, does not solve all problems. If you perform a few obfuscation cycles, you will eventually end up with mark-up similar to the one accompanying this step:

```
var a0_0x31e5=['bGVuZ3Ro','dGVzdA==',
ZnJvbUNoYXJDb2R1,'Y2hhckNvZGVBdA==',
dG9VchBlckNhc2U='];
...
...
var myString='Hello\x20from\x20Future\x20plc'
```

11. Understanding randomisation

Obfuscators work in a pressure field between high performance and code protection. One way to



Aggressive defensive measures

While obfuscation does make cloning code difficult, the algorithm still lives on the client. Should this be completely unacceptable, the architecture shown in the figure can help out. Sensitive mathematical processes live solely on your server, and the client gets relegated to the role of a dumb terminal displaying the computed results.

In theory, users should rebel against such systems due to the requirement of a permanent internet connection. Interestingly, this is not the case in practice - this writer's company marketed a scientific calculator for Firefox OS based on this concept and did not receive a significant amount of complaints about the practice.

address the problem involves 'randomising' elements. The runtime-expensive obfuscations are not added to all nodes, but only to a subset. Detecting if nodes are affected or not is usually done via a random number generator, whose sensitivity can be tuned.

12. Go after strings

The above-mentioned random number generator emits numbers ranging from zero to one. If the number is larger than the threshold, the modification will not take place. Setting `stringArrayThreshold` to a value of one means that all numbers are smaller than the threshold, ensuring that each and every string gets mangled.

```
tamhan@tamhan-thinkpad:~/Future0bfuscate/code$ 
javascript-obfuscator . --output ./obfusca
--string-array-encoding base64
--stringArrayThreshold 1
```

13. Inject random code

Analysis tools such as JSNice profit from having a small code base. Given that `javascript-obfuscator` breaks the code into an AST anyway, nothing speaks against inserting 'garbage code' on the fly. As this feature adds significant bloat, developers must activate it manually via the two parameters shown next to this:

```
--dead-code-injection <boolean>
--dead-code-injection-threshold <number>
```

14. Enjoy the chaos

Running the obfuscator with both `-dead-code-injection true` and `-dead-code-injection-threshold 1` set leads to a filesize of about 2.5KB. Attempting to nicely the code leads to an almost indecipherable wall of mostly tautological JavaScript.

15. Aggressively modify program flow

Breaking code into an AST allows for deep transformations. Setting the `controlFlowFlattening` attribute to true tells the program that it can de-inline function calls. This leads to a significant expansion of the generated code - keep in mind that the results can take a 150 per cent performance hit.

16. Annoy the debugger

Artefacts from debugging are the world's greatest gift to hackers. A few calls to `console.log()` and its friends can give an attacker valuable information on what happens inside the program - a good example would be the snippet shown:

```
function doTheTrick()
{
    var chiffrat;
    console.log("Preparing to encrypt
    chiffrat");
    chiffrat = rot13(myString);
    console.log("Returning chiffrat");
    return chiffrat;
}
```

17. Evaluate technical problems

We can attempt to re-obfuscate the program with the command below. It disables the string caching feature and should disable console logging via redirection.

```
tamhan@tamhan-thinkpad:~/Desktop/
DeadStuff/2018Nov/Future0bfuscate/code$ 
javascript-obfuscator . --output ./obfusca
--string-array-encoding false
--disableConsoleOutput true
```

18. Play cat and mouse games

Obfuscator and browser vendors fight a long and bitter battle about the debugger function. Due to this, 'aggressive' measures, such as console redirection of the program flow interruptors shown accompanying this step, usually don't work for long.

```
debugProtection: false,
debugProtectionInterval: false,
```

19. Quick online obfuscation

Installing the entire Node.js package for obfuscating one or two files is pointless. Visit obfuscator.io to access an online version of the program that lives in your browser. Check and comboboxes below the main input let you modify program behaviour as outlined in the steps above.

20. Learn more

The developer team maintains relatively detailed documentation explaining the way the various

Tutorials

Protect your JavaScript code

The screenshot shows the JScrambler web interface. On the left, the 'FILE TREE' panel lists files like 'common.js' and 'index.html'. The 'SOURCE CODE' panel contains the original JavaScript code for 'common.js'. The 'PROTECTED CODE' panel shows the obfuscated version of the same code, where variable names and function signatures are heavily altered. The 'APPLICATION SETTINGS' sidebar on the right includes options for 'Language Specification' (ES5, ES6, ES7), 'Platform' (Browser App, Desktop App, Server App, Hybrid Mobile App, Native App, HTML5 Game), and 'Archive'.

command line parameters interact with one another. Simply visit github.com/javascript-obfuscator/javascript-obfuscator if the 'short help' output shown in the figure next to this step does not help you reach your goal.

21. An obfuscator's mortal enemy

The ETH Zurich provides a de-obfuscation service hosted at jsnice.org. It uses neural networks and a knowledge database made up of existing code to determine variable names. While sprucing up the formatting usually works pretty well, some of the names - such as pixelSizeTargetMax in the encryption routine accompanying this step - can be rather amusing.

20

```
tamhan@tamhan-thinkpad:~  
tamhan@tamhan-thinkpad:~$ javascript-obfuscator  
Usage: javascript-obfuscator <inputPath> [options]  
Options:  
-V, --version  
-O, --output <path>  
--compact <boolean>  
--config <boolean>  
--control-flow-flattening <boolean>  
--control-flow-flattening-threshold <number>  
--dead-code-injection <boolean>  
--dead-code-injection-threshold <number>  
--debug-protector <boolean>  
--debug-protector-interval <boolean>  
--disable-console-output <boolean>  
--domain-lock <list> (comma separated, without whitespaces)  
--enable-strict-mode <boolean>  
--enable-strict-mode-for-variables <boolean>  
  
function rot13(PL$120) {  
    /** @type {!RegExp} */  
    var insufficientRegExp = new RegExp  
    (a0_0x395d("0x1"), "i"); var  
    pixelSizeTargetMax = "A" ["charCodeAt"]()
```

JS NICE STATISTICAL RENAMING, TYPE INFERENCE AND DEOBFUSCATION

ABOUT 21

ENTER JAVASCRIPT

NICIFY JAVASCRIPT

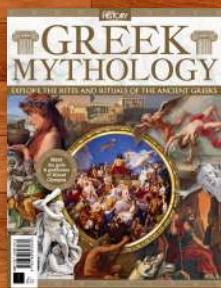
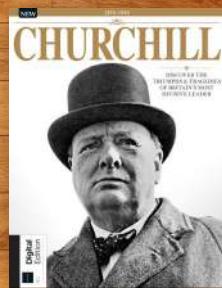
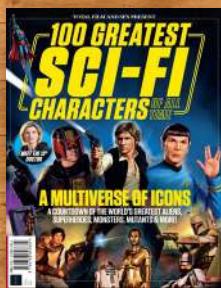
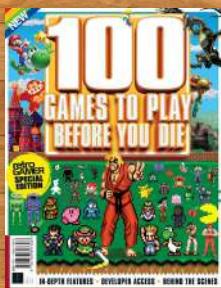
RESULT. ARE YOU SATISFIED?

YES NO

```
1 'use strict';
2 /**
3  * @type {Array}
4  */
5 var a0_0xb9e2 = ["Hello\x20from\x20Future\x20plc", "[a-z]", "test", "charCodeAt"];
6 function(a0_0x58e063, _0x1b231a){var _0x43c4c5=function(_0x39e842)
7 {_while(-- _0x39e842){_0x58e063['push'](_0x58e063['shift']());}};
8 a0_0x395d=function(_0xa31e6d, _0x26d65d){_0xa31e6d=_0xa31e6d-0x0;var
9 _0x62f8d=a0_0xb9e2[_0xa31e6d];return _0x62f8d;};var myString=a0_0x395d('0x0');
10 function doTheTrick(){var _0x28d05c; _0x28d05c=_0x28d05c||rot13(myString);return
11 _0x28d05c;};function rot13(_0x55414e){var _0x332d03=new
12 RegExp(a0_0x395d('0x1'),"i");var _0x142336=_A ["charCodeAt"](_0x0);var
13 _0xa92da=_Z ["charCodeAt"](_0x0);var _0x4c1303=_0xd;var _0x517c9a=_i;
14 _0x55414e=_0x55414e['toUpperCase']();for(var _0x4057bf=_0x0;
15 _0x4057bf< _0x55414e['length'];_0x4057bf++){_0x517c9a+=_0x332d03[a0_0x395d('0x2')]
16 (_0x55414e[_0x4057bf])_2String[_fromCharCode'](_0x55414e[_0x395d('0x3')]);
17 (_0x4057bf)-_0x142336+_0x4c1303)%(_0x2a92da-_0x142336);_0x55414e[_0x4057bf];};return _0x517c9a;};  
  
18 var zeroSizeMax = "Z"["charCodeAt"]()
```

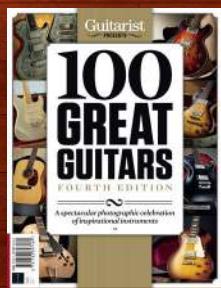
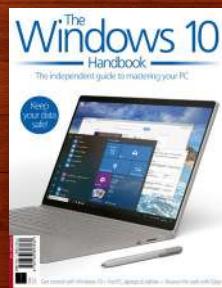
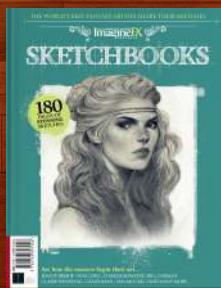
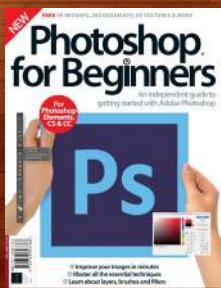
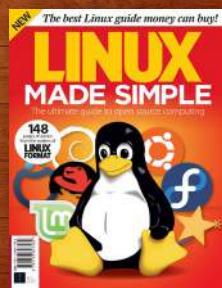
Online obfuscator

If running an obfuscator in Node.js is not your thing, take a look at docs.jscrambler.com. The somewhat pricey product has a subscription on media mentions from various organisations, lives in your browser and contains extremely advanced obfuscations. It describes itself as "a JavaScript interpreter and parsing engine that creates an Abstract Syntax Tree representing your source code". One really neat aspect is the 360-degree coverage - you upload your code to the cloud and harvest the finished results as an archive. This approach lets JScrambler look at both JS and HTML files - the small mark-up found in worker.htm would also be obfuscated.



Discover another of our great bookazines

From science and history to technology and crafts, there are dozens of Future bookazines to suit all tastes



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



www.myfavouritemagazines.co.uk
Magazines, back issues & bookazines.



WHAT'S NEW IN WORDPRESS 5.0?

Content editing with WordPress is getting a complete facelift, see what's changed in the new version of the world's most popular CMS



“

Gutenberg is going to reinvent the way we write and display content on the web, bringing powerful tools to the hands of all WordPress users, from editor to developer

Sarah Maynard, Freelance WordPress Developer
www.sarahmaynard.net

”

WHAT'S NEW IN WORDPRESS 5.0?

WordPress 5.0 is slated to bring the biggest change to the CMS in a long time. With the rollout of Gutenberg, a block-based content editor, WordPress is taking a huge step towards bringing easy, responsive layouts into content editing. It's the first step in a three-stage plan to revolutionise content, templates and eventually WordPress site-building as a whole.

Gutenberg enables users to form their content out of responsive blocks - similar to site-builder tools like Squarespace - to create posts and pages that mould easily with their theme to any screen size. Theme editors can style block types to match their templates, and non code-savvy users will find it easy to add elements like columns, cover images and social media embeds without the need for unwieldy plugins. It's even possible to embed widgets in posts and pages. Gutenberg aims to negate the need for shortcodes and custom fields by standardising the content creation process.

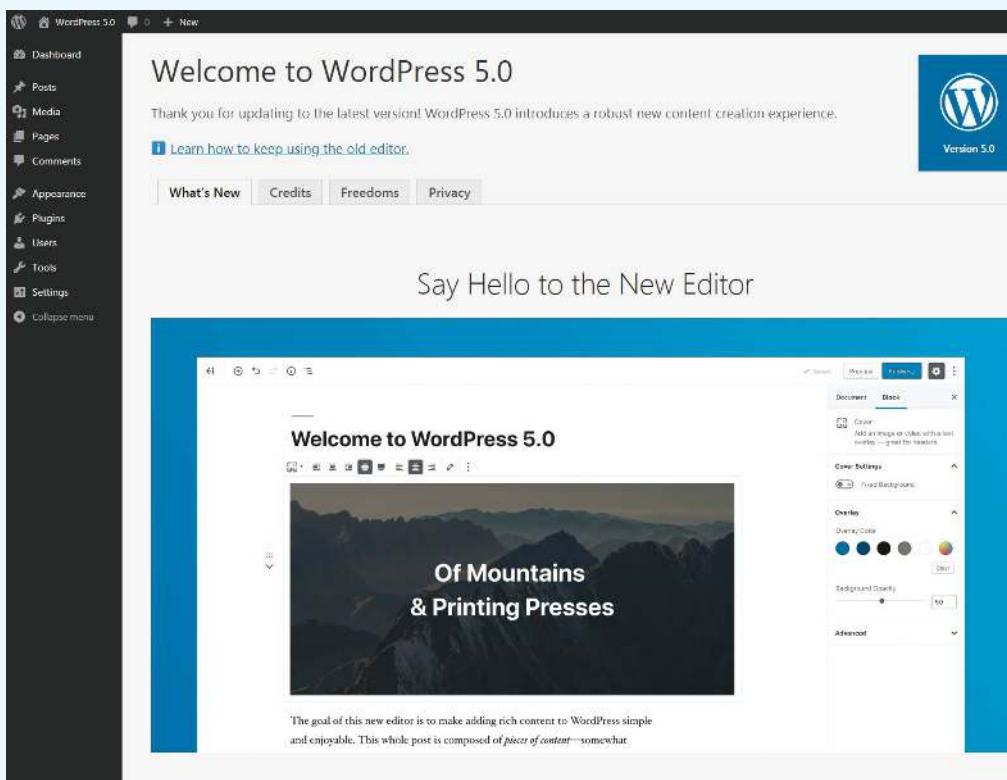
It makes publishing faster and more powerful, giving editors all-new tools to write and publish more efficiently than before. You can even write your content in other editors like Google Docs or Microsoft Word and paste into the Gutenberg editor, and it will translate your content into blocks automatically.

Coming with Gutenberg is the all new Twenty-Nineteen theme and updates to the previous WordPress themes to support the new Gutenberg editor. Suffice to say, the WordPress landscape is going to change massively in the next few years.

5 REASONS TO USE THE GUTENBERG EDITOR

- 1 IT'S EASIER THAN IT LOOKS** New things can be daunting, but Gutenberg is surprisingly easy to pick up. Within a few minutes of using the new content editor, you'll be creating content quicker than ever.
- 2 CREATE SOMETHING NEW** With all-new block types and support for things you never knew you needed, Gutenberg will encourage editors to create content that breaks out of the mould and discover new ways to display content and engage with viewers.
- 3 PROMOTE DEVELOPER AND USER SYNERGY** Because blocks standardise the way content is crafted, developers can get a better handle on what editors might be creating and predict how their themes and styles might be utilised to avoid disastrous user code clashes.
- 4 WYSIWYG IS REALLY WYSIWYG** Going forward, with a little dedication from developers and their styling, Gutenberg will more accurately represent what really appears in the published post. Editors can build their content in real time without having to rely on the preview.
- 5 STEP INTO THE FUTURE** The world of web content is changing fast, and what better way to keep up than to ride on the wave of new technology? Gutenberg is paving the way for exciting things to come.

IS WORDPRESS 5.0 BETTER THAN ITS PREDECESSORS?



With such a drastic change to the beloved content editor, Gutenberg has encountered resistance and negative feedback, so opinion on whether it's an improvement or not is highly contested by certain factions. However, the future of web design is dependent on change, and as Matt Mullenweg said, "It's worth a little discomfort to change the world."

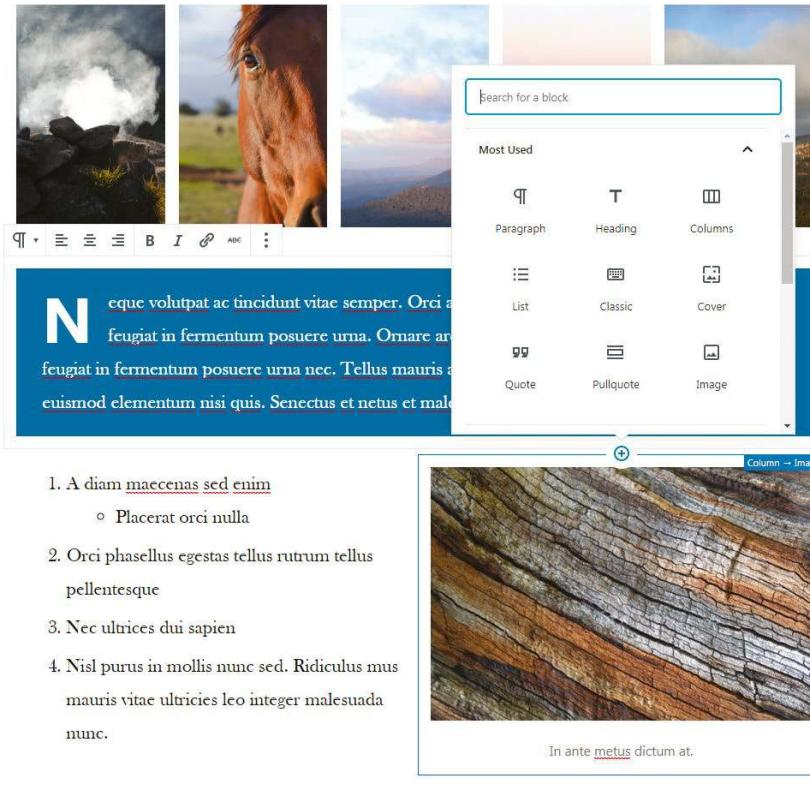
WordPress 5.0 is a huge step towards improving content creation for the layman, as well as enabling developers to maintain content unity. It makes it easy for editors to create content in different layouts without giving designers a headache, and is likely to be a boon for developers who have had trouble in the past adapting user content to fit nicely into their themes. The difficulty arises with the initial release; most if not all themes will need to be updated to utilise the new Gutenberg block styles.

Overall, opinions may be mixed in the early stages, but WordPress 5.0 should pave the way for a more flexible and user-friendly experience all round.

WordPress 5.0 New View Post Hi, sarah

Dashboard Posts All Posts Add New Categories Tags Media Pages Comments Appearance Plugins Users Tools Settings Collapse menu

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!



GUTENBERG'S BUILDING BLOCKS

Blocks are the foundation of the new Gutenberg editor. By splitting content up into different types – such as paragraphs, lists, images, quotes and more – blocks enable editors to insert, drag and drop, remove and swap parts of the post or page with ease. Compared to the previous TinyMCE editor, it may be somewhat alien to users who have never seen a block-based editor or page builder before, but after a little practice the benefits far outweigh the learning curve. So don't lose heart if it seems daunting at first.

The default blocks are split into a few different sections; inline elements, common blocks, formatting, layout elements, widgets, embeds and reusable blocks. There's even a 'Classic' block, if you miss TinyMCE, with the option to convert it later into Gutenberg blocks. Plugins can add blocks too, making integration of new blocks easy and streamlined, and reducing the amount of clutter on the editor where you have multiple plugins weighing in.

Theme and plugin developers will have some work to do to bring their code up to date to make full use of the new update, but it should be worth it

to see users creating content that is not only easy on the eye, UI-friendly and theme-matching, but composed of valid code too. Page-breaking user code will be a thing of the past.

Writing with blocks is a breeze, with new keyboard shortcuts to help you tab between existing blocks, insert, duplicate, move and delete blocks on the fly. Blocks also include a wide variety of default embed blocks, living up to the WordPress.org Gutenberg promise that content editing will support multimedia and content side by side. With such a comprehensive set of blocks right out of the box, Gutenberg is ready to tackle content of all flavours.

Blocks are going to standardise the way content is put together. Plugins, custom post types, meta boxes and shortcodes will be able to utilise the same UI, making the process of putting it all together seamless for every user. It also means code and content is finally completely separate. Updating code will no longer require any update to content, because developers can target the surrounding block behaviour.

Blocks also include a wide variety of default embed blocks, living up to the WordPress.org Gutenberg promise that content editing will support multimedia and content side by side

REUSABLE BLOCKS

Save blocks to use repeatedly across your site

Reusable blocks are a new integral function that enables you to save a block and reuse it in any content. When a reusable block is edited, it updates automatically wherever it has been used, without the need to manually edit every page where it appears. Reusable blocks can be updated inline or from their own administration panel by clicking the 'Options' button in the top right of any content editor and going to 'Manage All Reusable Blocks'. Creating a reusable block can be done from there, or by clicking the 'Options' button on any block menu and clicking 'Add to Reusable Blocks'. You can also save multiple blocks into one reusable block by highlighting content, clicking the 'Options' button that appears in the top left of your selection and selecting 'Add to Reusable Blocks'.

Reusable blocks can be exported from the 'Manage All Reusable Blocks' menu by hovering over a block title and clicking 'Export as JSON' or imported via the 'Import from JSON' button. This makes it possible to share blocks between sites.

The screenshot shows the WordPress 5.0 editor interface. On the left is a sidebar with navigation links: Dashboard, Posts, All Posts, Add New, Categories, Tags, Media, Pages, Comments, Appearance, Plugins, Users, Tools, Settings, and Collapse menu. The main area displays five numbered steps (1-5) illustrating different text blocks:

- 1** **Enim lobortis scelerisque fermentum**: A large heading block.
- 2** At quis risus sed vulputate odio ut enim blandit. Ultrices eros in cursus turpis massa tincidunt dui. Purus sit amet luctus venenatis lectus.
- 3** At quis risus sed vulputate
Egestas erat imperdiet sed euismod
Tempus urna et pharetra pharetra
- 4** Pellentesque dignissim enim sit amet venenatis
urna cursus eget. Lacus luctus accumsan tortor
posuere ac ut consequat. Imperdiet nulla
malesuada pellentesque elit eget.

John Smith
- 5** Blandit volutpat maecenas

Faucibus purus in massa tempor nec feugiat nisl
premium fusce. Purus in massa tempor nec feugiat
nisl premium. Sit amet mauris commodo quis
imperdiet massa tincidunt nunc.

• Fames ac turpis egestas sed
• Tortor consequat id porta nibh
• Facilisis gravida neque convallis a

A toolbar above the blocks includes icons for Paragraph, Bold, Italic, Alignment, and other editing tools. To the right, a panel shows "Document" and "Block" options, with "Classic" selected. The status bar at the bottom right shows "Switch to Draft", "Preview", and "Update".

There should be a block for every occasion, even with just the default block types in 5.0, with the addition of the custom HTML block and the original TinyMCE Editor-like 'Classic' block

BLOCKS FOR TEXT FORMATTING

Format your text with a different block for each type

1 The heading block formats headings from H1 through to H6. Inserting a heading block into the content will by default select H2, which is ideal for steering content editors into maintaining the content's waterfall heading structure and avoiding multiple H1s on a page. Twenty Nineteen's default settings enables editors to change heading alignment (left, centre or right-aligned) and add a HTML anchor for linking directly to a section on the page.

2 The paragraph block will make up most of your content, it is the 'building block of all narrative'. Each paragraph has its own block, and Twenty Nineteen comes with the default text settings for text size, drop caps, background and text colour. It can be transformed easily into a quote, heading, list or verse. As you're writing, pressing enter will create a new block, ready for the next paragraph. Text formatting is limited to inline styles like bold, italic, links and strikethrough.

3 Lists are separate blocks to paragraphs, which can cause some confusion at first because you could mistakenly look under 'paragraph' instead. List blocks by default have

simple options, such as ordered (numbered) and unordered (bullet-point) lists, as well as a tool to nest lists underneath other list items. Basic text formatting, like the paragraph block, includes bold, italic and links.

4 Quote blocks format blockquotes with the option to add an easy citation. This used to be difficult for the non tech-savvy in the classic TinyMCE editor, as adding a citation would require additional coding in HTML view. Default options include quote text alignment, as well as regular or large quote styles. There is a separate Pullquote block option for featuring quotes in left or right-floating blocks, or full width spreads. Swapping between the two is easy and can be done with a single click.

5 Other default formatting options include 'Classic', code, preformatted, table and custom HTML blocks. There should be a block for every occasion, even with just the default block types in 5.0, while the addition of the custom HTML block along with the original TinyMCE Editor-like 'Classic' block should cover most of the edge-cases.

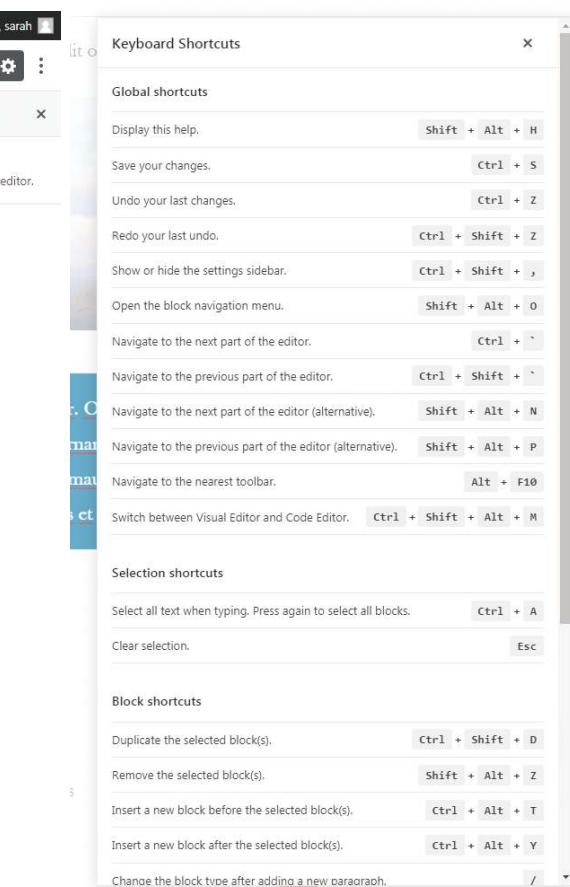
BLOCKS IN COLUMNS

Use the column block type to sit content side by side in responsive columns

Perhaps one of the most powerful new blocks for content layout is the column block type, enabling users to create up to six columns and insert paragraphs, lists, images and more in a responsive format. With the option to make the column block full width, the page's space is well-utilised and content can flow as the user likes.

There's a bit of a knack to find the column settings again once you've created and edited the block, but using Tab and Shift+Tab will help if it's proving difficult to bring up the options menu with the mouse hover. Beware, reducing the number of columns after filling them up will delete the content in the removed columns, so make sure to move your blocks out of the column structure before editing the column block.

As long as the theme supports it, the columns will stack on smaller screen sizes, maintaining responsive design.



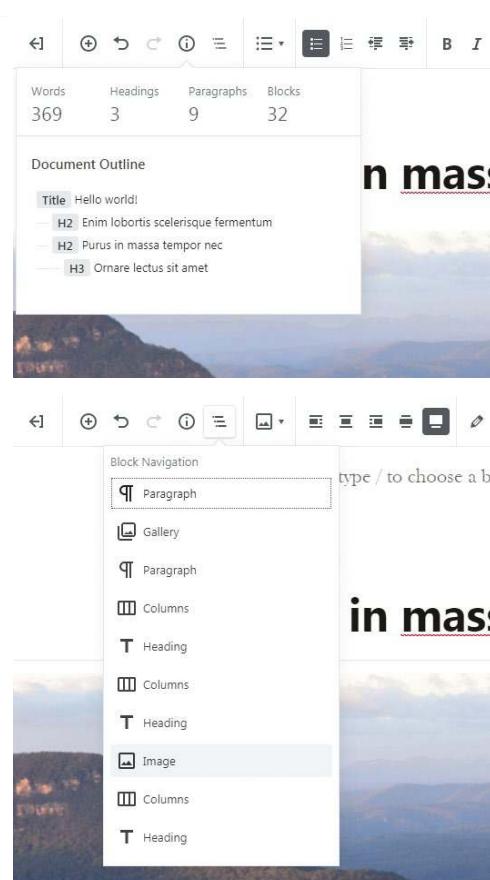
KEYBOARD SHORTCUTS

Keyboard savants will find the new Gutenberg editor to be a haven of time-saving shortcuts. An entire post can potentially be written without the use of the mouse. The WordPress team have been working hard to make the editor accessible for everybody. Inserting blocks, swapping between blocks and editing block settings can all be done with keyboard shortcuts. Use Tab and Shift+Tab to skip up and down the page options, and use enter to select. To open the block type chooser, start a line with '/' or select any of the '+' icons with Tab. You can swap between page sections with Ctrl+ or Cmd+ and see all the OS-native keyboard shortcuts with Shift+Alt+H.

CONTENT STRUCTURE AND BLOCK NAVIGATION

Gutenberg comes with some handy tools for content information and navigation. In the top left, the 'Content Structure' icon shows you the wordcount of your content, and how many headings, paragraphs and blocks it's split up into.

The block navigation (Shift+Alt+O) gives a table of contents for easily skipping to blocks without having to scroll up and down looking. This is particularly helpful for keyboard-only users, who won't need to tab through every block to get to the end of their content. The only downside is you can't currently name blocks, so there's some guesswork involved when you have a lot of blocks of the same type.



EDITOR VIEWS

The new editor has a few different views to cater to every editor's preferred writing environment. By opening the options in the top right, it enables you to choose between the visual editor and the code editor (Ctrl+Shift+Alt+M). The code view will display the block syntax denoted by HTML comment tags (i.e. <!-- wp:example -->). Be careful when editing the syntax - broken tags will trigger your blocks to be changed into the 'classic' block.

Each editor can choose one or more view options: 'Top Toolbar' (move all the block options to the sticky top menu), 'Spotlight Mode' (grey out the blocks you're not working on) and 'Fullscreen Mode' (hide the rest of the site options).

CONVERTING OLD CONTENT

When you upgrade to WordPress 5.0, your old content will not be lost or broken. When you go to edit the page, it will be displayed as the full article without blocks in the 'Classic' block type. You have the option to use the 'Convert to Blocks' tool to automatically update your old content into the new blocks format, or manually create blocks if the tool doesn't act as expected. There are even 'Shortcode' blocks to handle your old shortcodes if that plugin does not yet support blocks. Or you can leave your content as it is; the 'Classic' block editor gives you the option to create and manage content with the familiar TinyMCE format.

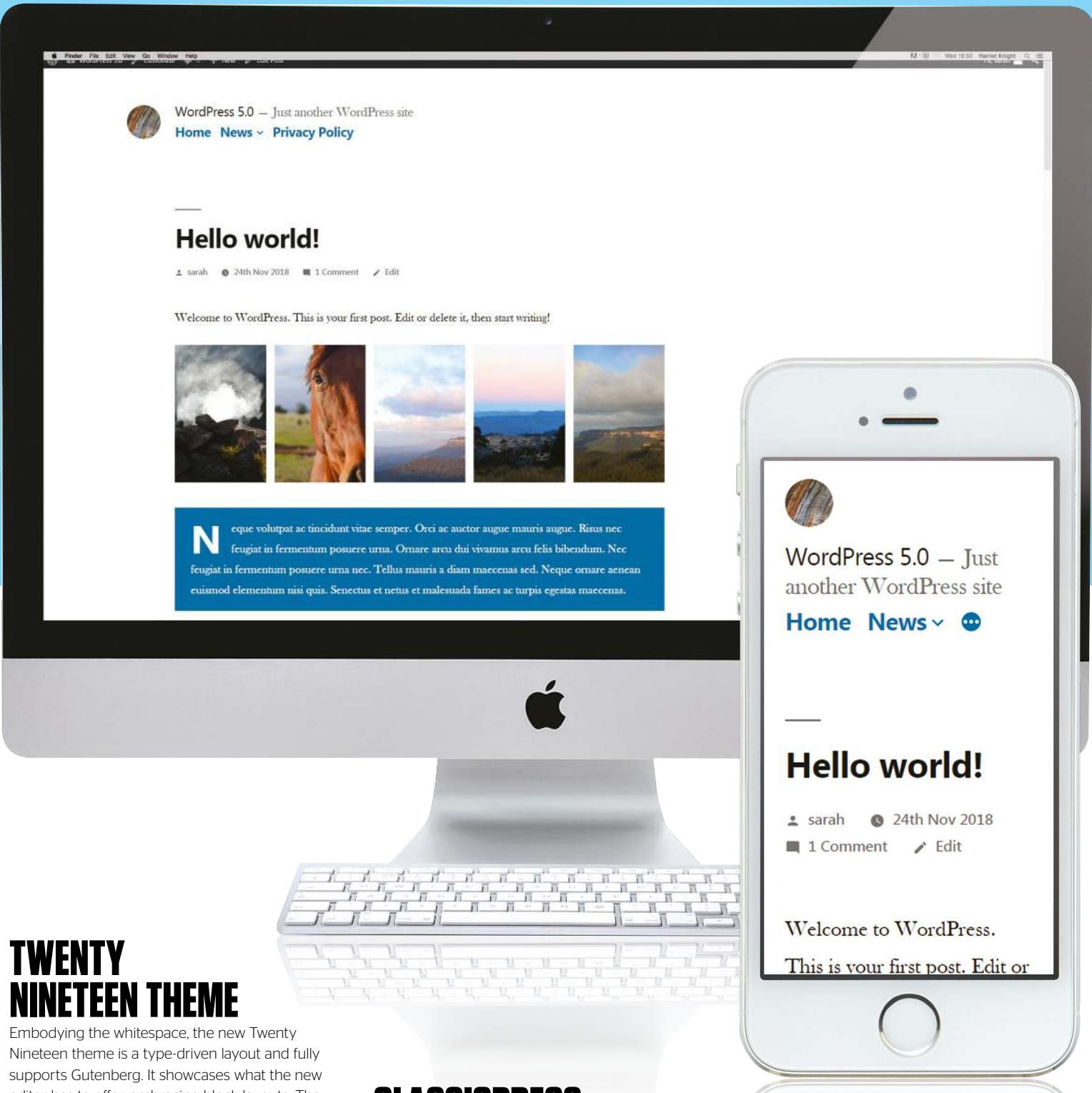
WHAT HAPPENS TO TINYMCE?

Blocks won't replace TinyMCE, it's still there

TinyMCE - the current tool for editing rich text in WordPress - is not being replaced. Instead blocks will use TinyMCE for editing their rich text with boundaries, such as limiting formatting to bold, italics and links. Most of the TinyMCE options will be extracted into their own unique block types. Formatting like paragraphs, lists, media, blockquotes, code and headings etc. will be split up into block types and inserted into content as self-contained sections. The original editing experience will still exist, however. The 'Classic' block will be very familiar to WordPress users, acting exactly like the current WordPress editor and adding a fallback for users who are not sure where to find the format type they are looking for within the myriad of new blocks. If Gutenberg or blocks really aren't for you, the Classic Editor is available as a plugin to completely replace the new editor or to be there as an optional alternative, and will be supported by WordPress until 2022.



If Gutenberg or blocks really aren't for you, the Classic Editor is available as a plugin to completely replace the new editor



TWENTY NINETEEN THEME

Embodying the whitespace, the new Twenty Nineteen theme is a type-driven layout and fully supports Gutenberg. It showcases what the new editor has to offer, embracing block layouts. The styles between editor and page are almost exact, so you can see what your page will look like as you edit. It's designed to work for a variety of cases, from bloggers to small businesses, and has a minimalist menu and fullscreen featured image area (like the Twenty Seventeen theme). It has a two-column widget area in the footer and no sidebar.

The theme comes with the option to change the primary colour for buttons and links etc. and the option to filter the featured images using the chosen primary colour. The inline menu is bold and minimalist, with the logo, title, tagline, menu and social menu sat together at the top. The theme adapts well to tablet and mobile sizes, although content in multiple columns looks a little cramped on medium-sized screens before it breaks down and stacks on smaller screens.

CLASSICPRESS

ClassicPress is a pre-Gutenberg spin-off of WordPress 4.8.9 aimed at business websites that want to maintain the original editing experience. The founders of ClassicPress fear that Gutenberg will disrupt the power and popularity of WordPress and launched the push-back initiative after the announcement that Gutenberg would become part of WordPress core. It's a community-led project and so no new features are added without the vote of the community. The casual admin bar greeting 'Howdy' and other colloquialisms have been stripped in favour of formal language better suited to more serious websites. This open-source fork is gaining traction and could actually be a valid resource for WordPress users looking for a more 'business-style' version of WordPress.

REST API UPDATES

The REST API caused a hubbub at its conception but until recently has not been used extensively in WordPress core. Now, with Gutenberg relying on the API to fetch and save content, the REST API is starting to be used more heavily, as it was intended. Many updates have been made to the REST API for 5.0, and importantly the same hooks have been used for saving and retrieving content with the API as were used in the original editor, to maintain integrity across plugins that work with the editor. The Gutenberg editor's seamless editing experience is powered by the REST API, saving and publishing without having to reload the page.

JANUARY SALE

The perfect gift from just \$23*

FREE 340 CREATIVE DESIGN RESOURCES

WEBSITE designer Expert tutorials, techniques and inspiration

JAVASCRIPT 2019

25 FRAMEWORKS, TOOLS & APIs YOU NEED FOR THE NEW YEAR

NEW MEDIA QUERY Implement 'prefers-color-scheme' with CSS custom properties

WORDPRESS 5 GET A CLOSER LOOK AT GUTENBERG

CSS LIQUID EFFECTS Apply SVG filters to add fluidity

REACT AND REDUX How to manage asynchronous actions with Thunk

BUILD A WEBVR SCENE Use the power of three.js to add another dimension

ISSUE 282 U.S. \$14.99 CAN. \$18.99 FUTURE 02>
0 70992 51891 9

FROM \$23 EVERY 3 MONTHS (\$93 PER 13 ISSUES YEARLY)

Delivery included in the price

Buy as a gift or treat yourself!

Free personalised e-card when buying for someone else

www.myfavouritemagazines.co.uk/pin/xmas187

ORDER HOTLINE: +44(0) 344 848 2852

PLEASE QUOTE XMAS187 WHEN ORDERING BY PHONE

LINES ARE OPEN MONDAY - FRIDAY 8AM TO 7PM AND SATURDAY 10AM TO 2PM (GMT)

***Terms and conditions:** Savings calculated against the full RRP (single issue price x frequency). Dollar prices quoted are for the United States, other global territory dollar pricing may vary. This offer is for new subscribers only. You can write to us or call us to cancel your subscription within 14 days of purchase. Your subscription is for the minimum specified and will expire at the end of the current term. Payment is non-refundable after the 14 day cancellation period unless exceptional circumstances apply. Your statutory rights are not affected. Prices correct at point of print and subject to change. Full details of the Direct Debit guarantee are available on request. For full term and conditions please visit: bit.ly/magtandc. Offer ends 28th February 2019

Get your listing in our directory
To advertise here call the ads team on 01225 442244

HOSTING LISTINGS



Featured host: Netcetera

netcetera.co.uk
03330 439780

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 per cent uptime, multiple internet routes with the ability to handle DDOS attacks, ensuring your site doesn't go down when you need it.

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.

2. Secure and 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.

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.

3. 24/7 technical support

Netcetera has a committed team of

5. Eco-friendly
Netcetera's environmental commitment is backed by use of eco-cooling 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"

Suzi 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 per cent"

Supreme hosting



cwcs.co.uk
08001777000

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 per cent network uptime

UK-based hosting



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



fasthosts.co.uk
08081686777
UK-based and operating 24/7 from dedicated UK data centres. Fasthosts keep over 1 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

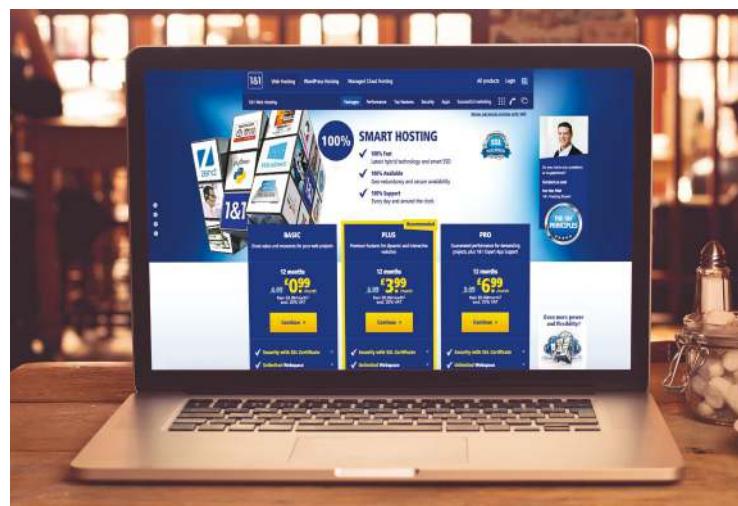


hetzner.com
+49 (0)9831505-0

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 nationally and internationally.

- Dedicated/shared hosting
- Colocation racks
- SSL certificates



All-inclusive hosting



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



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

Agency hosting specialist



nimbushosting.co.uk
02031266781

Nimbus Hosting have partnered with agencies to develop our revolutionary platform STORM. With a team dedicated to outstanding support, our 5-star Google reviews truly speak for themselves. Join the thousands of agencies and freelancers who are benefitting from a control panel that speeds up your website development as well as your client's websites. Supercharge your digital projects today with STORM.

- 30-second WordPress install
- Deploy directly from GitHub
- Easy team management

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, by email and with a ticketing system.

- Cloud servers with any OS
- Linux OS containers
- 24/7 expert support

Get your listing in our directory
To advertise here call the ads team on 01225 442244

COURSE LISTINGS



About us

Northcoders is the coding bootcamp for the north, based in the heart of Manchester and built upon northern values of grit, determination and community spirit. No matter what your background, you can fast-track your career and become a web or software developer in 12 weeks at their

Featured: **Northcoders**

northcoders.com
Twitter: @northcoders
Facebook: Northcoders

full-time bootcamp, or fit their course around your life with their 24-week part-time bootcamp. Their internal career support team will help find you work as a developer, setting up interviews with your choices of Northcoders Hiring Partners across the north of England.

What we offer

- **Full-time:**
Fast-track your career in just 12 weeks
- **Part-time:**
Fit our curriculum around your life in 24 weeks

5 tips from the pros

1. Get started with coding

The best way to know if coding is for you is to just try it! We recommend the free, online JavaScript track of Codecademy to get you started with the basics.

for you, set aside a few evenings each week to really start making progress! If coding is for you, this should be fun.

4. Be prepared

We'll be with you every step of the way when you apply. Make sure you go through all the materials we recommend and ask for help if you're stuck.

5. Get social

With Northcoders, you're not just on a course, you're part of a community that will stay with you long after you graduate. Make the most of it!

2. Do your research

Make sure you read plenty of student reviews to make sure you're applying somewhere reputable. Read their blog and have a look at their social channels.

3. Throw yourself in

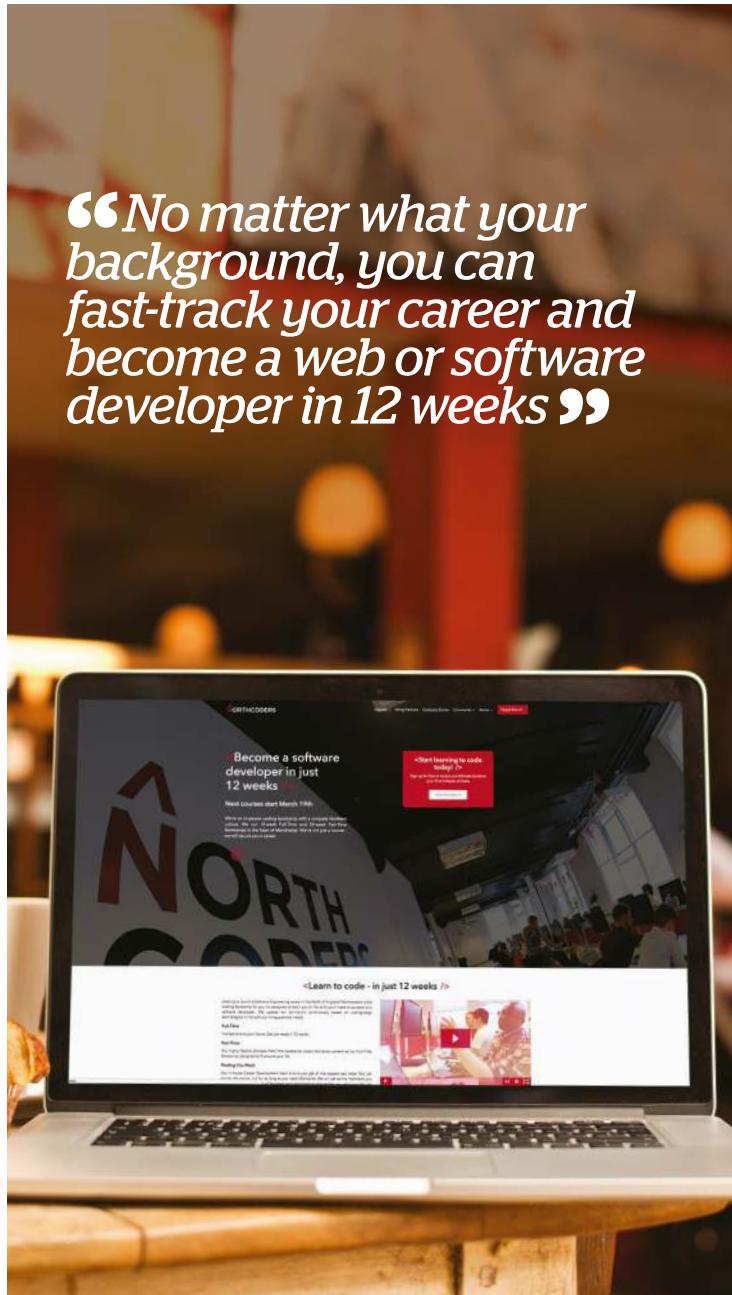
Once you've decided it's right

“ Becoming part of this vibrant, caring community was something I hadn't expected before the course, but now I couldn't be without it. To be a Northcoder is to be enlightened, inspired and supported

Joanne Imlay

Primary school teacher to software developer at Careicon

“No matter what your background, you can fast-track your career and become a web or software developer in 12 weeks”



“ Northcoders delivered their part of the bargain in spades. They provided tremendous assistance in turning me into the full product - a well-rounded, capable future tech employee - and they have the contacts to deliver the opportunities for such people.

Joe Mulvey

Maths teacher to software developer at Auto Trader

udemy

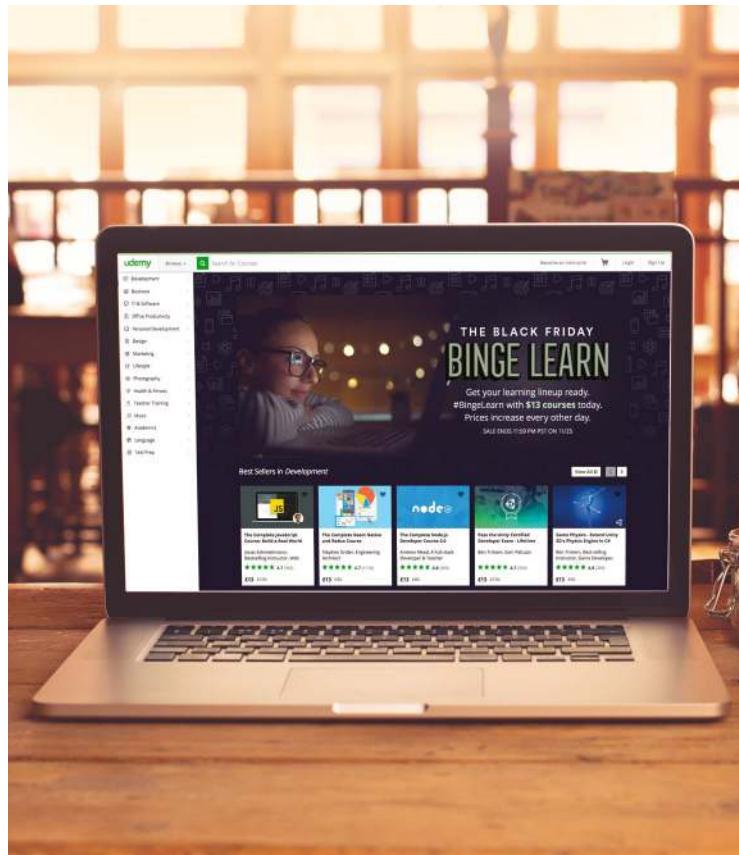
UDEMY

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.



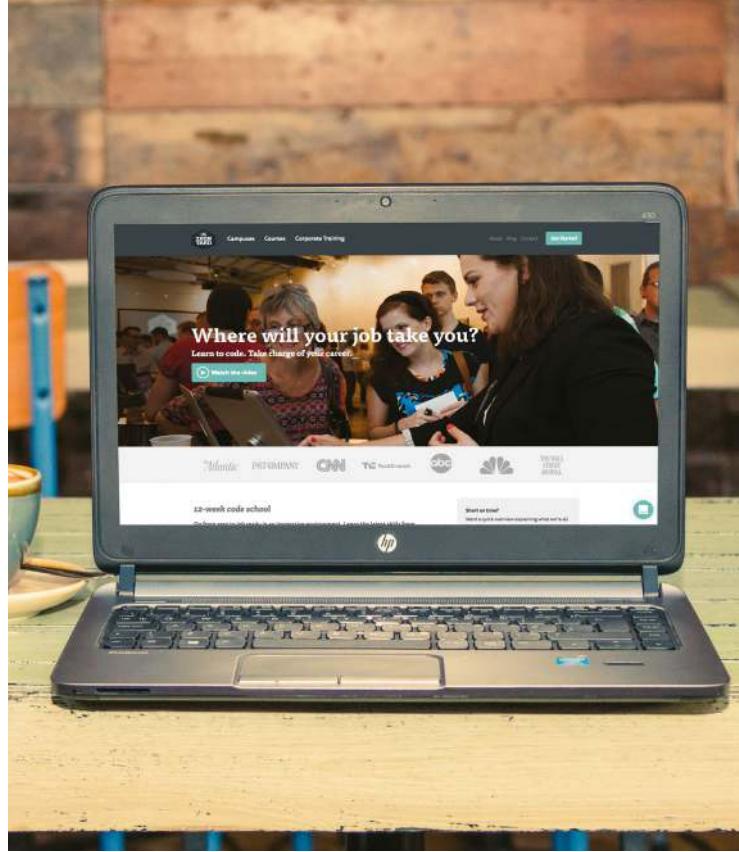
THE IRON YARD

theironyard.com

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.



WE GOT CODERS



WE GOT CODERS

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 courses 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



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



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

Free with your magazine

Instant access to these creative resources...

Essential assets and resources

Get textures, fonts,
backgrounds and more



Exclusive video tutorials

Learn to code/create
with HTML, CSS & JS



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
- 200 Vibrant Overlays, 16 minimalist business card mockups, 61 actions, hi-res textures and double exposure gradients, 50 glossy blood splatter brushes from Sparklestock (www.sparklestock.com)

→ 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



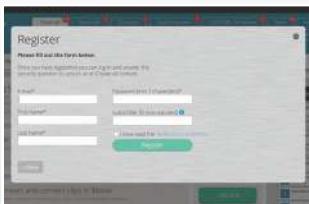


The home of great downloads – exclusive to your favourite magazines from Future!

- 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@futurenet.com

An incredible gift for subscribers



Subscribe today & unlock the free gifts from more than 70 issues

Access our entire library of resources with a money-saving subscription to the magazine – that's more than 900 free resources

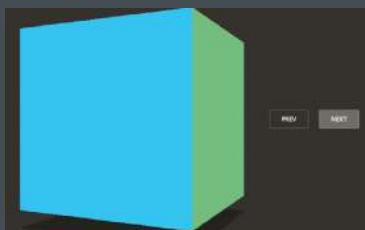
Over 70 hours of video guides

Let the experts teach you to create and code



More than 400 tutorials

Get the code you need to get creative



Over 250 creative assets

Templates, fonts, textures and backgrounds



Head to page 28 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 ten-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

NEXT MONTH

THINK FONTS THINK COLOURS THINK DESIGN

How to choose the right fonts and colours to make your designs stand out from the crowd

OPTIMISE YOUR KEYWORDS

Get the most from meta titles, meta descriptions, headers, image alts and URLs

DESIGN WITH CSS SHAPES

Recreate print style layouts and designs with the power of CSS

USE IMMUTABLE.JS WITH REDUX

Working with Redux means updates must not edit the state directly – Immutable.js makes it easier

Visit the **WEB DESIGNER** online shop at



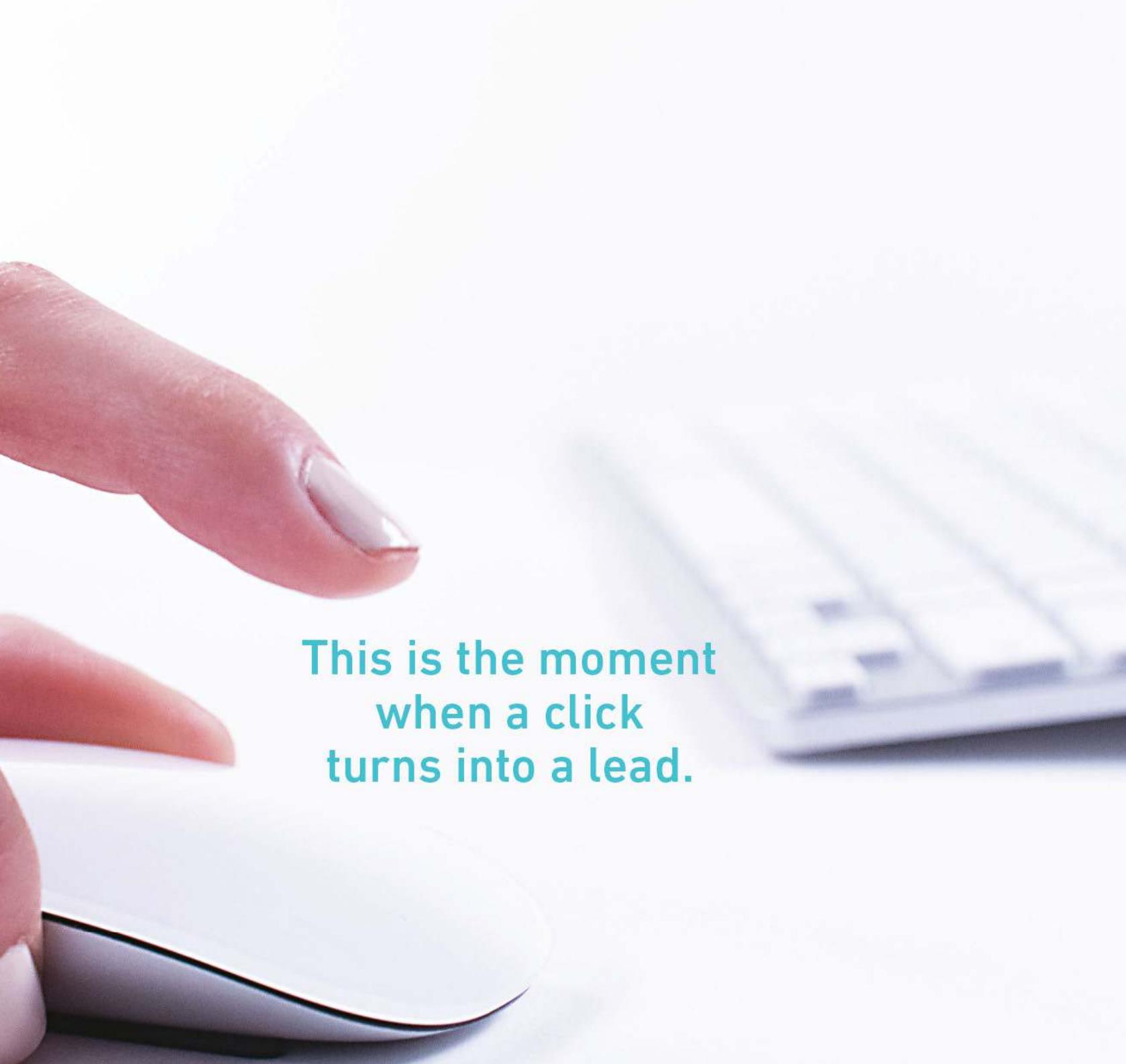
myfavouritemagazines

myfavouritemagazines.co.uk

for the latest issue, back issues and specials

ALL IN YOUR NEXT
WEB DESIGNER
Issue 284 on sale
Tuesday 5 February 2019

SUBSCRIBE TODAY Go to [page 28](#) to learn more



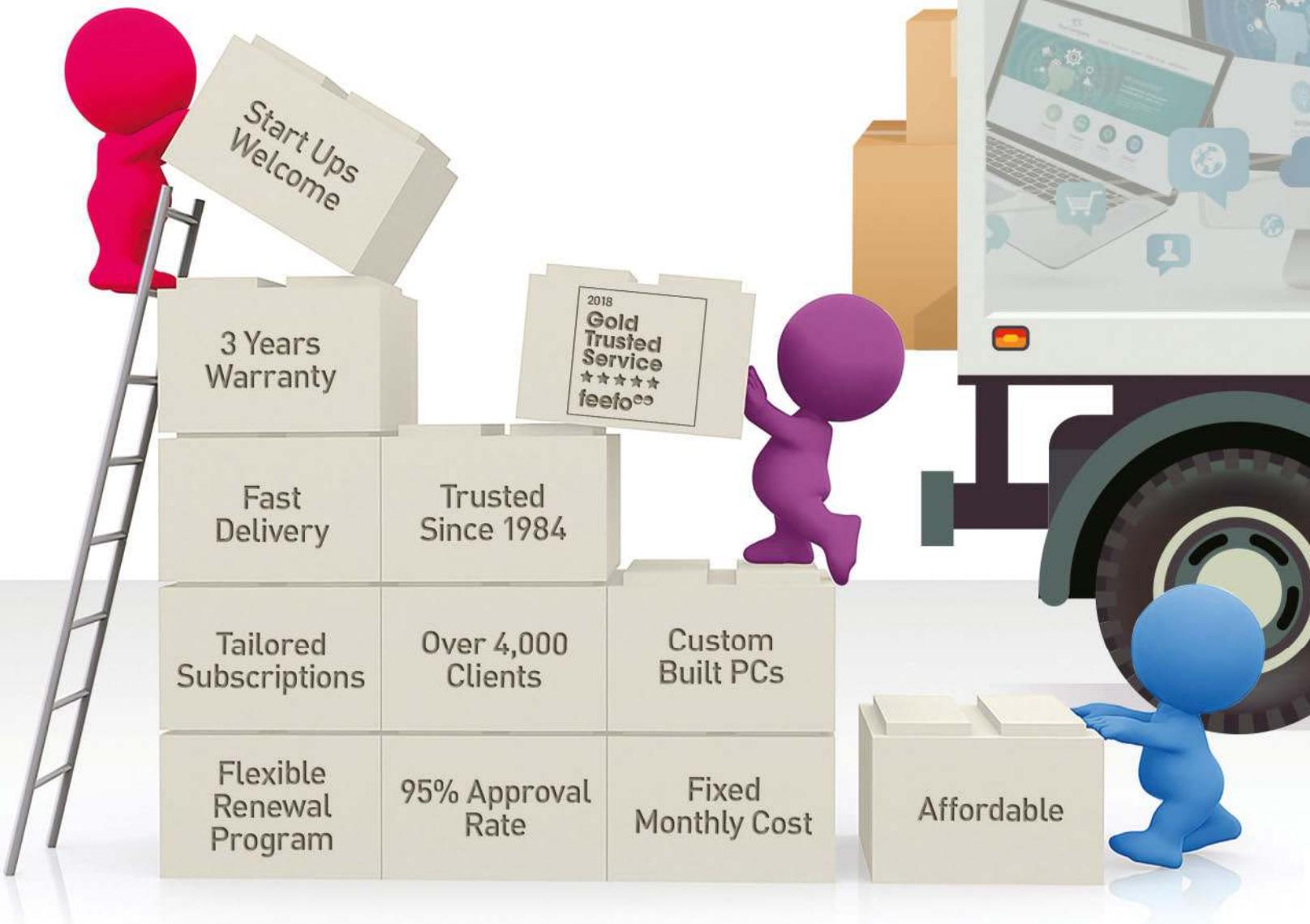
This is the moment
when a click
turns into a lead.

PRESS AHEAD

WP Engine's digital experience platform drives your business forward faster. wpengine.co.uk

WP engine*

Take delivery of **more** than just a computer...



Get so much more with **Flexi-Lease**

Business users only



HARDSOFT
ONE STOP COMPUTER LEASING



Authorised
Reseller

Discover the benefits of leasing computers for business
at www.hardsoft.co.uk Email info@hardsoft.co.uk