

FREE 60 MINS OF FREELANCE VIDEO GUIDES

Expert tutorials, techniques and inspiration

designer™

PWA POWER TIPS

23 ways to make
your apps shine

VARIABLE FONTS

Introduce multiple styles and
weights with one typeface

ANIMATE FOR THE WEB

PRO TIPS
TO INSPIRE
AND BUILD

FEATURING:
GET STARTED WITH
GREENSOCK

DISCOVER
expressJS

Get onboard the fast Node.js
application framework



GET STARTED
WITH THREE.JS

WHAT'S NEW IN
NODE 10?

CODE DYNAMIC
UI TRANSITIONS

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 100

5★

GOOGLE REVIEWS

30 DAY FREE TRIAL

We are offering you a 30 DAY FREE TRIAL of our control panel, STORM*.

Call, email or visit us at

0203 308 2889

sales@nimbushosting.co.uk

nim.host/trial

 **NimbusHosting**

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

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



imation has become a first-class citizen of the web. It is not some novelty looking for support from browsers and creators. It can be found in every corner of the web and is lighting up the user experience.

Web animations come in many forms.

There are extravagant full-page designs that offer the wow factor, and there are functional UI animations that ensure the user is getting a little extra to improve their experience.

In our latest lead feature we have employed a trio of talent that have been working in the world of animation for as long as we can remember: Chris Gannon is an award-winning freelance

interactive designer, illustrator and animator who offers excellent advice on how to approach the art of animation. Keeping him company we have regular Richard Mattka who shows how to get started with the HTML animation library GreenSock. And finishing up we have Supremo designer Dan Heywood who offers some inspirational examples and the best sources to follow on Twitter.

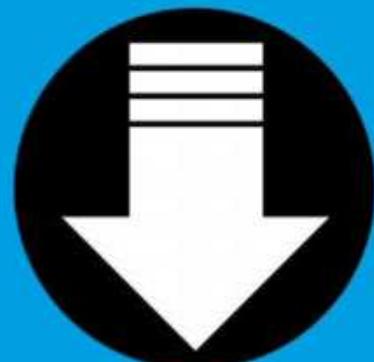
Elsewhere, we move on to mobile and provide a host of PWA tips to power up your mobile (and desktop) apps to ensure that users get the best experience possible.

On the development side we take a closer look at Express.js, the fast, unopinionated, minimalist web framework for Node.js. And on page 56 you can find out why variable fonts are such big news.

As always, enjoy the issue and see you soon.

❶ A picture tells a thousand words and a carefully crafted animation turns those words into a meaningful story ❷

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
68 Designer resources

Video Tuition - 60 minutes of the Complete Freelancer Course video guides from Killersites (shop.killervideostore.com)

Assets - 40 Travel and portrait PS filters and 17 Animated banner ads from Sparklestock (sparklestock.com)
- 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...



Animate for the web

This issue's contributors include award-winning interactive designer Chris Gannon; **Web Designer** regular Richard Mattka, who shows you how to get started with GreenSock; and Dan Heywood, frontend developer and designer at Supremo, who offers inspiration and seven to follow on Twitter.

Page 42

More and more it's becoming important to design interactive, animated content as a framework rather than a linear experience

Maximiliano Firtman



Max is the author of High Performance Mobile Web, a conference speaker and trainer with a focus on mobile apps, PWAs and web performance. This issue he reveals his essential tips to get the best from your mobile designs. **Page 66**

Mark Shufflebottom



Mark is a professor of Interaction Design. In this issue Mark is looking at variable fonts in CSS and how they allow for one font to be manipulated to create different styles, and eliminate slow page load times. **Page 56**

Richard Mattka



Richard is an award-winning interactive director, designer and developer. In this fourth tutorial in an ongoing 3D programming series, he shows you how to interact with Objects in 3D space using the Three.js library. **Page 52**

Frank Kagumba



Frank is a frontend developer and tech writer based in Nairobi, Kenya. In this tutorial, he will teach you the basics of the Pts JavaScript library and show you how to create a captivating landing page with the library. **Page 86**

Neil Pearce



Neil is a long-time frontend developer and designer who works with HTML, CSS and JavaScript. In this issue he demonstrates how to develop an animated UI element, ideal for eCommerce, with help from CSS and JS. **Page 60**

David Howell



David is a journalist with over 20 years' experience in publishing and runs his own business, Nexus Publishing. In this issue he heads to Hampshire to talk to the talented crew at Studio Republic and see how they operate. **Page 34**

Tam Hanna



Tam describes himself as an old process computer head who quickly found a taste for Node.js. In this issue he reveals how Node.js 10 introduce nifty tweaks intended to appeal to all technical audiences. **Page 80**

Leon Brown



Leon is a freelance web developer and trainer who assists web developers in creating efficient code for projects. This issue he recreates a host of techniques as inspired by the top-class sites seen in Lightbox. **Page 14**

Follow us!

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

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

Editorial

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

Designer **Harriet Knight**
Editor in Chief **Amy Hennessey**
Senior Art Editor **Will Shum**

Contributors

Andy Culbert, Chris Gannon, Dan Heywood, Mark Billen, Leon Brown, David Howell, Mark Shufflebottom, Richard Mattka, Frank Kagumba, Neil Pearce, Maximiliano Firtman, Tam Hanna, Philip Morris, Rob Mead-Green

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
Account Manager **Chris Mitchell**
chris.mitchell@futurenet.com

International

Web Designer is available for licensing.
Contact the International department to discuss partnership opportunities
International Licensing Director **Matt Ellis**
matt.ellis@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

Managing Director **Aaron Asadi**
Commercial Finance Director **Dan Jotcham**
Editorial Director **Paul Newman**
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 9060

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 paper 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 2008885) 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**
Chairman **Richard Huntingford**
Chief financial officer **Penny Ladkin-Brand**
Tel +44 (0)1225 442 244

MAKE INSPIRATIONS COME ALIVE

PD Series Designer Monitors



technicolor



Colour Certified

UHD/QHD* | 100% sRGB | Rec. 709

*models may vary

BenQ.co.uk

BenQ UK

BenQ_UK

Order now from

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 Generate London 2018

Who is speaking at the conference and what are they speaking about? Find out more

10 WebKit: The best must-try resources out there

Discover the libraries and frameworks you need.

11 Designing for digital

Andy Culbert, cofounder of digital-first branding agency MERÓ, talks about change in design.

14 Lightbox

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

26 Bankable assets

Find out how agency CODE D'AZUR constructed an AR escape-room experience.

34 Ethical Design

Studio Republic are shaping the digital spaces of non-profits and charities.

42 Animate for the web

Essential pro advice, inspirational examples and how to get started with GreenSock.

66 PWA power tips

A host of tips to ensure that the mobile experience is the best it can be.

74 Discover Express.js

Five things you need to know about the minimalist web framework for Node.js.

92 Hosting listings

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

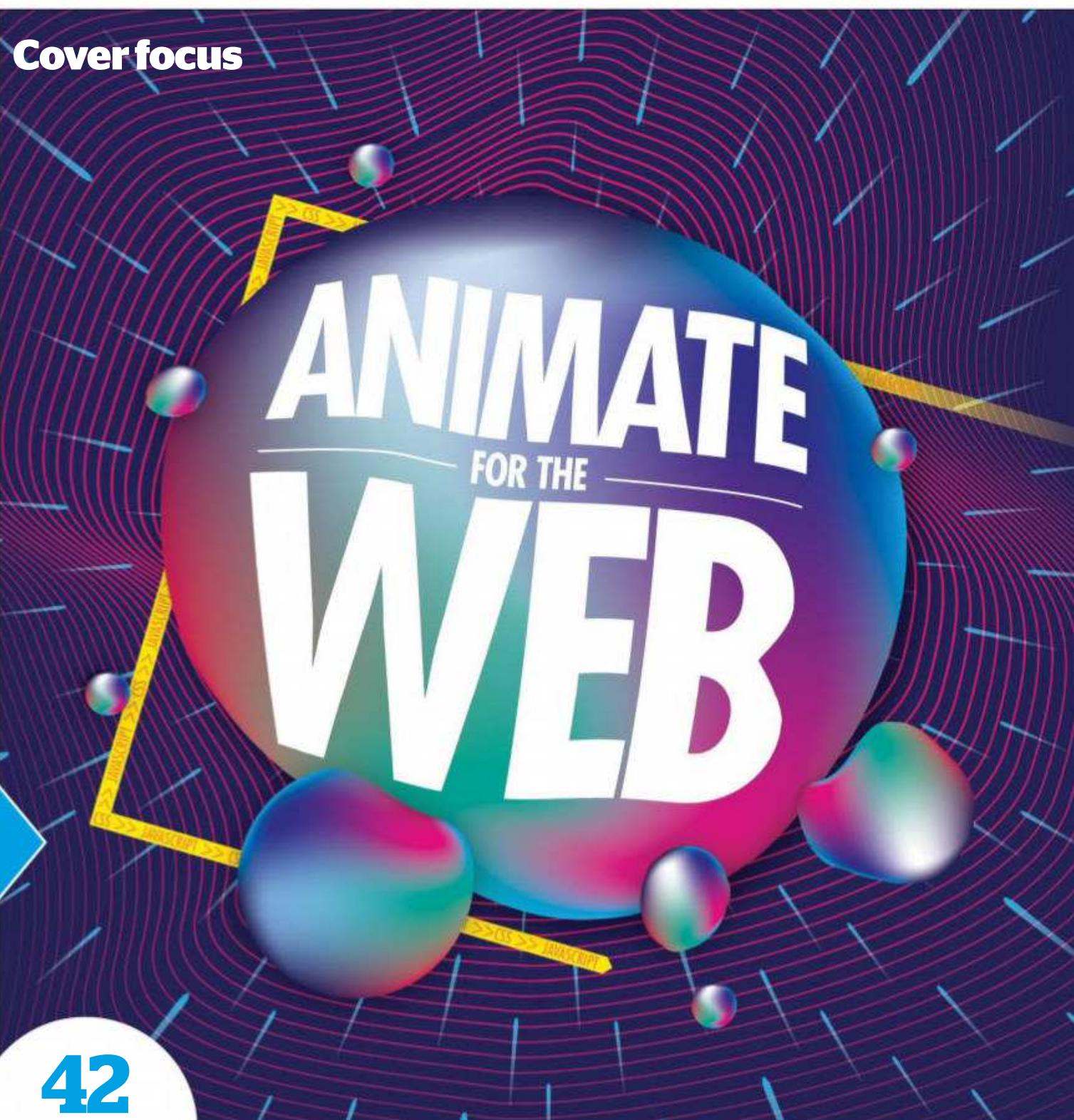
94 Course listings

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

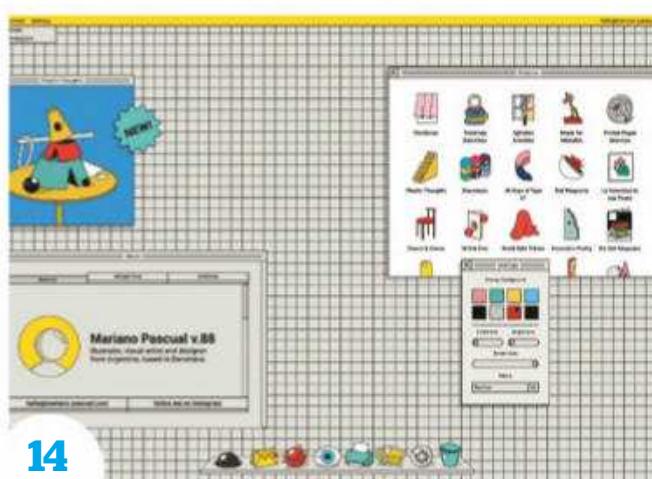
98 Next month

What's in the next issue of **Web Designer**?

Cover focus



42



14

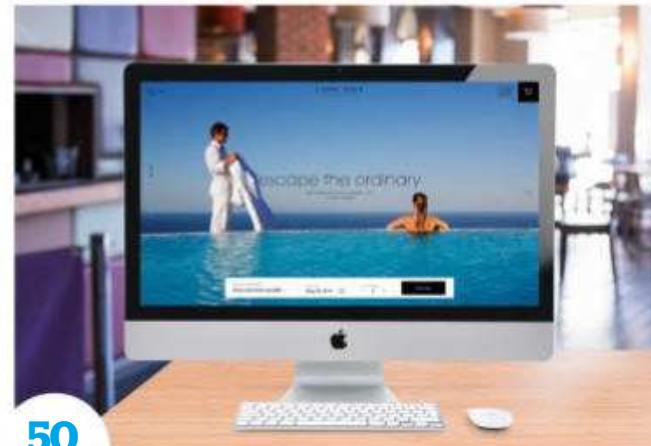


18

TUTORIALS

50 Creating cursor effects

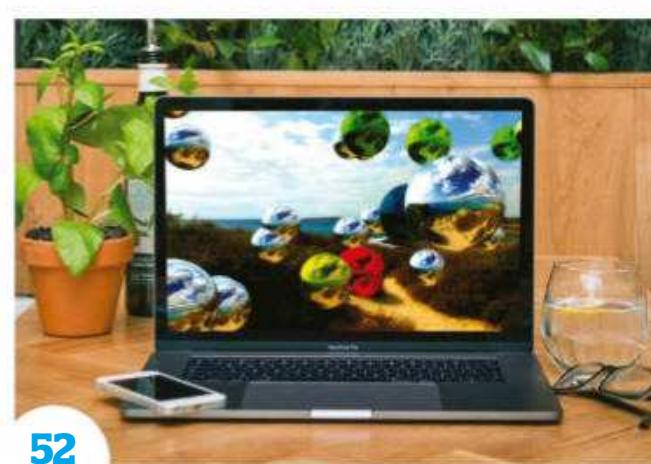
Learn how to add some background glow to the mouse cursor



50

52 Get started with Three.js

The latest in the series shows you how to interact with 3D objects in WebGL.



52

56 Design with variable fonts

Get excited. Variable fonts enable you to load one font and manipulate it through CSS.



64

74 Five reasons to love Express.js

Discover what the minimalist web framework for Node.js has to offer the development process.



34

80 What's new in Node.js 10

Take a closer look at the JavaScript runtime's latest updates and what they can do.

86 Get creative with code

Discover the fundamentals of working with the Pts.js JavaScript library and how to create an engaging landing page.

FILESILO

96 Get the latest must-have resources and videos

- 60 minutes of freelance video guides from Killersites.
- 40 Photoshop filters and 17 animated banner ads 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 20%

<https://bit.ly/2sGwB3h>

Header

The tools, trends and news to inspire your web projects

Are you going to Generate London?

Last chance to get tickets to see a stellar line-up of speakers



eb conferences are a great way to listen and learn and this year's Generate London (19th-21st September) is no different. It has a superb line-up of well-established, upcoming and spot-on speakers who will impart their words of wisdom and pass on their vast bank of knowledge.

So what does Generate London 2018 have to offer? It has 16 speakers and three workshops spread over three days, all hosted at the impressive The Royal Institution. You can catch a quick glimpse of the location at venue.rigb.org/.

Keynote speakers include Web Standards Lovegod (his words not ours) Bruce Lawson; super-smart designer Sarah Parmenter; frontend genius Sara Soueidan; and the creator of the web's most popular 3D library – Three.js – Ricardo Cabello AKA Mr. Doob.

Keeping this illustrious selection of web gurus and experience makers company are AR/VR evangelist Alexandra Etienne; React Academy founder Kristijan

Ristovski; digital artist Marpi Marcinowski; Asembler creative developer Steven Roberts; Clearleft cofounder Richard Rutter; CodePen senior software engineer Cassidy Williams; and UX consultant Laura Yarrow. Head to generateconf.com/london/speakers to see the complete line-up.

So what are the key subjects that the speakers are concentrating on in 2018? Naturally, there's a strong focus on core

AR can bring, and what you can do with AR.js library. Elsewhere, Web Standards Lovegod Bruce Lawson will be talking web standards and the mindset required to be a 'shokunin' of the web. What's a 'shokunin' of the web? Head to the Generate site for more info. And frontend UI developer Sara Soueidan will be extra busy taking on a talk and a workshop. Her workshop 'Frontend Meets UX: Designing & Creating Universal UI Components' will

“Keynote speakers include designer Sarah Parmenter”

design and development processes, but you'll also be able to take a closer look at the technologies that are shaping today's and tomorrow's web experiences. Three.js creator Ricardo Cabello will be giving an overview of the current state of the APIs and how to starting building your first VR and AR projects with Three.js. AR/VR evangelist Alexandra Etienne will be revealing the advantages that web-based

teach best UX considerations and practices, and help you apply those lessons so you can work your way through building components.

There is so much more to see and learn, but you'd better hurry: there are only a few days left for you to get your ticket and join this smorgasbord of web talent. Head to bit.ly/2PvVUOF to get yours. See you there.

CREATIVE BLOQ
creativebloq.com

In-depth tutorials, expert tips, cutting-edge features, industry interviews, inspiration and opinion. Make sure to get your daily dose of creativity, design and development.



Subscribe today and save 20%
<https://bit.ly/2qLxVI4>

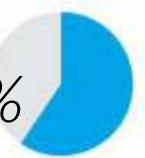
STAT ATTACK BROWSER SHARE

Who's in the top five worldwide? Any surprises?

Google

59.69%

The Chrome browser is still on the rise.



Safari

14.49%

Popular on mobile. Staying steady.



UC Browser

6.27%

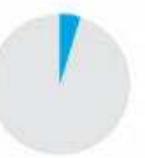
Down 1% compared to the same time last year.



Firefox

4.93%

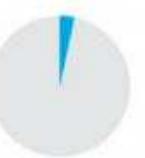
Giving up 1% to other browsers compared with last year.



Opera

3.52%

Popularity waning in Africa. Down 6% on last year.



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

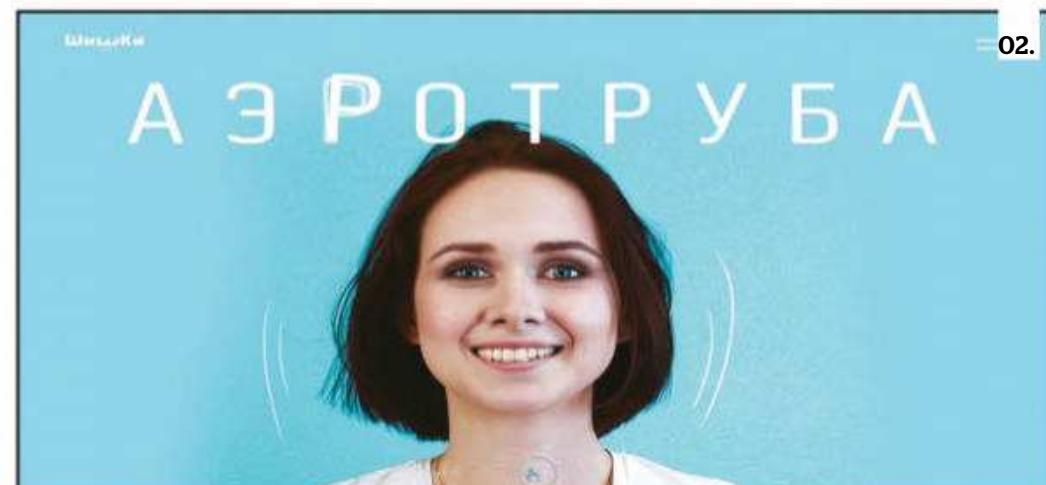
Sites of the month

風景

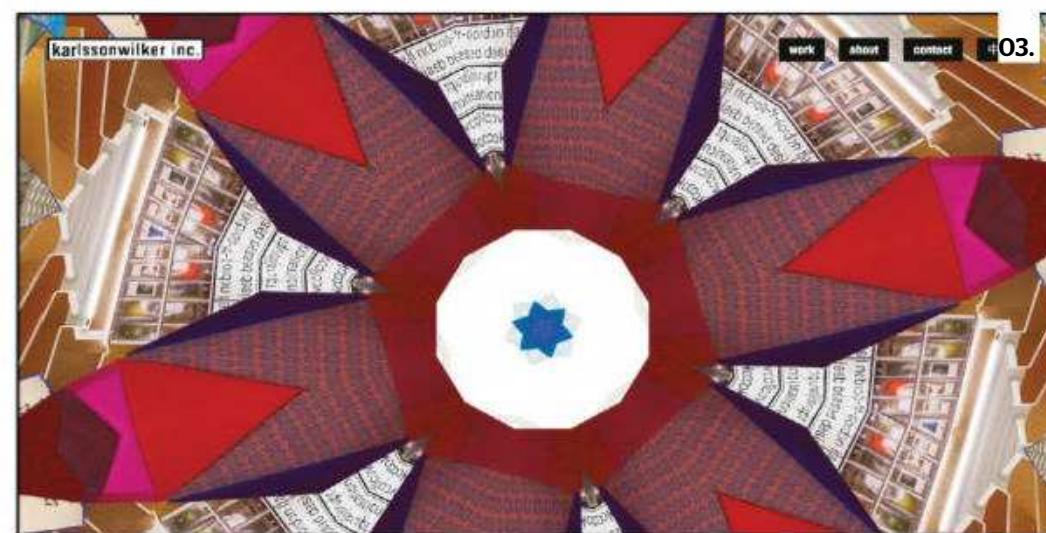
Landscapes
& cities



01.



= 02.



= 03.



04.

01. Get the picture

getthepicture.tours/galleries

Animated circle transitions abound as they present picture galleries.

02. Shishki Park

shishkipark.ru/aerotube

Inventive press-and-hold image and text animations that engage.

03. karlssonwilker

karlssonwilker.com

Kicks off with an interactive kaleidoscope. What more do you want?

04. Gabrielle Dolan

gabrielledolan.com

Horizontal navigation that goes vertical on small screens and neat typography.

Graphics THE END

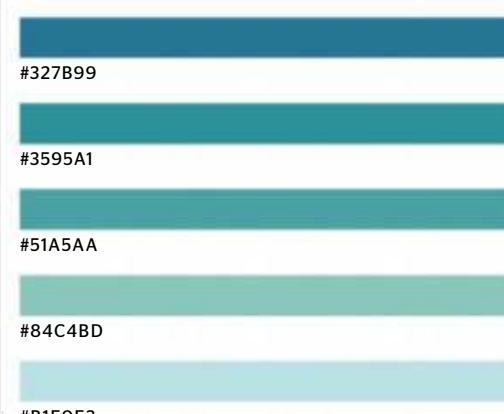
bit.ly/2wkZi7F

Smart illustrations with a limited colour palette and a story to tell.



Colour picker Window

bit.ly/2MUDTvj



Typesetter Hideout

bit.ly/2MDppXN

A retro, antique font that looks to the past for its inspiration. Ideal for branding.

ABCabc
0123456

WordPress Structura

bit.ly/2Lzma83

A collection of minimalist one-page designs that keeps it simple and the user focused.

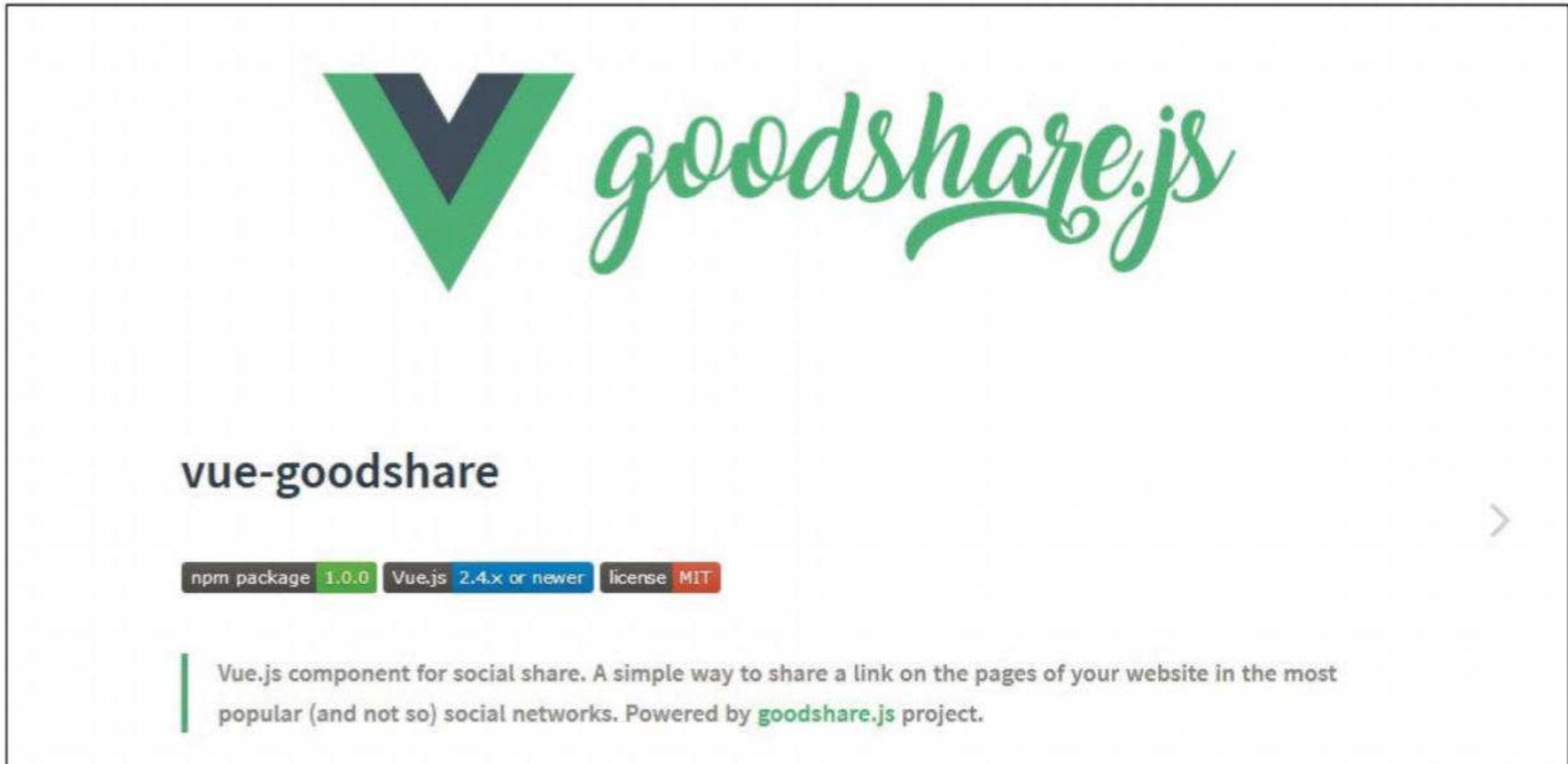


Header Resources

CONTACT US AT: webdesigner@futurenet.com | @WebDesignerMag

webkit

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



The screenshot shows the GitHub page for the `vue-goodshare` component. At the top, there's a large green 'V' logo followed by the text "goodshare.js". Below this, the component name "vue-goodshare" is displayed. A navigation bar includes links for "npm package 1.0.0", "Vue.js 2.4.x or newer", and "license MIT". A description box states: "Vue.js component for social share. A simple way to share a link on the pages of your website in the most popular (and not so) social networks. Powered by [goodshare.js](#) project." A green vertical bar is positioned to the left of the description.

Goodshare.js

bit.ly/2MKGLvt

This is a Vue.js component that allows for social sharing. Includes a simple install, extensive documentation, is SEO-friendly and has UI customisation options. Supports all the major social networks including Facebook, Twitter and tumblr. Plus, a host of lesser-known options. Can't see what you want? Ask and the authors will try to add.



F2

antv.gitbook.io/f2

F2 is a flexible, elegant and interactive charting library with a focus on mobile. It is HTML Canvas-based and compatible with Node.js, Weex and React Native.



Evergreen

evergreen.surge.sh

This is a React UI framework that describes itself as a Design system for the web. It offers out-of-the-box functionality and full customisation.



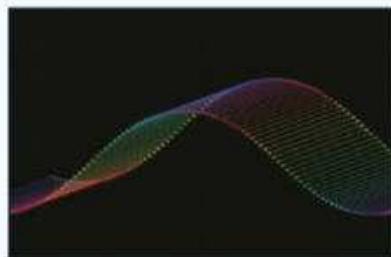
lazyestload.js

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

Images can be bandwidth heavy. This library only loads images when they come into the viewport. If the user scrolls past an image it will not load.

TOP 5 CodePens

Be inspired by this collection of smart and interesting codebases



CSS Rainbow

bit.ly/2QOQlBk

A mesmerising, animated rainbow-coloured wave that's constantly on the move.

Powered by some simple CSS.



3D Insta Washer

bit.ly/2PZPZIB

Inspired by the old Instagram logo this CodePen uses CSS to create a constantly moving animated washing machine.



Text animation

bit.ly/2CcvUFR

A great example of how text can be animated to good effect. Multiple animations are created with HTML, CSS and JavaScript.



The Carlton dance

bit.ly/2PWgRTN

If you don't know the Carlton dance, then why not? This 8-bit style animation includes music.



Hover effects

bit.ly/2wDd3id

A host of different animated hover effects that are well put together. Change the options by using the Sass variable.

Designing for digital

Digital-first branding has finally overtaken offline brand design



Andy Culbert

Co-founder of digital-first branding agency MERÓ
<http://worksbymero.com>

“ Coca-Cola and Audi are just some of the household names rethinking their branding from a digital-first perspective ”

The world is finally switching on to the fact that designing for digital channels is an essential way to future-proof a business. Digital-first design is ‘second-nature’ to designers who have come from a digital background; we know we have to create something that works even on the smallest screens, yet up until now, designers – including ourselves – have still experienced being handed a brand identity or design that simply doesn’t work in some digital channels and has to be totally recreated.

Part of the problem is shaking off old ways of thinking and approaching design. Sometimes, it’s communicating the importance of digital to a wider organisation, or gaining senior level buy-in that can be the problem. But what about historic brands created before the digital age? Well Coca-Cola and Audi are just some of the household names rethinking their branding from a digital-first perspective – because going forward, they need to.

Legacy branding that was designed prior to today’s ‘digital world’ can be problematic. As to be expected, where the predominant representation of the brand (and audience’s experience) was offline, assets and identities were designed to best cater for these channels to market. Then as new digital channels emerged, the branding was ‘shoehorned’ into online environments. Many are now finding that, as they expand their digital activities, their brand is actually not fit for purpose. It simply doesn’t ‘work’ digitally. The problem with designing for offline first is that digital branding has a different set of considerations versus print. Fonts need to be available for digital and legible at all sizes, colours need to be accessible (we aim for anything between AA and AAA compliant) and images, logos and other design components need to be carefully considered. Overly complicated elements get lost when scaled down to the smallest spaces. Something that looks amazing in print might not be usable or functional when applied to a website or app. And it’s easier to scale a brand upwards and out if you have considered the smallest first rather than vice versa.

Video is another consideration and an increasingly important tool to engage with consumers. By 2020, it’s predicted that online videos will make up more than 80% of all consumer internet traffic, yet many organisations give little or no thought to how their brand will move when animated.

And on top of this, digital experiences are constantly evolving. There are more channels than ever before, each with different requirements that can alter overnight. A brand identity needs to perform in a multi-screen, multi-channel world, where each device and platform requires different sizes, fonts and image specifications. Even tone of voice needs are different for online audiences and in different channels; think of the vastly different way brands need to be represented on Instagram vs Twitter – it’s similar to how you alter your behaviour when visiting an art gallery compared to a nightclub.

The digital landscape is prompting a need for brands to be much more versatile. Yet many brands still have a set of static, offline brand guidelines that leave digital teams scratching their heads! One way that marketing teams and designers can better prove the worth of digital-first design is by getting to know the needs and reactions of the customer better and using this to prove effectiveness. Digital-design allows this to happen much more easily. It’s more effective to test a brand online than offline. We can use a variety of tools to help make some really informed decisions. Digital design means we can look beyond whether the majority, or even one person, has a preference to the overall look and feel. It has to be tested to ensure ‘it works’ and resonates with the end user. Tangible and measurable results will drive the end product rather than opinions of few.

A digital-first approach is also vital for those brands wanting to future-proof their business by connecting with the next generation of consumers. Millennials and their Gen Z successors have high standards when it comes to the digital expectations they put on brands. Gen Z especially have been raised by the internet. They’re multi-taskers, sometimes juggling up to five screens at a time, which means they’re easily distracted. A brand’s online experience needs to be slick and seamless across all touch points. And above all it needs to be flexible to stay relevant to their audience. Digital-first is the only approach that can achieve this.

Header Resources

CONTACT US AT: webdesigner@futurenet.com | @WebDesignerMag

webkit

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

The grid-based cheat sheet covers the following properties:

- display**: Establishes a new grid formatting context for children.
- grid-template**: Defines the rows and columns of the grid.
- grid-gap**: Specifies the size of column and row gutters.
- justify-items**: Aligns content in a grid item along the row axis.
- align-items**: Aligns content in a grid item along the column axis.
- justify-content**: Justifies all grid content on row axis when total grid size is smaller than container.
- align-content**: Justifies all grid content on column axis when total grid size is smaller than container.
- grid-auto-flow**: Algorithm for automatically placing grid items that aren't explicitly placed.

GRID

grid.malven.co

Grid layouts are big news and are the ideal way to create super-flexible and unique layouts. Any thing that helps to make the process easier is always worth a look. GRID is a simple, straightforward interactive cheat sheet. Pick the layout you want and the code gets copied to your clipboard ready to use.



Vivid.js

webkul.github.io/vivid

Need some neat SVG icons that are ready to go? Vivid provides an easily customisable set of over 100 icons that look good anywhere.



Bit

bitsrc.io

Bit is cloud component hosting. Hosting components in the cloud means teams can discover and organise the components of their projects.



WorkerDOM

bit.ly/2CoulRB

In short worker-dom is “The same DOM API and Frameworks you know, but in a Web Worker”. Make sure you pay a visit to find out more.

TOP 5 multi-purpose WordPress themes

Need a theme that has more than one look and design. Check out this selection



Avada

bit.ly/2dQQSOv

Powerful options and tools, unlimited designs and a responsive framework make this great for customisation.



Impreza

bit.ly/2NE3mWV

A multi-purpose WordPress theme, created with the best practices of modern site building. Check out the demos.



Canvas

bit.ly/2LRct05

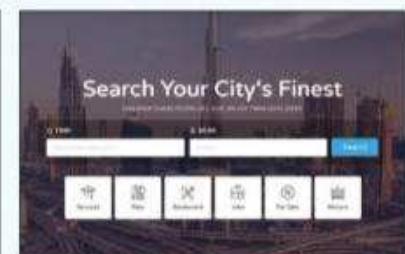
A multi-purpose Bootstrap-based template with 850+ niche, multi-page and one-page layouts with 1,000+ UI features.



Bridge

bit.ly/2DqehPt

Whatever type of site you want to start creating this theme has an option for it. Check out the 376+ demos that it has on offer.



Directory Box

bit.ly/2PwLoX1

This theme has a focus on directory-style sites. It has a number of pre-designed templates for you to use.

HEAR FROM THE PROS

SIGN UP TO THE

**web
designer**TM

NEWSLETTER TODAY!



Get weekly news, tips & inspiration

SIGN UP NOW!

bit.ly/2KI5b7Y

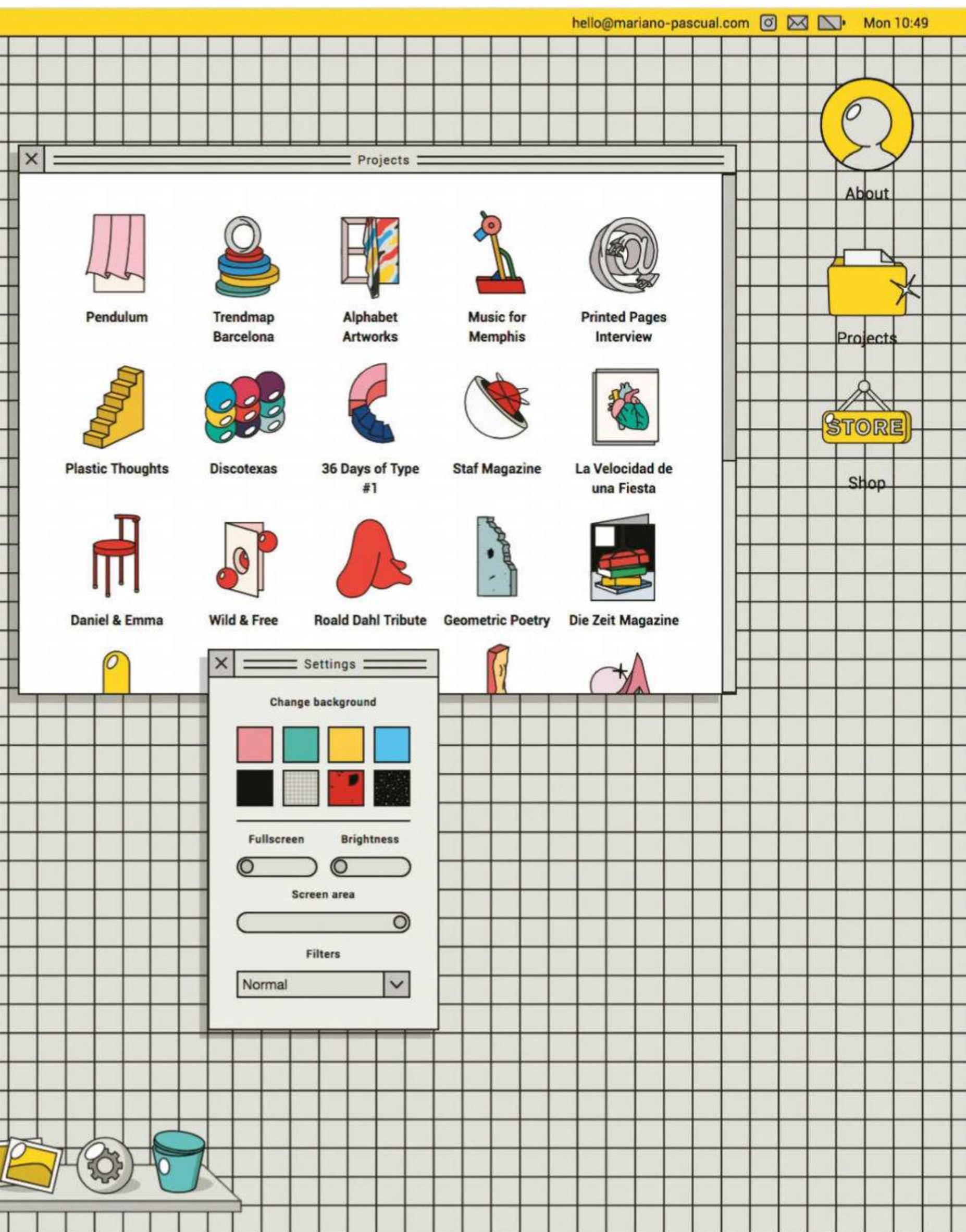
Mariano Pascual

mariopascual.me

Designer:
Achos achos.xxx

The image shows a screenshot of Mariano Pascual's website. At the top, there is a navigation bar with links for 'Mariano Pascual', 'File', 'Contact', and 'Settings'. Below the navigation bar is a small sidebar with links for 'Email' and 'Instagram'. The main content area features a large blue window titled 'Plastic Thoughts' containing a cartoon illustration of a character with a yellow cone head, red shirt, and teal pants, standing on a yellow palette. A blue starburst next to the character says 'NEW!'. Below this window is another window titled 'About' with tabs for 'About me', 'Selected Press', and 'Exhibitions'. The 'About me' tab is active, showing a large yellow and white circular logo on the left and the text 'Mariano Pascual v.88' followed by 'illustrator, visual artist and designer from Argentina, based in Barcelona.' at the bottom. At the very bottom of the page, there is a footer with links for 'hello@mariano-pascual.com' and 'follow me on Instagram'. The entire website is set against a background of a light gray grid.

“Barcelona-based artist Mariano Pascual’s online gallery portfolio is reimagined as a stylised operating system full of surprises”



Colours

#DADAD1	#4196CD
#FAD843	#E64F47

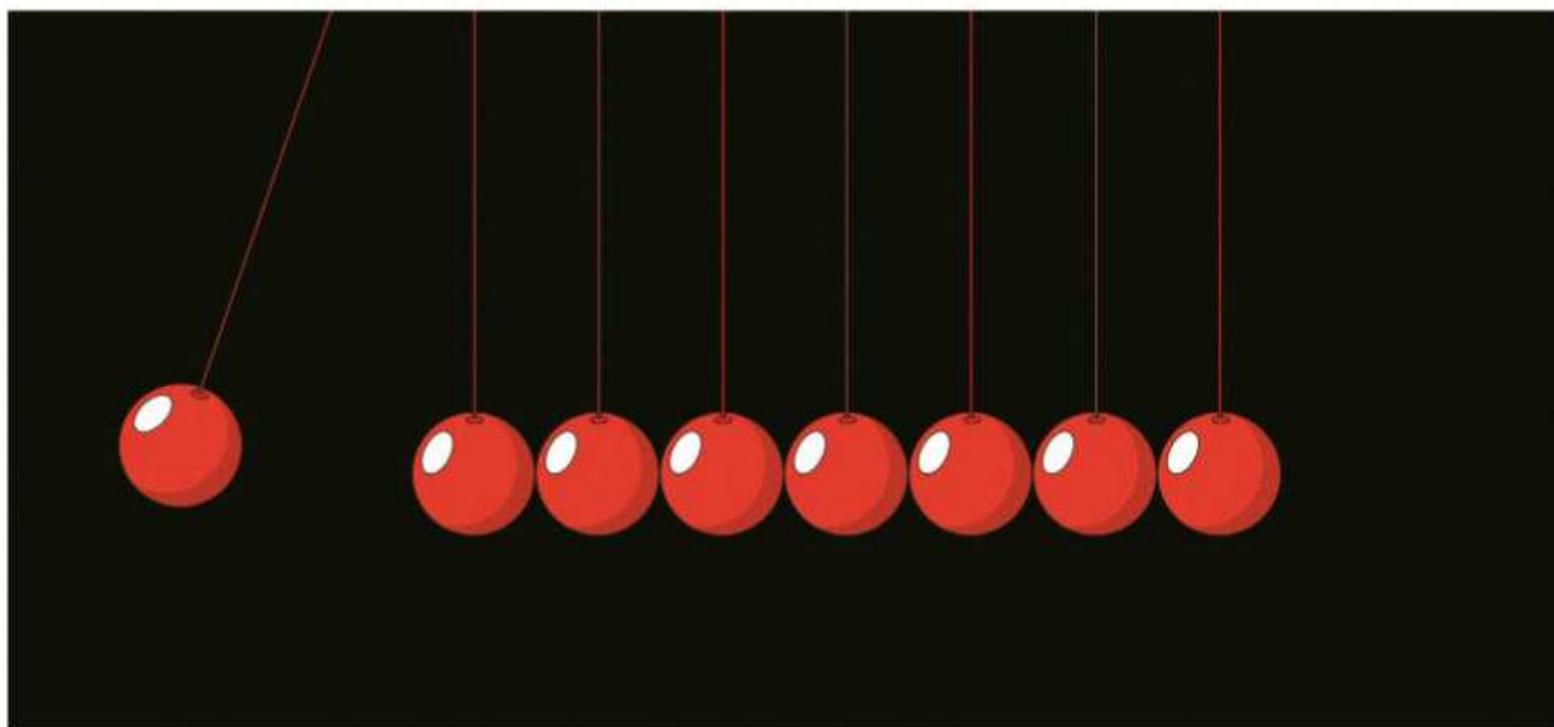
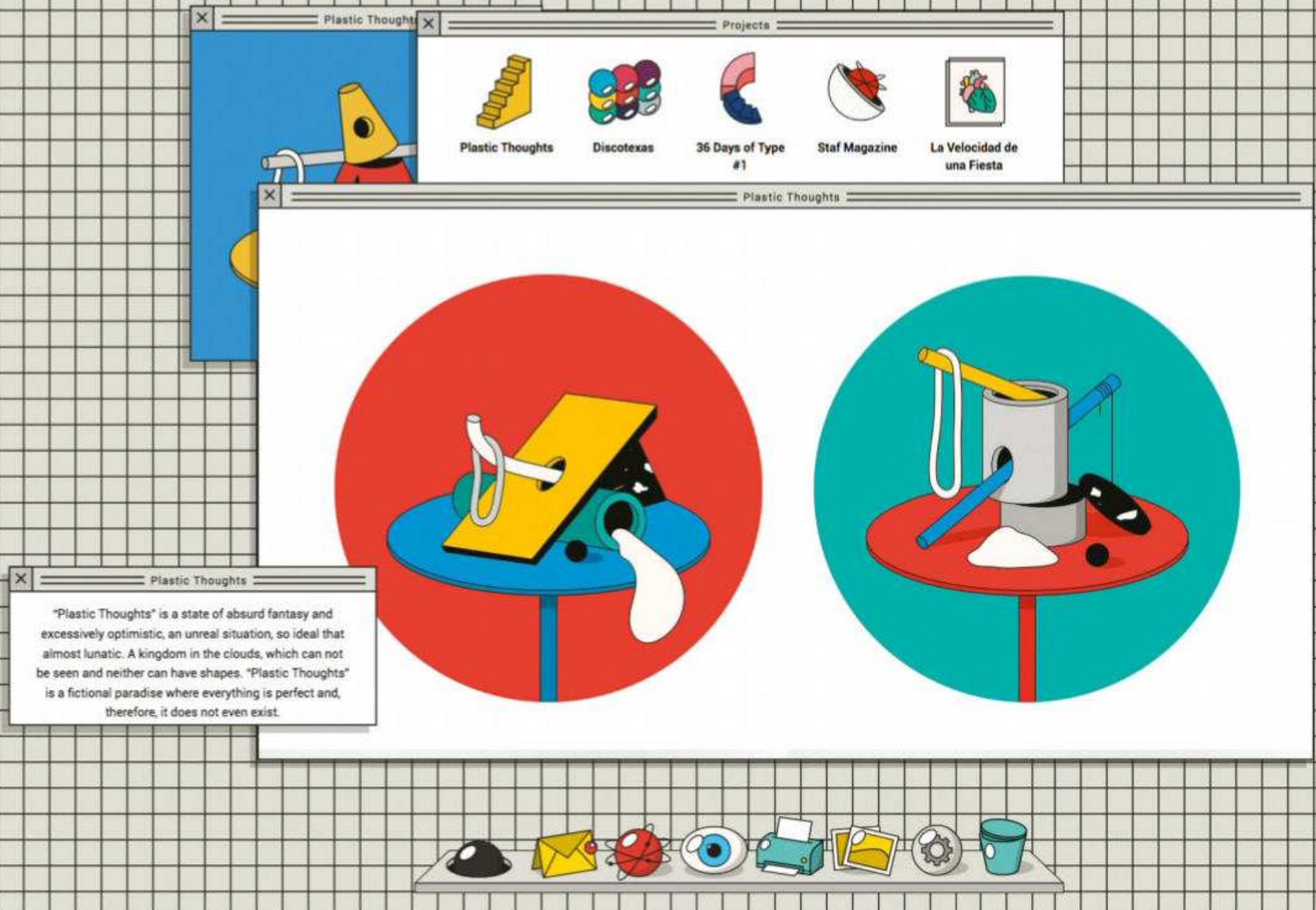
Tools

HTML5, CSS3, jQuery, SVG

Fonts

abcABC
1234567890
abcABC
1234567890

Roboto by Christian Robertson and available via Google Fonts is used in Regular and Bold varieties.



Above

Plastic Thoughts and other projects can be viewed within floating windows that can be dragged, dropped and closed

Middle

Packed with humour, the site has 'settings' for switching desktop backgrounds, screen size and the application of colour filters

Bottom

Leave the site idle for a moment or two and the desktop shows an animated desk toy screensaver

How to create an interactive pop-up box

Use CSS and JS event listeners to create pop-up boxes activated from regular link references

1. Initiate HTML document

The first step is to initiate the HTML document structure. This consists of the HTML document that contains two sections – head and body. While the head is used to contain references to the external CSS and JavaScript resources, it is the body section that will be used in step 2 for storing the page content elements.

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

2. HTML content

The HTML content consists of a sample navigation and an article container used as the pop-up box. Links will activate their associated pop-up box by using their 'href' attribute to reference the box's ID. A 'data-popup' attribute is applied to any container element that is to be used as a pop-up box.

```
<nav>
    <a href="#box1">Open Lightbox</a>
</nav>
<article id="box1" data-popup>
    <h1>Hello</h1>
</article>
```

3. CSS initiation

Create a new file called 'styles.css'. The first step of the stylesheet initiates the presentation of the page document. Default colours are applied for the page background and content colour, as well as a guarantee for no visible border spacing. The default font is also applied for all content to inherit.

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

4. Pop-up definition

Container elements using the 'data-popup' attribute are set to display with a width and height that's proportionate to the browser window. Similarly, fixed positioning is used to position the element with co-ordinates calculated in proportion to the browser – guaranteeing centred positioning. A negative z-index results in the element being hidden below the visible page – and to become visible when the value of the 'data-popup' attribute is set to 'open'.

```
[data-popup]{
    position: fixed;
    display: block;
    width: 50vw;
    height: 50vh;
    top: 25vh;
    left: 25vw;
    z-index: -1;
    border: .2mm solid #fff;
    padding: 1em;
    overflow: auto;
    box-sizing: border-box;
}
[data-popup="open"]{
    z-index: 99999; }
```

5. Close control

The close button is made from an element created in step 7 that contains the 'data-control' attribute set to 'close'. It appears at a specification co-ordinate calculated in relation to the browser window size. The content size of this element is set to double the size of the pop-up box content size. A white background and black content colour are also applied.

```
[data-popup] [data-control="close"]{
    position: fixed;
    top: 25vh;
    right: 25vw;
    font-size: 2em;
    font-weight: bold;
    background: #fff;
    color: #000;
    padding: 0 .25em; }
```

6. JavaScript initiation

Create a new file called 'code.js'. This step applies an event listener to the browser window that will be used to activate the JavaScript code after the page has loaded. This is important to avoid the JavaScript returning errors due to content elements not being available at the time of its execution.

```
window.addEventListener("load", function(){
    *** STEP 7 HERE });
});
```

7. Pop-up box code

A search for every page element with the 'data-popup' attribute is performed. Each item found has a closed button applied to – ie the one referenced in step 5. This button has a 'click' event applied that will result in the parent 'data-popup' attribute being set to 'closed' – ie not 'open' results in the pop-up box disappearing.

```
var nodes = document.querySelectorAll('[data-
popup]');
for(var i=0; i<nodes.length; i++){
    var ctrl = document.createElement("span");
    ctrl.innerHTML = "&times;";
    ctrl.setAttribute("data-control", "close");
    ctrl.addEventListener("click", function(){
        this.parentNode.setAttribute("data-popup",
        "closed");
    });
    nodes[i].appendChild(ctrl); }
```

8. Make it visible

A search is performed for all links containing an href attribute beginning with '#' – ie those referencing an on-page ID element. Each of these links has a 'click' event listener applied, which will check to see if the referenced page element has the 'data-popup' attribute. Its value will be set to 'open' – triggering the visibility of the found element.

```
var nodes = document.querySelectorAll('a
ef^="#"');
for(var i=0; i<nodes.length; i++){
    nodes[i].addEventListener("click", function()
    {
        var box = document.querySelector(this.
        getAttribute("href"));
        if(box != null && box.hasAttribute("data-
popup")){
            box.setAttribute("data-popup", "open");
        }
    });
}
```



Beyond Beauty

beyond-beauty.co

Designer:
Louis Ansa louisansa.com

= BEYOND
BEAUTY

FOUR

REVOLUTION
OF DESIRE

Natalie Portman • The Revolution of Desire

TWO

“Four recognisable female faces unite for an inspiring endorsement of inner beauty, applying engaging multimedia design techniques”



• SOUND

Colours



Tools

GSAP, PixiJS

Fonts

**abcABC
1234567890**

EF Hawthorn is a font designed by Esselte Letraset and Mike Daines and used here for key headings and paragraphs.

**abcABC
1234567890**
The Messina Sans font by Luzi Gantenbein for Luzi Type styles the main story text in Light, Regular and SemiBold varieties.



We must do better

— 28.01.2018

Hi I'm Jen! I'm not here to tell you how to vote. I'm here because as corny as it sounds I love my country and I want to protect it. And there is a problem that is much bigger than political parties. It's a problem that is at the root of our political system and it's spreading like a rock through our democracy and it's called corruption. Our political system is the one that's supposed to work for us, the American people. It's broken and it affects basically every issue that you and I care about, health care, public education, the environment, you name it.

I love my country
and I want to protect it.



Above

Visitors drag an opening carousel to select between the stories of Natalie Portman, Emma Watson, Jennifer Lawrence and Cara Delevigne

Middle

Middle
Spoken word audio is played back in sync with text highlight animations to narrate the experience and give the content humanity.

Bottom

Bottom
By colouring the text to match the background while applying a thin stroke, certain words are made hollow to emphasise other phrases

Create an animated bubble background effect

Make empty space more appealing with this eye-catching animated effect

1. Document template

The first step is to define the template structure of the HTML document. This consists a HTML container for describing the document, consisting of sections for head and body. The head section is used to reference the JavaScript and CSS resources, while the body section is used in step 2 to store the visible content elements.

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

2. Body content

The page content consists of a 'main' container used for storing the effect. Visible text is placed inside the child article, allowing its content to avoid interference from any manipulation that will be applied via JavaScript and CSS. The use of the article as a container for content will also provide control for keeping all content to be displayed above the bubbles.

```
<main>
  <article>
    Hello
  </article>
</main>
```

3. Hide overflow

Create a new file called 'styles.css'. The first step is to define the presentation settings for the 'main' container that the effect will be applied to. A black background is applied, along with hidden overflow to cut out content not intended to be visible. Width and height are set relative to the browser window.

```
main{
  position: relative;
  display: block;
  width: 70vw;
  height: 50vh;
  overflow: hidden;
  background: black;
}
```

4. Main article

The article container inside 'main' is used to control the presentation of the foreground content. Absolute positioning is used to allow z-index positioning, which is set to 1 in order to be above the bubbles. To keep this tutorial short, a default font size and colour is applied for text to inherit.

```
main > article{
  position: absolute;
  font-size: 50vh;
  margin: 0;
  color: #777;
  z-index: 1;
}
```

5. Bubble elements

Bubble elements will be defined using the 'data-bubble' attribute. These will require absolute positioning in order to move – achieved through applying the 'move' animation. Each bubble element is placed out of view towards the left using the same space as its width. Applying a maximum border radius results in each bubble element appearing as a circle.

```
[data-bubble]{
  position: absolute;
  z-index: 0;
  left: -10vh;
  z-index: 0;
  height: 10vh;
  width: 10vh;
  background: #333;
  border-radius: 100%;
  animation: move 10s forwards infinite
  linear;
}
```

6. Applying the animation

Bubble movement speeds are modified using the 'nth-child' selector, which is used to alter the animation speed applied to groups of bubbles. The 'move' animation is defined using just the final 'to' frame, describing the co-ordinates that the bubbles should animate to. With starting frame in the animation, each bubble will animate from their default position.

```
[data-bubble]:nth-child(4n){ animation-
duration: 25s; }
[data-bubble]:nth-child(4n+1){ animation-
duration: 15s; }
[data-bubble]:nth-child(4n+2){ animation-
duration: 10s; }
[data-bubble]:nth-child(4n+3){ animation-
```

```
duration: 20s; }
@keyframes move {
  to { top: -20vh; left: 100%; }
}
```

7. Search

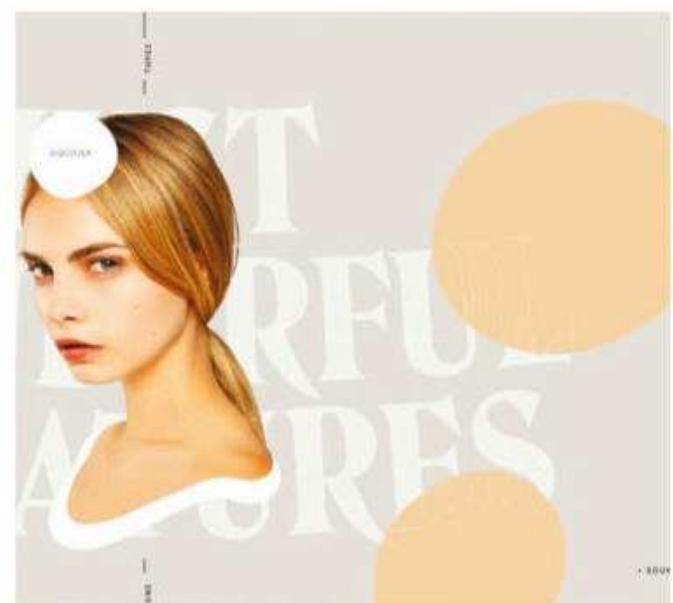
Create a new file called 'code.js'. The first step of JavaScript is to apply a 'load' event listener to the browser window that will trigger function. The function is used to modify the page's 'main' container – which wouldn't be available until after the page has loaded. The 'amount' variable is set to control how many bubbles are created using the following 'for' loop. Each bubble is created in step 8.

```
window.addEventListener("load",function(){
  var parent = document.querySelector("main");
  var amount = 10;
  for(var i=0; i<amount; i++)
  {
  }
});
```

8. Create bubble

Each bubble is created as a span element, with the 'data-bubble' attribute applied. Unique vertical positioning is achieved by setting the 'style.top' attribute with a random number up to the parent main height. After these updates, the new element is appended to the parent container using the 'parent' reference carefully defined in step 7.

```
var node = document.createElement("span");
node.setAttribute("data-bubble","");
node.style.top = (Math.random()*parent.
offsetHeight)+"px";
parent.appendChild(node);
```



DC Shoes | Lids The BlackOps Collection dcshoespromotions.com

DC | Lids
THEBLACKOPS
COLLECTION

ENTER TO WIN THE ULTIMATE TRIP
TO

ENTER
TO WIN



No purchase necessary to enter or win.

Participation is subject to the Official Rules.
A purchase will not increase your chances of winning.
All prizes subject to the terms and conditions of the [Official Rules](#).

Designer:
Ekkomedia ekkomedia.com

“ Shoe brand DC’s slick catalogue for its BlackOps collection marries video clip backgrounds with the chance to win an LA trip ”



Colours



Tools

Laravel (PHP MVC), Polyfill
IO, Vue.js, Instagram API

Fonts

abcABC
1234567890

Helvetica Neue font is employed almost exclusively by the site, using the font-weight property to apply multiple typefaces.

abcABC
1234567890
abcABC
1234567890

Helvetica Neue is also used in its Extended family form, appearing in both Thin and Black weightings.

How to code a circular expansion container

Enable content containers to act as navigation components by allowing them expand into view

1. Document initiation

Initiate the page using a description of the document. This consists of the HTML document container, which stores the `<head>` and `<body>` sections. While the `<head>` section is used to reference the required JavaScript and CSS resources, the `<body>` section will be used in the next step to store the page content.

2. Page content

The page content consists of two content containers using the 'data-expand' attribute. Each container has a 'data-control' to be used as an interaction button, along with a section container. Placing content inside the section container enables easy CSS rules to be created that can show or hide the content.

```
<article data-box="1" data-expand>
  <span data-control="expand"></span>
  <section>
    <h1>Hello</h1>
  </section>
</article>
<article data-box="2" data-expand="closed">
  <span data-control="expand"></span>
  <section>
    <h1>Hello</h1>
  </section>
</article>
```

3. Unique container styles

Create a new file called 'styles.css'. This step uses the unique values applied to the 'data-box' attribute of the containers to set a custom background. Box 1 has a background image that will cover the full size of the container, while box 2 has a plain colour for its background.

```
[data-box="1"]{
  background: url('background.jpg');
  background-size: cover;
}
[data-box="2"]{
  background: red;
}
```

4. Initial expand settings

Containers using the 'data-expand' attribute are initially set to appear using normal HTML content flow. Their sizes are set in proportion to the browser window using width and height, with 100% 'border-radius' to appear as a circle. The applied transition will ensure that any changes to these defined settings will appear animated over a duration of one second.

```
[data-expand]{
  position: relative;
  display: block;
  top: 0;
  left: 0;
  width: 10vw;
  height: 10vw;
  border-radius: 100%;
  transition: all 1s; }
```

5. Expand section

The section container inside the expansion element is set to be hidden by default using zero opacity. The default size is set, with central positioning within the container guaranteed with auto values applied to the left and right margins. The 'transition' attribute makes sure that all changes are animated over a duration of one second.

```
[data-expand] > section{
  opacity: 0;
  width: 50%;
  height: 50%;
  margin: 10% auto 0 auto;
  transition: all 1s; }
```

6. Control options

Settings for elements inside the expansion container that have the 'data-control' attribute are set to appear as a 1cm silver circle. A virtual 'before' element is used to provide CSS with control of content to display over the data control. This step also sets default content character as '+' for the 'data-control' element set as 'expand'.

```
[data-expand] > [data-control],
[data-control] > [data-control]:before{
  position: absolute;
  display: block;
  background: silver;
  color: #000;
  width: 1cm;
  height: 1cm;
  font-size: 1cm;
  text-align: center;
  border-radius: 100%; }
[data-expand] > [data-control="expand"]::before{
  content: "+"; }
```

7. Open status

The expansion container has different settings applied to it and its section, and 'data-control' elements when its value is set to 'open'. New size and co-ordinates applied

to the expansion container using fixed positioning guarantee its visibility regardless of page scroll location. Section opacity set to '1' will animate the content into view, while the 'data-control' element's content changes the icon symbol.

```
[data-expand="open"]{
  position: fixed;
  width: 50vw;
  height: 50vw;
  top: 20vh;
  left: 25vw;
}
[data-expand="open"] > section{
  opacity: 1;
}
[data-expand="open"] > [data-control="expand"]::before{
  content: "-"; }
```

8. JavaScript: Control search

Create a new file called 'code.js'. This step waits for the page to complete loading, upon which it will execute the JavaScript. The first step of the JavaScript is to search for the controls inside the expansion container – specifically those with the 'expand' value. A 'for' loop is used to apply Step 9 to each item found.

```
window.addEventListener("load", function(){
  var nodes = document.querySelectorAll('[data-expand][data-control="expand"]');
  for(var i=0; i<nodes.length; i++){
    *** STEP 9 HERE
  } });
```

9. JavaScript: Click event

Each of the elements found from Step 8 has a 'click' event listener applied – triggering a function to be executed whenever the user clicks or taps on the item. This function will toggle the value of the 'data-expand' attribute between 'open' and 'closed', which in turn triggers visual changes defined within the CSS.

```
nodes[i].addEventListener("click", function(){
  var parent = this.parentNode;
  if(parent.getAttribute("data-expand") != "open")
    parent.setAttribute("data-expand", "open");
  else
    parent.setAttribute("data-expand", "closed");
});
```

generate

The conference for web designers

LONDON

19-21 SEPTEMBER 2018

3 DAYS OF INSIGHT AND INSPIRATION

TICKETS ON SALE NOW

www.generateconf.com
#generateconf

BROUGHT TO YOU BY

web
designer CB CREATIVE BLOQ net

THE LOCKDOWN

thelockdown.com

iOS: <https://apple.co/2wCZmyW>

Android: <https://play.google.com/store/apps/details?id=com.abnamro.thelockdown>

AGENCY

CODE D'AZUR

codedazur.com

@codedazur

PROJECT DURATION

15 weeks

PEOPLE INVOLVED

20

INTERVIEWEES

JEROEN THISSEN

Creative Director

PRISCILLA DE GIER

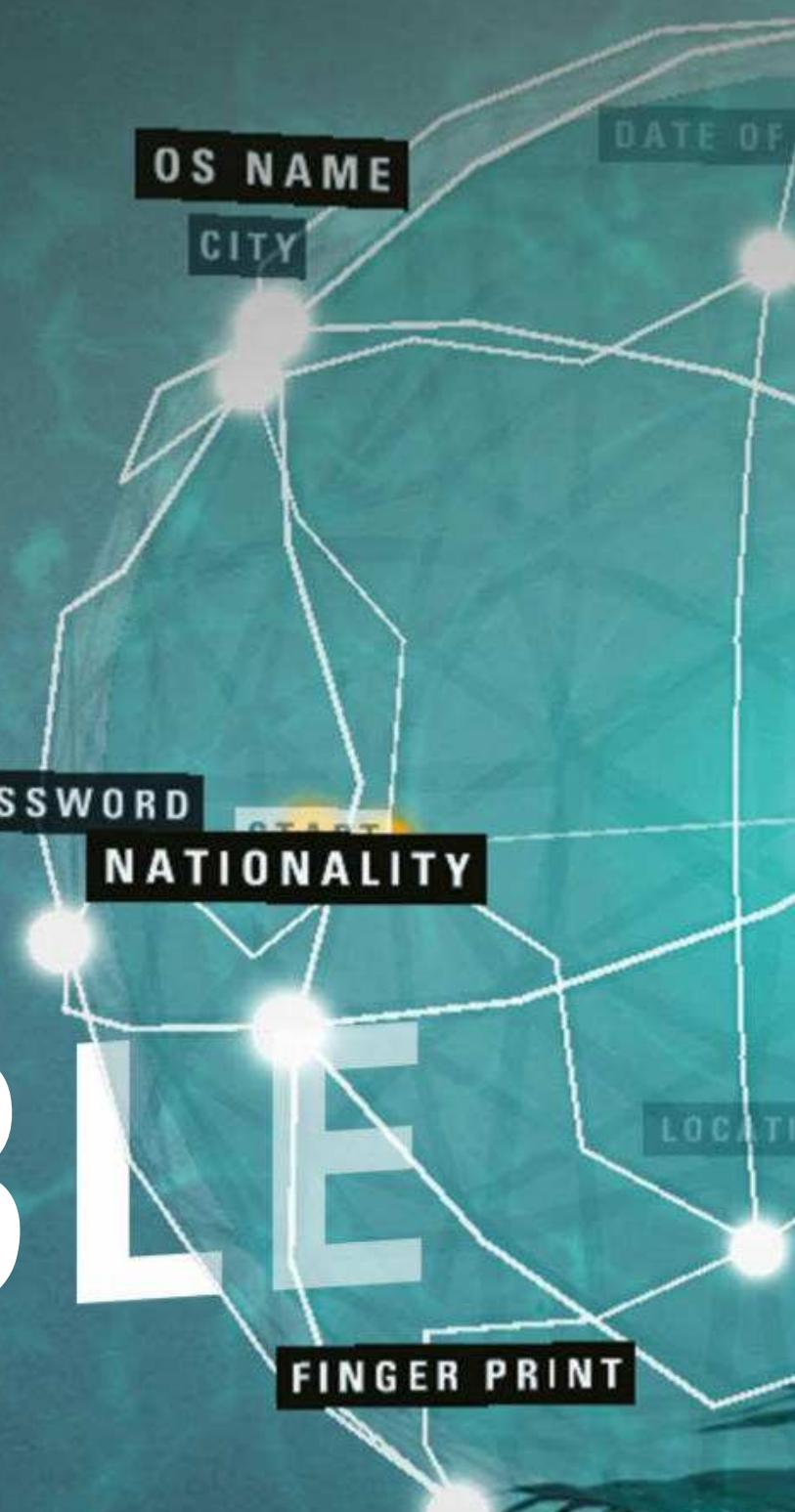
Art & Design Director

ERIK RAVE

Technology Director

BANKABLE ASSETS

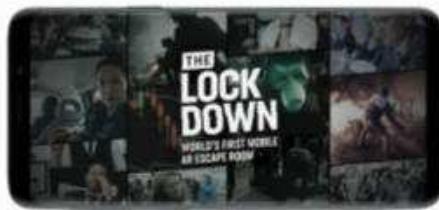
AS PART OF DUTCH BANK ABN AMRO'S QUEST TO RECRUIT THE HOTTEST IT TALENT, WE REVEAL HOW AGENCY CODE D'AZUR CONSTRUCTED AN AR ESCAPE ROOM EXPERIENCE WITH A TECHY TWIST





A S THE CONNECTED WORLD CONTINUES TO PROGRESS AT LIGHT SPEED, THE BANKING

sector is a critical arena for innovation. This is, of course, largely reliant on the attraction of new talent into the industry and the attraction of the brightest IT prospects from each generation. The inherent challenge then is convincing these kinds of fledgling technologists that “traditional” banks are modernising enough to excite them, while providing a worthy platform for their skills. So when Dutch bank ABN AMRO was facing a huge recruitment drive for the brightest IT minds, it needed something special. In a campaign to find over 400 of such IT and digital professionals, it decided to eschew posting the normal job ads for something more ambitious and progressive, resulting in a partnership with creative agency CODE D'AZUR and an incredible Augmented Reality (AR) experience. Billed as the world's first mobile AR escape room, The Lockdown is an immersive online game that uses ARCore and ARKit on Android and iOS platforms to render an inspiring storytelling adventure. Set in 2028, money has been wholly digitised and secured on the blockchain only for the system to crash and call on the players' IT prowess to fix the leak over a series of challenges. “Blockchain was considered innovation technology a few years back, but now the bank already employs it for several processes,” begins Creative Director Jeroen Thissen. “For that technology, as well as other emerging tech, it needs specialists, but also developers and data analysts. The brief was, therefore, to create a recruitment campaign that stands out, and is compelling, while fitting the interests and frame of reference of the target group.” Thanks to intense collaboration between ABN AMRO and CODE D'AZUR with its partners, The Lockdown was created in under just 15 weeks and by speaking to the team involved we discover how. ►



SUCCESSFUL SUBTLETY

One of The Lockdown's boldest strategies was the willingness from the client to not make it overtly branded. From the outset ABN AMRO appreciated a goal to avoid cynical marketing and obvious promotion, in favour of a more subtle approach. "Again, the idea was to serve the objective for positively changing the player's perception of the bank, without the bank being too much in their face," explains Design Director Priscilla de Gier. "Branding might cause people's original perception of the bank to be hesitant to play the game." This foresight, and also a commitment for not talking down to a tech-savvy target audience, could be cited as reasons for the game's subsequent success since launch. Some of the impressive numbers here include a 73 percent target audience reach in the Netherlands of 311k, downloads exceeding 10k across iOS and Android, plus over 10,641 unique visits to ABN AMRO's recruitment website. With reviews of the app averaging 4.4 out of 5 stars, a whopping 42k minutes were spent playing the game over a six-week period. "The story was quickly picked up by the IT community and media, showing to IT professionals in The Netherlands and the rest of the globe that ABN AMRO is an innovative and challenging IT employer," confirms Creative Director Jeroen Thissen. "The story featured in Dutch leading tech platform Bright.nl as well as media like VR Focus, Android Police, Contagious, The Next Web, Computable and iCulture." Not to mention Web Designer Jeroen!

PARTNERS AND PERCEPTIONS

Based in Amsterdam and Barcelona, CODE D'AZUR are an agency that creates digital services and campaigns for global brands such as KLM, Red Bull and Tele2. In addition, a long-standing working partnership with ABN AMRO bank suggested a trust between both parties for tackling this latest project. CODE D'AZUR's mantra of "Stand Out. Fit In." seemed immediately ideal for something so innovative, while inevitably requiring additional expertise to pull off. "It was obvious that a project like this, being the first of its kind, needed the combined knowledge and skills of a great many professionals and areas of expertise," continues Thissen. "We involved the experts of Sherlocked, for example. They are the team behind some of the most successful and award-winning escape rooms, with two of them being here in Amsterdam." To undertake this project in such a short period of time, the agency would need to work together closely with the client and these other partners, opting for an agile process. This enabled ABN AMRO to be very involved in every step of the project. "Specifically this meant that we worked closely with the bank to tackle the themes, information in the challenges and the methodology of them. Being a bank that produces a game that is specifically targeted towards IT and digital professionals, the facts and challenges that you present them with need to be correct as well as realistic." A common discussion throughout would be how recognisable the bank's brand will be, how prominently it should preside. Being such an open client, ABN AMRO recognised that for the game to most effectively change the perception of the bank as an employer, its brand shouldn't be overly present in the game. "This was translated to where the game would feature the bank's iconic green colours, but

VIDEO FRAGMENT

RECALIBRATE AR

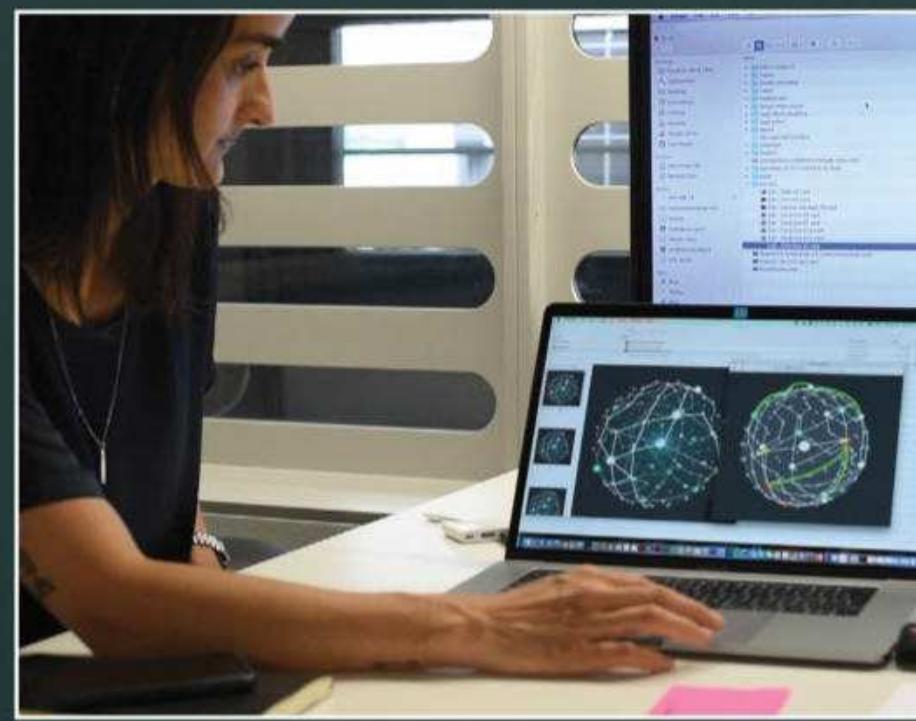


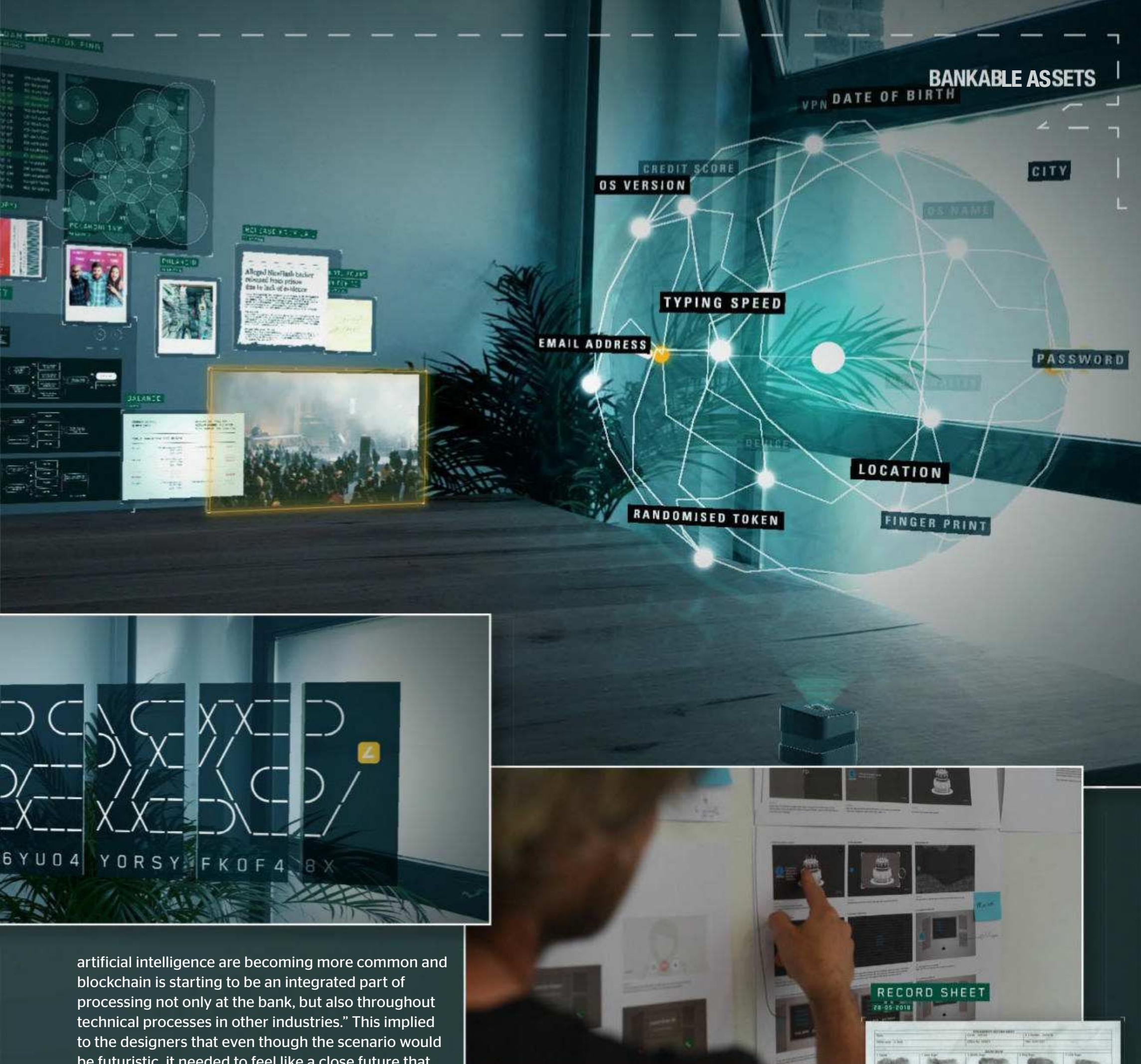
"EVEN THOUGH THE SCENARIO WOULD BE FUTURISTIC, IT NEEDED TO FEEL LIKE A CLOSE FUTURE"

nothing more. The only reference to the bank is featured at the very end of the game, when it's completed, to invite players to put their skills to work and apply for a job."

COMBINED EXPERIENCE

As an initial concept, the escape room idea had been something the client had used in previous recruitment campaigns. They had always been physical locations however, so the amount of engagement was limited by the amount of people that could be accommodated. The idea for The Lockdown was to introduce themes like blockchain, cryptocurrency, artificial intelligence and infosecurity into a virtual AR escape room open to all. "We wanted to create a scenario for this game that allowed players to dive into challenges that the bank faces today and will face in the close future. Challenges in infosecurity and





artificial intelligence are becoming more common and blockchain is starting to be an integrated part of processing not only at the bank, but also throughout technical processes in other industries.” This implied to the designers that even though the scenario would be futuristic, it needed to feel like a close future that could conceivably take place a decade from now. “In terms of how the game should look and feel, we immediately knew that AR changes the playing field completely when it comes to user experience,” explains Design Director Priscilla de Gier.

“Experiencing something ‘flat’ or 2D, can’t be compared to an experience where you can walk up or even through elements that are in front and around of you. In AR there are so many more and other ways to solve challenges.” Indeed, this distinct difference in user experience meant that from the outset the team had to approach design, technology, development and creation cohesively to ensure user flow was right. “We’ve wanted to use the AR technology to its full capacity. The AR user experience was not only integral to thinking about how the challenges were solved but also to details about how the player would enter their answers in the system.”

PIXEL PERFECTION

When it came to the visual and graphic design work, efforts were purposefully steered away from giving the game too much of a sci-fi feel. Favouring a more realistic futuristic setting that would be easy for players to dive into, the team looked to references such as TV series Black Mirror for inspiration around what 2030 might look like. “It’s been important to match all art direction of the videos, 3D and UI, to bring this vision to life,” Priscilla continues. “The many mood boards at the beginning of the project were key in crystallizing the visual style of the game. For the optimal experience and the style to be ultimately realistic, we’ve worked closely with A.I. and blockchain experts to ensure every detail was where it’s supposed to be. All visual details in the game are correct. There is no bogus code or details in there”



EYE CLOSE-UP

28-05-2018

1. IRIS COLOR

6. SCLERA COLOR

THE LOCK DOWN

WORLD'S FIRST MOBILE
AR ESCAPE ROOM



A WORTHY PLATFORM

The Lockdown was always designed to be the main focal point of client ABN AMRO's recruitment campaign. This would naturally be supported by other supplementary marketing efforts alongside, all geared to driving traffic toward the game and the message at its heart. "The game itself is ultimately about raising awareness of the bank being much more of a challenging IT employer than players would expect," underlines Creative Director Jeroen Thissen. "A place where talent can contribute to the bank's and societies challenges regarding artificial intelligence, infosecurity, blockchain and cryptocurrency. So it's been especially great to see the game taking off so well within the target market, and even see game reviews asking for a second part of the game. Plus it has also proven to be a great example of what can be done with AR to create an immersive and engaging brand experience. Perhaps even specifically proving that AR, which such high engagement potential, is the ideal technology to shape the way your potential customers or recruits see your brand."

that wouldn't work in real life." This approach would also acknowledge the decision to keep branding subdued and passive, using subtle styling like ABN AMRO's corporate typography and colour palettes across certain assets. "The starting point was to simulate game elements analogously, testing what an AR rendering of the feature would look like displayed in the real world. This meant a lot of back and forth between the digital and the analogue world to ensure the high level of realism was achieved. The game is built in Unity, which also meant we had to push the boundaries to ensure the game was realistic, looked beautiful, but these choices were not detracting from the performance of the game." The close collaboration between the in-house design team and the development team has meant that lines were incredibly short and conducive to maintaining a high quality in visual result. "The design team has had control in ensuring the design would be executed properly, by actually making the designs themselves in Unity. This way the developers had all feedback and continuous support, and together created pixel-perfect execution."

ENGINEERING AUTHENTICITY

In terms of the development side, this was always a project where development considerations had to run right through it. The awareness for the technology was hugely important therefore, aided by the recent focus on AR platform support from Google and Apple. With ARCore and ARKit respectively showing major advancements, the overall technical challenge was made fairly smooth sailing. "However, one of the challenges that we did encounter was a memory issue with all the high resolution textures we had to include in the holodesk and 'info-wall' in the game," as Technology Director Erik Rave concedes. "We wanted players to not only view the many documents, pictures and other key items from far away, but also be able to zoom in. This meant that all assets had to be high resolution, but this would not be able with the memory. What we then decided is to make certain items clickable and then expand for the player. Most items are still very high resolution, to maintain the level of realism, of actually being able to walk up to your info wall and have a close look at what's up there. With this solution we've found a way to adjust the UI to best fit both the design and the concept goals, as well as to elevate the usability of the game." That pursuit of high-resolution realism also translated into the challenge of making the world authentic. Often a gripe of Hollywood's portrayal of technology is those computer screens that apparently a hacker is using to hack something, only for the display to look blatantly false or fake. Since IT and digital professionals were the target group of the game, the team knew it needed to avoid this at all cost. "What ties into that is the additional contribution of the technology team in the shape and difficulty level of the puzzles. The team is the target audience, so it's been great to test and play around with the development of the challenges. This, in combination with help from the experts from



SITE HIGHLIGHT

Technology Director Erik Rave gives us his instant response to what it is about the project that makes him and the rest of the crew most proud and why.

"I'm very proud of the realism we achieved in the overall experience and look of the game. Though it's set a decade from now, it doesn't feel overly sci-fi and truly feels like any space you're in is transformed to your own escape room"

"SCREENS WENT BLACK AND OUR HACKER DELIVERED A SPECIAL MESSAGE TO THE AUDIENCE"

the bank in creating realistic scenarios that people encounter there, as well as escape-room experts, has been key in successfully creating challenging and engaging puzzles throughout the game."

NUMBER CRUNCHING

The launch process was suitably geared to meet the game's target audience, too. Unveiled during one of Europe's leading technology conferences, The Next Web, it simultaneously had its landing page, banner ads and even cinema trailers go live before showings of the new Solo Star Wars movie.

"During the opening of the conference, with 3,000 people in the audience, we hacked the conference," laughs Thissen. "Screens went black and our hacker delivered a special message to the audience that challenged them to play the game. This was supported by a stand at the conference where people could not only chat to the bank, but also play the game on sample phones. It was great to see that we had a few bright minds actually solve the complete

game under 30 minutes, naturally, they were invited for a coffee." The weekend after this launch event yielded thousands of downloads and very quickly saw the story get picked up by the IT community and media. Spreading the desired message at home and internationally, a total reach of 311,000 was achieved within the Netherlands — in other words, around 73 percent of the working IT talent in the Netherlands had been reached solely with the earned media of the campaign. So with such promising engagement figures to date, do the CODE D'AZUR guys anticipate more to come? "Since we've left it open ended, who knows what will lay in store for The Lockdown," grins Thissen, cryptically. "With the development of AR now by Apple and Google, we expect so much more to be possible in future. Unlike VR, AR has a very low participation threshold, you aren't closing yourself off to your environment, and actually can share your experience with more people than yourself. It wouldn't be surprising if this and next year, we'll come across many more AR experiences." ■



SUBSCRIBE

TODAY AND RECEIVE AN ECOFFEE CUP



- Each cup is made out of bamboo fibre - the world's fastest growing, most sustainable crop
- Each cup is BPA and phthalate free
- A re-sealable no-drip lid means it can be taken everywhere you go

IT'S EASY TO SUBSCRIBE
myfavouritemagazines.co.uk/WDRCUP

OR CALL: 0344 848 2852 AND QUOTE WDRCUP

SUBSCRIBING FROM OUTSIDE THE UK?
HEAD TO [BIT.LY/2UREG5D](https://bit.ly/2UREG5D) FOR UP TO 15% OFF WORLDWIDE



PLUS with your subscription
you have access to
900+ FREE RESOURCES
from previous issues
with FileSilo. See page 96

CHOOSE YOUR PACKAGE!*

PRINT EDITION ONLY



3 months of **Web Designer** in
print, plus **ecoffee cup**

DIGITAL EDITION



3 months of **Web Designer** on
your iOS or Android device

PRINT & DIGITAL BUNDLE



3 months of **Web Designer**
in print and digital,
plus **ecoffee cup**

* OFFER NOT AVAILABLE FOR DIGITAL EDITION ONLY PACKAGE

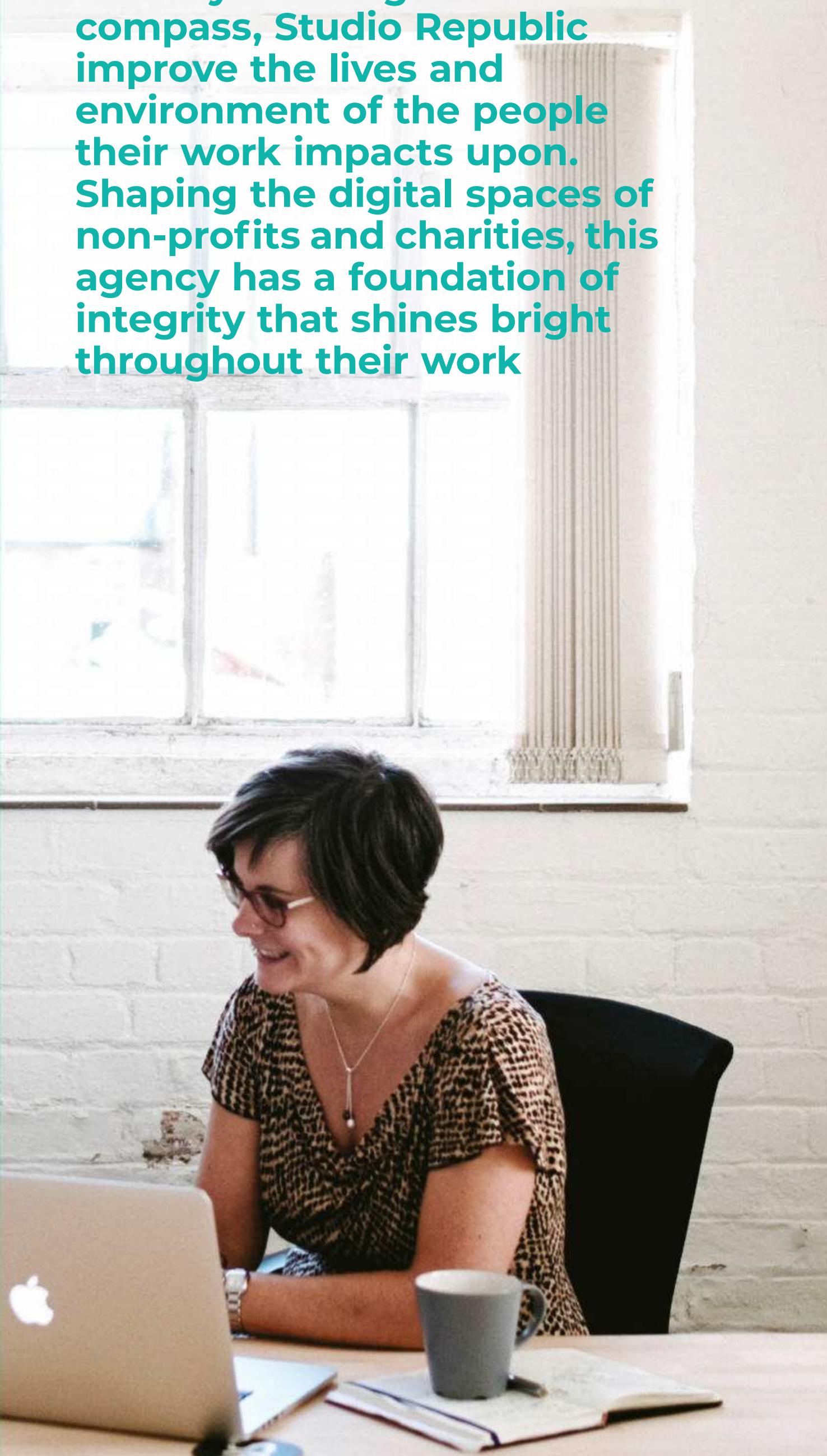
Terms and conditions: This offer entitles new UK Direct Debit subscribers to pay just £17 every 3 months plus receive a branded reusable coffee cup worth £8.95. Gift is only available for new UK subscribers. Gift is subject to availability. Please allow up to 60 days for the delivery of your gift. In the event of stocks being exhausted we reserve the right to replace with items of similar value. Prices and savings quoted are compared to buying full-priced print issues. You will receive 13 issues in a year. Your subscription is for the minimum term specified and will expire at the end of the current term. You can write to us or call us to cancel your subscription within 14 days of purchase. 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. UK calls will cost the same as other standard fixed line numbers (starting 01 or 02) or are included as part of any inclusive or free minutes allowances (if offered by your phone tariff). For full terms and conditions please visit: www.bit.ly/magterms. Offer ends November 2018.

Ethical Design



Jack De Wolf (Client Director) and Dee Russell (Brand and Marketing Lead) putting their heads together for an upcoming project pitch.

Lead by a strong ethical compass, Studio Republic improve the lives and environment of the people their work impacts upon. Shaping the digital spaces of non-profits and charities, this agency has a foundation of integrity that shines bright throughout their work



Who
Studio Republic

What
Web design, Website development, MODX CMS development, Branding and corporate identity, Graphic design, Charity branding and campaigns, and Advertising

Where
10 Charlecote Mews, Winchester, Hampshire, SO23 8SR

Web
studiorepublic.com

Key Clients
ismybillfair.com
The Green Party
Ordnance Survey
Chartered Institution of Water and Environmental Management (CIWEM)
Sailors' Society

IN 2003, STUDIO REPUBLIC, LIKE MANY DESIGN AGENCIES, BEGAN IN A SPARE ROOM. Chris Todhunter, the Digital Director and the Founder of Studio Republic, originally created the business as a digital freelance resource for local agencies. However, Chris's journey to that point wasn't all plain sailing.

Dropping out of his Software Engineering Management course at Bournemouth University, Chris spent his spare time touring as a drummer with various bands, playing gigs across the country – and even having the opportunity to play at Glastonbury. It wasn't until his meticulous attention to detail scored him a job as an artwork that his path in the creative field started to take hold.

When the digital era started, Chris found his true passion – web development. Excited by the new advancements in the tech world, he spent his evenings teaching himself how to code. As the digital industry grew, Chris's talents grew alongside it. Chris's flair for tech generated enough business for him to start working as a freelance developer and, before long, he had so much work he started thinking about expansion, so Studio Republic was born.

"For the next eleven years, Studio Republic worked to provide its clients with high quality, ahead-of-the-curve digital services – doing such a good job that some of our very first clients still work with us today," Chris explains. "But it wasn't until 2014, when Jack De Wolf joined Studio Republic as Client Services Manager, that the business started to become what it is today."

Jack De Wolf graduated from university with a first class BA Hons in Design and went ▶

Ethical Design



"As well as keeping up-to-date with new technologies, we're keen to roll up our sleeves and develop new tools ourselves. In an effort to balance our collaborative approach with clients and get rid of any bottlenecks in our planning, designing, building, managing and training processes, we've started using a new system that we've designed and built as a plugin for our MODX CMS systems"

Chris Todhunter
Digital Director and Founder

straight into the industry to work as a designer for a London agency. Not enjoying London life, and realising that working as a designer wasn't the right fit for him, he spent the next few years travelling the world.

Upon his return to the UK, Jack worked in a variety of industries – from recruitment to footwear – trying out various roles before realising that he still had a love for the creative sector, but wanted to take on more of a client-facing role. In a Business Development Management role for a company on the South Coast, he realised he'd found the right fit.

"When Jack joined the team here at Studio Republic, he took our client list and grew it significantly," says Chris. "Jack and I worked hard to realise the potential of Studio Republic as a full service, digital creative marketing agency specialising in the charity, not-for-profit and ethical arena. In 2017, when Jack became Partner, Studio Republic really took off as an ethical creative digital agency. Since then, the agency has gone from strength to strength. Our team has grown significantly, and we've never been happier doing the work we love, for the people who make the world a better place."

Chris explains how he named the business: "It's always the way when you think of a great business name that someone has already purchased the domain name. Competition wasn't as fierce back then as it is now, but out of the long list of 20 or so name options, half had been taken. To me, Studio Republic best

Halina Myers (Designer) and Harley Southwick (Lead Designer) at work on their latest project



summed up the independent nature of the business, and the rest is history."

Agencies approach their websites as calling cards and often as a first point of contact for potential clients. As Chris explains, keeping their own site up-to-date is often a challenge: "Our clients' websites always take priority, which leaves very little time for our own site and has often pushed our own marketing to the back of the queue.

"In the past, we've been pretty slack when it came to updating our own site, but now, with our new focus and larger team, we ensure that we regularly update the website – we're going

Russell, our Marketing Lead, has invested into memberships, events, awards, SEO, PPC, etc., in order to attract clients in the charity, not-for-profit and ethical business sectors.

"We don't creatively pitch for work, as anything you present before getting to know your client and their business needs would just be a 'pretty picture'. Instead, we lead each pitch with our tailor-made strategy and impressive credentials."

Having the ability to choose which clients they take on hasn't limited the breadth of accounts that Studio Republic have worked on, as Chris explains: "We're fortunate to choose

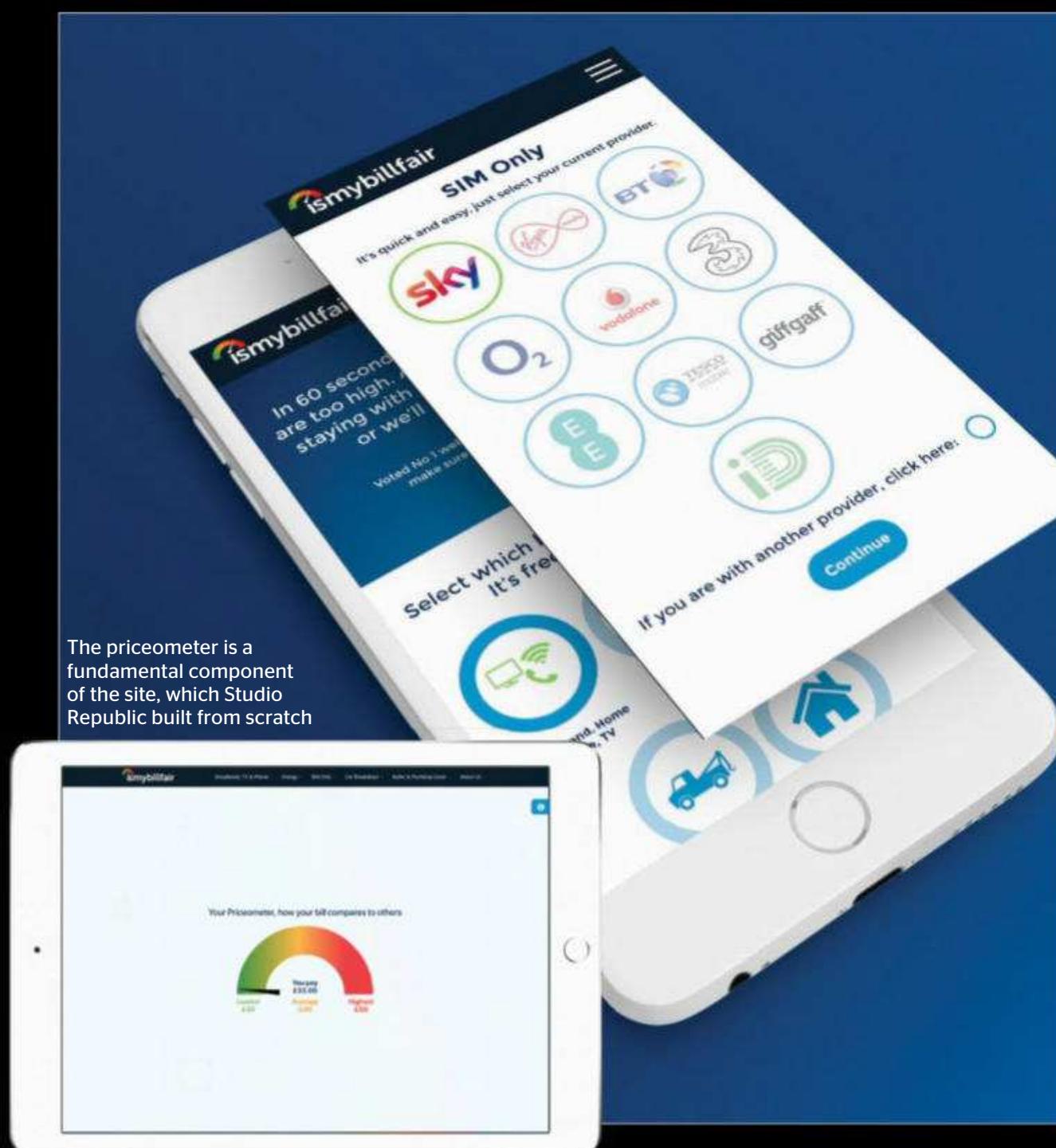
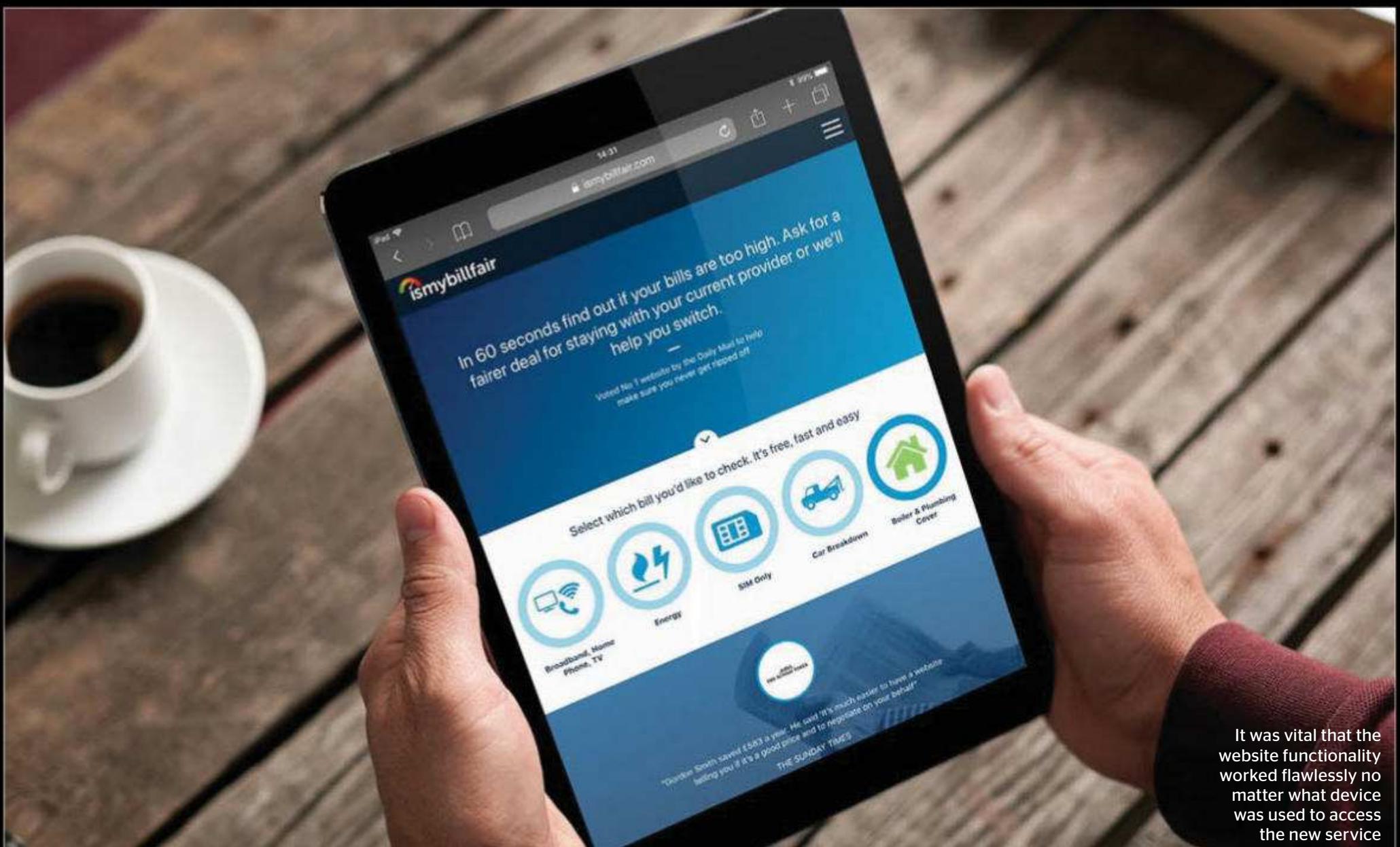
"When Jack became Partner, Studio Republic really took off as an ethical creative digital agency"

through another re-fresh as we speak. Our website should reflect who we are and what we can do, and that takes work. We treat ourselves as another one of our clients: we set a budget and assign time each month for that well-deserved TLC."

The channels that agencies use to gain their clients can often be diverse. "Over the years, we've been very lucky to receive the majority of our clients through referral," Chris explains. "Each new project we launch acts as Studio Republic's new ad campaign. This year, however, we've mixed things up, giving ourselves a marketing budget, which Dee

the projects and clients we get involved with. We live by our values and are privileged to work with a wide range of clients who share our outlook."

Chris also shares: "There have been a few occasions where we have politely declined working for businesses simply because they don't share our ethos. It's not so much the size of a project that we decline, more the values of the people we'd be working with, or uncertainty that we can bring something to the party. We will only ever work on a project if we are certain that we can add value. We're not an app, PR or media-buying agency, and we



ismybillfair

ismybillfair.com

Each client requires their own targeted set of services. For ismybillfair, they needed innovation. Ismybillfair came to Studio Republic with a new business that was set to disrupt the price comparison market. There is significant price disparity for UK customers of utility and service providers, from energy to broadband.

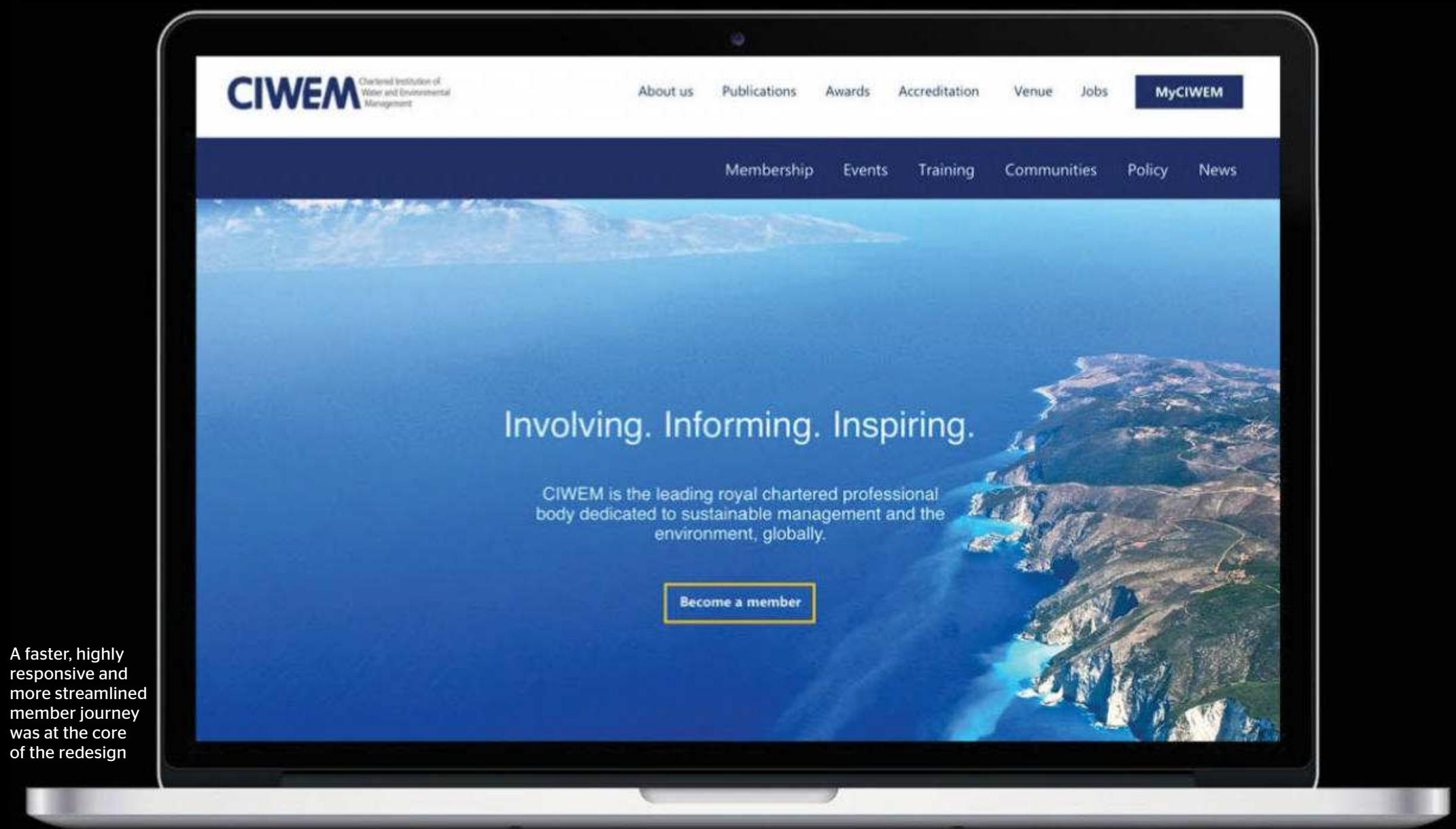
The founders saw this unfairness and wanted to challenge it, supporting people to remain with their providers whilst still getting the fairest price. The task was to brand the business (which only had a working title at the time) and build the UK's first price transparency site offering users the opportunity to stay with their providers for a better deal.

To achieve the client's goals, the team needed to innovate and utilise new technologies as well as create a highly tailored site that would match the vision for ismybillfair. Everything was designed and developed bespoke for them.

Studio Republic's designers worked closely with the development team to carefully create and refine a bold, user-friendly site. The central journeys of the site – the pages which lead users to the 'priceometer' that tells them how much other customers with the same services are paying – were carefully designed and developed with ease of use in mind and were rigorously tested.

The resulting site guides a user through a series of questions about their usage of various services (such as broadband, energy or mobile phones) and then presents them with an easy-to-understand priceometer, which shows whether what they are paying is high, low or average compared to other users with the same levels of service. Users are then offered a chance to ask their supplier for a better deal, or to see a range of alternative suppliers if they wish to switch.

Ethical Design



A faster, highly responsive and more streamlined member journey was at the core of the redesign

CIWEM

Chartered Institution of Water and Environmental Management
ciwem.org

At the end of last year, the Chartered Institution of Water and Environmental Management (CIWEM) approached Studio Republic to pitch for their website redesign, and the agency jumped at the opportunity. Since refocusing their efforts at the end of 2014 to work only with clients that share their values, Studio Republic have wanted to work with an organisation that's prioritising the environment.

Since winning the pitch, Studio Republic have worked very closely with CIWEM, getting to know them and their organisation inside out, in order to pinpoint and understand the issues CIWEM's website presented to their users as well as revealing opportunities within their business that could be developed or improved.

As an international organisation with members in 89 countries and hopes to expand their member numbers, CIWEM required a site that drove membership and a simplified member journey, which was more user-friendly. Their original membership sign-up and journey were lengthy and complicated, and the Studio worked very closely with the team at CIWEM to streamline this process to make the user experience as easy and hassle-free as possible.

CIWEM's new site shows a simplified membership journey of just three steps (reduced from the original seven) and also enables users to save their application midway through and return at a later date to complete the process.

It was vital to understand the business in order to develop a new site that supported its membership.

would never agree to take on services that are not our core specialism, but there are no digital projects that we can't deliver."

As an agency that designs on an ethical basis, ***Web Designer*** asked Chris if there was one project that really encapsulated the ethos of his business: "The Green Party website would be one of the projects that best defines our purpose, capabilities and scalability," Chris added. "They are the epitome of ethical and we really align with their values. The website we designed and built for them was independently ranked the winner of the 2017 general election; according to CBR, 'The Green Party is the winner of the General Election, at least when it comes to digital performance'."

"From one single installation of the MODX CMS we built the main website and 180 local microsites through a control panel that gives 200 users granular access levels. In the four weeks leading up to the General Election, the site received a huge amount of visibility achieving 823,000 visits."

Tools? Chris explains which tools Studio Republic favour for their work: "As with most agencies in the industry, we use a lot of Adobe software but, over the last few years, the market has become a lot more competitive, with a lot of other third-party software being launched – obviously an amazing thing for everyone in the industry!"

We use both Sketch and InVision for designing and prototyping on all of our digital projects. InVision seems to be going from strength to strength, with new tools, product launches and even documentaries coming out

all the time. They're a great resource for designers. Although we love the latest software, sometimes you can't rule out good old-fashioned pen and paper. We tend to start a lot of our wireframes off on Post-it notes or in our notepads as it's such a quick way of prioritising information and working things out."

Chris continues: "HTML5, CSS3 and JS are constantly evolving to provide us with new and interesting techniques to use on websites. Being able to utilise technologies such as CSS grid, WebGL, SVG animations and even HTML canvas elements creates unlimited possibilities. If we could have seen the way we build our websites now a few years back, our minds would have been blown. In such a fast-moving arena, it's hard to predict where web technology will go – even in just a few years! What we do know, however, is that wherever it goes, and wherever it takes us, we will always look for the most innovative and creative applications.

"As primarily MODX developers, we are always looking for new plugins and features for that framework. New features and extras are constantly being created and improved and, as we're very active within the MODX community, we often suggest updates for the creators to implement.

"And as well as keeping up-to-date with new technologies, we're keen to roll up our sleeves and develop new tools ourselves. In an effort to balance our collaborative approach with clients and get rid of any bottlenecks in our planning, designing, building, managing and training processes, we've started using a new system ▶

"HTML5, CSS3 and JS are constantly evolving to provide us with new and interesting techniques"



Timeline

2003

After a stint as a drummer, Chris Todhunter decides development is his future and begins freelancing.

Employees: 1

2004

Understanding that his business needs additional skills, Studio Republic hires a designer.

Employees: 2

2005

Studio Republic move into a Winchester office to be closer to some bigger clients.

Employees: 2

2010

In five years Studio Republic have outgrown their original office, moving into their new and current offices.

Employees: 2

2012

A major milestone is reached for the ethical business with The Green Party becoming a client.

Employees: 3

2014

A major step forward with the development of Studio Republic as a business is taken when Jack De Wolf joins the studio as Client Services Manager.

Employees: 4

2015

Lead Designer Harley Southwick joins the team at Studio Republic.

Employees: 5

2017

Studio Republic celebrate their 14th birthday.

Employees: 8

2018

The team at Studio Republic continues to expand to meet the demands of their expanding client roster.

Employees: 10

Agency breakdown

1x Head of Digital

3x Developers

2x Designers

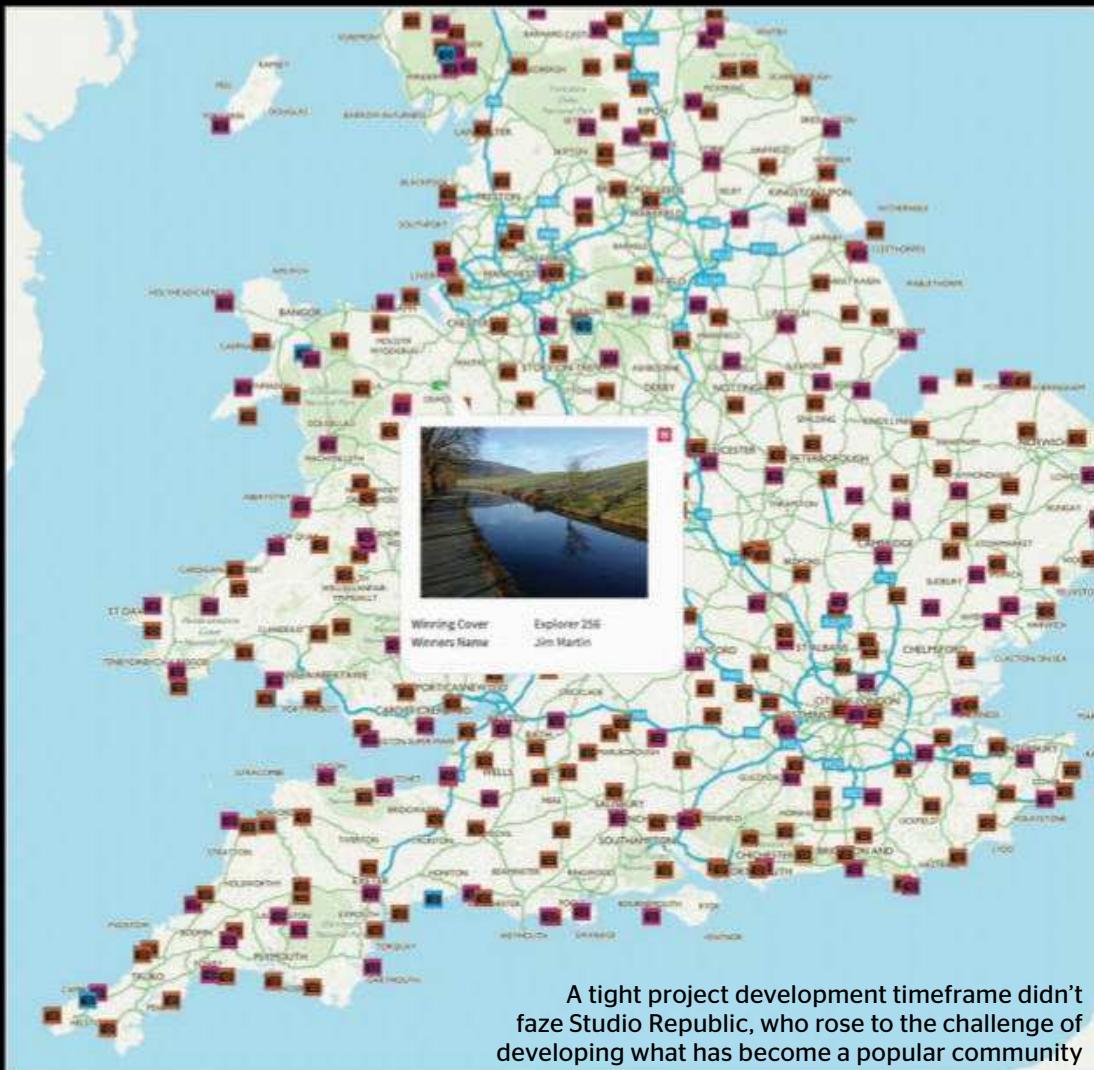
1x Digital Strategist

1x Marketing Lead

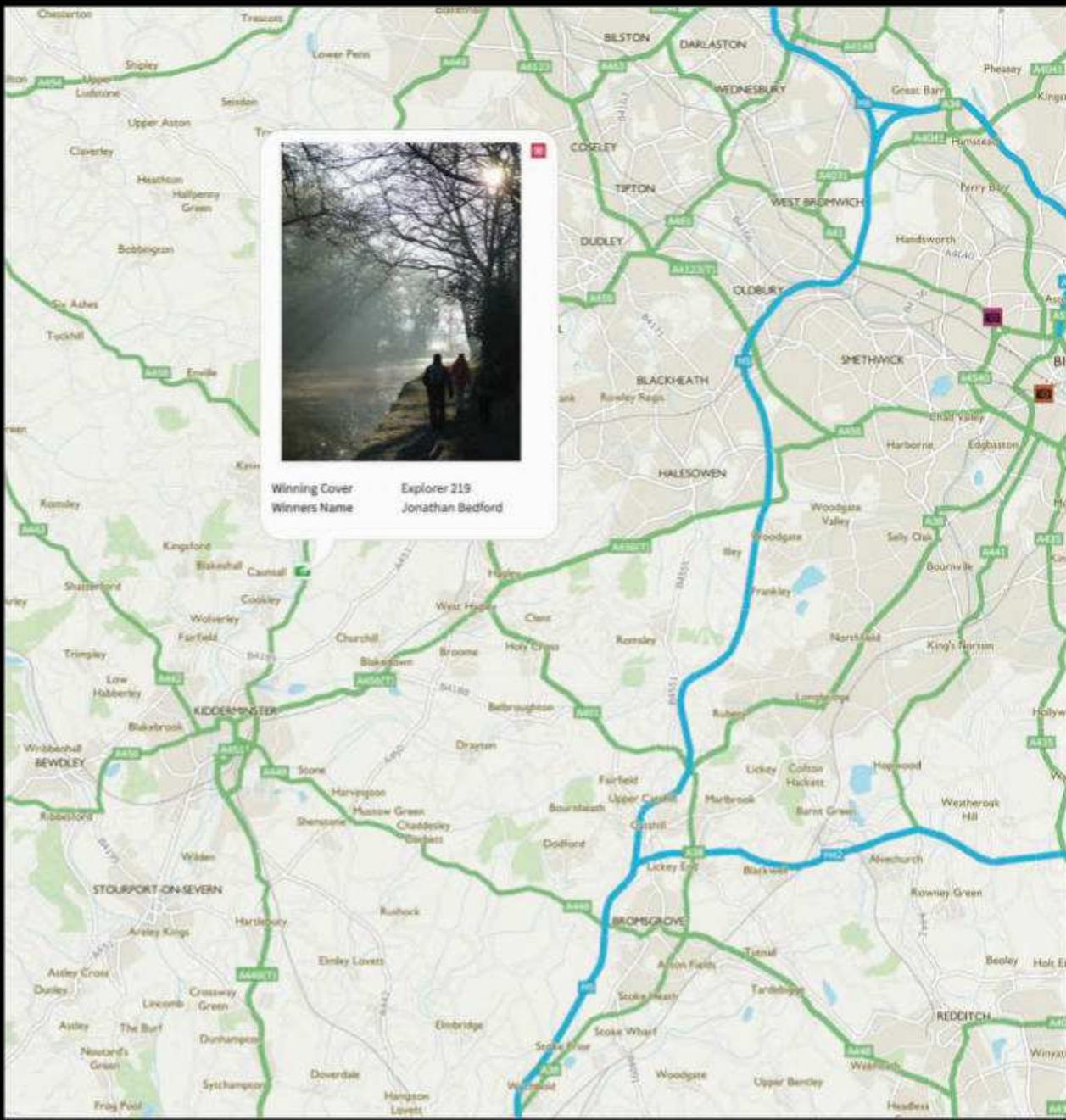
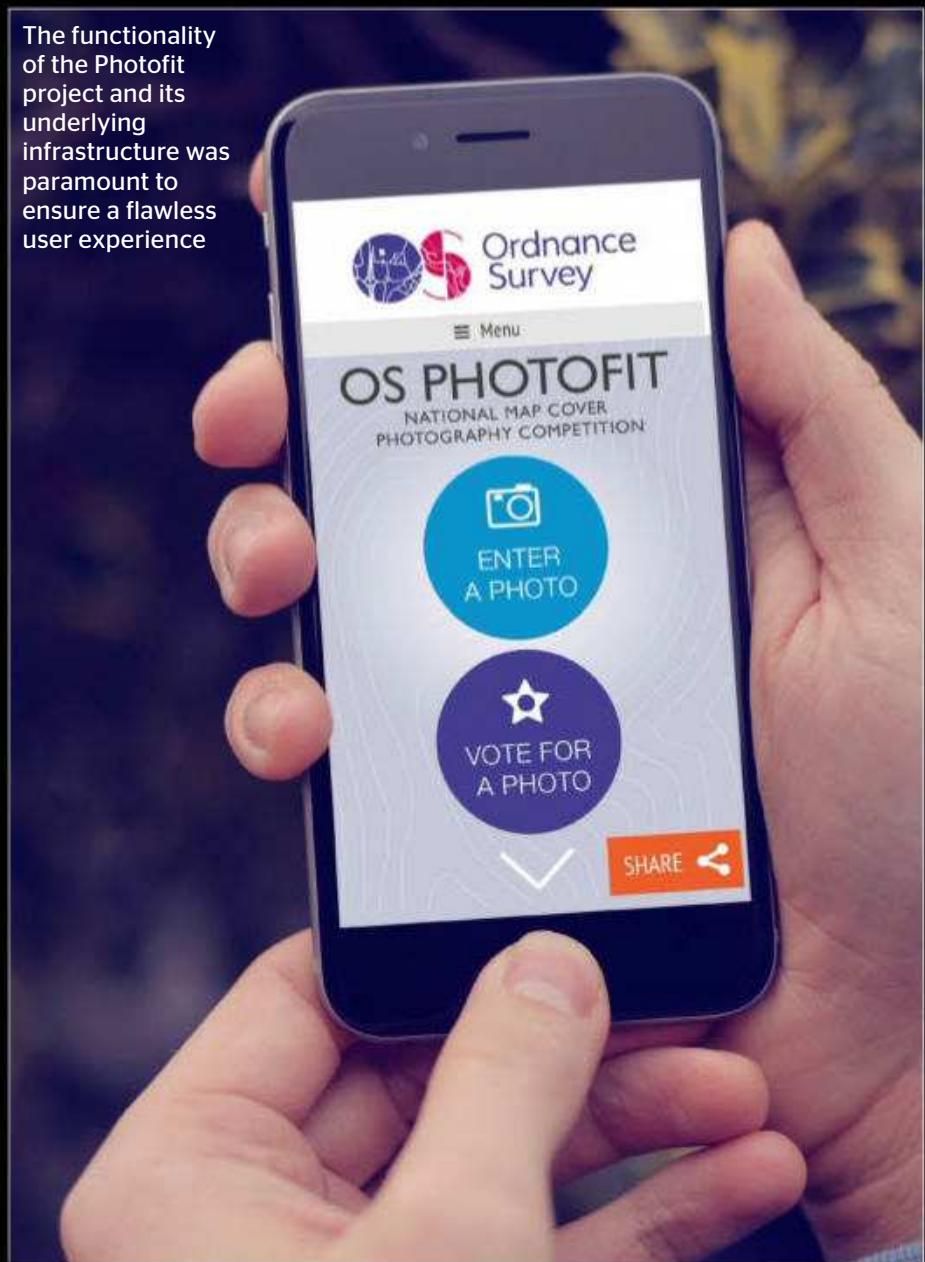
1x Content Producer

1x Client Services

Ethical Design



The functionality of the Photofit project and its underlying infrastructure was paramount to ensure a flawless user experience



Visitors are able to zoom into the incredibly detailed maps that OS is famous for in order to locate Photofit winners from any part of the country

Ordnance Survey Photofit

ordnancesurvey.co.uk/blog/tag/os-photofit

Studio Republic been working with Ordnance Survey (OS) for a number of years now, creating engaging campaigns and related software that has always inspired large numbers of people to get involved.

Ordnance Survey had an idea that needed to be delivered within a very tight timeframe. The project was Photofit, which aimed to encourage customers to engage more with the OS brand.

Together, the Studio and OS developed a national online competition that enabled members of the public to upload their own photographs using the bespoke GeoTagging service we designed and developed specifically for them, as well as allowing users to vote for their favourite shot. The GeoTagging functionality enabled the winners from each region to be selected with ease, with the winning photographs being featured on the iconic Ordnance Survey paper maps.

The Studio team had an extremely tight project timeline of just two months to build the interface, deliver complex functionality, and to fully test the site before it was launched. The tight-knit team pulled together to ensure the deadline was met and that the product the client received was exactly what they needed.

Studio Republic didn't stop there: Ordnance Survey had a scheduled appearance on The One Show, which they anticipated would lead to a high influx of traffic to the site, so they ensured that the infrastructure of the site would be robust enough to deal with high visitor numbers and content upload.

The site received 15,000 hits and 700 uploads within 24 hours of going live, which it handled without a hitch. The campaign generated 471,000 page views, 46,000 users, 11,480 entries, and 59,889 votes cast.

that we've designed and built as a plugin for our MODX CMS systems.

"Now, when a client logs in, the first thing they see is a simple interface that shows exactly what's in the pipeline, what's completed, and what they can be getting on with. We introduce them to the system when the first templates have been signed off and built, which enables them to start using MODX a lot earlier, rather than having to learn everything just before the site goes live. In the not-too-distant future, we're hoping to give this software away for free to help other developers in the MODX community and their clients."

The tools in use are not the core driver for how a design is approached and executed. "Our designs are always created with the primary users in mind – if mobile is the primary device used to access the site, we design mobile first. Responsive designs utilising a fluid layout are key for a consistent experience," says Chris. "This has been important for some time but, with Google now indexing by mobile first, and with the way that people now consume the web, this has never been truer. With adaptive designs being less flexible and not allowing for scalability for all mobile screen sizes, we feel



"We spend more time with our colleagues than we do our own families"

the extra effort given to designing and developing responsively always pays off for our clients and their end users. It just provides a more polished and engaging product.

"We try to always have the end user in mind. The extensive initial research period that we undertake on all of our projects means that we can start to outline and analyse who will be using the website or service and how they'll be using it. From there, we plan how best to serve them the content they need. Working with a lot of charities means we're constantly trying to improve and refine how people interact with online services – especially the way that they are able to make donations. Whilst a desktop screen provides loads of room to convey the impact of giving money, keeping the donation experience simple but informative on a mobile device is where the real challenge comes in."

Chris also explains Studio Republic's approach to social media: "With billions of people using social media channels every day, it's a vital way of driving traffic to a site. Use of social media is bespoke to the project; each channel obviously comes with differing audiences and varying ways for them to be targeted," he says. "Our first step is always to work out who we want to attract and, from there, we identify the best way to engage with them. The great thing about being a full-service agency is that we can offer a wide range of

solutions. Each of our campaigns take multi-faceted approaches with social media being one, among content creation, PPC, guest blogging and printed material."

The ethical stance the agency takes influences the people that want to work for the agency. Chris concludes: "Studio Republic is made up of the people that work within it. When looking for new team members, we look for passionate people who have researched and understood why Studio Republic exists. In applying for a role here, we hope that they've connected with our ethos and approach. We spend more time with our colleagues than we do our own families and, in a tight-knit team of like-minded individuals, it can be disastrous to get it wrong. Whilst skills are important, they can be learned and improved, so it's the right personality and outlook that we try to find."

Studio Republic not only create sites that enables clients to make a positive impact, they craft digital spaces that speak to their own values, too. The close connection to these ideals colours every aspect of the striking work the agency has created over the last 15 years. With their fingers on the pulse of what digital design means today, Studio Republic's ethical compass ensures that their clients aren't just clients, but partners who want to develop new digital spaces to improve the world that we live in. □



studiorepublic.com

Founders
[Chris Todhunter](#)

Year founded
2003

Employees
10

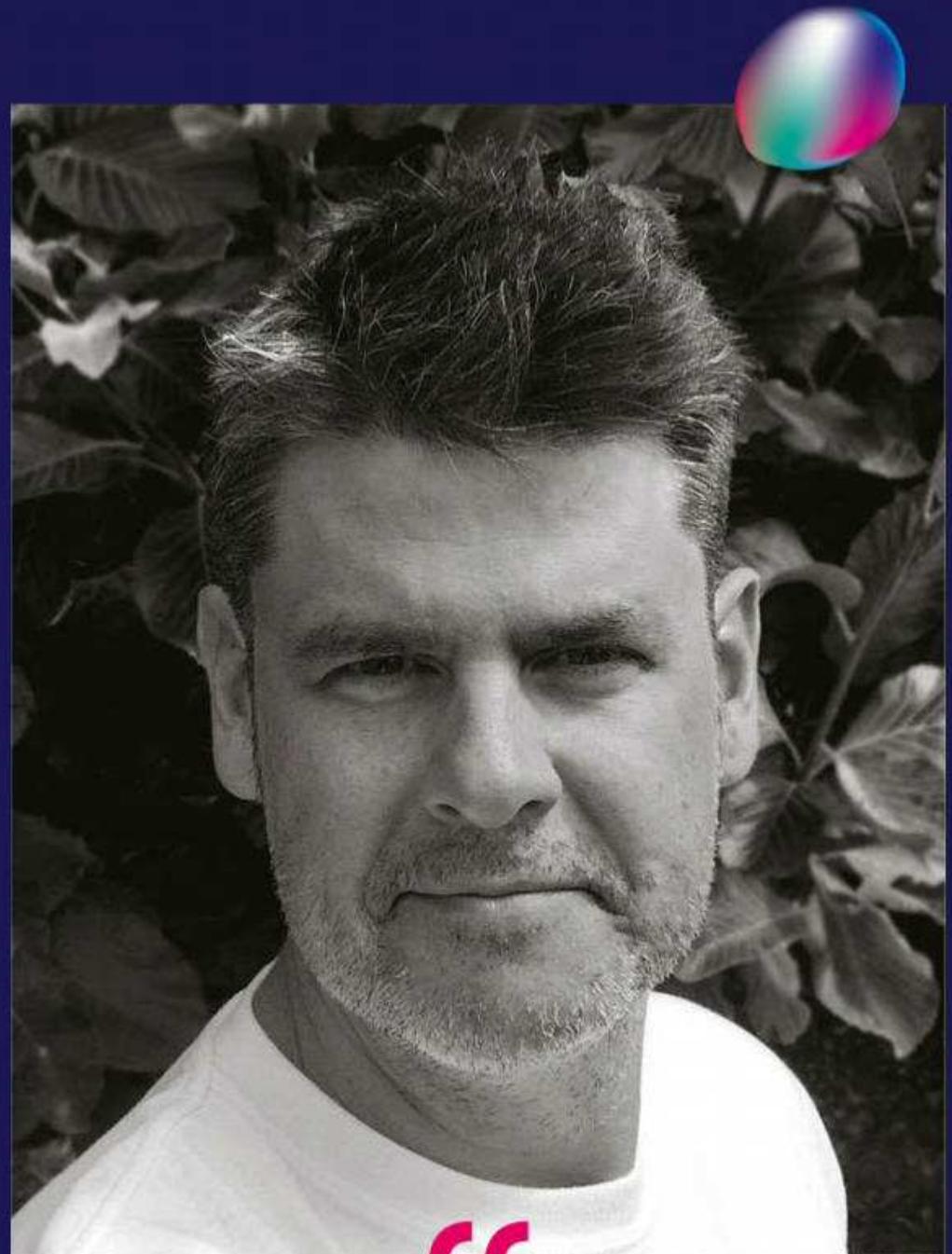
Location(s)
[Winchester and London](#)

Services
[Web design](#)
[Web Development](#)
[Campaign strategy](#)
[SEO / PPC](#)
[Content Creation](#)



ANIMATE FOR THE WEB

The background features a dark blue gradient with a grid of red wavy lines. Several 3D-rendered spheres in shades of blue, green, and purple are scattered throughout, some with motion trails. Yellow diagonal banners with the text '>> CSS >>', '>> JAVASCRIPT >>', and '>> JAVASCRIPT >> CSS' are positioned around the text.



Animation has become a key component of a great web experience. Here we reveal essential advice from the pros, inspiration examples and how to get started with the popular web animation platform Greensock. It's time to start tweening

EXCITING TIMES AHEAD

More tools and greater support means animation on the web is destined for greater things

IT'S AN EXCITING TIME TO BE PART OF THE WEB ANIMATION community and there are now more tools than ever to help you create your next engaging animation or interactive project. And no matter what language you use (imperative, declarative, and so on), or what software you use to build your project there's now likely to be an animation library that is designed to work specifically for that. You can even output your video animations to SVG from After Effects now using Lottie, enabling a whole new raft of animators to join the party.

Animation for the web is now a first-class citizen and, frankly, it's

about time. Traditionally animated content is a snapshot of motion – you watch it and the content never changes. This is fine when the animation is designed to get across an idea or a concept. A picture tells a thousand words and a carefully crafted animation turns those words into a meaningful story. But more and more it's becoming important to design interactive, animated content as a framework rather than as a linear experience. Our digital lives are fuelled by live data that constantly changes – interactive animations should be able to handle that data and react and adapt in a functional and aesthetically pleasing way.

Try something new

Don't get stuck using the same tools – try new animation frameworks especially if they have been built specifically for the kind of project you're working on (React, Vue, and so on). It's a great way to expand your skills, stay relevant and encourage new ways of thinking about your approach to animation.

This is not to say that you should keep up with every single new framework that comes out – that would be impossible, unnecessary and frankly ridiculous. Pick libraries and tools that will help you with your current project. You will learn how it works more easily if you have a real use case for it and know roughly what you need it to do. Working with new tools can also reshape how you think about the way you currently build and structure projects – they can also help you to understand more about the framework you're using. For example, animation tools designed specifically for React can really help you think in a more 'React' way.

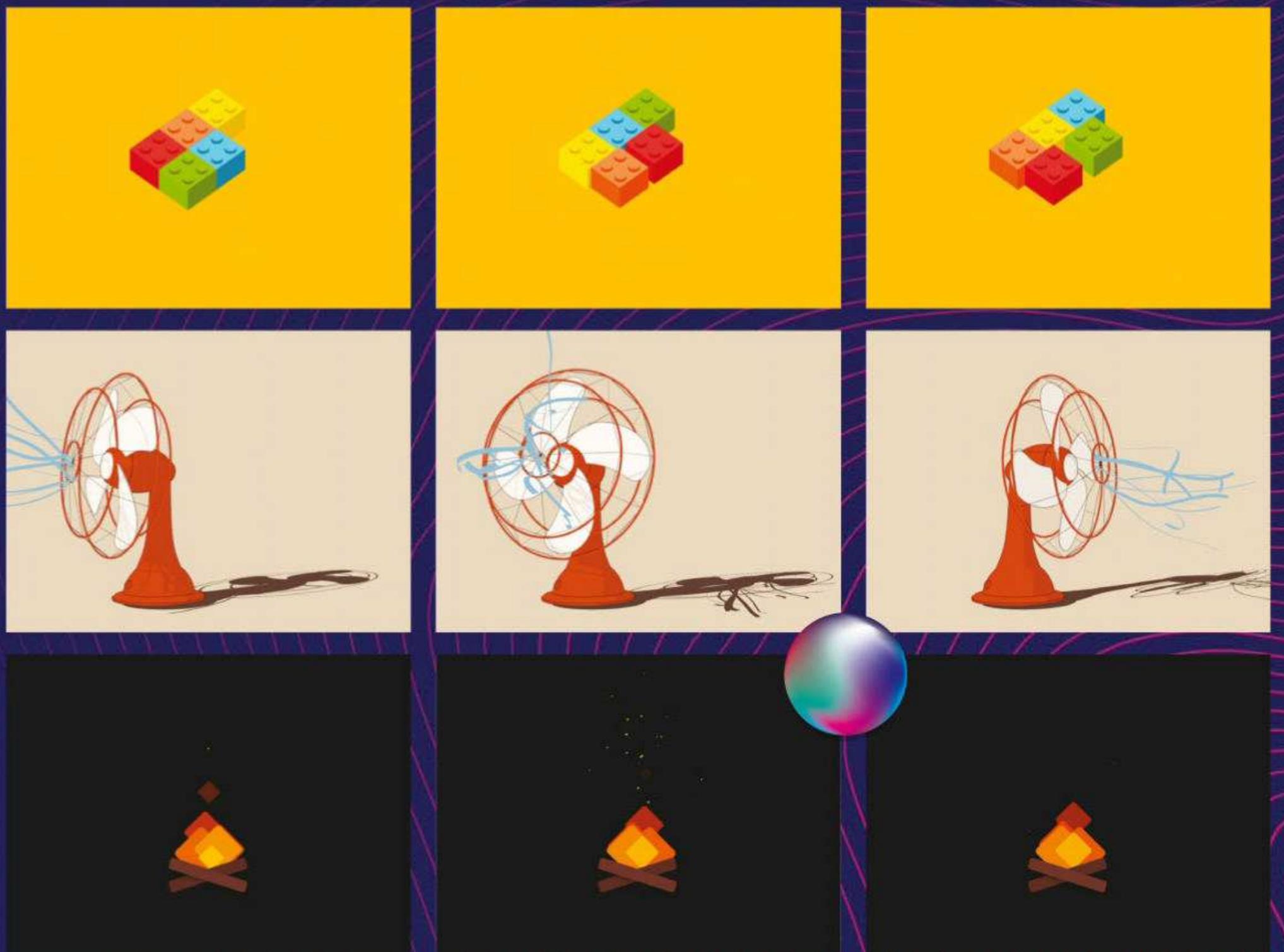
Always be a student. Remain humble and stay excited about learning. The more you know the more you'll realise that there is so much more to learn.

Chris Gannon
Independent Interactive Designer, Web Animator and Illustrator
gannon.tv [@chrisgannon](https://twitter.com/chrisgannon)

Get hired to do what you love

When prospective clients find your portfolio they'll often point to something you've already done and want that or a version of it. This means if you have a project in your portfolio you didn't particularly enjoy you might be hired to do it again! So it makes sense to only put the work you want to get hired for in your portfolio.

This is tricky when you're starting out as you won't have a huge backlog of work to choose from, so everything you've ever done tends to go in your portfolio. However, work in your portfolio doesn't necessarily have to be client work. Instead create a fake brief of, say, a logo animation (maybe of a company you admire) and get on and make the best animation you can. Hopefully your passion and enthusiasm for the project will shine through in your work and when you post it on your portfolio and social media, @ that company, too. They may not want that particular animation, but they'll see you're keen and talented and you'll be on their radar.



Look outwards

Take inspiration from random and disparate sources. Avoid looking at your own industry for your next idea/inspiration (unless you are practicing a specific technique). Looking inwards means you just regurgitate the same ideas. Looking outwards makes you form unusual connections and keeps your brain on its toes.

Have good taste

The very best designers and developers have a great sense of quality, whether it's regarding aesthetics, layout, functionality or code practices. And they devote their professional lives trying to attain or exceed their own tastes. In other words, if you have good taste, you'll produce great work.

Add your personality

Adding your own personality to your work is crucial. Whether it's visual puns, hidden meaning or quirky humour at the end of the day we are all humans and we all respond to human traits. Humour is important - it's a universal language and is always appreciated pretty much everywhere.

Ship it and share it

Even if it's just a silly little experiment or a technique you've practiced, finish it as best you can and share it. You can learn a huge amount from other people's comments – and sometimes you just need to get some closure and move on. Often the thing you think isn't that great is the thing that resonates most with others.

Practice, be excited then practice more

No-one is born being good at anything - it takes years of practice and dedication to reach the level you want. And this level is only possible by staying excited and practicing. Practice doesn't seem like practice if you're excited about what you're learning.

Sketch it

Sketch your ideas out on paper, even if you're a terrible artist. The simple act of drawing an idea on paper can help make that idea more of a reality. We all work in a virtual space so having a physical reminder of your idea can help you to stay focused and stop you from straying or getting distracted.

ABOVE Chris Gannon puts his own advice into action



Looking inwards means you just regurgitate the same ideas. Looking outwards makes you form unusual connections

Get started with the GreenSock Animation Platform (GSAP)

GSAP is an industry standard for web-based animation and is used by millions of sites worldwide

GreenSock Animation Platform (GSAP) enables you to animate anything you can access with JavaScript including DOM, Canvas and CSS, as well as your own custom objects.

GSAP also helps resolve browser inconsistencies, enables you to use objects to manage complex animations, and runs up to 20 times faster than jQuery. It has become a standard in the industry, and has been used in countless award-winning websites.

1. Set up a basic HTML file.

The best way to learn GSAP is to see it in action. In this tutorial you'll learn about the key features of the platform with working examples that you can put to use in your projects today!

Get started by setting up a basic HTML file, where you can drop in your JavaScript code. Include an image with a 'logo' class. You'll use GSAP to animate properties to see how it works.

```
<html lang="en">
<head>
  <style>
    .logo { width:150px; }
  </style>
</head>
<body>
  
  <script>
```

```
// code here
</script>
</body>
</html>
```

2. Include GSAP Library
Next, you'll need to add GSAP to your web project. Between your `<image>` and `<script>` tags, insert a link to the GSAP library. You can download the ZIP directly from bit.ly/2Myg5OC, or grab it from GitHub at github.com/greensock/GreenSock-JS.

Files are also hosted on the superfast Cloudflare CDN, so the simplest way is to use the hosted files like this:

```
<script src="https://cdnjs.
cloudflare.com/ajax/libs/
gsap/2.0.1/TweenMax.min.js"></
script>
```

3. Animate with tweening

Tweening is the process of changing a value over time to create an animation. For example, moving an object from A to B, scaling it, or rotating it. You can also tween your own custom values, as you may need.

The simplest way to tween a property is to use the 'TweenMax.to()' function. This needs a target object, a duration and the property/value pairs you are animating. To see the function in action, try out each of the lines of

code below using the 'logo' as the target object:

```
// tween x position from current
to 400 over 2 seconds
TweenMax.to(".logo",2,{x:400});
// tween y position from current
to 200 and opacity to 0, over 1
second
TweenMax.to(".logo",1,{y:200,
opacity:0});
// tween x and y to 100, scale to
1.5, and rotate 90 deg, over 2
second
TweenMax.to(".logo",2,{x:100,
y:100, scale:1.5, rotation:90});
```

4. Tween from and FromTo

You can tween a property from its current value to a new one using 'to()', but you can also tween 'from' a value to its current value.

For example, if your 'logo' is starting at x position '0' and you want it to end there, you could do this: 'TweenMax.from("logo",2,{x:400});'

In this case, your logo will immediately move to x=400 and then tween to x=0.

You can even define both the start and end values, ignoring the current values using 'fromTo()' like this: 'TweenMax.fromTo("logo",2,{x:400},{x:200});'

5. Easing

'Easing' is the 'style' of animation, as values transition from A to B.





Instead of a constant rate of speed, which is called ‘linear’, you can apply functions to curve the rate of speed. Do they start slowly and gradually speed up? Do they come to an abrupt stop and bounce a little at the end? In animation these styles add character and emotion to your work. You can apply an ‘easing’ function like this:

```
TweenMax.to(".logo", 2, {x:100, y:100, rotation:90, ease:Circ.easeIn});
```

6. Easing functions and In / Out

GSAP includes a variety of ‘easing’ functions such as ‘back’, ‘bounce’, ‘elastic’, ‘sin’, ‘circ’, and ‘expo’. You may have also noticed that the ‘ease’ function has a property of ‘easeIn’, which we used in the step above. You can also use ‘easeOut’ and ‘easeInOut’. These indicate where the easing curve is applied in the animation. Try out the following to see the results:

```
// easing Out with a bounce
TweenMax.to(".logo", 2, {x:100, y:100, rotation:90, ease:Bounce.easeOut});
// easing In and Out with elastic
TweenMax.to(".logo", 2, {x:100, y:100, rotation:90, ease:Elastic.easeInOut});
```

7. Delay a tween

Sometimes you will want to delay the start of an animation, either to synchronise it with another animation, or to make it wait for an event to occur. You can use another one GSAP’s ‘special properties’ called ‘delay’ to do this. Try out the following code to see how you can delay tweens for specific timing:

```
TweenMax.to(".logo", 1, {y:100, ease:Bounce.easeOut});
// delay this tween by 1 sec
TweenMax.to(".logo", 1, {rotation:90, ease:Circ.easeOut, delay:1});
```

8. Callback functions

To integrate animations with your code, it’s important to know when events occur, especially when an animation ends or begins. You can use the ‘onComplete’ event callback for this. You may want to know when a tween starts, using ‘onStart’. And you may also want

to sync your tween with another animation, or use the tween and its easing to update some other custom object. Use the ‘onUpdate’ callback for this. Here is how you use the callback:

```
TweenMax.to(".logo", 1, {
  y:100,
  ease:Bounce.easeOut,
  onComplete:tweenComplete
});
function tweenComplete() {
  console.log("tween complete");
}
```

9. Callback parameters

When firing callbacks you may want to pass information along with the callback to the function that handles it. You may need to pass information about the caller or specific values. This enables you to integrate your animations with the rest of your code base. All parameters are passed via ‘callback+“Params”’, that is ‘onUpdateParams’.

In the following example, you can track progress of the tween using the keyword ‘{self}’ and can see how to pass multiple parameters using an array:

```
TweenMax.to(".logo", 1, {
  y:100,
  ease:Bounce.easeOut,
  onUpdate: tweenUpdate,
  onUpdateParams:[{"self"}, "completed"]
});
function tweenUpdate(tween, message) {
  var percentage = tween.progress() * 100; // progress() is a value 0-1
  console.log(percentage + " " + message);
}
```

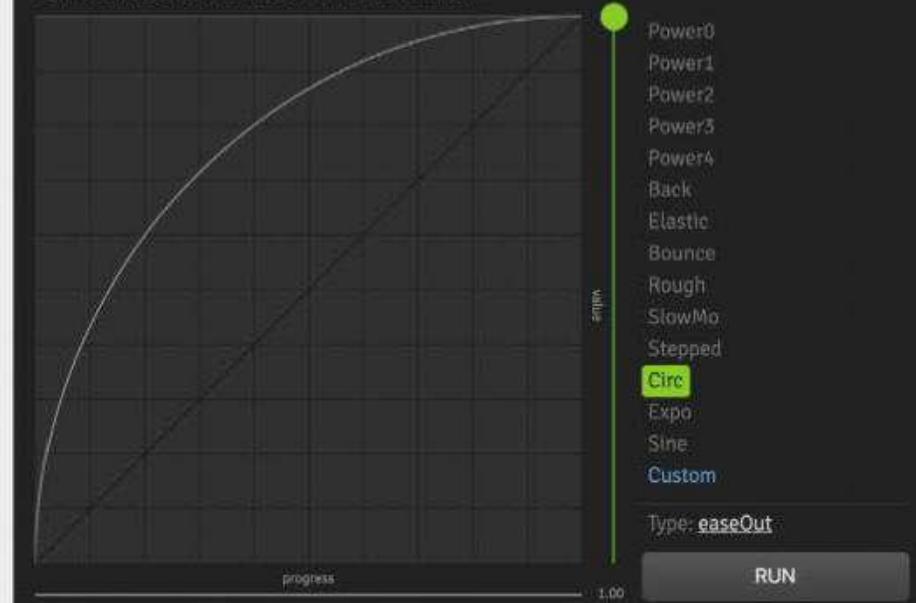
10. Controlling animations

OK, so you can now create tweens and ease the tweens, fire callbacks and parameters. But how do you control animations? Often you want to start or stop them based on other events.

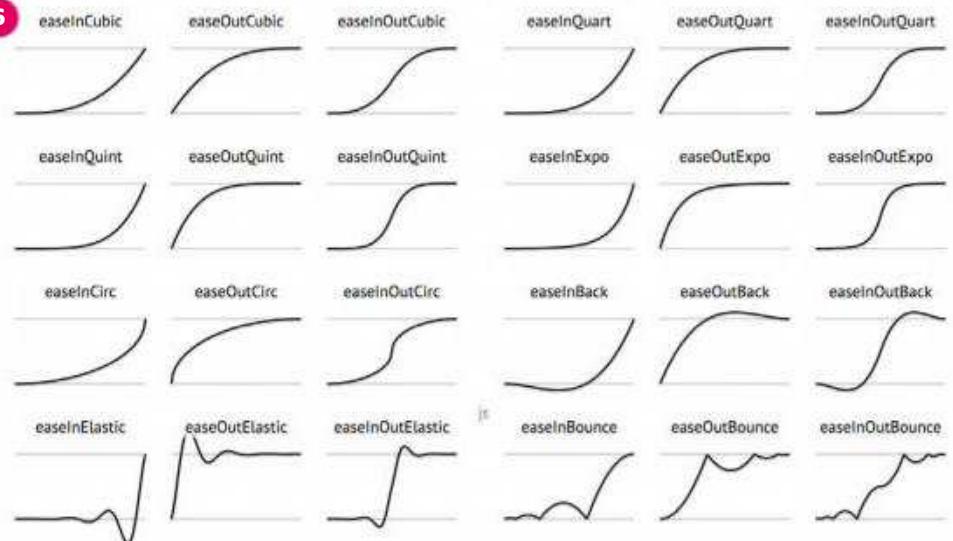
If you need to stop an animation you can use the ‘pause()’ method. If you want to resume a tween, use ‘resume()’. To totally destroy a tween use ‘kill()’. Try out the following and see what happens:

5

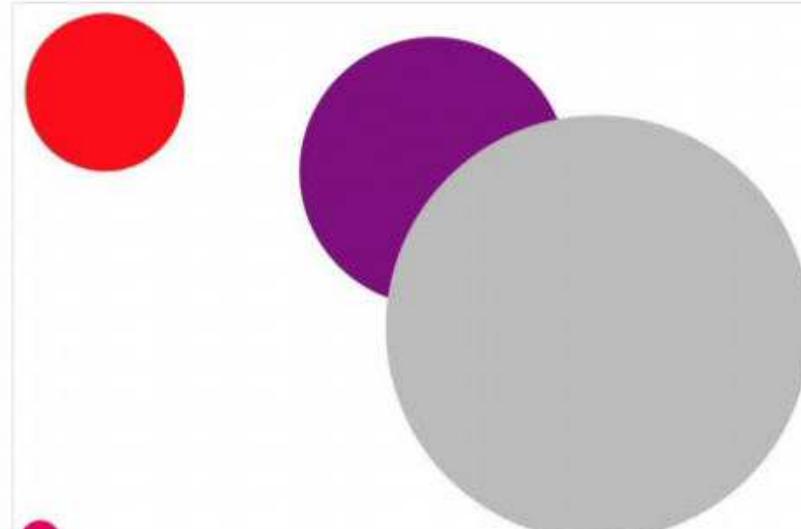
GreenSock Ease Visualizer



6



11

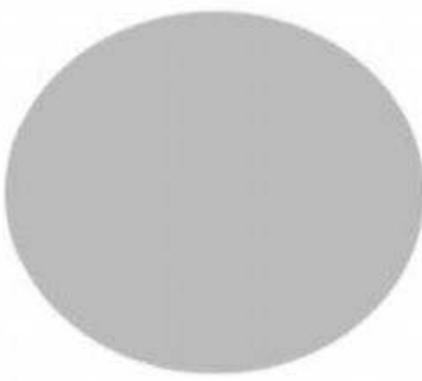
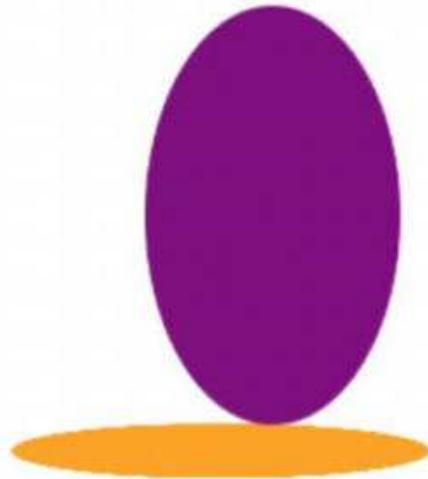
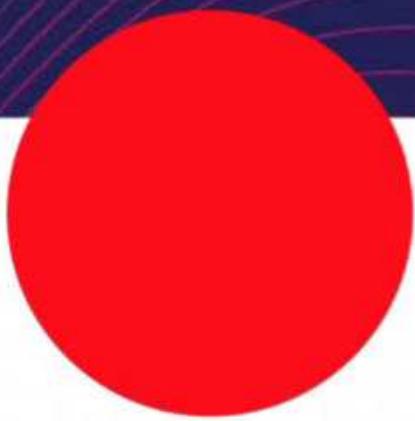


11

```
// var to hold reference to tween
var tween = TweenMax.to(".logo", 2, { y:100, ease:Bounce.easeOut });
// pause immediately
tween.pause();
// start on event
setTimeout(function(){tween.resume()}, 1000);
// reverse animation on event
setTimeout(function(){tween.reverse()}, 3000);
```

11. CSS and CSSPlugin

The CSSPlugin (see ‘GreenSock plugins’ box on the opposite page) is included when you use ‘TweenMax’. It is a huge time-saver, in that it normalises behaviours across browsers and auto-detects when it is needed for animations. CSSPlugin can handle colour tweens, SVG animations and optimised performance with caching and other internal tricks.



12

13

HTML



13

Result

duration: 7.70 1 iteration
totalDuration: 36.80 all iterations including repeatDelays
repeat: 3 of 3
time: 7.70
totalTime: 36.80
progress: 1.00
totalProgress: 1.00

Burger Boy
Free delivery in NYC
Open 24hrs
[order now](#)

[restart\(\)](#)

- and it's often more efficient to translate positions via CSS.

You've already seen this in action when you used the 'opacity' and 'position' animations. It works without any special coding!

12. 2D and 3D transforms

CSS transformations enable you to scale, rotate, translate and skew. They work in both 2D and 3D to create beautiful effects quickly. GSAP includes built-in 'transform' properties such as 'x', 'y', 'rotation', 'rotationX', 'rotationY', 'skewX' and 'scale'. Try applying the following tween to your image instead of the ones we've already tried:

```
TweenMax.to(".logo", 3, {
  x:100,
  y:100,
  scale:1.5,
  rotationY:360 ,
  ease:Bounce.easeOut});
```

13. Timelines

Once you move beyond one or two tweens you may be wondering how to manage multiple animations. GSAP includes a 'timeline' object to do exactly that. You append tweens to the 'timeline' object and can use the 'position' parameter after the tween to time them. You can have tweens run one after another, or have gaps, or even overlap them. Add a couple more images to your HTML with classes 'logo2' and 'logo3' respectively.

To see how it works, try out the following 'timeline' code:

```
//create a timeline instance
var tl = new TimelineMax();
tl.add( TweenMax.to(".logo", 1,
{x:50}) );
// note the final "0" to make
this one start at 0 sec.
tl.add( TweenMax.to(".logo2", 1,
```

```
{y:100} ),"0" );
//note the "+.25" to make this
one start at .25 sec
tl.add( TweenMax.to(".logo3", 1,
{rotationY:180, y:50,
X:50}),"+.25" );
```

14. Timeline control

As well as controlling animations, you may want to control timelines, too. Luckily, you can pause, resume and reverse just like you can with animations. You can also add parameters to the timeline to 'repeat', 'yo-yo' and add callbacks for the entire timeline. You can also control a timeline's speed using the 'timeScale' property. Try replacing your timeline declaration with the following code to see how it works:

```
var tl = new TimelineMax({
  // repeat infinitely
  repeat:-1,
  yoyo:true, });
```

15. Get and set properties

One feature that's really useful is getting and setting properties of tweens and timelines. That way, you can get to know the overall progress and duration, or gather other information, about a GSAP object. In this code example you can 'get' the duration of the timeline, and then 'set' its duration to change it. This works the same for tweens as well. It's another great way to react in real time to events, and modify animations on the fly. Add the following after your previous timeline code:

```
// get the current duration of
the timeline
console.log(tl.duration());
//sets the duration to 5 seconds
after a 2 sec wait
setTimeout(function(){
  tl.duration(5);
}, 2000);
```

Move beyond one or two tweens you may be wondering how to manage multiple animations



Greensock plugins



Plugins are special properties that get added to GSAP to extend the core functionality. This keeps the core JS small, but gives you the flexibility to add features you might need.

CSSPlugin

bit.ly/2BVacGm

Enables you to animate almost any CSS property, including 2D and 3D transforms and colours.

MorphSVGPlugin

bit.ly/2Nr2NOv

This plugin enables you to control tweens that morph SVG paths. This lets you morph one shape to another and much more!

DrawSVGPlugin

bit.ly/2MDYvsD

Enables you to reveal or hide SVG strokes progressively, to create a 'drawing' animation of SVG paths.

jquery.gsap.js

bit.ly/2LyIgsv

If you're already using jQuery in your project, this plugin will automatically take over the jQuery animations and improve performance.



“

Animations have become a powerful tool for a number of reasons. They are important to grab the users attention whether it be to tell stories, guide users through a process or to simply make something look cool and engaging.

Dan Heywood
Designer / Front-end Developer
www.supremo.co.uk

Inspiration

Brilliant examples of web animation today



Julia Muzafarova

codepen.io/miocene/

Julia creates beautiful CSS animations and then shares them on Codepen. Her work is amazing and shows exactly what you can achieve just by using CSS keyframes. With over 100,000+ views on the pens, Julia's work is definitely worth a follow.

Nike React

nike-react.com/

Here we have another 3D WebGL experience, created by Netherlands agency DPDK. This interactive site shows us what Nike's new Reactor trainers feel like through the use of interesting 3D renders and smooth typographic animations.

Olaian – Lookbook Summer

lookbook-olaian.fr/

This is a great example of what you can do to make static images much more interesting. Along with its subtle parallax and micro animations, images are treated with water distortion animations and transitions.

Foosball World Cup 18

bit.ly/2mdTc3e

To celebrate the 2018 World Cup, Italian agency AQuest created this amazing interactive WebGL game. From 3D rendering to slick typographic animations, it's a joy to play and showcases the power that modern browsers can accomplish.



Seven to follow

Darin Sennett

twitter.com/dsenneff

Sharing how he builds his amazing animations with SVG and GreenSock, Darin is a must-follow.

GreenSock

twitter.com/greensock

Talking of GreenSock, their Twitter is a great place for inspiring designs created with the GSAP library.

Chris Gannon

twitter.com/ChrisGannon

Chris is another designer who shares great content on creating wonderful web animations.

Green Chameleon

twitter.com/CraftedbyGC

A Bristol-based creative studio that often shares tasty animated designs.

Val Head

twitter.com/vlh

Author of Designing Interface Animation, Val shares fun examples of simple to advanced web animations.

Manoela Ilic (Mary Lou)

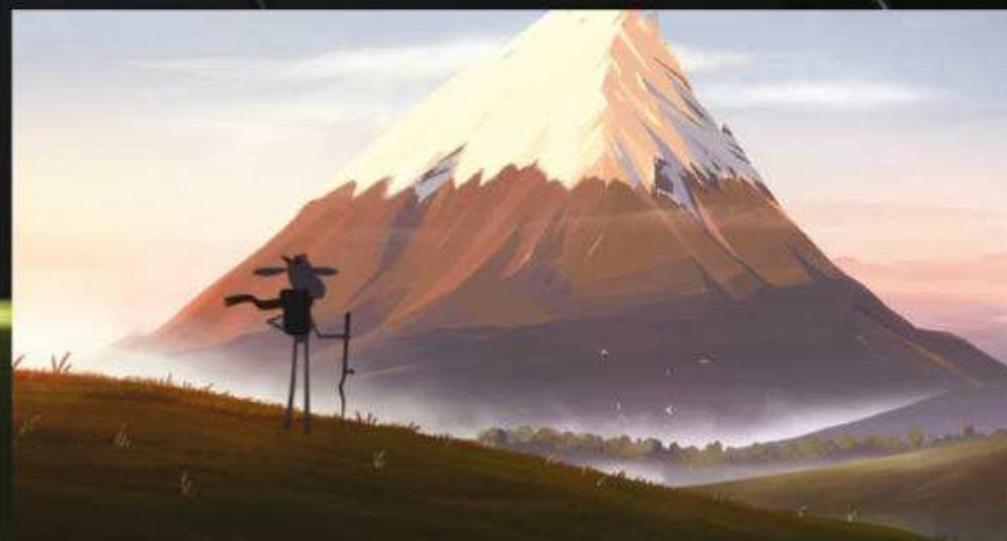
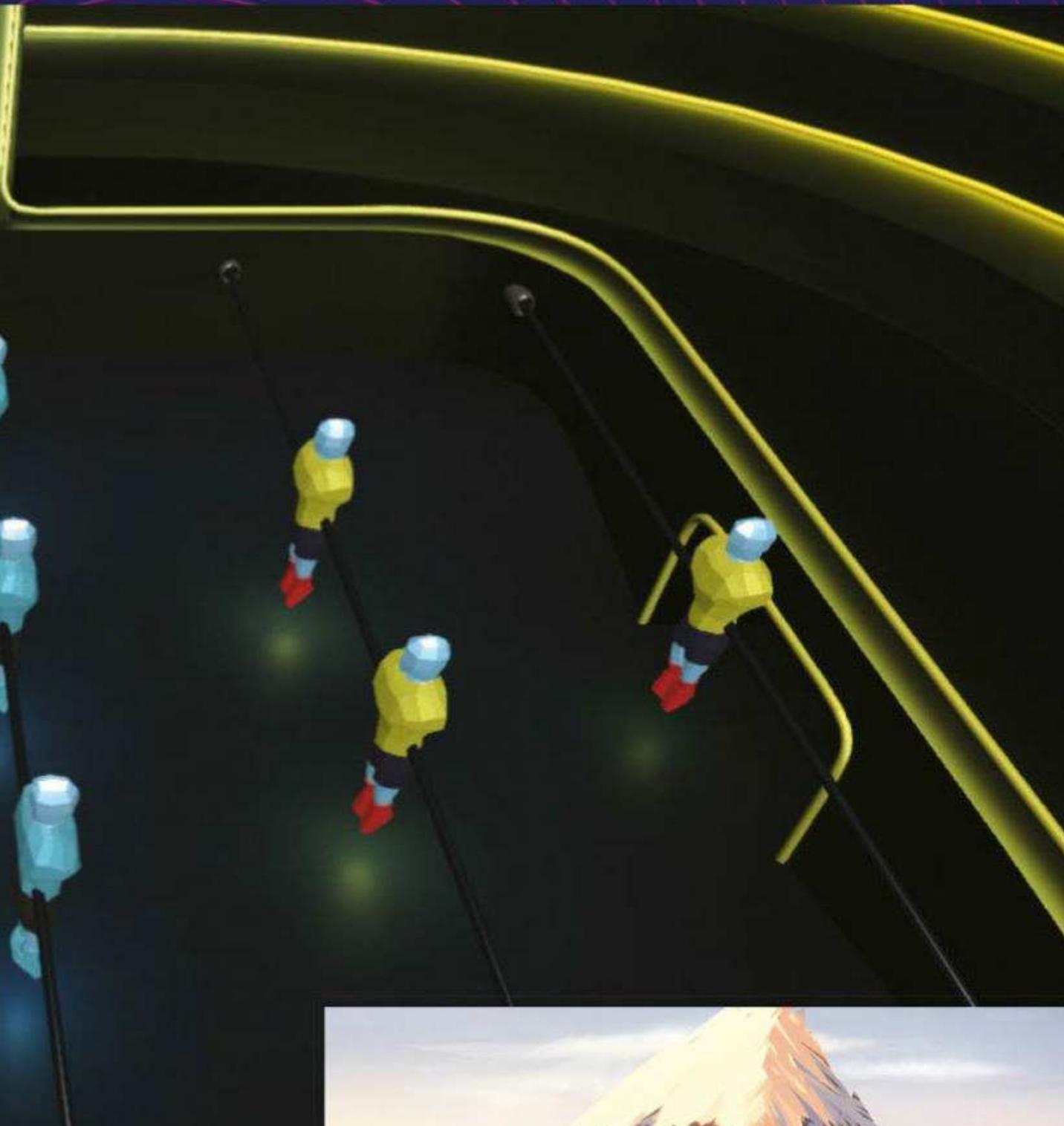
twitter.com/crnacura

One of the curators of Codrops, ML creates and shares her work, tutorials, and collections of web inspiration.

Muzli

twitter.com/usemuzli

Providing quick access to all sorts of design, Muzli share lots of animated inspiration hourly.



In Pieces

species-in-pieces.com/

Created way back in 2015, In Pieces was one of the first projects that stood out for me. It grew by experimenting with CSS clip-paths and CSS transitions. Every one of the 30 endangered animals you see is built with the same 30 pieces as the last.

Oat The Goat

oatthegoat.co.nz/

This beautiful interactive story utilises HTML Canvas and the GreenSock animation library to create something special. Created by Assembly, the piece was part of a bullying prevention campaign launched by New Zealand's Ministry of Education.

UI Movement

uimovement.com

UI Movement is a community of design enthusiasts and developers, which focuses exclusively on web and mobile interfaces, and is where members share and discuss the interesting designs. A great source of animation inspiration.

web workshop

Creating cursor effects

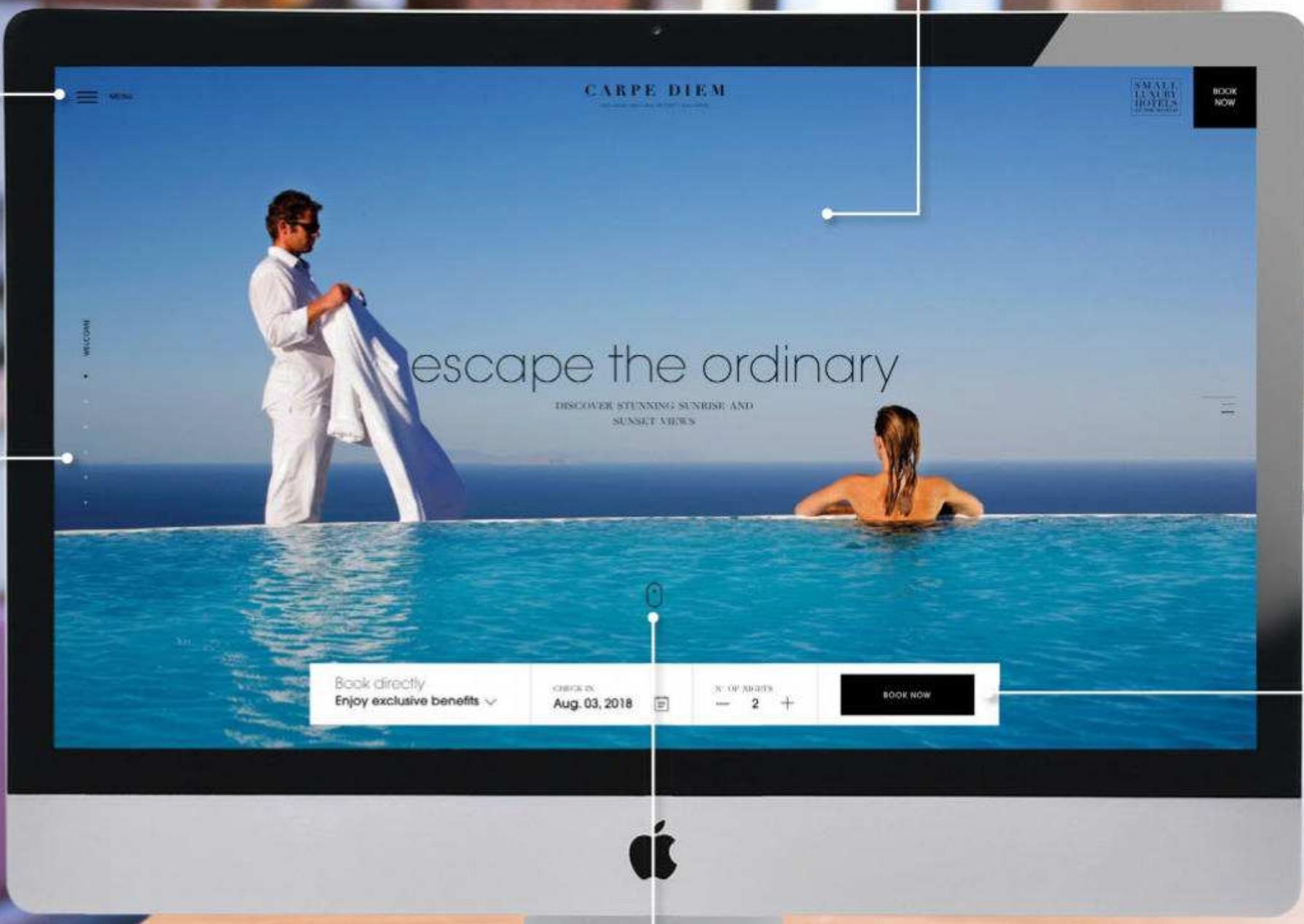
Inspired by carpediem.santorini.com

Menu overlay

The menu reveals an overlay with a glowing cloud that follows the user's mouse around as they navigate the options on the menu panel.

Background slide show

The image in the background is a transforming slideshow, with animated text for each slide explaining the breaks.



The quick access menu

The content for the page is available from this quick access menu, with each section being displayed as a point to orient the user.

Animated hint

The user is prompted to scroll down with the animated mouse icon on the screen. The point inside the mouse moves up and down to show the page has scrolling.

Context information

Information about the vacation breaks are located directly on the screen to enable the user to see options that are available.

EXPERT ADVICE**Shimmering image trends**

A recent trend with having shaders for WebGL is that images are often found to shimmer on the screen. Don't just add these because they are cool effects, but add them because it adds meaning to the content. Here the shimmering reflects the heat or the water movement of the swimming pools.

**<comment>**

What our experts think of the site

Making the menu

The menu of the Carpe Diem site features a glowing cloud that follows the mouse around and this connects the physical space of clear skies and light, wispy clouds with the digital locations of the web presence. The movement following the mouse with easing gives subtle interest to the navigation.

Mark Shufflebottom, Professor of interaction design

Technique**1. Create the menu**

The menu sits as an overlay on the page and has a glowing cloud follow the mouse around. Here is the page structure for the menu. Add as many list items as necessary for the left and right menus.

```
<div class="container">
  <div id="glow"></div>
  <div class="holder">
    <div class="left">
      <ul><li>LEFT Menu</li> </ul>
    </div>
    <div class="right">
      <ul><li>Right Menu</li></ul>
    </div>
  </div>
</div>
```

2. Position the content

Using either an external CSS document or style tags in the head, add the positioning code for the content. This displays the container as a grid element and enables the glow effect to be positioned absolutely so it can follow the mouse.

```
.container {
  display: grid;
  width: 100%;
  height: 100vh; }

#glow {
  position: absolute;
  z-index: 1; }
```

3. Menu elements

The elements for the menu are placed in another grid over the top of the glowing cloud. It has two columns for the left and right menus. The menu items are in an unordered list, so the bullet point is removed.

```
.holder {
```

```
display: grid;
margin: auto auto;
grid-template-columns: 1fr 1fr;
width: 50%;
z-index: 500; }

ul {
  list-style-type: none;
  margin: 0;
  padding: 0; }
```

4. Move the glow

To make the glowing cloud follow the mouse, add in either an external JavaScript file or add script tags before the closing body tag. These variables control the movement and cache the 'glow' DIV in a variable.

```
var speed = 12, _x = 0, _y = 0, endX = 0,
endY = 0;
var glow = document.getElementById("glow");
```

5. Update positions

Whenever the mouse moves within the document the following function is called. This updates the destination position that the glowing cloud is to move towards, and this is stored in 'endX' and 'endY'.

```
document.onmousemove = function(e) {
  endX = e.pageX;
  endY = e.pageY; }
```

6. Make it move

The last code is run every frame and updates the position of the glowing cloud. It moves it using an easing equation so that it slowly reaches the mouse position and eases into place.

```
function animate() {
  requestAnimationFrame(animate);
  _x += (endX - _x) / speed;
  _y += (endY - _y) / speed;
  glow.style.top = _y - 33 + "px";
  glow.style.left = _x - 33 + "px"; }
```

HOME
SPA
WEDDINGS
EXPERIENCES
PACKAGES
GALLERY
ABOUT CARPE DIEM
PRESS

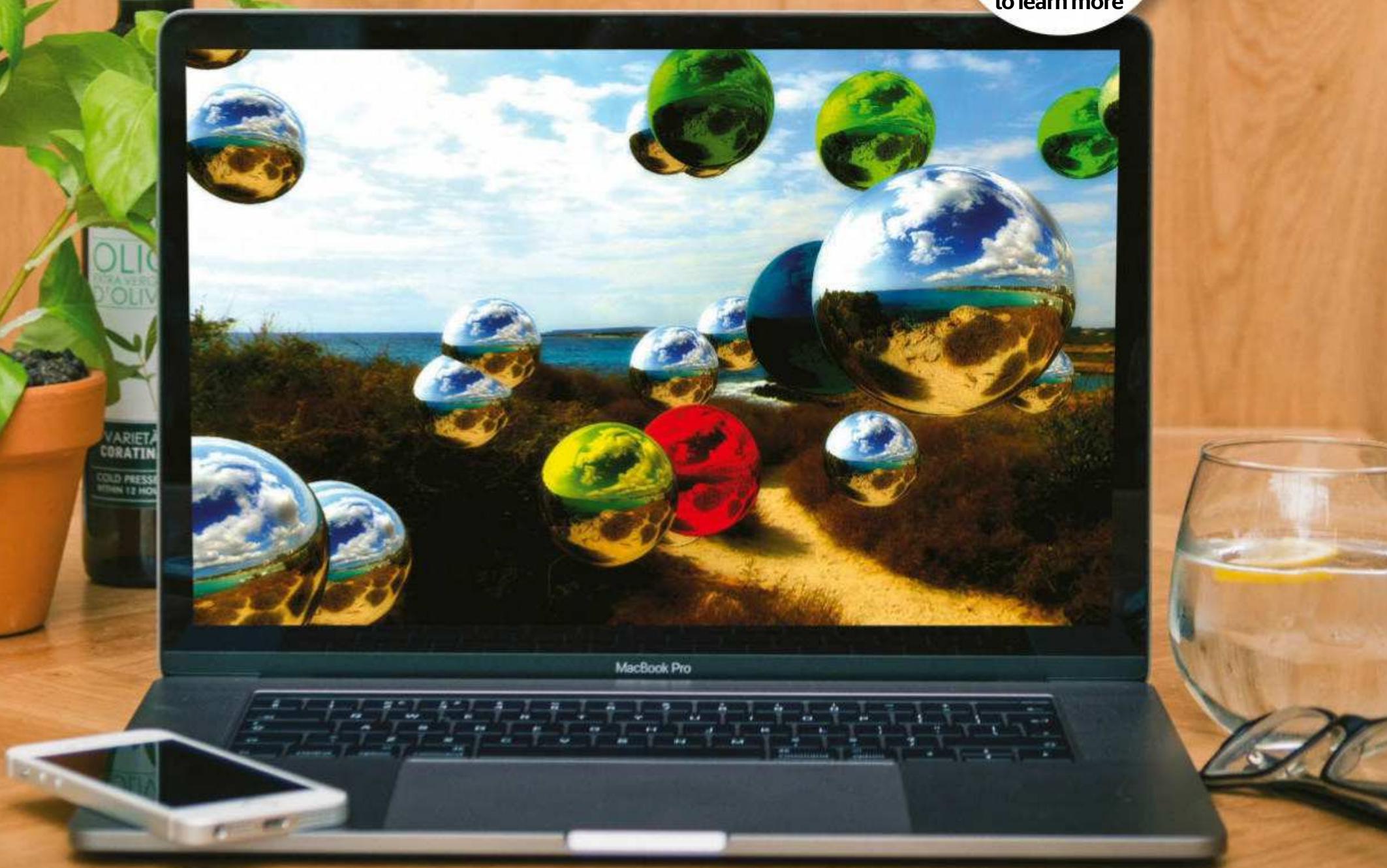
SUITES
ALL SUITES AT A GLANCE
ALBA SUITE
AURA SUITE
ASTRA SUITE
LAPIS SUITE
TERRA SUITE
VITA SUITE

Get started with Three.js – Part 4

In this fourth tutorial, you'll learn how to interact with 3D objects in WebGL using Three.js



Subscribe today
Go to page 32
to learn more



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



WebGL 3D leverages the graphics pipeline, for highly optimised, interactive experiences. Near-universal browser and device support makes it a perfect approach for real-time rendering. No plugins are required and you can start learning these technologies right away. WebGL is based on the powerful OpenGL language.

You can create complex models with high levels of detail, reflections, environment maps, and shadows. Users can interact with your experiences in real-time and in this tutorial you'll learn how to do just that!

You'll be using the popular 3D library Three.js to dive into creating scenes, and animating objects. Three.js is free, open source and lightweight, and has been used by countless award-winning websites. Even Facebook 3D objects are also now powered by this 3D library, so you're in excellent company.

Continuing from previous tutorials, you will move onto learning about interaction with objects in 3D space. You'll learn how to detect clicks (or touches on touchscreens) and mouse-over events. You'll learn about raycasting from 2D space to 3D space. You'll also learn some cool tricks for handling multiple objects and some maths to plot orbits. Other than having a JavaScript background, you can dive into this tutorial with no prior knowledge and get some great results. The goal is to demystify 3D web programming and get you inspired.

1. Create a basic HTML file

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

```
<!DOCTYPE html>
<html>
<head>
  <style>
    html, body { margin: 0;
padding:0; overflow: hidden; }
  </style>
</head>
<body>
  <script>
  </script>
</body>
</html>
```

2. Include the Three.js library

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 can find the library and minified JavaScript at github.com/mrdoob/Three.js/. Note: The code in this tutorial has been tested on Three.js v91.

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

3. Add global variables

Between your script tags, add the following global variables to keep track of your mouse, raycaster and object collection. You'll use many objects in this tutorial, to help demonstrate the interactivity:

```
// global vars
var objects=[]; // collection of objects
var num=20; // number of objects
var raycaster = new THREE.Raycaster();
var mouse = new THREE.Vector2();
```

4. 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 in some form. What's 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();
```

5. 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 indicate the near-clipping plane and far-clipping plane distances, which define what is to be visible to the camera. You will also push the camera back in 'Z' space a little to make things easier to see.

```
// create camera
var camera = new THREE.PerspectiveCamera( 45,
window.innerWidth/window.innerHeight, 0.1,
1000 );
camera.position.set(0.0,-1.0,10.0);
camera.rotation.y=.5;
```

6. Add a renderer and canvas

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 our object. You can also define the size of the draw area to full screen. The renderer creates a 'domElement' – which is actually an HTML <canvas> element – that you can then append to the body. Optionally, you could specify an existing canvas element to draw to if you prefer, via the 'canvas' attribute of the renderer.

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

7. Load the environment map and set 'skybox'

Physical materials work best with environment maps applied. These maps are 'skyboxes' that surround the object so they can affect it from all directions accurately, impacting the colour and intensity of the colour on the surface texture. A great resource for cube maps can be found at humus.name/index.

```
php?page=Textures. Add this code to load your map:
// create environment map
```

```
var envMap = new THREE.CubeTextureLoader()
.setPath( 'assets/' )
.load( [ 'posx.jpg', 'negx.jpg', 'posy.jpg',
'negy.jpg', 'posz.jpg', 'negz.jpg' ] );
// set as skybox
scene.background = envMap;
```

Note: The order of the cube map images is important, so follow that pattern when setting up your own.

8. Create a loop for object creation

You will be using a collection (array) of objects in this tutorial, to demonstrate the interactivity in a more dynamic way than using a single object. Set up a simple loop defined by the global 'num' variable that you created previously:

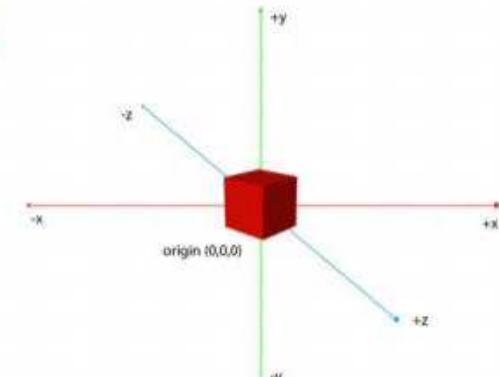
```
// create collection of objects
for (i=0;i<=num;i++){
}
```

9. Create a 3D object

Next, you need to create your object. Try using a sphere or another object you like. As with previous tutorials in this series, you use geometry and a material together to create a mesh, which can then be added to the scene. Add the following code inside your 'for' loop:

```
// create new mesh
var geometry = new THREE.SphereBufferGeometry(
1,30,30 );
var material = new THREE.MeshPhysicalMaterial(
{ envMap:envMap, metalness:1.0, roughness:0.0 });
var object = new THREE.Mesh( geometry,
material );
```

10



10. Position the object randomly

To spread the objects out in 3D space, you can apply random x, y and z position values. 3D co-ordinates have an origin of 0,0,0. Objects placed there would be in the exact centre of the scene. To distribute evenly we want the randomness to range from -10 to 10 in each direction. Try this code:

Refresher: Defining a 3D object mesh

Remember, 3D objects meshes are like physical objects. They are comprised of a geometry and materials. Geometry defines the shape through vertices, faces, and so on. Materials are the 'skin' that textures the geometry. The mesh is the result of combining geometry and materials together.

Tutorials

Get started with Three.js – Part 4

```
// set random position  
object.position.set(Math.random() * 20.0 -  
10.0 , Math.random() * 20.0 - 10.0, Math.  
random() * 20.0 - 10.0 );
```

11. Calculate the distance from the origin

To create some useful motion, you will create orbits in 3D space for each object. This will give them a predictable path to move around the scene over time. To do this we simply need a distance to store as a constant on our object. Using the origin and the random position you just set, it is easy to do this.

Add this code next:

```
// 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;
```

12. Define initial angles for orbits

To create a simple orbit over time, you need a constant distance and an angle (in radians). You will set two starting angles for your orbits, to add randomness to your scene. The two angles will enable you to orbit in three dimensions. Add this code next:

```
// define 2 random but constant angles in  
radians  
object.radians = Math.random()*360 * Math.  
PI/180; // initial angle  
object.radians2 = Math.random()*360 * Math.  
PI/180; // initial angle
```

13. Add the object to the scene and array

The last bit of code you need in the object creation loop is to add the object to the scene and to the array you created. This will enable you to easily iterate over the objects later for checking for interactions and for animation. Add this code next:

```
// add object to scene  
scene.add( object );  
// add to collection  
objects.push( object );
```

14. Render the scene for each 'requestAnimationFrame'

Next, you call the renderer's 'render' function inside a loop bound to the 'requestAnimationFrame' function. You also add a little rotation animation to the object to see it spin. Add this new code, and run your scene:

```
// render the scene
```

Object arrays

It's a great practice to start using arrays to manage collections of objects once you have more than one. Raycasting methods expect an array to check against. Animating and handling common functions are also advantages of using collections of objects.



```
var animate = function () {  
    requestAnimationFrame( animate );  
    for (i=0;i<num;i++){  
        var o = objects[i];  
    }  
    renderer.render(scene, camera);  
};  
animate();
```

15. Update object's position

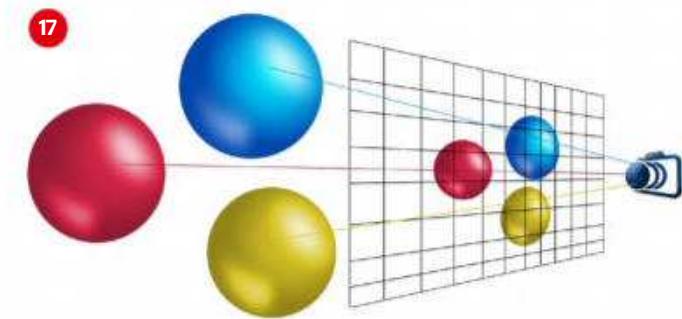
Inside your 'for' loop, update the angle of rotation for your object, which will cause it to orbit around the origin over time. You can use some simple trigonometry to calculate the new positions using distance and angle (in radians). For variety you will orbit odd objects in one direction and even objects in the other. Add this code:

```
if( i % 2 == 0 ) {  
    o.radians+=.01;  
    o.radians2+=.01;  
} else {  
    o.radians=-.01;  
    o.radians2=-.01;  
}  
o.position.x = (Math.cos(o.radians) * o.  
distance);  
o.position.z = (Math.sin(o.radians) * o.  
distance);  
o.position.y = (Math.sin(o.radians2) * o.  
distance*.5);
```

16. Add a click event listener and handler function

Interacting with 3D objects in WebGL requires a few steps. First, you need to add an event listener to the document of the WebGL canvas. Then you'll need to assign a handler function to it. Add the following code:

```
document.addEventListener( 'mousedown',  
onDocumentMouseDown, false );  
function onDocumentMouseDown( event ) {  
}
```



17. Get the mouse position and ray cast

Next, you need to get the 2D position of the mouse. You set up a vector '2' (x,y) to use for mouse position. You then need to translate this 2D position into 3D space by drawing a ray or line. You can also prevent the default click behaviour to better control the event in your own code. Add the following code in your event handler:

```
event.preventDefault();  
mouse.x = ( event.clientX / renderer.  
domElement.clientWidth ) * 2 - 1;  
mouse.y = - ( event.clientY / renderer.  
domElement.clientHeight ) * 2 + 1;  
raycaster.setFromCamera( mouse, camera );
```

18. Check to see if the ray intersects with objects

The raycaster's 'intersectObjects' method returns an array of objects that it intersects. If it's empty, then nothing was intersected. Otherwise you have objects to check. The second parameter indicates it should also check nested objects. Add the following code in your event handler:

```
var intersects = raycaster.intersectObjects(  
objects , true);  
if ( intersects.length > 0 ) {  
}
```



Interactivity in 3D

Screens on mobile and desktop are two dimensional. They have x and y axes. A first step in 3D interactions is to translate the 2D co-ordinates of a touch or mouse position, into 3D space. This is done by casting a ray or line and capturing where it intersects with objects. Three.js has a 'raycaster' class that handles this and can return the array of objects intersected. You can analyse the intersected objects and react to the event. This can be triggered on clicks, touches, mouse movements or other events. Dragging, which is a combination of clicking and moving a finger or mouse, is common as well. Also using accelerometer or gyroscope data makes for exciting interactions as well.

19. Change the clicked-on object's colour

Now that you have a reference to an object that was clicked on, you can confirm that interaction visually, or handle it in any way that you need. Try chaining the colour of the object to a new random colour, so you can see it working. Try out this code inside your 'if' block:

```
active = intersects[ 0 ].object; // get the
first object intersected
// change material to random color
active.material.color.setHex( Math.random() *
0xffffffff );
```



Turning the mouse cursor into a pointer when it's over an object that can be clicked is a useful UI technique

20. Add a mouse-over handler

Another useful interaction is mouse movement, specifically mouse-over and mouse-out events. You can use a similar technique to the click-handler you just made, with a couple of small adjustments for mouse movement. Try this code out to make the cursor icon change to a 'pointer' when you mouse over an object:

```
document.addEventListener( 'mousemove',
onDocumentMouseMove, false );
function onDocumentMouseMove( event ) {
    mouse.x = ( event.clientX / renderer.
domElement.clientWidth ) * 2 - 1;
    mouse.y = - ( event.clientY /
```

```
renderer.domElement.clientHeight ) * 2 + 1;
    raycaster.setFromCamera( mouse,
camera );
    var intersects = raycaster.
intersectObjects( objects , true);
    if ( intersects.length > 0 ) {
        document.body.style.cursor =
"pointer";
    } else {
        document.body.style.cursor =
"default";
    }
}
```

21. Mobile click-event handler

Another essential interaction is capturing 'touch' events for mobile and touch screens. This works exactly like the 'click' event, but uses the 'touches' array. First, check for touches, then use the 'targetTouches' array to grab 'pageX' and 'pageY' values, which are similar to the 'mouse.clientX' and 'Y' values. Add the following code to check it out:

```
// add mobile/touch event listener
```

```
document.addEventListener( 'touchstart',
onDocumentTouchStart, false );
function onDocumentTouchStart( event ) {
    if ( event.touches.length === 1 ) {
        event.preventDefault();
        mouse.x = +(event.
targetTouches[0].pageX / window.innerWidth) *
2 - 1;
        mouse.y = -(event.
targetTouches[0].pageY / window.innerHeight) *
2 + 1;
        raycaster.setFromCamera(
mouse, camera );
        var intersects = raycaster.
intersectObjects( objects , true);
        if ( intersects.length > 0
) {
            active =
intersects[ 0 ].object;
            active.material.
color.setHex( Math.random() * 0xffffffff );
        }
    }
}
```



Applying your code to mobile and touch screens is straightforward once you have the basics down

Design with CSS4 variable fonts

It's time to get excited. Variable fonts allow you to load one font and manipulate it through CSS





Typography on the web allows for graphic designers to be expressive with their client's messages. The web has come a long way with typography, from only allowing fonts that are installed on the user's computer, to now having a whole raft of typefaces available from online content delivery networks. If you've ever used Google Fonts, then you might notice that when you add a number of weights for the same typeface, because you would like to use it in bold etc, then Google generally warns you that adding these makes the page slow to load. This problem is because a whole typeface of every single character has to be downloaded for each of those weights, regardless of whether you use all the characters. By contrast, a CSS variable font uses variables in the CSS to manipulate the font properties. This means that one variable font will have all weights, so as the designer you can become far more expressive in the way that you use typography in your design. It doesn't have to end there though, because some typefaces allow the designer to alter more than just the weight of the typeface, and as such a great deal of flexibility can occur. We are using the open source Amstelvar font, which has no less than 17 variables associated with it, so that you can get some really interesting options from just one typeface.

1. Opening the project

Open the 'start' folder in your code IDE and open 'index.html' for editing. In the body section of the page, add the code below to give some structure and content for us to work with using the CSS variable fonts. You can change the text to suit your own needs.

```
<div class="featured">
  <div class="featuredInner">
    <h1>Variable</h1>
    <h2 class="subhead">CSS FONTS</h2>
    <h2 class="kicker">Multiple styles from just </h2>
    <h2 class="base">ONE TYPEFACE! </h2>
```

2. Finish the content

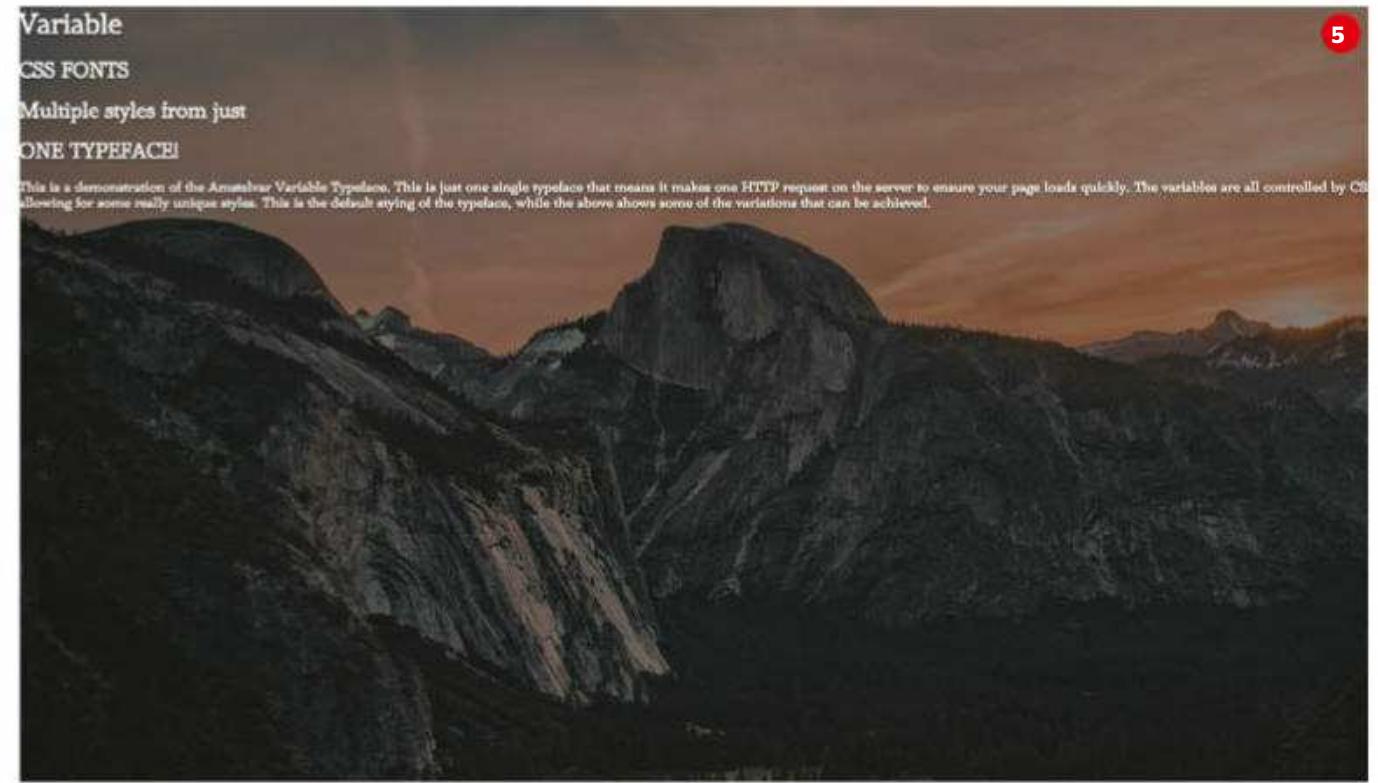
Now add the remaining code shown below. In our example we have a larger paragraph than shown here in order to allow you a description of CSS variable fonts in the example. You can copy that text from the finished folder or add your own as necessary.

```
<p class="desc">Add a paragraph description here.</p>
</div>
</div>
```

3. Link up the style

All of the CSS is going to be placed in its own separate file in the 'CSS' folder. As such the link to this is placed anywhere in the 'head' section of the code on the page. Save the 'index.html' page now as you are done with this and all further code will be in CSS.

```
<link rel="stylesheet" href="css/design.css">
```



4. Loading the typeface

From the 'CSS' folder, open 'design.css', which will be empty. Add the code as shown here at the top of your CSS. As you can see this is identical to how you would load any locally stored typeface with current CSS.

```
@font-face {
  font-family: Amstelvar;
  src: url(..../fonts/AmstelvarAlpha-VF.ttf);}
```

5. Styling the page

Now the HTML and body of the page are styled with the margin and padding removed. The font that was loaded in the previous step is now applied as the default font to all text on the page. A background image is added to cover the background and the text is set to white with a slight shadow to help it stand out.

```
html, body {
  width: 100%;
  margin: 0;
  padding: 0;
  font-family: Amstelvar;
  background: url(..../img/bg.jpg) no-repeat center center fixed;
  background-size: cover;
  color: #fff;
  text-shadow: 0px 3px 5px rgba(0, 0, 0, 0.4);}
```

6. Centring the text

The easiest way to centre text both horizontally and vertically is to use the newer CSS grid as the display object. This wrapper, which encapsulates all other tags, is set to take 100% of the vertical height with the 'vh' property.

```
.featured {
  width: 100%;
  height: 100vh;
  display: grid;
```



7. Auto centre

Now the content inside can make use of the auto for both the margin at the top and bottom as well as left and right. This means we get a text box that is going to take 60% of the screen and centre it. The text is also centred and this will help. As you can see in the browser the variable font is applied as any normal font.

```
.featuredInner {
  margin: auto auto;
  text-align: center;
  width: 60%;}
```

8. Varying the font

Now the font for heading 1 is going to be given some varying properties. The W3C would like us to use font-weight, stretch and optical sizing, but for this typeface the optical sizing doesn't work. We'll work around that in the next step but just check your progress in the browser.

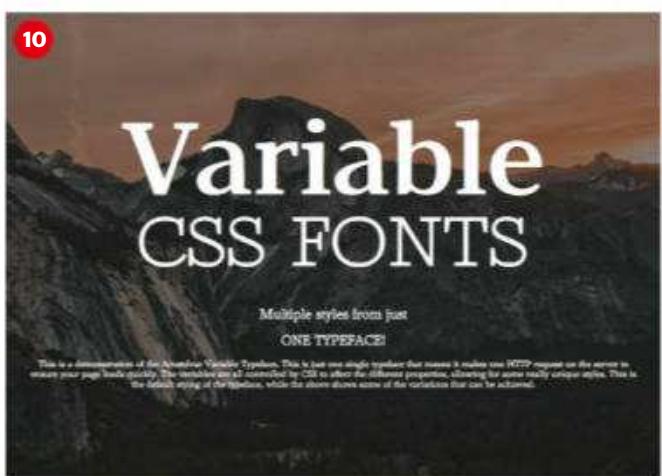
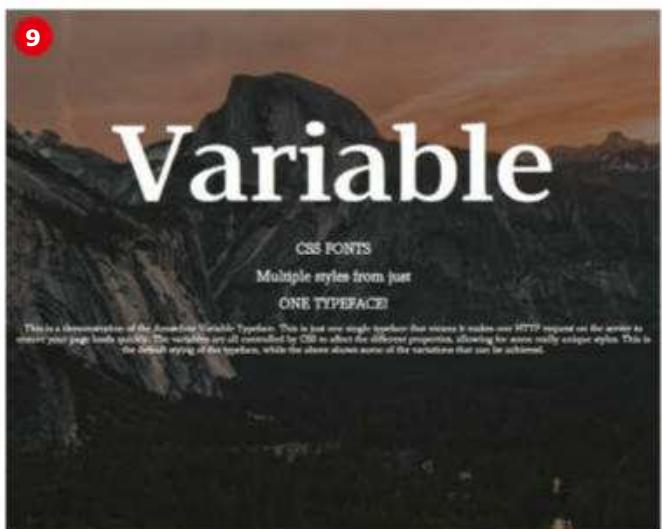
```
h1 {
  font-size: 10vw;
  line-height: 0.1;
```

The 'vh' property

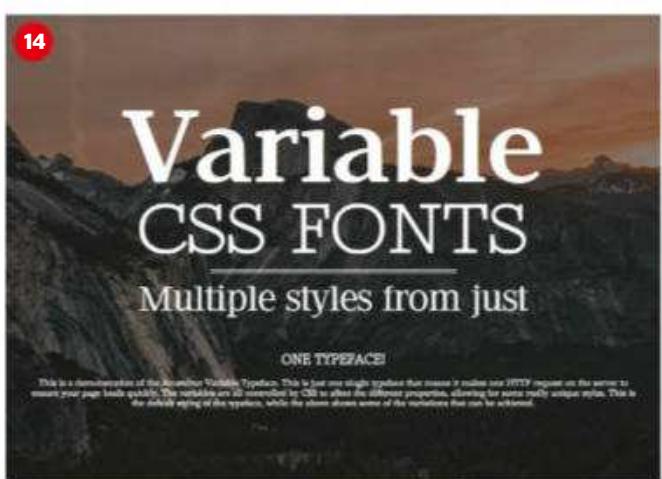
When using 'vh', 100 refers to the height of the display window. The same principle applies to width, 'vw'. This will give the viewport width and will scale according to those width and heights.

Tutorials

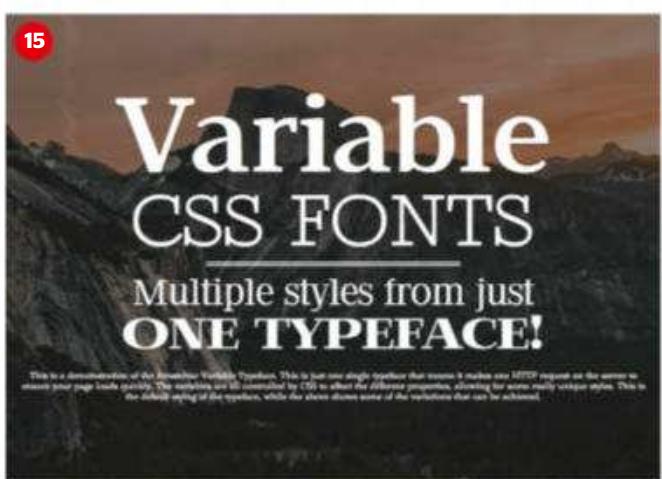
Design with CSS4 variable fonts



The second line is styled up, and as you can see here the typeface looks very different, despite being the same font



Once the third line is in, the variety of variables for the Amstelvar font makes for differing possibilities



The final large line of text looks very different from the others. This could be mistaken for a different typeface

Reduced line height

You may wonder why the line height is reduced for most of the typography. This is simply to make the text fit closer together and not have large gaps between the lines of type for the headings.

```
font-weight: 550;  
font-stretch: 100;  
font-optical-sizing: 24;  
}
```

9. Customising the font

The Amstelvar font has so many variables that aren't controlled by W3C commands, but there is a way to access them. Add this line of code that also adds the height of the y-ascender to 700 and the height of the y-upper case. In both instances, they are reduced making the type have a reduced overall height.

```
font-variation-settings: "wght" 550, "wdth"  
100, "opsz" 24, "YTAS" 700, "YTUC" 720;
```

10. Subtle changes

Now the styling of the subhead will be added. As you look at this you will see that the weight has been reduced as well as the optical sizing. When you look at this in the browser it almost looks like a completely different typeface given the characteristics have changed so much.

```
.subhead {  
font-size: 7vw;  
letter-spacing: 0.03em;  
line-height: 0.2;  
font-variation-settings: "wght" 100,  
"wdth" 100, "opsz" 20, "YTUC" 700;  
}
```

11. Creating a separator

Now a double line separator will be added between the text. This will only be 40% wide, so slightly less than the text. This is also given a slight shadow behind it. This is just to help aesthetically add a break between the first two lines and the second two lines.

```
.kicker::before {  
content: "";  
display: block;  
margin: -4% auto 0%;  
width: 40%;  
border-bottom: 8px double #fff;  
box-shadow: 0px 3px 5px rgba(0, 0, 0,  
0.4);  
}
```

12. The next line

The next styling is added to help make a difference in the text. As it stands now it doesn't look too different to anything done in the previous step, but in the next few steps more variables will be added to enhance the way this looks with some subtle changes.

```
.kicker {  
font-size: 3.3vw;  
line-height: 1.5;  
font-variation-settings: "wght" 156,  
"wdth" 100, "opsz" 32, "YTUC" 795;  
}
```

13. More variables

The new variables are marked in bold below. XOPQ is the x width of the letter, XTRA is the width of the curve.

YOPQ is the y height of the letter, YTLC is the y height of lower case letters. YTSE is the serif height, GRAD is the grading of the letter. YTAS and YTDE is both the ascender and descender y height.

```
.kicker {  
font-size: 3.3vw;  
line-height: 1.5;  
font-variation-settings: "wght" 156,  
"wdth" 100, "opsz" 32, "XOPQ" 117, "XTRA" 402,  
"YOPQ" 45, "YTLC" 600, "YTSE" 9.5, "GRAD" 90,  
"YTAS" 750, "YTDE" 250, "YTUC" 795;  
}
```

14. Yet more variables

The last few variables shown here in bold are the overall y-height of the letter with the YTRA property and then the paragraph weight and paragraph width. Save this and refresh your browser to see the changes take hold. The changes are subtle, but enough to make it look like a different typeface.

```
.kicker {  
font-size: 3.3vw;  
line-height: 1.5;  
font-variation-settings: "wght" 156,  
"wdth" 100, "opsz" 32, "XOPQ" 117, "XTRA" 402,  
"YOPQ" 45, "YTLC" 600, "YTSE" 9.5, "GRAD" 90,  
"YTAS" 750, "YTDE" 250, "YTUC" 795, "YTRA"  
900, "PWGT" 92, "PWDT" 402;  
}
```

15. All change

The next line of text is styled up with this CSS now. Again the settings are being adjusted to give the appearance of a different typeface. Save this and look at the effect that has been generated in the browser. The payoff here is that you have only made one HTTP request for the typeface but generated a different style.

```
.base {  
margin-top: -2.5%;  
font-size: 4vw;  
line-height: 0.1;  
font-variation-settings: "wght" 156,  
"wdth" 100, "opsz" 32, "XOPQ" 247, "XTRA" 402,  
"YOPQ" 45, "YTLC" 600, "YTSE" 9.5, "GRAD" 90,  
"YTAS" 750, "YTDE" 250, "YTUC" 795, "YTRA"  
900, "PWGT" 92, "PWDT" 402;  
}
```

16. Last text

The final section of text is being left in the default setting for the Amstelvar font. All that is going to happen here is that the line height is adjusted to make it much more readable on the screen. This completes the styling of all the text. Next is to show how to animate some of the properties.

```
.desc {  
line-height: 1.8;  
}
```

17. Making it animate

Let's revisit the initial heading tag and add some animation to that. In order to do that, add in the code

Amstelvar

Height: 100
Width: 23
Optical Size: 10
x-axisvar: 1
x-transparent: 42
y-axisvar: 4
y-transparent: 443
baseline: 10
Grade: 25
x-transparent Chinese: 800
x-transparent Chinese: 800
y-transparent ascender: 630
y-transparent descender: 150
y-transparent spacer: 450
y-transparent: 600
Para Weight: 38
Pwgt Wght: 402

Amstelvar is an exercise in variable weight, contrast, and proportions. Positive and negative forms can be manipulated as separate axes, as well as a large array of other variables. It is an experimental work in progress so can produce unusual results.

Design: David Berlow
Publisher: Font Bureau
Character: Latin (Metal), East Asian (Limited)
Licensing: Open source
Website: amstelvar.com

Selawik Variations

Height: 300
A demonstration font with special debugging glyphs for testing support of various variable font features and providing readouts of current settings.

ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghi
nopqrstuvwxyz 0123456789 |o+0.0000

shown in bold. The animation name refers to the keyframes that will be defined in the next step. It will take four seconds and hold on the last keyframe.

```
h1 {
    font-size: 10vw;
    line-height: 0.1;
    animation-name: anim;
    animation-duration: 4s;
    animation-fill-mode: forwards;
    font-variation-settings: "wght" 550,
    "wdth" 100, "opsz" 24, "YTAS" 700, "YTUC" 720;
}
```

18. Defining the start

The keyframes for the 'anim' are created. Here the starting keyframes are added. The weight of the typeface is changed to its lowest value of 100 and the paragraph weight is changed to 0 so that it becomes bolder as it animates over the duration of the four seconds.

```
@keyframes anim {
    from {
        font-variation-settings: "wght" 100,
        "PWGT" 0;
    }
}
```

19. Ending point

The keyframes are animated from one point to another in this example. The final ending point of those keyframes is defined, which as you can see takes it back to the default paragraph weight and the weight of the typeface is set to how it was originally defined earlier in the tutorial. Test this in the browser.

```
to {
    font-variation-settings: "wght" 550,
    "PWGT" 100; }
}
```

20. Refining the animation

When you test the animation, you will see that it is a little jittery because of the extreme change in the paragraph width. Here this is changed to 50 and the opacity is changed to 0 so that it fades in and makes the transition smoother.

```
@keyframes anim {
    from {
        font-variation-settings: "wght" 100,
        "PWGT" 50;
        opacity: 0;
    }
}
```

21. Finishing off

The final amendment is to add the opacity of 1 so that it fades in. Refresh the browser and this works much smoother. Experiment by animating any of the properties over time and you will get some interesting results for your type. And this is all from one typeface!

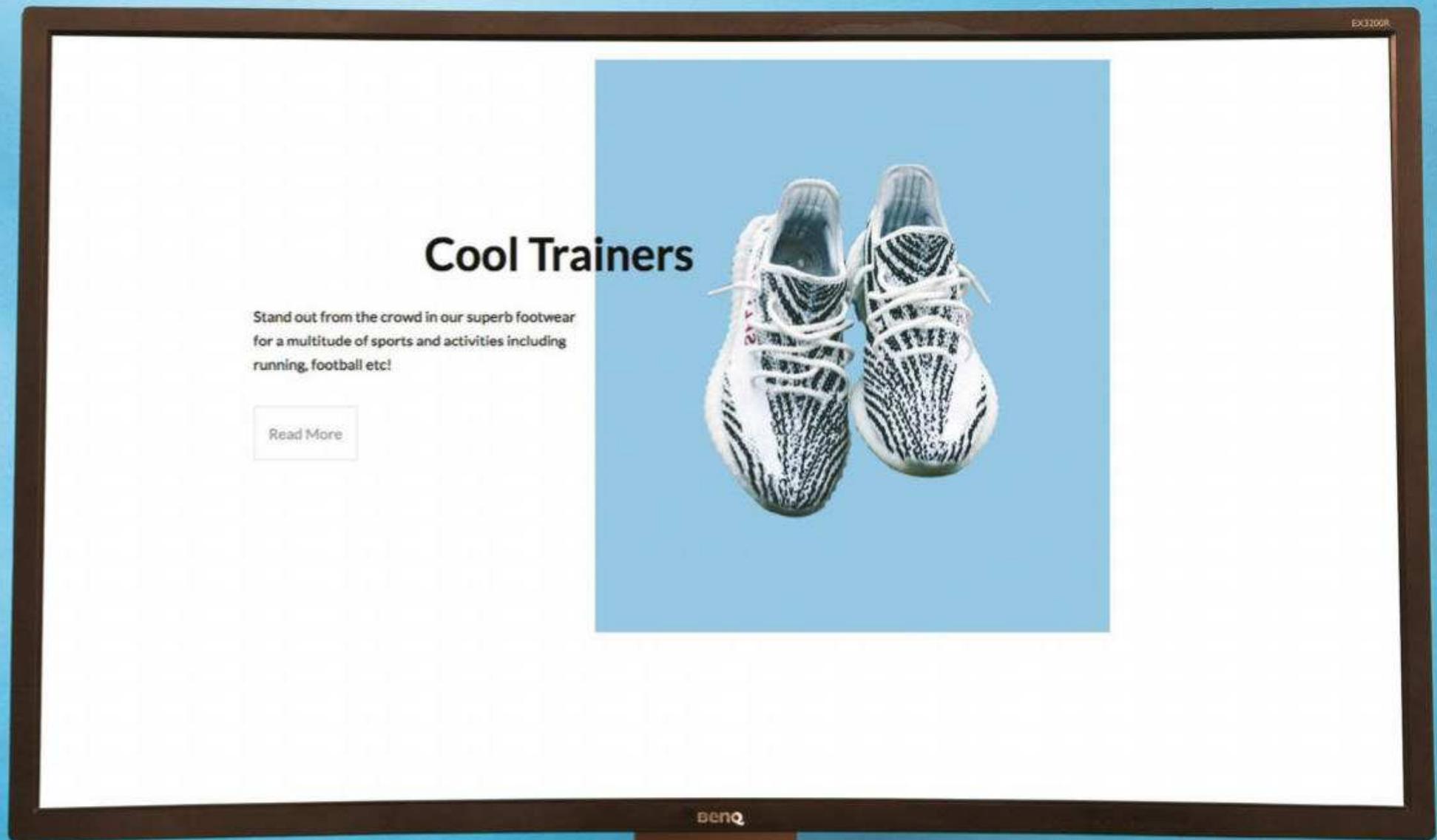
```
to {
    font-variation-settings: "wght" 550,
    "PWGT" 100;
    opacity: 1;
}
```



With the design in place, the first line is animated so that its variable font properties animate into position, while fading onto the screen

Code CSS transitions using Sass and JS

Use HTML, Sass and JavaScript to develop a UI animation effect for any eCommerce website





Animations, and movement in general, have come a long way in terms of web design. Animations that were once limited to GIFs, JavaScript and Flash designs are now possible with CSS3. This helps designers and developers build unique elements while cutting down on clutter and page speed.

That being said, there are a number of things to understand regarding CSS3 animations and their use. With a little bit of knowledge, however, you can use properly placed animations to encourage user interaction and direct the user flow. Understanding the user experience too is vital in today's industry and having confusing, slow and pointless UI animations, is not a good idea just because we can!

So in this tutorial, we will look at how we can build a simple UI animation that we could use for a fashion brand for an eCommerce store. We will be giving the user a nice pleasant design with a nice large image and animated transitions that happen on the click of a button. We'll be using HTML, Sass (.scss) and a touch of vanilla JavaScript that uses the ES6 syntax. To make things easier for us, we will develop this project within CodePen (<https://codepen.io>). If not then you can use your own local setup.

However, we will be using Sass, so if you want to follow along to everything without using CodePen, then you'll need Sass installed and running before we can get the ball rolling.

1. Getting set up

The very first thing you need to do (after you've opened up your code editor) is to create or find an image that you can use for the animated section. What we will do is use a transparent .png image so we can create a nice animation effect when the read more button is clicked.

2. Get started with the HTML

Create a new 'index.html' file and open it up in your code editor. All of our HTML will be within a wrapping element, which we will call 'container'. Then the left-sided content will be added next and will include a title, some text and a 'read more' link which we will end up styling as a button.

```
<div class="container">
  <div class="item has-content">
    <div class="item inner">
      <h2 class="section-title"><span>Cool Trainers</span></h2>
      <div class="outer-content">
        <p>text goes here....</p>
      </div>
      <p class="more-btn">Read More</p>
    </div>
  </div>
```

3. Right-sided HTML

The right-sided content will contain our image and some text, which we will hide on default and then reveal once the animation has occurred. The image we're

using has a transparent background and is at a size of 400 x 500 pixels.

```
<div class="item has-img">
  <div class="display">
    <figure class="trainers">
      
    </figure>
  </div>
  <div class="img-content">
    <div class="img-content_inner">
      <p>Lorem ipsum...</p>
      <p>Lorem ipsum...</p>
    </div>
  </div>
  <div class="bg"></div>
</div>
```

4. Sass variables

To help us stay tidy within our CSS code, we will make use of Sass variables. So if you're using CodePen, make sure the CSS section is set to SCSS and at the very top of our CSS we will add three variables: one for our primary colour, font family and easing values that we will use for animations.

```
$blue: #8dc3e2;
$mainFont: "Lato", sans-serif;
$easing: cubic-bezier(0.68, 0.11, 0.66, 0.27);
```

5. Body and title

The main body text will be the same throughout the page, so we can make use of our Sass variable '\$mainFont' and set the line height and some margin at the top. The title will be positioned slightly off centre, and we can do this by using the 'translateX' property. We also want the title to animate into a different position once the learn more button is clicked, so we will add a transition.

```
body {
  font-family: $mainFont;
  line-height: 1.6;
  margin-top: 3em; }

h2 {
  font-size: 3em;
  margin: 0;
  text-align: left;
  position: relative;
  transform: translateX(50%);
  transition: all 500ms ease; }
```

6. Centring

Using our container element, we are able to position everything to the centre of the page. We will then use flex to align centre all our elements together. Doing this will now place our main content underneath our image, ready for us to hide and reveal in a later step.

```
.item-inner {
  position: relative;
  right: 0em; }

.container {
  max-width: 900px; }
```

```
margin: auto;
display: flex;
align-items: center; }
```

7. Introduce items

The two main elements that hold both sets of content are given a class name of 'item' and we want to give these some width and padding. Also, we need to make sure the <figure> element that holds our main image, doesn't have any margin.

```
.item {
  width: 40%;
  padding: 6em 0em; }

figure {
  margin: 0; }
```

8. Hide the title

To give us some extra animation and slickness to our element, the section title that is shown on the left side will first be hidden, and when the page loads it will animate up into view. So using 'translateY(100%)' and 'overflow: hidden', we can move this out of view until the page loads (which we will do in a later step).

```
.has-content {
  position: relative;
  z-index: 4; }

.section-title {
  overflow: hidden;
  position: relative;
  margin: 0;
  span {
    display: inline-block;
    transform: translateY(100%);
    position: relative; }
```



Using cubic-bezier

Using cubic-bezier (x1, y1, x2, y2) will allow you to define points in the curve of your animations that will give you more control.

Tutorials

Code CSS transitions using Sass and JS

```
transition: transform 400ms ease-in-out;
} }
```

9. Left content and More button

What we did with the section title, we will also do to the left content text. So this will animate into view once the page is loaded/refreshed. The read more button will be given some styling so it actually looks like a button. You can give it some fancy over effect if you want, but we think it's nice without.

```
.outer-content {
  overflow: hidden;
  position: relative;
  p {
    position: relative;
    opacity: 0;
    transition: all 500ms ease;
    transform: translateY(30%);
  }
  .more-btn {
    cursor: pointer;
    display: inline-block;
    color: #999;
    text-decoration: none;
    border: 1px solid #ddd;
    padding: 15px;
  }
}
```

10. Right side

The right-side content will obviously have our main image and some text. We will give this some width of 60% but then the main background of our image - which will be blue - has no width to it but is positioned top left. However, we can now manipulate this and animate it on both page load and when the learn more button is clicked.

```
.main-img {
  width: 60%;
  position: relative;
}
.bg {
  position: absolute;
  top: 0;
  left: 0;
  width: 0%;
  height: 100%;
  background: $blue;
  transition: all 600ms $easing;
}
```

11. Hidden trainers

Again like we did in the last few steps, we will hide the main image (trainers) as we also want this to animate into view on page load. This is all achieved by using the 'transform:translateY' property and setting the opacity to zero.

```
.display {
  display: flex;
  max-width: 60%;
  margin: auto;
  position: relative;
  z-index: 6;
}
.trainers {
  width: 100%;
  img {
    width: 100%;
```

```
transform: translateY(20px);
transition: all 500ms ease-in-out;
opacity: 0; } }
```

12. Page load transitions

To get some nice animated transitions we will make good use of the transition delay property. This will allow us to have greater control of when certain elements will animate into view, and you can play around with the values to see if you can create something slightly different.

```
// on load transitions
.load-page {
  .bg {
    width: 100%;
    transition-delay: 300ms; }
  .section-title {
    span {
      transition-delay: 600ms;
      transform: translateY(0%); }
  }
  .outer-content {
    p {
      opacity: 1;
      transition-delay: 1000ms;
      transform: translateY(0%); }
  }
  .more-btn {
    opacity: 1;
    transition-delay: 1200ms;
    transform: translateY(0%); }
  .trainers {
    img {
      transition-delay: 600ms;
      transform: translateY(0px);
      opacity: 1; } } }
```

13. Adding the JavaScript

Now with all this talk of transitions when the page loads, at this point we can't see this happening because we need to control this with a touch of vanilla JavaScript. So in the JS section in CodePen (or using a external script file) we're going to begin by adding in some constant variables.

```
const body = document.body;
const button = document.querySelector(".more-btn");
```

14. Load page

Now that we have the document body element, which is stored in the 'const' variable 'body', we can now use that to add a class called 'load-page' to the body when the 'loadPage()' function gets invoked (at the end of the script). We've also used an arrow function here to keep things nice and tidy.

```
const loadPage = () => {
  body.classList.add("load-page");
};
```

15. Reveal content

Even though we haven't created it yet, we will have a click event on the 'learn more' button, which will invoke

9

11

16. Changing button text

Now inside the 'initReading' function we can use an 'if' and 'else' statement to check whether or not the body contains a class called 'is-reading', and if it does, set the text of our read more button to 'show image' or else, set it to 'read more' - which will be our default state.

```
if (body.classList.contains("is-reading")) {
  txt.innerHTML = "Show Image";
} else {
  txt.innerHTML = "Read More";
} ;
```

17. Page load

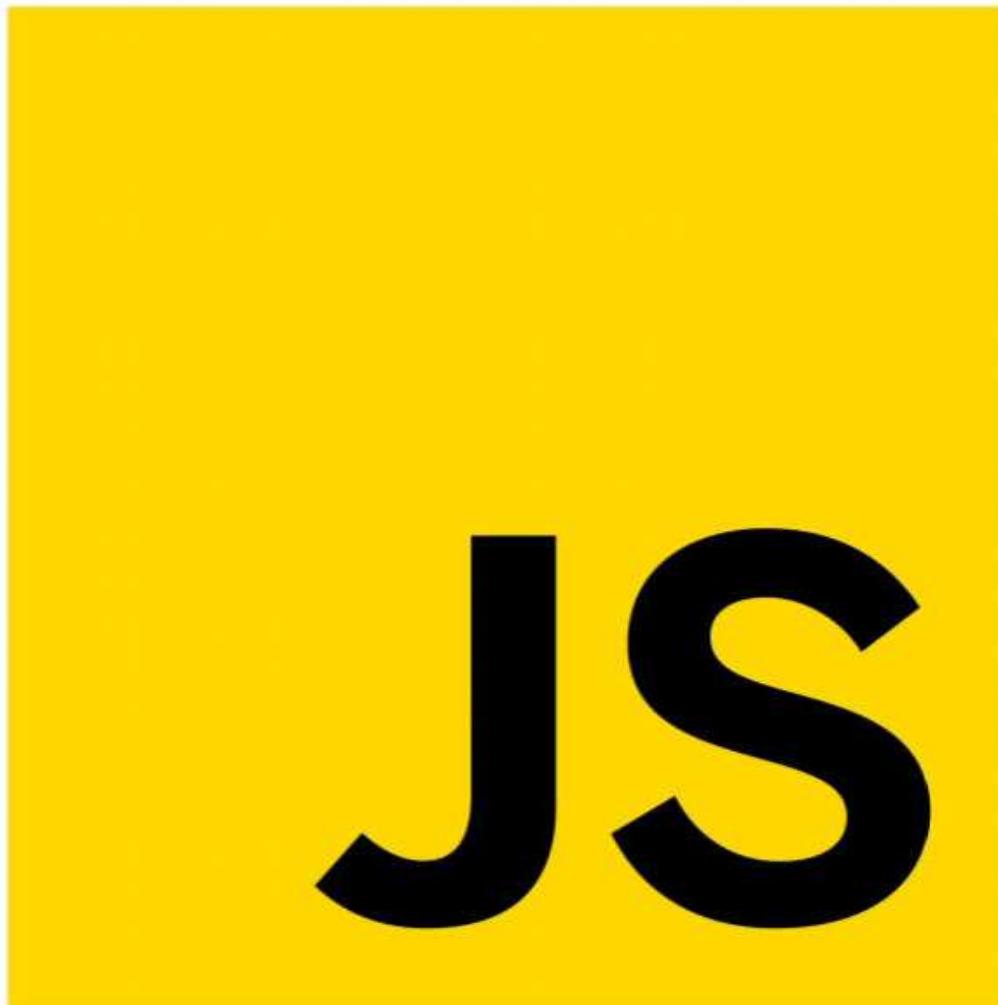
Outside and below the 'initReading' function, we can add the event listener to the read more button and fire off the 'initReading' function when the click event happens. Lastly we will use the 'window.onload' function and set that equal to our 'loadPage' function - this will execute as soon as the browser window is loaded up.

```
button.addEventListener("click", initReading);
window.onload = loadPage;
```

18. Image content

Back in our CSS, we will create some default styles to the text that will appear once the trainer's image has animated off. We want this text to be positioned at the centre of our content area and we can achieve this by using 'transform: translate(-50%)'. Then we will set the visibility of this to 'hidden' so we can reveal it in a later step together with some animation.

```
.is-img--content {
```



Using ES6 syntax

JavaScript is everywhere these days and it's a skillset that most (if not all) front-end developers should at least know the basics of. As we all know, JavaScript was given a significant update from ES5 to ES6. Throughout this tutorial we used the ES6 syntax for the little bit of JavaScript we wrote. So in case you haven't learned the basics of ES6, then what we did might have been a bit confusing to you, so let's clear a few things up.

Arrow Functions

```
const loadPage = () => {
  body.classList.add("load-page");
}
```

Arrow functions are a more concise syntax for writing our functions and happen to be the most used feature of ES6. They are one-line mini anonymous functions and help keep our code tidy. One of the features of arrow functions is when you only have one parameter to pass in, such as the event (e), then you don't actually have to use the parenthesis.

```
const initReading = e => {
  let txt = e.target;
  body.classList.toggle("is-reading");
}
```

Const and Let

These are two new ways of declaring variables. Let is block scoped, meaning it is only available to use within the block of code it is in. Variables declared with const are also block scoped but cannot be re-declared.

```
position: absolute;
top: 50%;
transform: translateY(-50%);
z-index: 5;
visibility: hidden;
```

19. Paragraph animations

When the content area is made visible and the main image has animated off, we want the text to animate up and into view. However, at this point, we will set the opacity to zero and then in a later step we will set it back to one, which will give us a nice transition.

```
.is-img--content_inner {
  padding: 0em 1em;
  p {
    transform: translateY(60px);
    position: relative;
    opacity: 0;
    transition: all 500ms ease;
  }
}
```

20. Finishing up

There's quite a lot going on in this last piece of CSS and it will glue everything together. The delay in transitions gives us a lot of control of how our elements appear, and the main one to take notice of is the 'is-img--content' rule where we set the visibility back to visible.

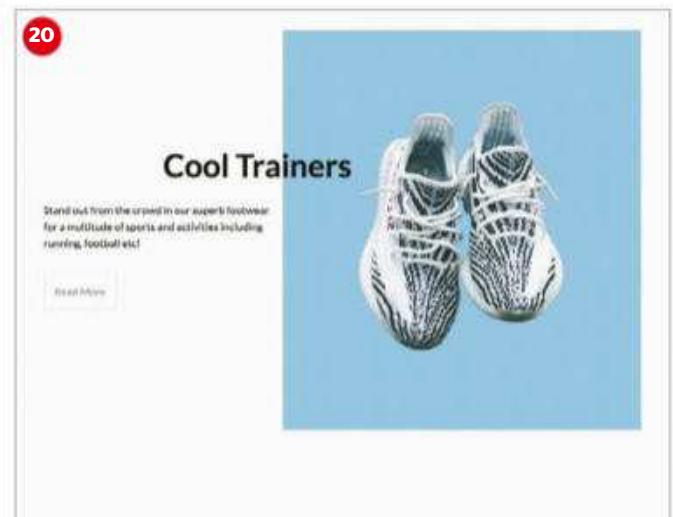
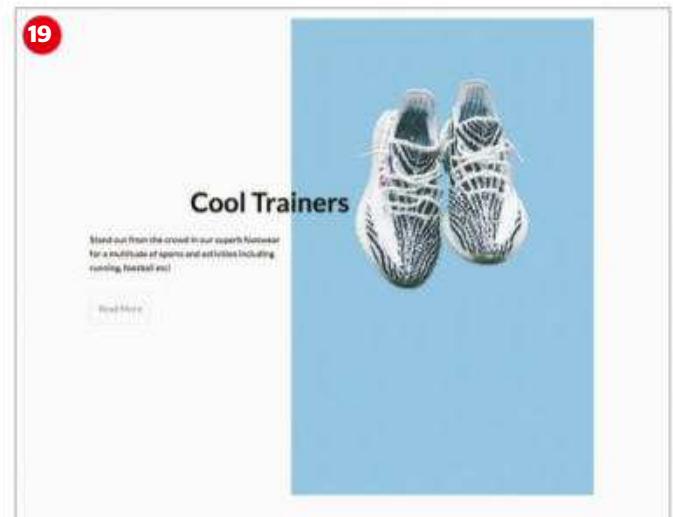
```
.is-reading {
  .bg {
    width: 0%;
    left: auto;
    right: 0;
```

```
transition-delay: 400ms; }
h2 {
  font-size: 2.3em;
  margin: 0;
  position: relative;
  transform: translateX(0);
  transition-delay: 300ms;
  transition: all 400ms ease; }
.trainers {
  img {
    transition-delay: 00ms;
    transform: translateY(20px);
    opacity: 0;
    visibility: hidden; } }
.is-img--content {
  visibility: visible; }
.is-img--content_inner {
  padding: 0em 1em;
  p {
    transform: translateY(0px);
    position: relative;
    opacity: 1;
    transition-delay: 800ms;
    &:nth-child(2) {
      transition-delay: 900ms;
    }
  }
}
```

21. Reactive animations

As you've seen from following along to this tutorial, UI animations are pretty easy to create just from using CSS and a touch of JavaScript. There's two types of UI

animations: static or reactive. What we've done is a reactive UI animation as it's based on an event that triggers the animation, and these can create a truly interactive user experience.



web workshop

Create an animated wavy line effect

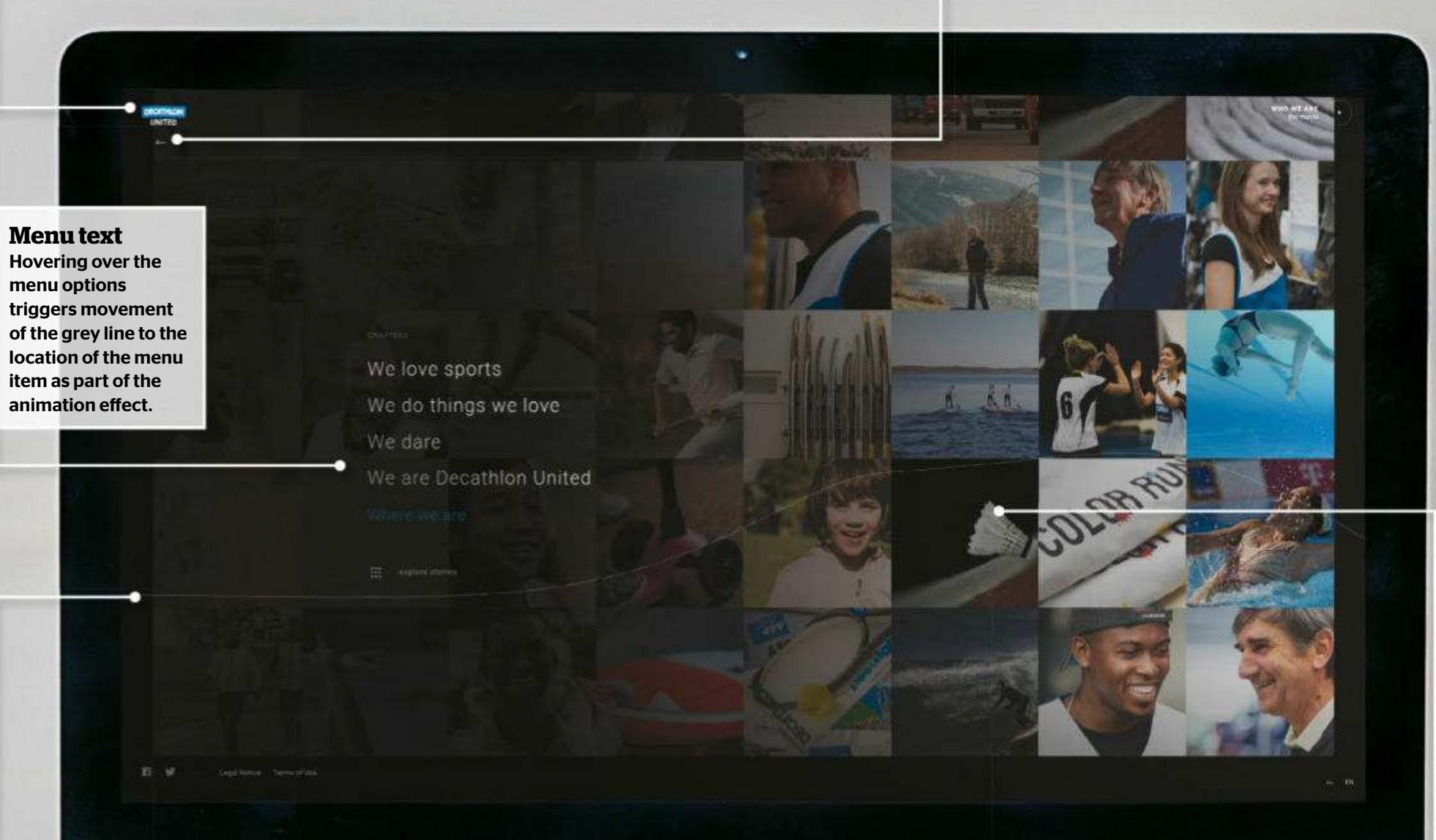
Inspired by www.decathlon-united.com/en/chapters

Logo's placement

The brand logo is placed in the top-left corner of the web page - allowing users to return to the menu from any selected page.

Home return arrow

Allows the user to return to the home page of the website. An arrow pointing in this direction is understood to mean 'go back'.



Menu text

Hovering over the menu options triggers movement of the grey line to the location of the menu item as part of the animation effect.

Wavy line

This line becomes an animated wave pattern that provides a response to the user's mouse movement interactions with the page.

Background image

The background image also responds to the user's mouse interactions, moving in the opposite direction to provide a sense of depth.





<comment>

What our experts think of the site

Illusion is a matter of perspective

The trick for finding simple answers to complicated problems can often be found through looking at the details with a different perspective. In this case, the illusion of wavy lines was simplified by looking for CSS features that include curves that can be presented with the required ‘wonkiness’.

Leon Brown, developer and author of e-learning content at nextpoint.co.uk

```
transform-origin: 50% 45%;  
animation: spin 5s infinite linear;  
}
```

Technique

1. Document initiation

The first step is to initiate the HTML document. This is the HTML consisting of the document container used to store the head and body sections. While the head section is used to reference the external CSS stylesheet, the body section is used to store content created in step 2.

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

2. HTML content elements

The HTML content consists as a navigation container that stores two span elements. The first span will be used for the wavy line effect, while the second span is used to contain the list of links to be made visible.

```
<nav>  
    <span></span>  
    <span>  
        <a href="#">One</a>  
        <a href="#">Two</a>  
        <a href="#">Three</a>  
    </span>  
</nav>
```

3. Initiate the CSS

Create a new file called ‘styles.css’. The first step of this file initiates the presentation of the main document containers – ie its HTML and body. These are defined to cover the full screen with a black background without border spacing.

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

4. Navigation container

The navigation container is set to display at a specific size

and with overflow set to hidden. This is important for the effect to work, as the wavy line is an illusion created from only part of the spinning box being visible.

```
nav{  
    position: relative;  
    width: 300px;  
    height: 300px;  
    overflow: hidden;  
}
```

5. Position size

The illusion is created using the navigation’s first child element as a box that’s sized and positioned to be partially visible. Percentage-based measurement is used for co-ordinates and sizing to guarantee this.

```
nav > *:first-child{  
    position: absolute;  
    top: -80%;  
    left: -30%;  
    width: 160%;  
    height: 160%;  
}
```

6. Presentation

The box is only required for its border outline, hence using a transparent background and slim solid border. The illusion of the wavy line is made from the box having curved edges using the border radius. A slightly off-centre origin is applied in order for a ‘wonky’ spin animation to be applied.

```
nav > *:first-child{  
    background: transparent;  
    border: 1px solid rgba(255,255,255,.5);  
    border-radius: 30%;  
}
```

7. Top-left corner

The second span that contains the text content is required to be positioned in the top-left corner of its parent nav – above the previously defined box element. This step also defines other properties for text elements to inherit such as the font size and text alignment.

```
nav > *:last-child{  
    position: absolute;  
    left: 0;  
    top: 0;  
    width: 100%;  
    z-index: 1;  
    text-align: center;  
    color: #fff;  
    font-size: 2em;  
}
```

8. Navigation links

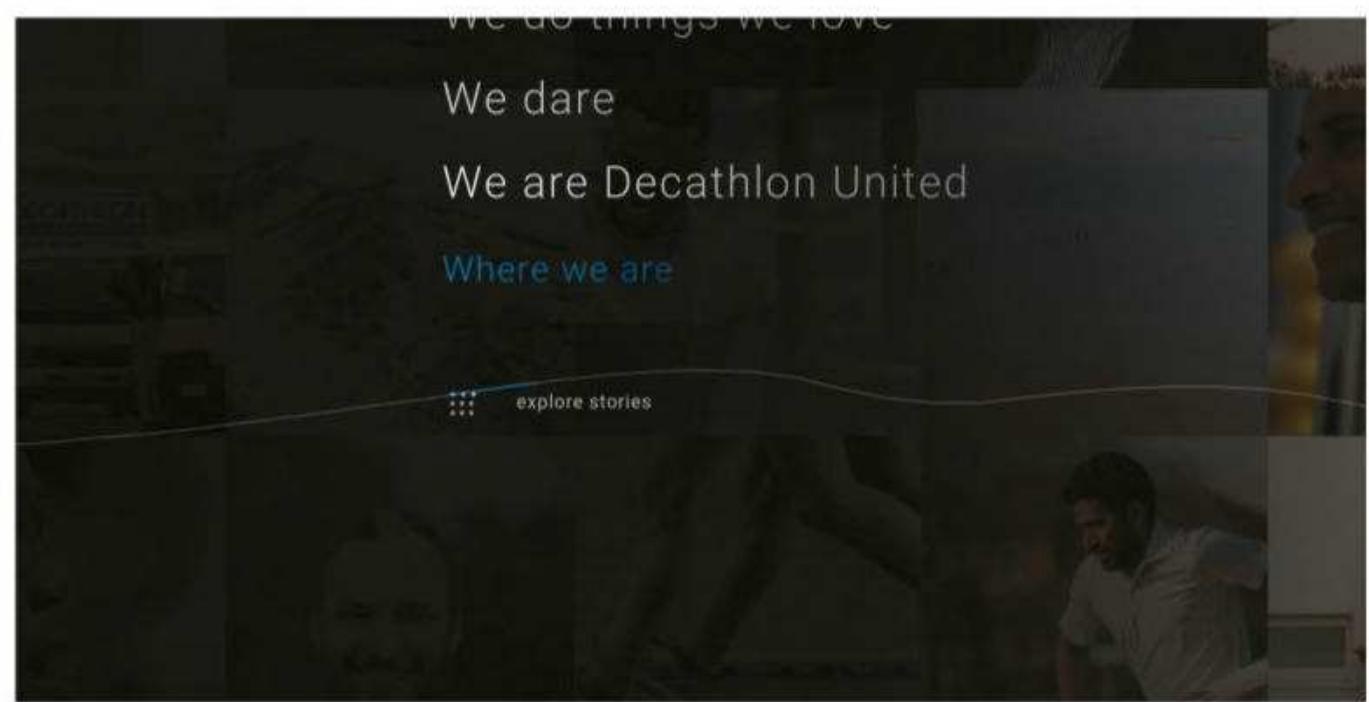
With this example being to display a menu, this step specifically sets link items inside the second/last child of the navigation to be displayed as a block element. This sets each link to appear on a new line, along with a margin applied for spacing and colour set to white.

```
nav > *:last-child a{  
    display: block;  
    text-decoration: none;  
    color: #fff;  
    margin-top: .5em;  
}
```

9. Spin animation

The spin animation referred to in step 6 is made from two frames that result in backwards rotation from 360 degrees. An illusion of a wavy line is presented when this animation is applied to the partially visible box with off-centred origin.

```
@keyframes spin {  
    from{ transform: rotate(360deg); }  
    to { transform: rotate(0deg); }  
}
```



POWER UP YOUR PROGRESSIVE WEB APPS

**Get yourself 23 tips to ensure that your mobile designs
are the very best they can be**

Progressive Web Apps (PWAs) are part of the current DNA for the mobile, web and desktop app space. Any website can now be an installable offline-capable app thanks to new technologies and APIs. They even offer similar experiences to native apps once installed. To provide the best possible experience, designers need to make an effort to provide a fast and mobile-optimised user experience, helping the user to navigate, install and use their apps. With app store distribution support now available, web design professionals need to pay close attention to every single detail when they offer a web-based progressive app. Here's how...

1 Understand the reach of your PWA

To create the best possible experience, testing your user interface in all possible user platforms is essential. While a PWA will work on every browser, to leverage the advantages of PWA as a standalone, offline-capable app, there are a list of compatible platforms and browsers. On iOS, it will be compatible with iPhone, iPod touch, and iPad since iOS 11.3 using just Safari. On Windows 10, Microsoft Store will publish your PWA if Bing crawls it. On Chromebooks, they are fully installable from Chrome OS 67. On Android, the best experience comes with Chrome, but PWAs can also be installed from Samsung Internet, Firefox, Edge, Opera and UC. On feature phones, KaiOS supports PWAs from an internal store. Finally, Chrome for desktop (Windows, macOS, and Linux) will have support for installable PWAs soon; they're available to test today by enabling the flag 'Desktop PWA' from `chrome://flags`.

2 Help the user install the app

When a user browses a site the resources from the PWA will be installed thanks to Service Workers, but the app is not installed in the operating system automatically. Not every user will know that installation is available or how to trigger that action. Promoting the action through a non-intrusive hint might help installation if you show that advice on the 'display-mode' browser media query.

In some browsers, triggering the installation dialogue is now possible from the web's user interface in the form of an 'Install' menu or button. The 'beforeinstallprompt' event on the window object fires when the browser recognises a PWA, and it's ready to show the installation prompt. Using the event's argument, the handler should cancel default browser's behaviour with 'preventDefault()' and save the argument for later in a global variable. When the user clicks on the custom UI element, the app should call 'prompt()' on the saved variable.



3 A new responsive breakpoint: Mini mode

Typical responsive breakpoints expect a minimum viewport of a mobile phone – 320 x 400 CSS pixels. But with PWAs on the desktop, there is a whole new dimension of breakpoints we need to consider. An installed PWA uses a resizable window of any size, including a tiny widget-like size, such as 200 x 100 CSS pixels. A mini mode can be created for this

unique situation, which will only render a summary of the content and quick actions for the user to keep it visible all the time.

Small viewports don't end with desktop PWAs – typical KaiOS feature phones supporting PWAs, such as the latest Nokia 8110, are exposing a viewport of 240 x 228 on a non-touch screen. And also, from Apple Watch Series 3 with watchOS 5, there is a web browser available when the user clicks on a link, exposing small viewports when adding the tag '`<meta name="disabled-adaptations" content="watch">`'.

4 Provide good PWA icons

A PWA will be recognisable by its icon and the short name defined in the manifest. On today's platforms, only PNG icons are supported. For app stores and launch screens, a 512 x 512 pixel icon is mandatory, and a 1024 x 1024 version is recommended. Android uses 48dp transparent icons; here the resolution defines the actual size. On most Android devices, 192 x 192 (4x density) and 144 x 144

(3x density) will be needed. Adaptive Icons available from Android 8 is a new way to create icons using a variety of shapes to be used across different device models, but this format is not yet available on PWAs.

iOS is entirely different as it doesn't support transparent icons. The PWA must supply square icons with no transparency and no rounded corners (iOS will round them). And the icons must be defined through a '`<link rel="apple-touch-icon">`' element in the HTML, and accept the standard sizes property.

5 Rendering the right content for other apps

Users of mobile devices share content on social networks and messaging apps, such as Facebook, Twitter or WhatsApp. All mobile browsers have a share action item that will trigger the native share dialogue in the operating system. A PWA can also use WebShare, so it's important to keep your content compatible to increase click opportunity. The first important

definition you need is the canonical URL. Sometimes apps add data to the URL, such as tracking information or have different URLs for mobile/PWA and desktop content.

The ideal situation is to share a canonical URL that will be multi-platform; to define that URL a PWA can add a '`<link rel="canonical" href="">`'. Then, Open Graph is the standard of defining metadata that all social networks use for creating the content visual snippet users see when browsing social apps. The following meta tags are crucial: '`og:title`', '`og:description`', '`og:url`' and '`og:image`'.

6 Web Share

When a PWA is used in standalone or fullscreen mode there is no URL bar or browser's UI on mobile operating systems which means a user can't share the current content via message or social networks. On browsers, such as Chrome on Android the app can use the recent Web Share API that will trigger the native share dialog from a custom user interface element. On a click handler, the app will

trigger a code similar to `navigator.share({ 'title': document.title, 'url': location.href })`.

For other browsers, a PWA can use deep links to communicate with native apps. For example, if a link navigates to `https://wa.me/<phone-number>/?text=<text>`, the WhatsApp app will be opened and will open a chat to the specified phone number and will fill the specified text. If no phone number is specified, the user will pick a contact from a dialog.

7 App's lifecycle

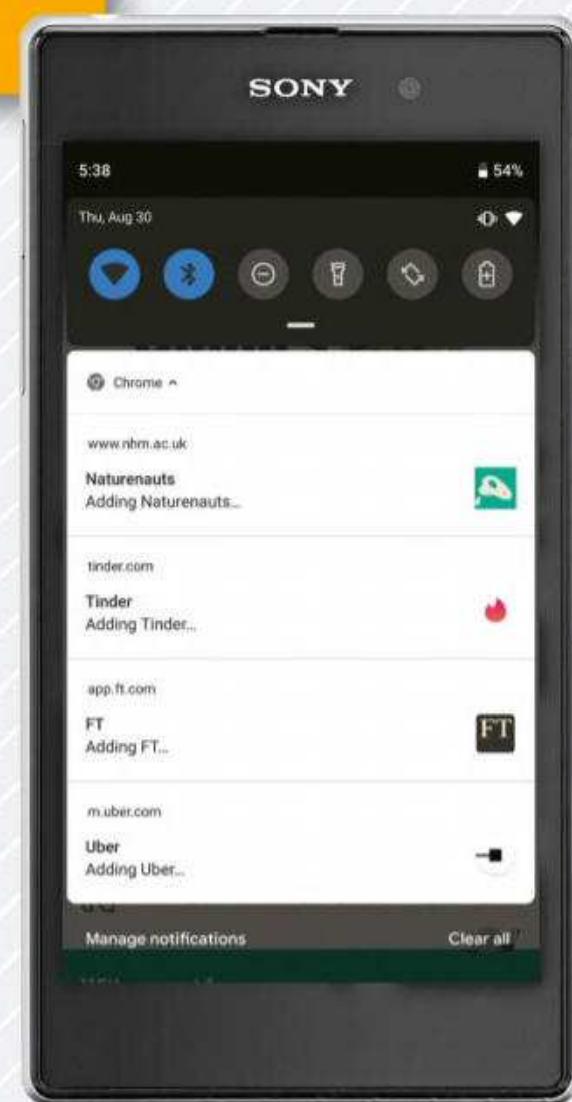
In traditional web development a website is in memory and execution, while the tab is opened even if it's not on the screen. In the PWA world users don't close apps, they get out of them with the home button or a gesture. Understanding the lifecycle is critical to providing a good user experience to your audience.

On iOS, a standalone PWA gets discarded every time the user moves to another app or home screen, and can be detected through Page Visibility events.

Using Chrome or a standalone PWA on Android, the web content will freeze when in the background (a 'freeze' event will be fired) and might resume later. Sometimes the user gets back to the app after several hours or days; at that point, the content is discarded from memory, and it will load from scratch. Checking '`document.wasDiscarded`' and reloading the previous state of the app will create a consistent experience.

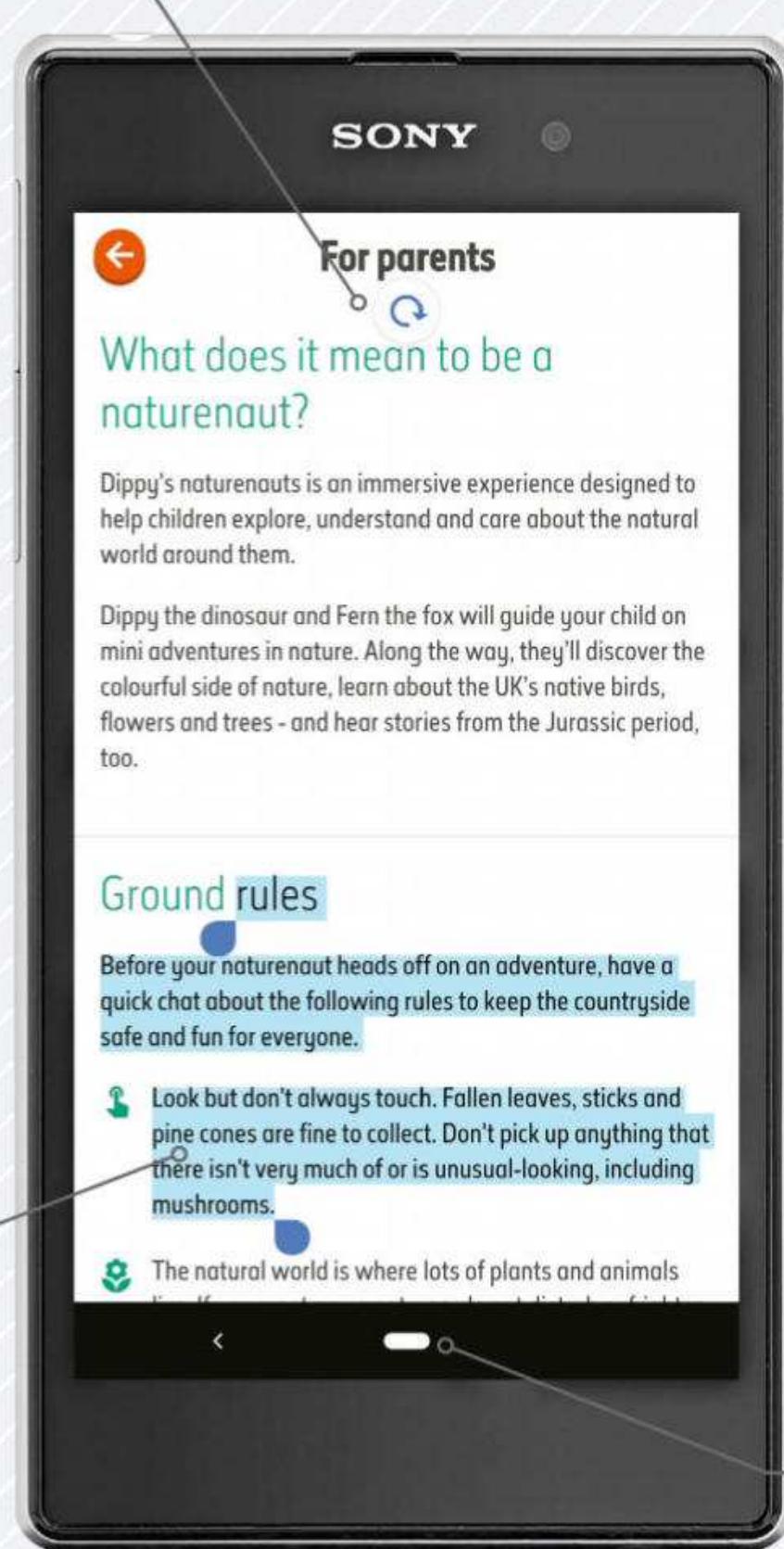
8 Storage limits and persistency

When a Service Worker installs all the assets from your PWA, these questions typically appear: how much space can the app use and for how long? On iOS, each Service Worker can only store up to 50MB, and they will be uninstalled from the system after a few weeks of inactivity. The app's icon will still be there, so next time the content will be downloaded again. On Android, the maximum package size is a percentage of the user's available space and will be stored until there is no space. On some browsers, a PWA can request Persistent Storage via '`navigator.storage.persist()`'.



9 Control reload gesture

Some browsers offer a reload gesture when the user pulls down from the top. You can manage the pull-to-refresh gesture with the overscroll-behaviour style on the body element. Using 'contain', the reload gesture will be disabled and using 'none' will also disable the bounce effect.



11 Disable user selection

When you design an app-like web experience, several sections of the UI shouldn't be selectable, such as icons, sidebars and tab bars. To avoid selection of such elements or to trigger unwanted selections, the PWA can define the CSS style with 'user-select: none' on the elements that are not rendering data.

When you design an app-like web experience, several sections of the UI shouldn't be selectable, such as icons, sidebars and tab bars



When you need a mobile or desktop app that typically consumes web content, PWA is the ideal platform; if you go native, you will be developing an expensive browser

Maximiliano Firtman
Mobile & web developer, book author
<https://firt.mobi>, @firt

10 Provide navigation within UI

When a user accesses a PWA in standalone mode, there is no browser user interface, and some platforms, such as iOS, don't have back buttons or gestures to offer navigation. Therefore, it's essential to provide all the possible navigation of the app within the boundaries of the web content.

12

Prioritising resources

Web performance is one of the critical aspects of a great Progressive Web App and helping the browser to render the content faster should be a designer's top priority. Using '`<link>`' elements to specify resources that are important for the rendering critical path will make the browser start downloading them as soon as possible, even if it doesn't know where to use them yet. For example, if the app uses Web Fonts, to save precious milliseconds and set a high priority for it within the browser's engine, it can add `<link rel="preload" as="font" href="font.woff" crossorigin>` to the HTML and the font file will be downloaded before even the browser has parsed the CSS file with the '`@font-face`' declaration. If the app doesn't know the actual file it will need initially, but it knows that it will come from a third-party server, the `<link>` can use '`rel="preconnect"`' or '`rel="dns-prefetch"`' to speed things up.

13

Reactive web performance

Responsive Web Design typically takes the viewport size, if it's a touch device and if the high-contrast mode is enabled to change layout, but in a dynamic world with so many devices and connections, we need to provide a consistent and pleasant user experience across all users no matter what. For example, if the user is in a bad 3G connection, or with an LTE+ connection but with a cheap Android Go device with low CPU, memory, and GPU, we might not want to deliver the same content as to a high-end device on a good Wi-Fi, even if the available viewport size is the same. With modern APIs we can query client and server-side about total memory RAM available, current network bandwidth and latency, battery level and rendering metrics. Based on that information, we might want to react and apply a different layout, image quality or amount of content loaded.

14

Modern image and animations delivery

One of the first problems with images on a PWA is not sending the appropriate size for that viewport and pixel density, thus delivering a bigger file that is necessary. The app should use '`<picture>`' or the 'Client Hints' spec on the server to provide responsive images. But also using new formats and compression algorithms will save bytes and rendering times, such as using the WebP format (evolving into the EVIF format for the future) or encoding your JPEG with the new Guetzli encoder or your PNGs with Zopfli encoder. Regarding animations, designers should replace GIF files with Animated PNGs available on Chrome, Firefox and Safari, or should use muted MP4 video files saving more than 90% of data and creating a 20x faster experience. While some modern browsers will accept muted MP4 files in an image, a PWA should have a fallback using '`<video muted -webkit-plays-inline autoplay>`' for the animation.

Using new formats and compression algorithms will save bytes

Resources

Android Studio

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

It includes the SDK that let web designers emulate PWAs on Android devices and Chromebooks (additional plugin required).



PWA Builder

<https://bit.ly/2wwx34V>

It helps convert a website into a PWA, and it creates packages for Google Play, App Store and Microsoft Store.



15 Render cheap and fast

After the browser downloads and parses all your HTML, CSS and JavaScript, the next important thing for performance is the rendering process. Based on how you are designing the project, the rendering can be faster or slower, it may use hardware acceleration and it can create a smoother experience while scrolling or animating. One quick tip is to specify to the rendering engine which properties of an element will change in the future, because of an event, a transition or an animation. For example, a web designer can specify 'will-change: opacity' on every element that they fade in or out, so the browser can optimise the engine for that. Also, CSS Containment spec lets a designer limit a browser's ability to style, layout, and paint on some regions of your document. For example, 'contain: strict' can be used on areas with predefined dimensions that won't affect the rest of the page and 'contain: content' is useful as a default when the size is dynamic.

16 Use the platform's abilities

When creating PWAs authors don't have access to just a screen and an input method. There is a good range of sensors, hardware and native apps available for the app to interact with and to create a better user experience. PWAs can access on all platforms to the user's location, accelerometer, magnetometer (compass), gyroscope, orientation events, images from the gallery and to capture still pictures from the camera. Using Progressive Enhancement (that's the 'P' in PWA) we can also use other hardware that might not be available on all platforms, such as Bluetooth, one-click payments, authentication, push notifications, VR and AR, raw live video from the camera, audio output, speech recognition and more. On PWAs for Windows Store, JavaScript has access to any native API to integrate with the system, such as creating custom tiles for the start screen and accessing a user's contacts or calendar.

17 Emulate and debug your PWA

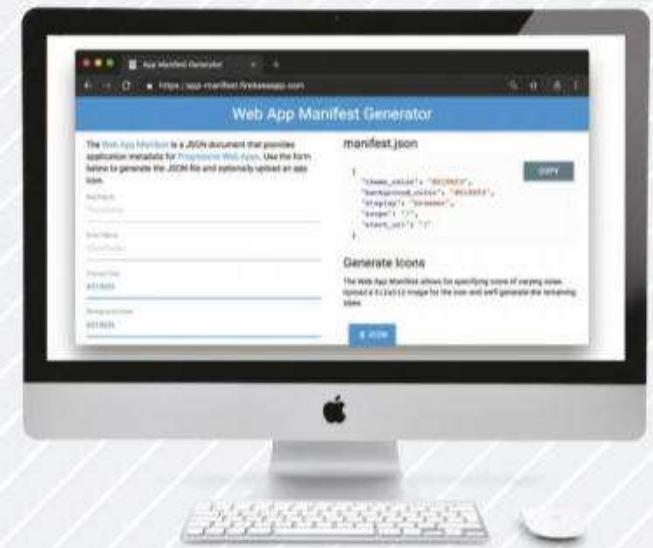
Testing the PWA on all the possible scenarios is a must-do for every web designer, but owning all the devices is almost impossible. That's why there are a range of tools that will let designers emulate and debug without a real device. For Mac users, Xcode (available in the Mac App Store) includes the iOS Simulator where PWAs can be tested on the whole range of iDevices. On every desktop OS, Android Studio includes the ability to create as many instances of devices as we want, including combinations of screen sizes, factor and OS version. However, not every version includes Chrome and the Play Store to install more browsers, so it's important to check if the emulator contains Play Store. As a plugin for the Android ecosystem, a Chromebook emulator beta is available to download so the full desktop PWA final experience can be tested and debugged there without an actual laptop.

There are a range of tools that will let designers emulate and debug without a real device

WebHint
<https://webhint.io>
 A tool that will help with PWA's accessibility, speed and security checking your code for best practices and common errors.



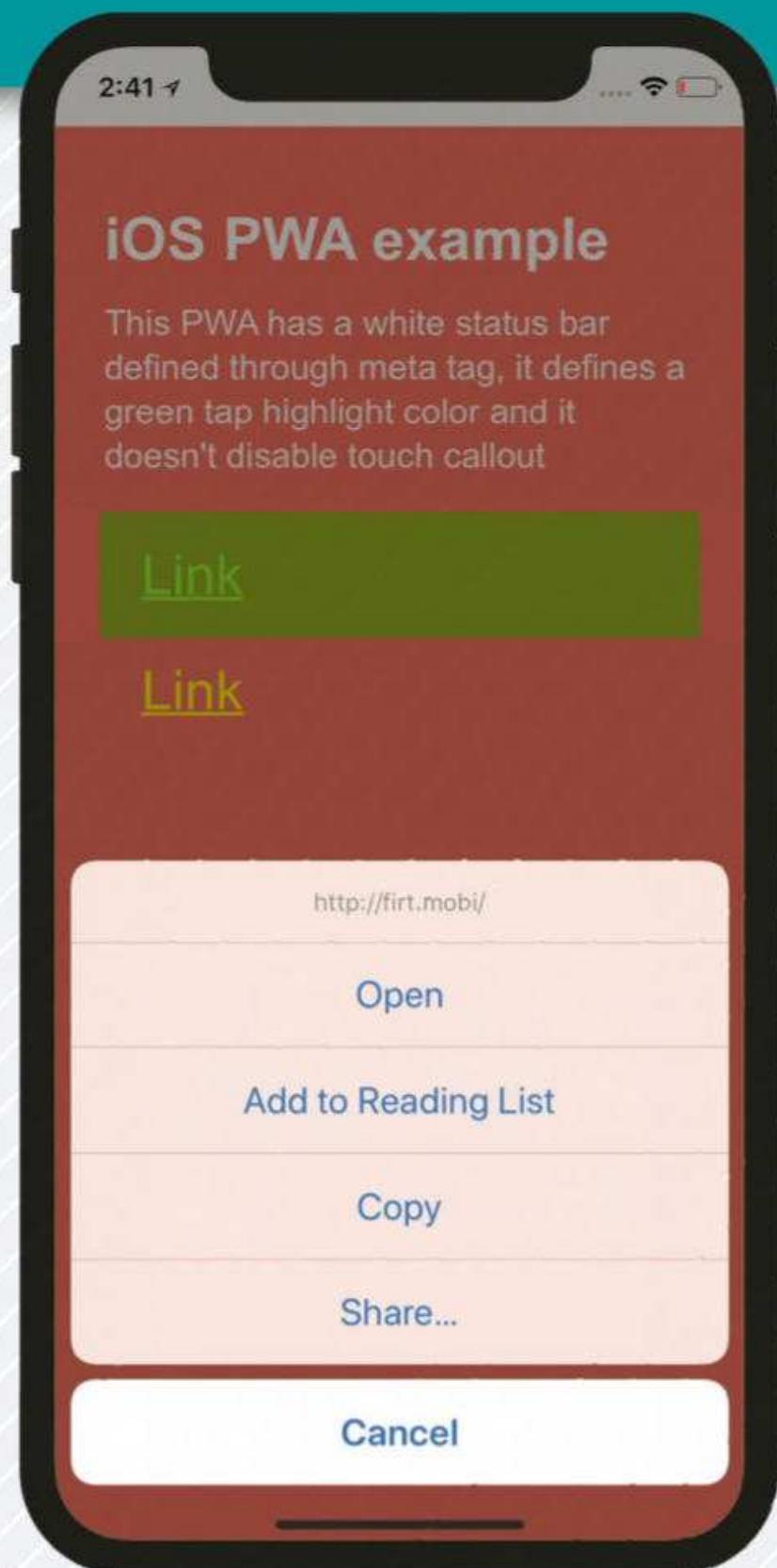
Web App Manifest Generator
<https://bit.ly/2eYETiN>
 It will create the Manifest file and effortlessly take one icon and create all the necessary versions for Android and Windows.



18 Launch images

All Android browsers compatible with PWAs will create a launch screen from your manifest's declarations automatically using an icon, the background colour and the app's name. But iOS doesn't generate those launch screens automatically, and PWAs need to set them up manually through link elements. That image is used on the opening animation and also in the multitasking mode, with a

white image as default. The PNG image has to be precise to the size of the actual screen, so several files must be served on different link elements with the right size property, such as '`<link rel="apple-touch-startup-image" sizes="750x1334" href="">`' for iPhone 6S, 7 and 8. For 7+, 8+ and iPads we also need to provide a landscape version and define it with a media query. If you don't want to offer all the versions manually, libraries such as AppCompat can help with generating them on the fly.



19 iOS user interface tricks

iPhone and iPad users need particular attention when designing a PWA to improve the user experience. Apple doesn't support the 'theme-color' property in the manifest, but we can specify status bar style through a meta tag: `<meta name="apple-mobile-web-app-status-bar-style">` accepting 'black' or 'white' as values. Then, when a user taps in a clickable area such as a

link, a highlight box appears over the element. That colour can be replaced with '`-webkit-tap-highlight-color` always' using a semi-transparent colour such as `rgba(255, 0, 0, 0.4)`. Finally, every link (including JavaScript internal links) in iOS will trigger a pop-up with more data about it using the long-press gesture, unless the designer applies the style '`-webkit-touch-callout: none`'. Finally, for Safari on iPad, we can specify not to render the content as a fullscreen iPad while in Side or Split View (multi-app view) using the viewport's declaration '`shrink-to-fit=no`'.

20 Limitations on iOS

While Service Workers and Web App Manifest are supported since iOS 11.3, there are some UX limitations compared with Android. It's essential for every designer to know the barriers to make decisions for iOS users. Apps will be reloaded every time the users get out of them, so authors should note that

two-factor authentications or OAuth login designs on different domains might not work correctly for PWAs sitting in the home screen. Some APIs that are available in Safari might not work correctly from the home screen, such as WebRTC, because it's using a slightly different engine. It's essential to test every feature and not to think that if it works in Safari, it will work in the PWA. Safari doesn't offer any install API or Web App Banner, so the only way to install it is to manually open the share sheet and pick 'Add to Home'.

It's essential to test every feature and not to think that if it works in Safari, it will work in the PWA

21 Avoid unwanted translations

Chrome supports web content's translation when the content and the user's language don't match. The problems appear when that behaviour also happens in a standalone app on every access to it and not just on web content. So if the PWA offers internationalisation within the app, or if the author doesn't want auto-translation, it can disable it through adding '`<meta name="google content="notranslate">`' in the HTML. If the author wants translation within the browser but not as an app, that meta tag can be injected dynamically only when a media query resolves, such as using the following code:

```
var isApp =
  matchMedia('(display-mode: standalone)')
    .matches;
var head = document.querySelector("head");
if (isApp) {
  head.innerHTML +=
    '<meta name=google content=notranslate>'; }
```

22 Notch and iPhone X support

If the PWA wants to take advantage of the notch available on iPhone X and some Android devices such as LG G7, it needs some work. By default, the designer can't use that space, and bars or just a colour will replace it. To use that space the viewport tag has to include '`viewport-fit=cover`' and then add proper padding values to avoid rendering content on the invisible notch area, rounded screen corners or non-clickable areas for global gestures. To make that padding compatible with all devices, CSS constants for the safe zone are available, such as '`env(safe-area-inset-top)`' for the top padding. Note that before iOS 12, the CSS '`env`' function was named '`constant`', so providing both names for some time will be necessary. To use the full screen and render the app from the very top of the display on home-screen apps, the meta tag '`apple-mobile-app-status-bar-style`' needs to use the value '`black-translucent`'.

23 Link capturing

On Android with Chrome, PWAs have enabled a feature known as Link Capturing thanks to WebAPK, a feature that creates an actual Android native package when adding the app to the home screen. With that feature, the scope of the PWA defined in the manifest becomes essential not just to open external links in the browser but also because the PWA will own that scope within the OS. So, let's say the user installs with Chrome on Android a PWA from <https://mydomain.app/mypwa>. After it was installed, every link in that device pointing to a URL will be opened in the PWA window and not in the browser. So developers must assure that the PWA should adequately manage every link sent on social networks or emails within that scope. The feature is also available on desktop OSs with Chrome under a flag at the time of publishing this article.



5 to follow

PWA List from Maxim Salnikov

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

A list of the most important 100 Twitter accounts on PWAs.

PWA Daily

[@pwadaily](#)

Curated content and articles on Progressive Web Apps every day.

Intent to Ship

[@intenttoship](#)

A quick way to get the latest information about new features on every browser.

Chromium Developers

[@chromiumdev](#)

The largest source for the latest APIs on Chromium-based and other browsers.

MDN Web Docs

[@mozdevnet](#)

Web Platform documentation updates published by the main browser's vendors.

`env(safe-area-inset-left)`

`env(safe-area-inset-right)`

`env(safe-area-inset-bottom)`

Safe Area

5

THINGS YOU NEED
TO KNOW ABOUT

expressJS



Creating browser-facing applications with Node.JS gets tedious. Express.JS is a framework dedicated to single-page and multi-page applications hosted in the Node.JS environment



```
Usage: express [options] [dir]
```

Options:

--version	output the version number
-e, --ejs	add ejs engine support
--pug	add pug engine support
--hbs	add handlebars engine support
-H, --hogan	add hogan.js engine support
-v, --view <engine>	add view <engine> support (dust ejs hbs hjs jade pug twig vash) (defaults to jade)
--no-view	use static html instead of view engine
-c, --css <engine>	add stylesheet <engine> support (less stylus compass sass) (defaults to plain css)
--git	add .gitignore
f, --force	force on non-empty directory
-h, --help	output usage information

1. Generate a workable structure

Introduce the right working options to get started

Express.JS prides itself in being ‘unopinionated’ - ie the framework allows the developer to mix and match in terms of architectures, templating and markup engines. Sadly, great power comes with great responsibility.

The Express developer team seeks to soften the blow by introducing a project generator. It comes to your workstation in the form of an NPM package, and will aid our experiments with the following framework:

```
tamhan@tamhan-thinkpad:~/Desktop/
Stuff/2018Aug/FutureExpressJS/
workspace$ sudo npm install
express-generator -g
```

The generator also contains dozens of project options - the figure accompanying this step shows the full help output. For simplicity’s sake, we will limit ourselves to a project based on the default settings. Kick off its generation process with:

```
tamhan@tamhan-thinkpad:~/Desktop/
Stuff/2018Aug/FutureExpressJS/
workspace$ express futuretest
```

Warning: the default view engine will not be Jade in future releases

When done, the current working directory contains a new folder called ‘futuretest’. It is home to our test project and must be configured using NPM’s package download command. At the time of writing, the generator includes the Jade view generator - the project plans to change this in

the near future, obligating you to pass in a parameter selecting the view engine intended. Alternatively, request the use of Pug - it is the official successor of the Jade engine:

```
cd futuretest/
npm install
```

Understand the application structure

Now that the project generator has done its thing, let us open App.js in an editor of choice. Its - much abridged - structure presents itself as following:

```
var indexRouter = require('../
routes/index');
var usersRouter = require('../
routes/users');
var app = express();
// view engine setup
app.set('views', path.join(__
dirname, 'views'));
app.set('view engine', 'jade');
app.use(logger('dev'));
app.use(express.json());
app.use(express.urlencoded({
extended: false }));
app.use(cookieParser());
app.use(express.static(path.
join(__dirname, 'public')));
```

Express.JS is highly modular. App.js serves as an entry point, where one or more ‘use()’ functions allow the adding of components intended to handle various requests. Invocations of ‘set()’ let you adjust parameters in the engine - one of which is the installation of the Jade view

engine mentioned in the previous step.

The actual emission of web content takes place in the router classes. For brevity’s sake, let’s limit ourselves to Index.js:

```
var express = require('express');
var router = express.Router();
router.get('/', function(req, res,
next) {
  res.render('index', { title:
'Express' });
});
module.exports = router;
```

‘get()’ is provided with a matcher string and an event handler which gets invoked whenever a corresponding event occurs. In our case, the render method of the chosen template engine is told to return content to the browser of the user who logged in.

Run the web page

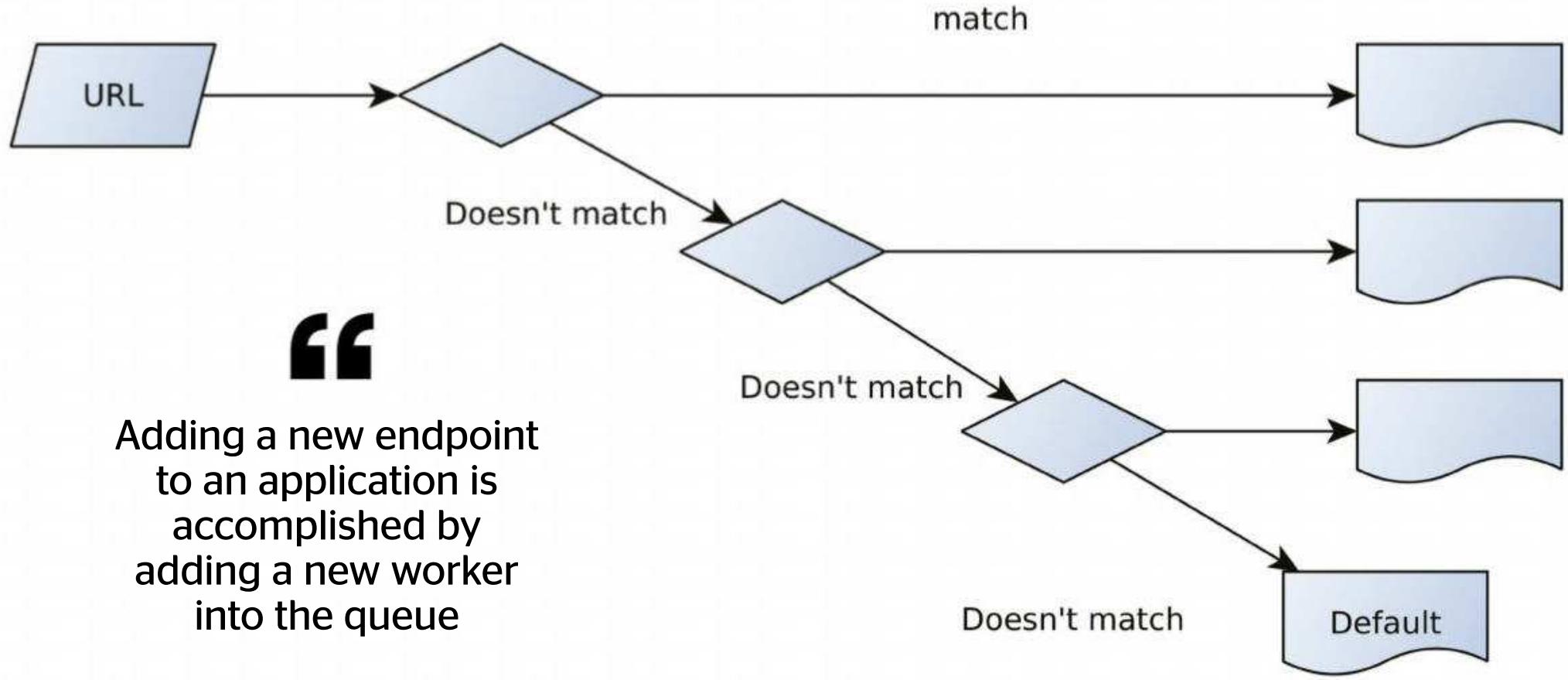
At this point, we are ready to take the website for a spin for the first time. Return to the terminal containing the Express.JS installation, and call on NPM start with the debug flag set:

```
DEBUG=myapp:* npm start
```

When done, enter http://localhost:3000/ into a browser of choice to look at the scaffolding created by the project generator. When done, press Ctrl+C to close the window and return control to the command line interpreter - keep in mind that this also closes the debugging web server.

**When done, enter
http://
localhost:3000/
into a browser of
choice to look at
the scaffolding
created by the
project generator**

“



2. It's all about routing and endpoints

Sort the right entry points, add new routes and introduce regular expression support

For the sake of simplicity, let us agree that a web application is usually made up of a sequence of entry points. Express.JS handles these via the router class - think of it as a repository of methods that get called upon in response to an incoming request (see above).

Adding a new endpoint to an application is accomplished by adding a new worker into the queue. Our auto-generated example creates two router types, each of which is raised using the 'require' method:

```
var indexRouter = require('./routes/index');
var usersRouter = require('./routes/users');
```

In the next step, 'app.use' registers the routers and connects them to the request strings. Our code furthermore adds an error handler that gets invoked if a non-existing URL is entered into the system:

```
app.use('/', indexRouter);
app.use('/users', usersRouter);
app.use(function(req, res, next)
```

```
next(createError(404));
});
```

Create a new route

Open Users.js, and modify its code as below:

```
router.get('/user1', function(req, res,
next) {
  res.send('Future says Hello
1');
});
router.get('/', function(req, res,
next) {
  res.send('respond with a
resource');
});
```

Adding new routes to Express.JS is a mechanical process. Grab the router object of choice, and invoke the method corresponding to the HTTP verb you seek to handle. Next, pass in a string that will be added to the 'offset' registered with 'app.use'. From that moment onward, both `http://localhost:3000/users/user1` and `http://localhost:3000/users/` return a valid response.

Keep in mind that Express.JS is not limited to handling 'GET' resources. 'post()', 'put()' and

'delete()' handle the traditional four HTTP requests, with dozens of additional verb methods catering to more unusual needs. Finally, the 'req' object provides access to the request header - put it to good use when parsing parameters or client information.

Advanced matching

Adding routes by hand gets tedious as program complexity grows. Express.JS caters for this problem by introducing both wildcard and regular expression support. For example, look at the following declaration which uses a regular expression to match against various strings containing the character sequence dog.

```
app.get(/.*dog$/, function (req,
res) {
  ...
});
```

Abnormal routing

While handling the four HTTP requests ought to be enough for anyone (hat tip to Bill Gates), Express.JS can also work with additional protocols. Express-WS, found at <https://www.npmjs.com/package/express-ws>, is an especially tasty candidate for this

section - it extends Express.JS's reach to include WebSocket communications.

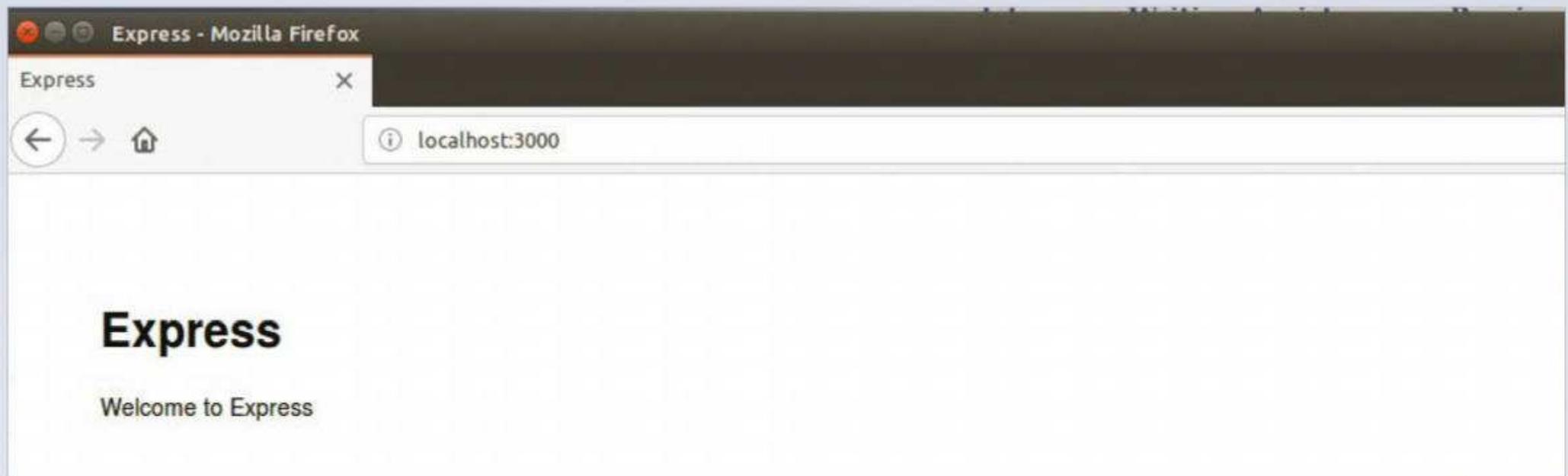
Once the plugin is added to the main Express.JS project, enabling it is accomplished via a 'require' call. It returns a helper object containing all but one method - call it to establish a connection between the router and the plugin:

```
var expressWs = require('express-
ws')
app);
```

After that, a new method called 'ws()' can be invoked to add new routes based on WebSocket technology:

```
app.ws('/', function(ws, req) {
  ws.on('message', function(msg)
{
  console.log(msg);
});
console.log('socket', req.
testing);});
```

Their prototype differs from normal routes due to the presence of the 'ws' object - it provides access to the underlying WebSocket instance connected to the client responsible for the connection.



3. Integrate databases and templating engines

Be sure to employ the power of the rich plugin ecosystem

Being based on Node.JS means that the rich plugin ecosystem is at your command when working on web-based applications. For example, accessing SQL and NoSQL databases – usually an extraordinarily tedious task – can be handled using plugins provided by the database vendors. The actual deployment is as easy as installing the needed NPM module – if your code is to access a Redis database, simply add the following:

```
var redis = require('redis')
var client = redis.createClient()
client.set('stringKey', 'aVal',
redis.print)
...
```

Of course, in-memory SQLite is also supported:

```
var sqlite3 = require('sqlite3').verbose()
var db = new sqlite3.Database(':memory:')
db.serialize(function () {
  db.run('CREATE TABLE lorem
  (info TEXT)')
```

Keep in mind that the Node.JS integration is not limited to database plugins. Adventurous developers could go as far as to include products like Tessel, thereby creating web applications which can also interact with Internet of Things devices.

Templating in style

One area where simple and real programs differ is the creation of

views. While a small example project usually uses hand-crafted strings, assembling large swaths of HTML with a string of connected things is highly annoying.

Template engines provide a neat workaround. They permit the creation of predefined schema files, which can be populated programmatically in execution.

In the case of our example program, the views lay in .jade files. Opening index reveals the following structure:

```
extends layout
block content
  h1= title
  p Welcome to #{title}
```

Expressions enclosed in curly brackets act as template fields whose values are to be replaced at runtime. Index.js invokes render with a parameter object, leading to the rendering of the start page shown in the figure accompanying this step:

```
router.get('/', function(req, res,
next) {
  res.render('index', { title:
  'Express' });
})
```

Most templating engines can also parse arrays when provided with an item template. In this case, every line of the array is displayed with one instance of the DOM model – similarities to the list display model found in Android are purely coincidental. Express.JS is not limited to the predefined

templating engines. Should you feel like rolling out your own for some reason, simply follow the steps outlined at <https://expressjs.com/en/advanced/developing-template-engines.html> – in principle, you have to override all but one function.

Handle static content

Express.JS applications tend to contain CSS files and pictures. Serving these via the Render function is inefficient – a smarter way would involve sending them on their merry way with a traditional HTTP request. This can be achieved via the 'express.static()' function, which can mark entire folders for export:

```
app.use(express.static('public'))
app.use(express.static('files'))
```

Modify the event flow

Finally, allow us to mention the term middleware shortly. In Express.JS parlance, middleware is a set of one or more components which integrate themselves into the flowchart shown opposite. They can, then, be used to modify requests as they pass through the routing system – when implemented correctly, limitless functionality can be achieved.

Furthermore, some ready components can be found at <https://expressjs.com/en/resources/middleware.html> – visit this site before embarking on a large-scale development project.

“

Keep in mind that the Node.JS integration is not limited to database plug-ins

4. How to host an Express.JS app

Consider the platforms where you can host your new creation

Testing Express.JS-based applications is easy. Problems occur once you want the page to become accessible to third parties - due to it being generated by the Node.JS environment, there is no way to get a static image fit for FTP deployment to web hosting services.

In theory, there is nothing against using a Raspberry Pi, an OrangePi, a dedicated server or a virtual machine rented from a cloud service or a web host provider that offers virtual hosting. However, renting a full virtual machine can burden you with the responsibilities of keeping the

The screenshot shows the Heroku Dyno pricing page. At the top, a banner states: "Dynos are smart, lightweight containers built for modern languages and developer productivity. Pay only for what you use, prorated to the second. Change dyno types at any time. Detailed Comparison." Below this, there are three main sections representing different usage levels:

- Free:** Ideal for experimenting with cloud applications in a limited sandbox. Features include Core Platform Features, NEVER SLEEPS, USES AN ACCOUNT-BASED POOL OF FREE DYNOS, and CUSTOM DOMAINS. It includes 512 MB RAM | 1 web/1 worker and costs \$0 per dyno/month.
- Hobby:** Perfect for small scale personal projects and hobby apps. Features include Core Platform Features, NEVER SLEEPS, FREE SSL & AUTOMATED CERTIFICATE MANAGEMENT FOR CUSTOM DOMAINS, APPLICATION METRICS, and MULTIPLE WORKERS FOR MORE POWERFUL APPS. It includes 512 MB RAM | 10 Process Types and costs \$7 per dyno/month.
- Professional:** Enhanced visibility, performance, and availability for powering your professional applications. Features include ALL HOBBY FEATURES +, SIMPLE HORIZONTAL SCALABILITY, THRESHOLD ALERTS, PREBOOT, LANGUAGE RUNTIME METRICS, and 512MB OR 1GB RAM. It includes MIX WITH STANDARDS IX, XX DYNOS, DEDICATED, AUTOSCALING, and 2.5GB OR 14GB RAM. It costs \$25 - \$500 per dyno/month.

execution environment and operating system up to date. If this task is not to your taste, a Platform-as-a-Service provider can be a more attractive (albeit, in most cases, pretty pricey) choice. Many developers consider Heroku, with its pricing shown in the figure accompanying this boxout, to be the gold standard for all things that are related to Node.JS hosting. This, however, is a bit unfair in

truth - Amazon's Elastic Beanstalk, Google's Cloud Platform and Microsoft's Azure all provide similar support for remote execution of Node.JS-based payloads. In all of these systems, the main issue is handling - while Azure is known for its slow deployments, other providers burden developers with difficult-to-use back-end services of extremely complex configuration systems.

Furthermore, the supported version of the Node.JS environment differs from provider to provider. Of course, a single page of **Web Designer** does not have enough space to cover the topic in depth. Visit <https://mzl.la/2BsAAap>, <https://bit.ly/2nSDf3k> and <https://bit.ly/2r61mwq> for some of the issues involved. Make sure to look at the provider's documentation to glean more best practices.

5. Future-proof your applications

A host of new additions need to be noted

Express.JS's development cycle is far from smooth: the developers are well-known for frequent API changes requiring rewrites of client code. The switch from 3.x to 4.x was especially painful, which is why the impending release of 5.x might leave quite a few of you feeling uncomfortable.

While Express.JS 5.0 brings along a few breaking changes, their impact is more limited. First of all, a set of already-deprecated functions is removed for real - if code still uses them, upgrading to 5.x requires maintenance.

Designers of view engines need to check 'res.render()' rank growth in regards to view renderers, which has led to some synchronous implementations slipping through. Version 5 of

the framework enhances performance by enforcing asynchronous rendering.

In addition to that, a set of sundry improvements and changes documented at <https://github.com/expressjs/express/pull/2237> sees the return of some extinct features from previous versions - furthermore, some long-standing bugs will be fixed in the new release.

Finally, be aware that you can already try the new version. Simply create a copy of your source code, grab a terminal and enter the following command to download an archive's worth of barely tested bleeding-edge JavaScript. Be safe.

```
$ npm install express@>=5.0.0-alpha.1 --save
```

The screenshot shows the Express.js 'Moving to Express 5' guide page. The title is 'Moving to Express 5'. Below it is an 'Overview' section stating: "Express 5.0 is still in the alpha release stage, but here is a preview of the changes that will be in the release and how to migrate your Express 4 app to Express 5." It notes: "Express 5 is not very different from Express 4: The changes to the API are not as significant as from 3.0 to 4.0. Although the basic API remains the same, there are still breaking changes; in other words an existing Express 4 program might not work if you update it to use Express 5." It provides instructions: "To install the latest alpha and to preview Express 5, enter the following command in your application root directory:" followed by the command: `$ npm install express@>=5.0.0-alpha.1 --save`. Below this, it says: "You can then run your automated tests to see what fails, and fix problems according to the updates listed below. After addressing test failures, run your app to see what errors occur. You'll find out right away if the app uses any methods or properties that are not supported." Finally, it lists 'Changes in Express 5' and 'Removed methods and properties'.

CREATE THE IMPOSSIBLE

www.photoshopcreative.co.uk

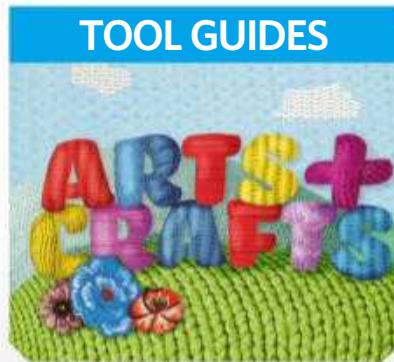
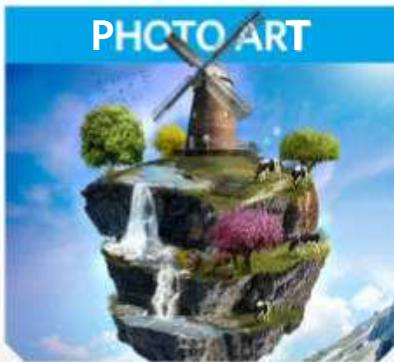


Photoshop creative

Available
from all good
newsagents and
supermarkets

ON SALE NOW

• Striking imagery • Step-by-step guides • Essential tutorials



BUY YOUR ISSUE TODAY

Print edition available at www.myfavouritemagazines.co.uk

Digital edition available for iOS and Android



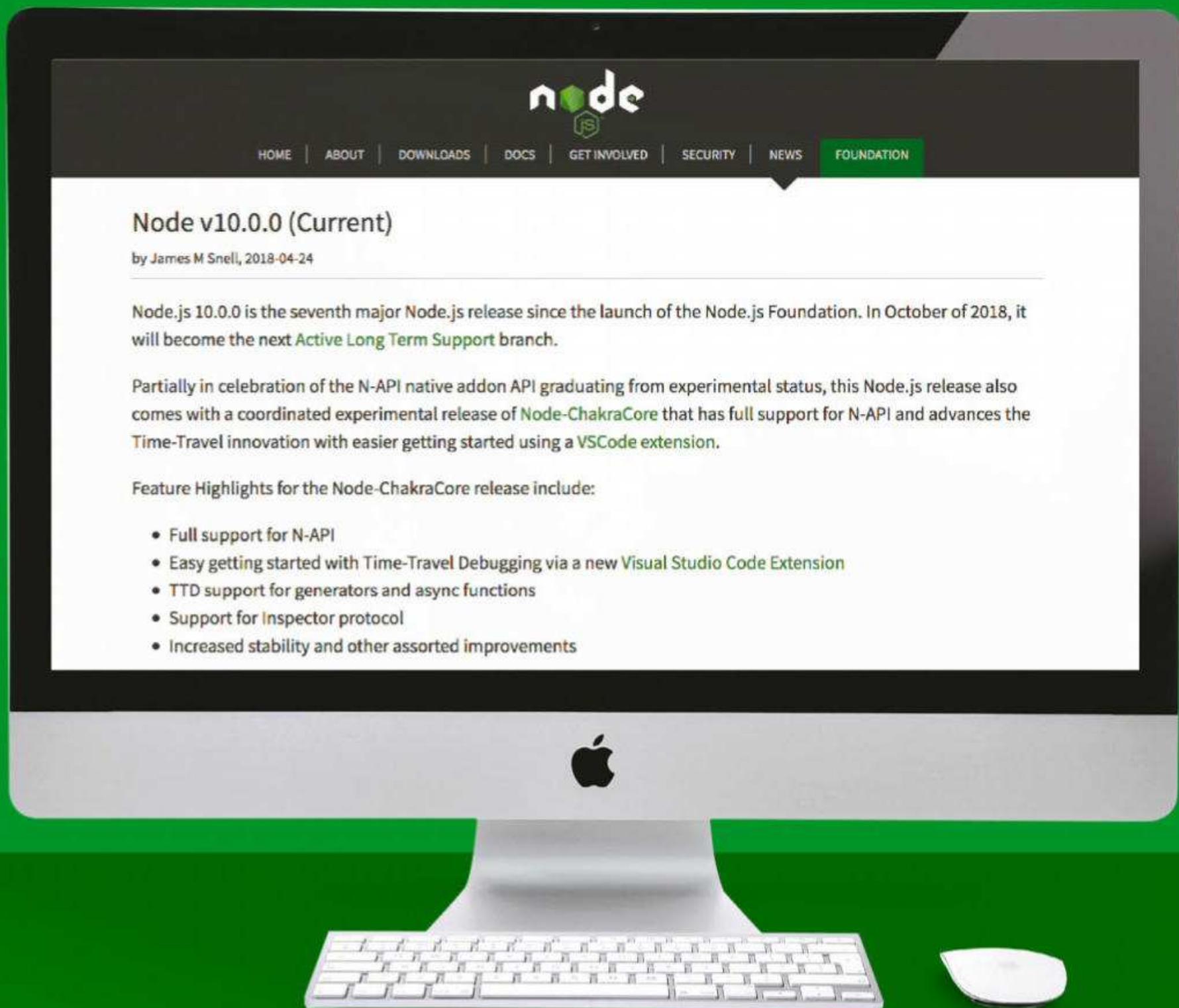
facebook.com/PhotoshopCreative



twitter.com/PshopCreative

Node.js 10: What's new?

Maintaining mature products is especially difficult. Take a look at tweaks and changes to the latest version



The image shows a laptop screen with a white background. On the screen, the Node.js website is displayed. The top navigation bar includes links for HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, NEWS, and FOUNDATION. The FOUNDATION link is highlighted with a green background. Below the navigation bar, the title "Node v10.0.0 (Current)" is shown, followed by the author "by James M Snell, 2018-04-24". A brief summary states: "Node.js 10.0.0 is the seventh major Node.js release since the launch of the Node.js Foundation. In October of 2018, it will become the next Active Long Term Support branch." It also mentions: "Partially in celebration of the N-API native addon API graduating from experimental status, this Node.js release also comes with a coordinated experimental release of Node-ChakraCore that has full support for N-API and advances the Time-Travel innovation with easier getting started using a VSCode extension." A section titled "Feature Highlights" lists several bullet points: "Full support for N-API", "Easy getting started with Time-Travel Debugging via a new Visual Studio Code Extension", "TTD support for generators and async functions", "Support for Inspector protocol", and "Increased stability and other assorted improvements".



ode.js bravely enabled JavaScript go to places its creators never intended the **language to go**. Many well-liked features seen in new ECMAScript revisions started out as features added to Node.js – sadly, the standardised APIs often were incompatible with the ones thought up by the Node.js developer team.

Thus, one of the recurring topics of the Node.js world revolves around reconciling differences between ‘web development’ and ‘node development’. For example, the module subsystem of Node.js does not work according to ECMAScript standards.

Node.js 10 updates the versions of various core components used in the product, thereby enabling developers to harness speed and security improvements. Furthermore, work on HTTP/2 speeds up delivery of content in bandwidth- and/or latency-constrained scenarios. A change in the native interface will simplify the life of Node.js plugin developers, leading to a more vibrant extension ecosystem.

The highly detailed technical nature of the changes makes providing individual examples difficult. Due to that, the following tutorial provides short tidbits of information.

1. Update your environment

Node.js can be compiled by hand. However, most Linux distros are able to run an automatic installer. The code here shows the installation process on Ubuntu 14.04 LTS. If your operating system shows up in Fig. 1, find a similar process at nodejs.org/en/download/package-manager/.

```
tamhan@TAMHAN14:~$ sudo apt-get remove nodejs
tamhan@TAMHAN14:~$ curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -
tamhan@TAMHAN14:~$ sudo apt-get install -y nodejs
```

2. Sniff out errors...

Developers have performed string comparisons to find errors ever since Node.js 1.0. Version 9.0 started to shift to easier-to-handle constants, however. The code accompanying this step is immune to spelling mistakes and is resistant to attempts ‘to improve usability by reformulating error messages’.

```
try {
  // ...
} catch (err) {
  if (err.code === 'ERR_ASSERTION') { ... }
  else { ... }
}
```

```
tamhan@TAMHAN14:~/Desktop/stuff/2018August/FutureNode10/workspace$ cat index.js
function /* demo */ sayHurz()
{
    console.log("Hurz!");
}

console.log(sayHurz.toString());
tamhan@TAMHAN14:~/Desktop/stuff/2018August/FutureNode10/workspace$ node index
function sayHurz()
{
    console.log("Hurz!");
}
tamhan@TAMHAN14:~/Desktop/stuff/2018August/FutureNode10/workspace$ node -v
v8.11.3
```

1

The screenshot shows the Node.js homepage with a navigation bar at the top. Below the navigation, there's a heading 'Installing Node.js via package manager'. A note below the heading states: 'Note: The packages on this page are maintained and supported by their respective packagers, not the Node.js core team. Please report any issues you encounter to the package maintainer. If it turns out your issue is a bug in Node.js itself, the maintainer will report the issue upstream.' To the right of the note is a list of supported platforms:

- Android
- Arch Linux
- Debian and Ubuntu based Linux distributions
- Enterprise Linux and Fedora
- FreeBSD
- Gentoo
- NetBSD
- nvm
- OpenBSD
- openSUSE and SLE
- macOS
- SmartOS and illumos
- Void Linux
- Solus
- Windows

3. ...understand them...

Process computer programmers balk at the idea of using strings to transmit error states – Node.js 10 does not break this tradition. Instead, the framework defines a few dozen strings, which will serve as a ‘magic constant’ and can be returned in the code field of the ‘Error’ object.

4. ...and describe them properly

The introduction of error codes does not deprecate traditional error messages. Code based on the presence of the ‘message’ field continues to perform just as it did – should you need to present a message end users can understand, you can trust the field just as you did before.

```
const err = new Error('The message');
console.error(err.message);
```

5. Speed comes with V8 updates

Node.js always used the V8 runtime for JavaScript execution. As with most project collaborations, Node.js usually lags behind V8’s progress. Node.js 10 is based on version 6.6 of V8, which brings significant improvements in caching and array-handling performance.

4

Node.js Error Codes
ERR_AMBIGUOUS_ARGUMENT
This is triggered by the assign() method in cases where the value [Err...Message] is used in a key until the message is the known error message. This is analogous because the message is not verifying the error message and will only be thrown in case no error is thrown.
ERR_ARG_NOT_ITERABLE
An iterable argument (i.e. a value that meets with the ... operator) was provided but not provided as a Node.js API.
ERR_ASSERTION
A special type of error that can be triggered whenever Node.js detects an exceptional logic violation that should never occur. This one is called typically by the assert module.
ERR_ASYNC_CALLBACK
An attempt was made to register something that is not a function as an asynchronous callback.

6. Headless try-catch!

‘Try-catch’ blocks are the bane of Java programmers: finding one marked with a comment instructing to shut the compiler up is common. V8 6.6 reduces the burden of unwanted ‘try-catches’ by letting developers omit the ‘exception’ object from the ‘catch’ block.

```
try {
  doSomethingThatMightThrow();
} catch {
  handleException(); }
```

7. Improved detail in function reflection

One little-known detail of JavaScript concerns functions: they can be considered objects, too. So far, the ‘toString()’ method just returned the method name. In Node.js 10,

Avoid root non-root mismatches

When installing Node.js by hand, make sure that the root and normal user account point to the same version of the framework! Parity can be checked by analysing the version numbers.

Tutorials

Node.js 10: What's new?

```
tamhan@TAMHAN14:~/Desktop/stuff/2018August/FutureNode10/workspace$ node index
function /* demo */ sayHurz()
{
    console.log("Hurz!");
}
tamhan@TAMHAN14:~/Desktop/stuff/2018August/FutureNode10/workspace$ node -v
v10.1.0
```

Node.js v10.8.0 Documentation

[Index](#) | [View on single page](#) | [View as JSON](#) | [View another version](#) | [Edit on GitHub](#)

Table of Contents

- N-API
 - Usage
 - N-API Version Matrix
 - Basic N-API Data Types
 - napi_status
 - napi_extended_error_info
 - napi_env
 - napi_value
 - napi_threadsafe_function
 - napi_threadsafe_function_release_mode
 - napi_threadsafe_function_call_mode
 - N-API Memory Management types

2018-08-01, Version 10.8.0 (Current), @targos

Notable Changes

- deps:
 - Upgrade npm to 6.2.0. #21582
 - npm has moved. This release updates various URLs to point to the right places for bugs, security, and documentation.
 - Fix the regular expression matching in `xcode_error.cc` in `node-gyp` to also handle very multiple-digit major versions which would otherwise break under use of XCode 10.
 - The npm tree has been significantly flattened. Tarball size for the npm package has gone from 1.2GB to 1.1GB.

Commits

- [325579e4bb] · benchmark: remove arrays benchmark (Peter Marshall) #21831
- [4200f8e51c] · build: create V8 code cache after script is run (Joyee Cheung) #21567
- [5ec0ca2a81] · build: increase macOS minimum supported version (Michael Zasso) #21883
- [5e1c6aa5a1] · build: remove redundant Makefile target (Rich Trot) #21915
- [4f865e2a11] · build: add new benchmark targets (Kenny Yuan) #20905
- [4d5fc5c0e2] · build: move to `nm -c`, where possible (Rich Trot) #21802
- [e97a5e3a] · build,win,v8: allow precompiling objects init.h (João Press) #21772
- [d790e055] · (SEMVER-MINOR) deps: upgrade rpm to 6.2.0 (Kai Marchan) #21582
- [f0a825c5] · deps: cherry-pick 804a693 from upstream V8 (Matheus Marchini) #21855

the program returns the whole first line – as can be seen in the figure, the return value even extends to all kinds of comment.

```
function /* demo */ sayHurz()
{
    console.log("Hurz!");
}

console.log(sayHurz.toString());
```

8. Remedy failed updates

Should your workstation be unable to run the new version of Node.js, check the package manager configuration. Yours truly had to delete the two files mentioned in the following code before normal program operation could resume:

```
tamhan@TAMHAN14:/etc/apt/sources.list.d$ sudo
rm nodesource.list
tamhan@TAMHAN14:/etc/apt/sources.list.d$ sudo
rm nodesource.list.save
```

9. ECMAScript modules

While Node.js influenced the idea of JavaScript modules, the ECMAScript implementation is not compatible with Common.js. While the '`--experimental-modules`' flag has been around for quite some time, Node.js 10 brings along a few new features intended to let ECMAScript-based modules work in Node environments.

```
tamhan@TAMHAN14:~/workspace$ node
--experimental-modules index
```

10. Understand native module architecture...

While JavaScript can do a lot of things, some jobs require the use of C++. So far, developers seeking tighter integration have usually placed their trust in the V8 API – leading to plugin rewriting whenever the version of V8 used in Node.js was changed.

11. ...and see N-API improvements

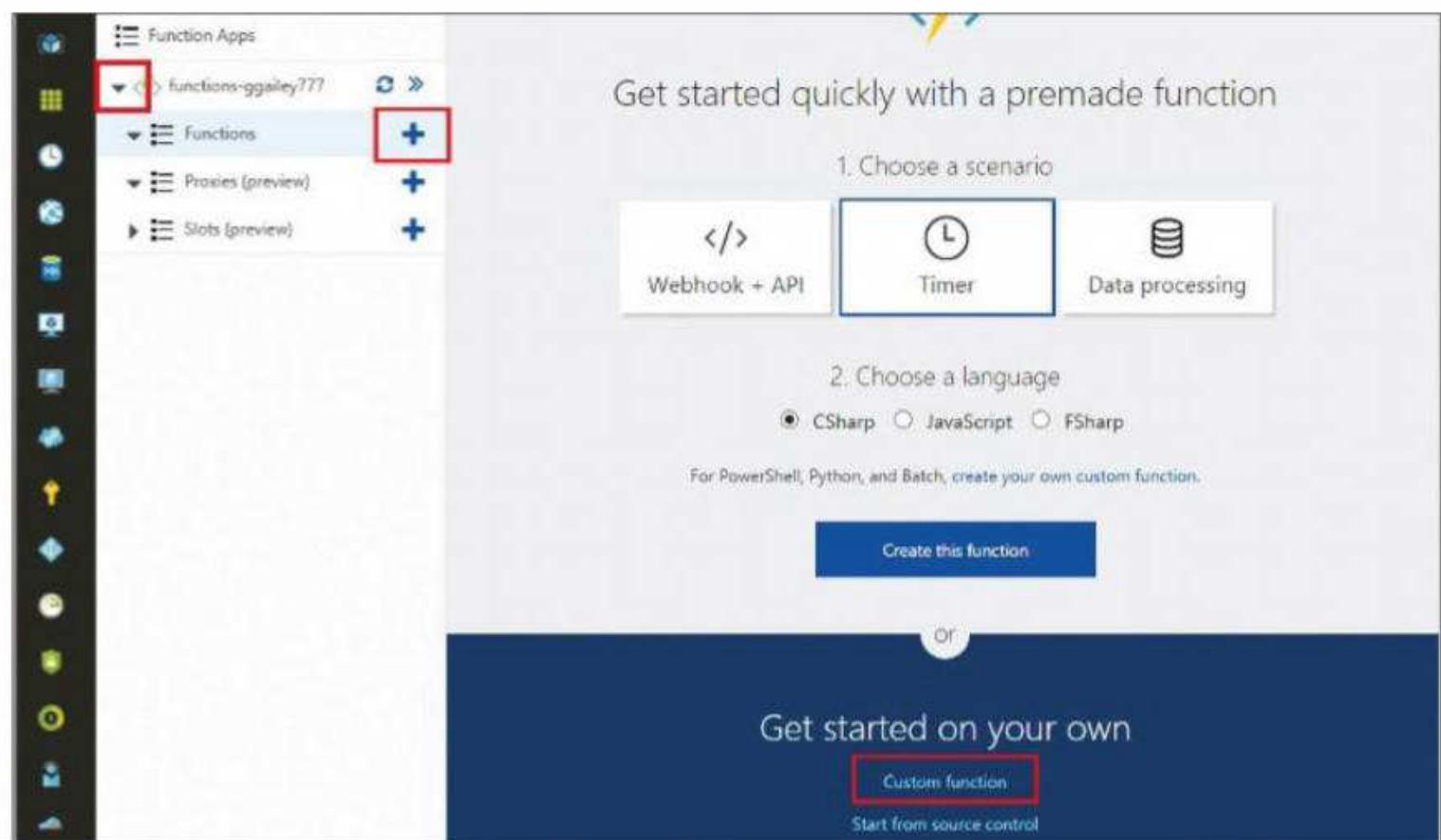
Thankfully, N-API solves this problem by providing a new abstraction layer. Any changes made to the underlying platform and tooling are no longer exposed to developers working on plugins, thereby lowering maintenance burdens and, hopefully, making Node.js upgrades smoother.

12. Learn about the API

As far as C APIs go, the N-API is clearly among the better-designed ones. As shown in excruciating detail at [nodejs.org/api/n-api.html](#), the API comes equipped with methods for handling object creation, callbacks and more. Furthermore, the header contains a 'lock' permitting you to disable features not found in older versions of the NAPI for compatibility reasons.

Move Node.js 10 to the cloud

Cloud service providers – be it Amazon with AWS, Microsoft with Azure or Google with its Web Services – tend to drag their feet when it comes to offering updates of Node.js. If past performance is an indicator, delays of up to one year are to be expected until the features discussed in this tutorial will become available. One way around the problem involves Babel ([babeljs.io](#)). It takes all kinds of JavaScript code, and outputs a version compatible with older versions. Some hackers – a hat tip to [hackernoon.com/so-i-still-can-t-run-my-node-js-10-code-on-aws-lambda-or-can-i-d2a9e8b1eeec](#) is in order – even go as far as to package Node.js via Babel, leading to a 'portable' version of the entire interpreter.



HTTP/2

This is the home page for HTTP/2, a major revision of the Web's protocol. It is maintained by the IETF HTTP Working Group.

Tweets by @HTTP_2

HTTP/2 @HTTP_2 Amazon CloudFront now supports HTTP/2! aws.amazon.com/about-aws/wha... Sep 8, 2016

What is HTTP/2?

HTTP/2 is a replacement for how HTTP is expressed "on the wire." It is not a ground-up rewrite of the protocol; HTTP methods, status codes and semantics are the same, and it should be possible to use the same APIs as HTTP/1.x (possibly with some small additions) to represent the protocol. The focus of the protocol is on performance; specifically, end-user perceived latency, network and server resource usage. One major goal is to allow the use of a single connection from browsers to a Web site. The basis of the work was SPDY, but HTTP/2 has evolved to take the community's input into account, incorporating several improvements in the process. See our charter for more details of the scope of the work, as well as our Frequently Asked Questions. See also [HTTP/2 JP](#), maintained by the Japanese HTTP/2 community.

Make the most of HTTP/2

When HTTP was specified, websites were, by and large, simple affairs. Re-establishing a TCP connection for each resource was a rather minimal overhead when taken in context. As page complexity increased, steps were made to eliminate protocol inefficiencies. This resulted in an upgraded HTTP protocol called HTTP/2, which lets the browser and the server recycle an already-established TCP connection in an attempt to reduce latency and overhead. Further information on HTTP/2's technical benefits can be found at [http2.github.io](#) – while GitHub hosts the page, its contents are maintained by the IETF and contains all kinds of valuable information.

```
#define NAPI_VERSION 3
#include <node_api.h>
```

13. Use ChaCha20 and Poly1305

Cryptography always was of significant importance on the web. Node.js 10 updates the underlying version of OpenSSL to 1.1.0, thereby enabling developers to use new encryption algorithms, which are expected to be more robust. This is especially helpful when seeking to implement AEAD-type encryption schemes.

14. Targeted tracing

Node.js 10 emits a set of additional tracing parameters, thereby letting developers zero in on performance problems. Trace objects can furthermore be enabled and disabled at will via the functions shown in the snippet accompanying this step – pinpoint accuracy is now at every developer's fingertips.

```
const trace_events = require('trace_events');
const t1 = trace_events.createTracing({
  categories: ['node', 'v8']
});

const t2 = trace_events.createTracing({
  categories: ['node.perf', 'node']
});

t1.enable();
t2.enable();
t2.disable();
```

15. Find out more about tracing

As of this writing, Node.js supports a total of eight trace types listed at nodejs.org/api/tracing.html#tracing_

trace_events. In addition to V8 events, developers can also register themselves for bootstrap, 'async_hook' and file system sync information at the command line or via the trace object.

```
const trace_events = require('trace_events');
const tracing = trace_events.createTracing({
  categories: ['node.perf'] });
tracing.enable();
```

16. HTTP/2 is now official...

While the API stability indicator in the Node.js

documentation still shows HTTP/2 support as being experimental, the API can be considered stable.

```
const http2 = require('http2');
const fs = require('fs');
const server = http2.createSecureServer({
  key: fs.readFileSync('localhost-privkey.pem'),
  cert: fs.readFileSync('localhost-cert.pem')
});
server.on('error', (err) => console.error(err));
```

Trace Events

Stability: 1 - Experimental

Trace Event provides a mechanism to centralize tracing information generated by V8, Node.js core, and userspace code. Tracing can be enabled with the `--trace-event-categories` command-line flag or by using the `trace_events` module. The `--trace-event-categories` flag accepts a list of comma-separated category names.

The available categories are:

- node - An empty placeholder.
- node.async_hooks - Enables capture of detailed `async_hooks` trace data. The `async_hooks` events have a unique `asyncId` and a special `triggerId` `triggerAsyncId` property.
- node.bootstrap - Enables capture of Node.js bootstrap milestones.
- node.fs.sync - Enables capture of trace data for file system sync methods.
- node.perf - Enables capture of Performance API measurements.
 - node.perf.userTiming - Enables capture of only Performance API User Timing measures and marks.
 - node.perf.timerify - Enables capture of only Performance API timerify measurements.
- node.promises.rejections - Enables capture of trace data tracking the number of unhandled Promise rejections and handled-after-rejections.
- node.v8.script - Enables capture of trace data for the `v8` module's `runInNewContext()`, `runInContext()`, and `runInThisContext()` methods.
- v8 - The `v8` events are GC, compiling, and execution related.

By default the `node`, `node.async_hooks`, and `v8` categories are enabled.

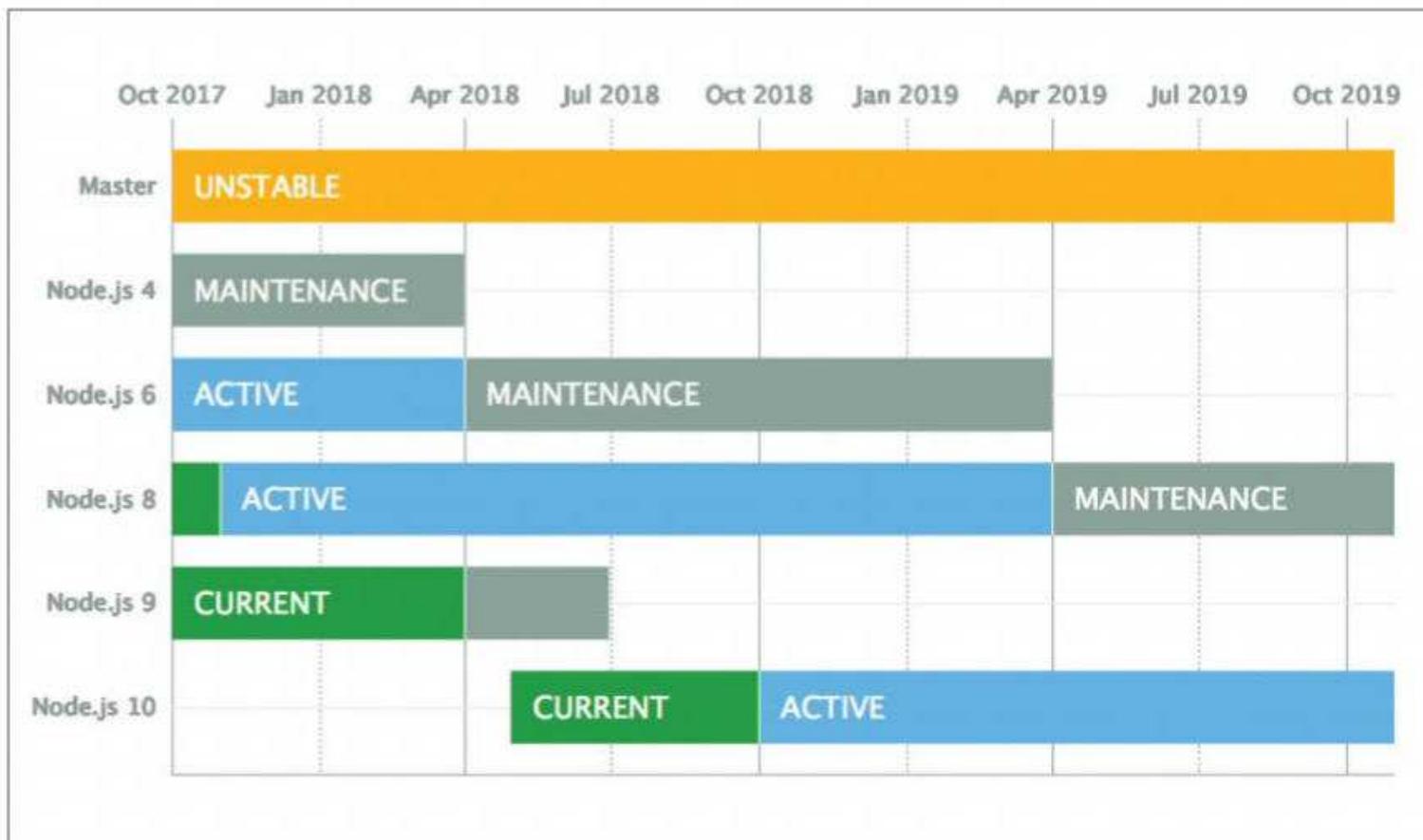
```
node --trace-event-categories v8,node,node.async_hooks:server.js
```

Prior versions of Node.js required the use of the `--trace-events-enabled` flag to enable trace events. This requirement has been removed. However, the `--trace-events-enabled` flag may still be used and will enable the `node`, `node.async_hooks`, and `v8` trace event categories by default.

```
node --trace-events-enabled
```

Tutorials

Node.js 10: What's new?



Understand the Node.js release schedule

Administrators loathe Node.js for the often frantic pace of development – hardly a month goes by without a new version of Node. A two-tier release schedule mitigates the problem: from time to time, a current version receives the LTS designation. From that moment onward, no new features will be added during an 18-month time frame. Maintenance, instead, limits itself to performing non-breaking changes. During the initial announcement of Node.js LTS releases, only “bug fixes for stability, security updates, possible npm updates, documentation updates and certain performance improvements that can be demonstrated to not break existing applications” were permitted – so far, that rule has stood the test of time.

17. ...but do mind the encryption

When implementing a Node.js server which uses HTTP/2, do bear in mind that the unencrypted version of the protocol is not supported by any web browser as the time of writing. Because of this, you should be prepared to deal with significant processor overhead when transmitting and receiving data in proxyless configurations.

```
server.on('stream', (stream, headers) =>
{
  // stream is a Duplex
  stream.respond({
    'content-type': 'text/html',
    ':status': 200
  });
  stream.end('<h1>Hello World</h1>');
});
server.listen(8443);
```

18. File system API with promises

Promises are all the rage in JavaScript. So far, developers have usually had to make do without promise-based file system APIs – a problem that's been remedied in Node.js 10. While the module ‘fs/promises’ is still experimental, it is already providing a nice alternative to ‘traditional’ file system accesses.

```
require('fs/promises')
```

19. More work has been done on async hooks

Node.js is single-threaded by default. Async hooks enable code to keep an eye on asynchronously-executed resources in a more comfortable fashion – while the API is far from stable, Node.js 10 provides significant improvements over the barely-working implementations found in previous versions.

```
const async_hooks = require('async_hooks');
const eid = async_hooks.executionAsyncId();
const tid = async_hooks.triggerAsyncId();
```

```
const asyncHook = async_hooks.createHook
({  
  init, before, after, destroy, promiseResolve  
});
```

20. Turbo-NPM

While NPM is usually seen as constituent part of the Node.js distribution, it is actually a standalone product. Current versions of Node.js 10 come with NPM 6 ([npmjs.com](https://www.npmjs.com/)), which offers significant speed increases when

installing and deploying packages. On this writer's machine, some processes saw ten-fold speed increases.

21. Find out more

Should you feel like always staying on top of the latest developments in Node.js, head to github.com/nodejs/node/blob/master/doc/changelogs/CHANGELOG_V10.md in a browser of your choice. This GitHub page provides an intimate level of detail on changes in the product.

Branch: master node / doc / changelogs / CHANGELOG_V10.md Find file Copy path
rvagg 2018-08-15, Version 10.9.0 (Current) 03b8258 14 days ago
9 contributors
2329 Lines (2252 sloc) 382 KB Raw Blame History

Node.js 10 ChangeLog

Current

10.9.0
10.8.0
10.7.0
10.6.0
10.5.0
10.4.1
10.4.0
10.3.0
10.2.1
10.2.0
10.1.0
10.0.0

• Other Versions
◦ 9.x

Special offer for readers in *North America*



6 issues FREE

When you subscribe*

FREE

resource
downloads
every issue



**"The only magazine
you need to design
and develop
stunning websites"**



Order hotline +44 (0) 344 848 2852
Online at myfavouritemagazines.co.uk/WEDPQ17

***Terms and conditions** This is a US subscription offer. 6 free issues refers to the USA newsstand price of \$14.49 for 13 issues at \$188.37, compared with \$103 for a subscription. You will receive 13 issues in a year. You can write to us or call us to cancel your subscription within 14 days of purchase. 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 upon request. UK calls will cost the same as other standard fixed line numbers (starting 01 or 02) are included as part of any inclusive or free minutes allowances (if offered by your phone tariff). For full terms and conditions please visit: bit.ly/magtandc Offer ends November 30 2018.

Expires
31 Nov
2018

Get creative with the Pts.js library

Introducing the fundamentals of working with the Pts.js JavaScript library and how you can use it to create a captivating landing page





First impressions matter a great deal, as it turns out. Within split seconds, people make snap judgments and form opinions on anything or anyone, and websites are no different. By spending a short time in your website, visitors are able to determine whether to continue browsing further or leave entirely. Consequently, enhancing the visual appeal of landing pages has received significant attention over the past few years. Companies acknowledge the importance of captivating visitors visually, and as such, seek to make their webpages unique and memorable. However, this puts pressure on front-end designers to keep learning new tricks to ensure their web projects not only deliver, but also offer a unique visual appeal. A modern JavaScript library that holds immense potential in meeting this need is Pts.js. The library, written in TypeScript and compiled to JavaScript ES6, is lightweight and modular and ships with many useful algorithms for data visualisation and creative coding. The library pushes the concepts of geometry to unimaginable heights and enables you to compose and express what you see in your mind's eyes – points as ideas, shapes, colours, motions, interactions and more. In this tutorial, the basics of the library are introduced and a practical application of the library demonstrated by creating a landing page.

1. Getting started

Begin by creating a folder, 'pts', on your desktop to store the tutorial files. Create two additional folders within it: 'css' to store the styling files, and 'js' to store JavaScript files. HTML files will be stored in the root folder ('pts'). Create a file 'styles.css' in the CSS folder and 'index.js' in the js folder. These will be used later in the tutorial.

2. Creating the landing page structure

Open your code editor and create an 'index.html' document to contain mark up for the main web page. Begin by creating the basic structure and give a suitable title to the page.

```
<html>
  <head>
    <title>Get Creative with PtsJs </title>
  </head>
  <body> </body>
</html>
```

3. Adding content to the landing page

The focus of the tutorial is to create a landing page and add captivating visual effects with the Pts.js library. As such, the landing page's content is added first. To do this, create a div in the body section to contain the header. Add the header title and its description within it.

```
<body>
  <div class="header">
    <h2>Creative Coding tutorial with Pts.js</h2>
    <p>Learn the basics and reach for the stars</p>
  </div>
```

Creative Coding tutorial with Pts.js

Learn the basics and reach for the stars

```
</div>
</body>
```

STYLING THE WEB PAGE

4. Styling the landing page

Open 'styles.css' we created earlier. Begin by setting the margin and padding to 0 for the body. This ensures the design remains consistent throughout the page. Next, add an overflow with the hidden parameter. This cuts off any content that breaks out of its bounds.

```
body {
  margin: 0;
  overflow: hidden;
  padding: 0;
}
```

Next, style the header and the descriptive paragraph. By using absolute positioning, centralise the header at the centre of the page.

```
.header{
  text-align: center;
  position: absolute;
  top: 40%;
  left: 20%;
  right: 20%;
  font-family: Trebuchet MS, Arial, Helvetica, sans-serif;
  font-size: 25px;
  color: #0bf5f5;
}

.header p {
  font-size: 20px;
  color: #c67508;
}
```

5. Linking the CSS file

Next, link the CSS stylesheet to the HTML document. The styles centralise the heading and its descriptions at the centre of the webpage. Create a link to this file in your

HTML document by adding the following code in the head section.

```
<link rel="stylesheet" href="css/styles.css" >
```

Currently, the landing page is very basic as it only contains coloured text on a white background.

ADDING JAVASCRIPT FUNCTIONALITY

6. Preparing to add JavaScript files

In 'index.html' create a new div where JavaScript will run below the header div in the body of the HTML document.

```
<div id="pt"></div>
```

Next, link the library to your project by either downloading the file from GitHub (<https://github.com/williamngan/pts>) or directly linking it from a CDN source such as 'unpkg' or 'jsdelivr'. In the former's case, save the file in the 'js' file created in step 1 and link the file as follows:

```
<script type="text/javascript" src="path/to/my_script.js"></script>
```

In the latter's case, simply add the link code below in the head section.

```
<script src="https://unpkg.com/pts/dist/pts.min.js"></script>
```

Finally, link the 'index.js' file created earlier, by adding the code below in the body section.

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

Exploring Pts.js example projects

The resource <https://ptsjs.org/demo> contains diverse examples of projects created using the library. The page is structured in a way that allows viewers to access the source code used in developing the different projects. Head onto the page to interact with the various projects to learn different uses the library can be put into.

Tutorials

Get creative with Pts.js library

7

Creative Coding tutorial with Pts.js

Learn the basics and reach for the stars

7. Adding Pts to the global scope

Open 'index.js' and begin by declaring the Pts library in the global scope by using the 'Pts.quickStart ()' function. By using this function, the need to call 'Pts.namespace' again is eliminated. Copy the code below.

```
Pts.quickStart("#pt", "#1e2f40");
```

Note that the div created in step 6 and the background colour are passed as parameters in order to target it directly and apply the given colour. The page should render as follows.

8. Declare the animation variable

The animation effect being created using the Pts library in this tutorial involves the collision of particles with one another in space. A viewer is able to move their cursor near the particles and hit them like billiard balls. As such, to control this behaviour, a variable is created to allow such manipulation. Declare this variable within the following function.

```
var world;
```

No observable changes are noted in the rendered webpage as functionality is yet to be added.

9. Understanding how the animation works

At this point it's important to understand the analogy Pts is built upon. Space provides the paper, Form provides the pencil and Point provides the idea. Currently, space has been created and its background properties

Understanding the reasoning behind Pts

The main idea behind the Pts JavaScript library is founded on three abstract concepts: Space, Form and Point. The resource <https://medium.com/@williamngan/pt-93382bf5943e> expounds further on the analogy behind its inception and introduces numerous possibilities in expressing ideas.

specified. The add function, with four callback functions: 'start ()', 'animate ()', 'action ()' and 'resize ()' is used to add players to the space, thereby giving it form and allowing the creation of points. These take the form below:

```
space.add ({  
    start: (bound, space) => {  
        // code for init  
    },  
    animate: (time, ftime, space) => {  
        // code for animation  
    },  
    action: (type, x, y, event) => {  
        // code for interaction  
    },  
    resize: (size, event) => {  
        // code for resize  
    }  
});
```

10. Setting up 'start ()' function

In initialising the animation, the 'start ()' function is declared. It takes two parameters: bound, which returns the bounding box, and space which returns its space. Carefully copy the code below to create the scene with 30 particles.

```
space.add ({  
    start: (bound, space) => {  
        // Create the scene and 30 random  
        points  
        scene = new Scene (space.innerBound,  
        1, 0);  
        let pts = Create.  
        distributeRandom(space.innerBound, 30);  
    }  
});
```

The first line of code utilises the method 'space.innerBound' to specify that the scene is being created in the inner bounding box of the space. 'distributeRandom' creates a set of 30 random points inside the boundary. Do not render the page yet.

11. Animating the random particles

Next, animate the particles in a random impulse movement in order to hit them. The 'for' and 'Math.random' functions are utilised in randomising the movement of the particles.

```
// Create particles and hit them with a random  
impulse  
for (let i=0, len=pts.length; i<len;  
i++) {  
    let p = new Particle (pts[i])  
.size( (i==0) ? 30: 3+Math.random() * space.  
size.x/50 );  
    p.hit (Num.  
randomRange(-50,50), Num.randomRange(-25,  
25));  
    scene.add (p);  
}
```

Finally, the created particles are locked so that they can be moved at a later point by using the cursor movement.

```
world.particle( 0 ).lock = true;
```

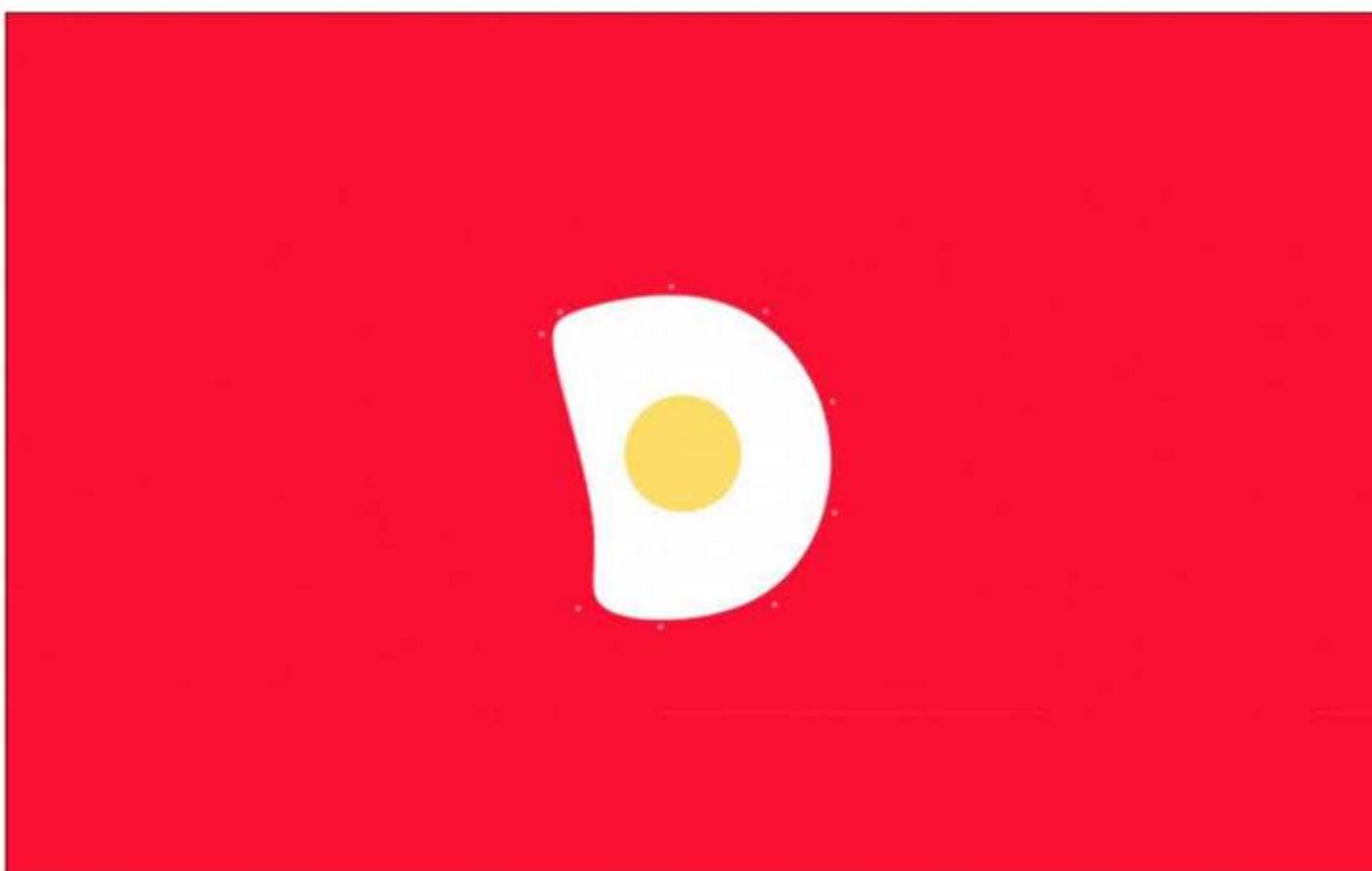
Do not render the page just yet as the function is currently incomplete.

12. Specify the animation timeline

Having already created the random particles and specified their random movements, call the 'animate' function in order to specify the timeline behaviour of the animation. Copy the code below.

```
animate: (time, ftime) => {  
    world.drawParticles( (p, i) => {  
        let color = (i==0) ?  
        "#1e2f40" : ["#ff2d5d", "#42dc8e", "#2e43eb",  
        "#ffe359"][i%4];  
        form.fillOnly( color  
).point( p, p.radius, "circle" )  
    });  
    world.update ( ftime );  
},
```

The code has two parameters: time, which gives the



Basic overview of creating with Pts library

Interaction with tons of example projects shared on the demo page <https://ptsjs.org/demo> reveals a similarity in the flow of logic in their development. As such, it is important to understand this logical flow before creating any project. In essence, only three key steps are involved. First, begin by declaring the Pts library in the global scope by using the 'Pts.quickStart ()' function. Pass the background colour and the targeted HTML element where the functionality is anticipated to be developed. This provides flexibility in creating with the library. Second, declare animation variables and add players to the space. This is the aspect that differentiates the different projects through the application of different maths logic. Finally, play the animation and decide whether it should respond to user activity.



FINALISING THE ANIMATION

14. Play the animation

Currently, the animation is complete. However, no activity can be observed because the functions to play the animation have not been called yet. Call the 'play ()' function as below.

```
space.play();
```

Refresh the page to observe the animation in play. Particles are now moving in random motion.

15. Making the animation sensitive to user activity

Despite the animation being complete, it is however not sensitive to user actions. Try moving the cursor on one of the particles. Note that no response activity takes place. To add this functionality, call the 'bindMouse' and 'bindTouch' functions in order to bind all touch or mouse events in the canvas element.

current running time, and 'ftime', which gives the time taken to draw the previous frame. The specific colours of the particles are also randomised and the shape specified to be circular.

13. Respond to user activity

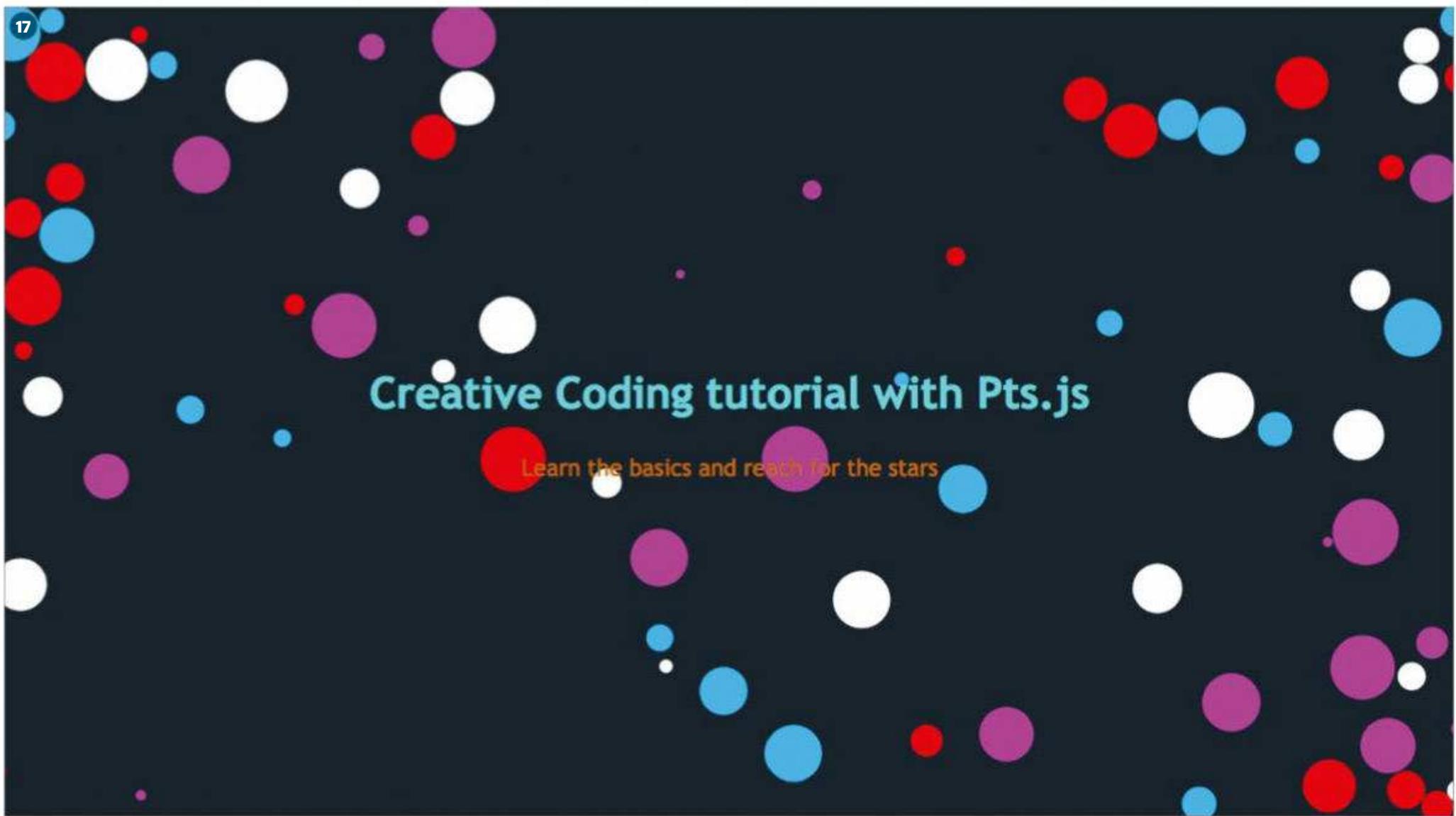
With the colour properties of the animation specified, call the action function to detect user activity. It includes four parameters. Type is a string that returns the action's name ('up', 'down', 'move', 'drag', 'drop', 'over' and 'out'). x and y returns the position at which the action happened, and event returns the actual event object. However, the event parameter is not declared in this tutorial.

```
action:( type, px, py) => {
  if (type == "move") {
    world.particle ( 0 ).position = new
Pt(px, py);
  }
}
```



Tutorials

Get creative with Pts.js library



`space.bindMouse().bindTouch();`

Refresh the page to observe the added interaction.
Particles are moving in response to mouse actions and
the cursor can be used to hit some balls against others.

TWEAKING THE ANIMATION

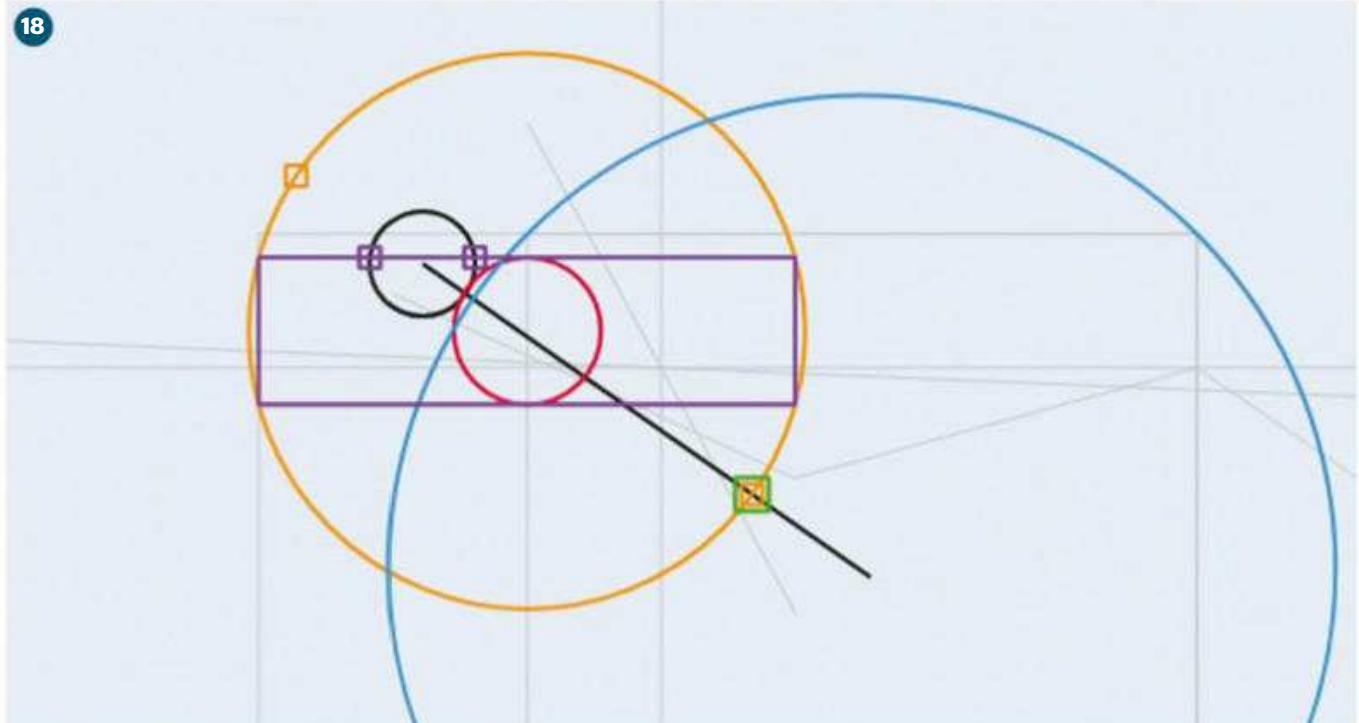
16. Tweaking particle number

Now that the entire animation is complete, tweak some features such as the number and colour of particles. To do this, edit the 'animate ()' function specified in step 12. Set the particle number to 80. Render should appear as shown on page 89.

17. Tweaking particle colour

Next, tweak the particle colours. Edit the specified colours in the 'animate ()' function specified in step 12. If these colours are specified in the parentheses: '#fff', '#e911f2', '#foo', '#15b6fd', the page should render as shown.

Note the key differences between the two.



differentiating the two with the
'Circle.Intersect2D' functionality.

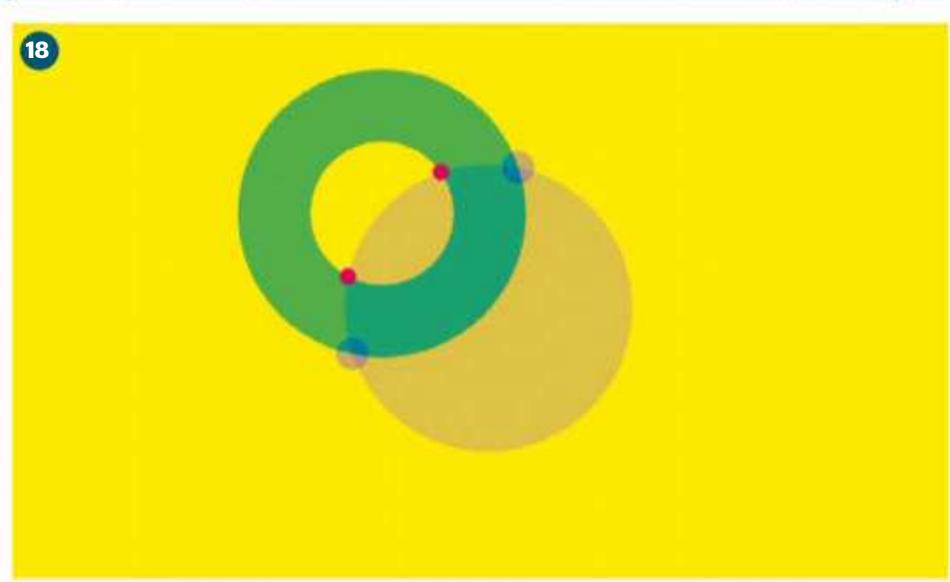
EXPLORING THE PTS LIBRARY IN DETAIL

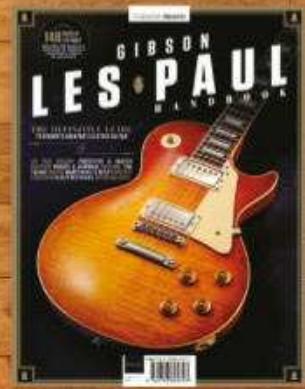
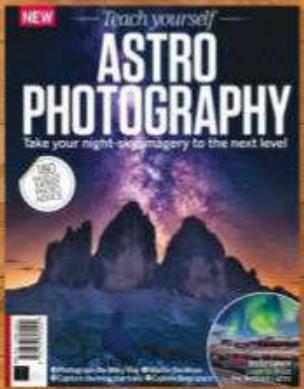
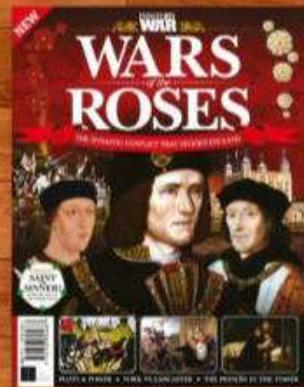
18. Interacting with studies

The demo page <https://ptsjs.org/demo> and the studies page <https://ptsjs.org/study> essentially describe similar projects. However, with the former, the projects are more 'polished', whereby the underlying functionality is abstracted. With the latter, the abstraction is removed and interaction between the different points is shown. Explore the different studies if interested in exploring the limits of the library. Here is a simple example

19. Interacting with docs

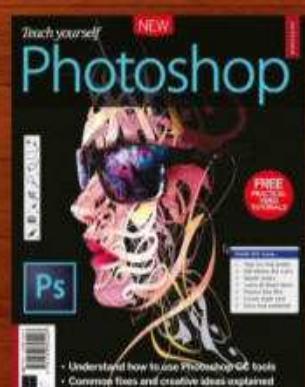
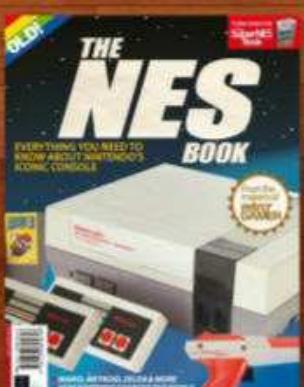
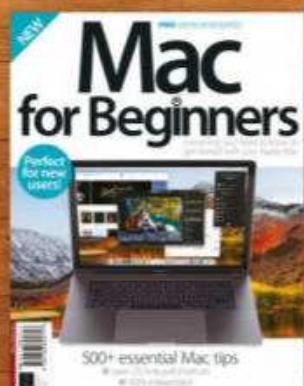
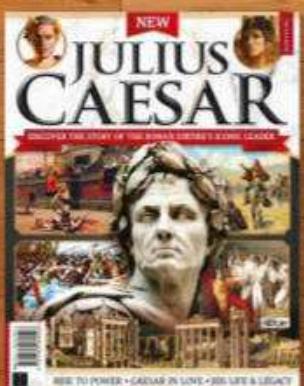
The documentation page <https://ptsjs.org/docs> provides detailed TypeScript/JavaScript API documentation of Pts.js. Though it is intended for advanced users, browse the resource to make any necessary references to functions utilised in different projects.





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.



Get your listing in our directory

To advertise here contact *Chris*

chris.mitchell@futurenet.com

+44 (0)1225 687832

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.

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.

3. 24/7 technical support

Netcetera has a committed team of

knowledgeable staff available 24/7 to provide you with assistance when you need it most. Our people make sure you are happy and your problems are resolved as quickly as possible.

4. Value for money

We do not claim to be the cheapest service available, but we do claim to offer excellent value for money. We also provide a price match on a like-for-like basis, as well as a price guarantee for your length of service.

5. 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."

Suzy 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
08455279345

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 one million domains running smoothly and safely each day. Services can be self-managed through the Fasthosts Control Panel.

- Dedicated servers
- Cloud servers
- Hosted email



Budget hosting



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
03333365509

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
08432892681

Since 2001, Bargain Host have campaigned to offer the lowest possible priced hosting in the UK. They have achieved this goal successfully and built up a large client database, which includes many repeat customers. They have also won several awards for providing an outstanding hosting service.

- Shared hosting
- Cloud servers
- Domain names

Value Linux hosting



patchman-hosting.co.uk
01642424237

Linux hosting is a great solution for home users, business users and web designers looking for cost-effective and powerful hosting. Whether you are building a single-page portfolio, or you are running a database-driven eCommerce website, there is a Linux hosting solution for you.

- Student hosting deals
- Site designer
- Domain names

Flexible cloud servers



elastichosts.co.uk
02071838250

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 contact *Chris*

chris.mitchell@futurenet.com

+44 (0)1225 687832

COURSE LISTINGS



Featured: **Northcoders**

northcoders.com
Twitter: @northcoders
Facebook: Northcoders

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

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

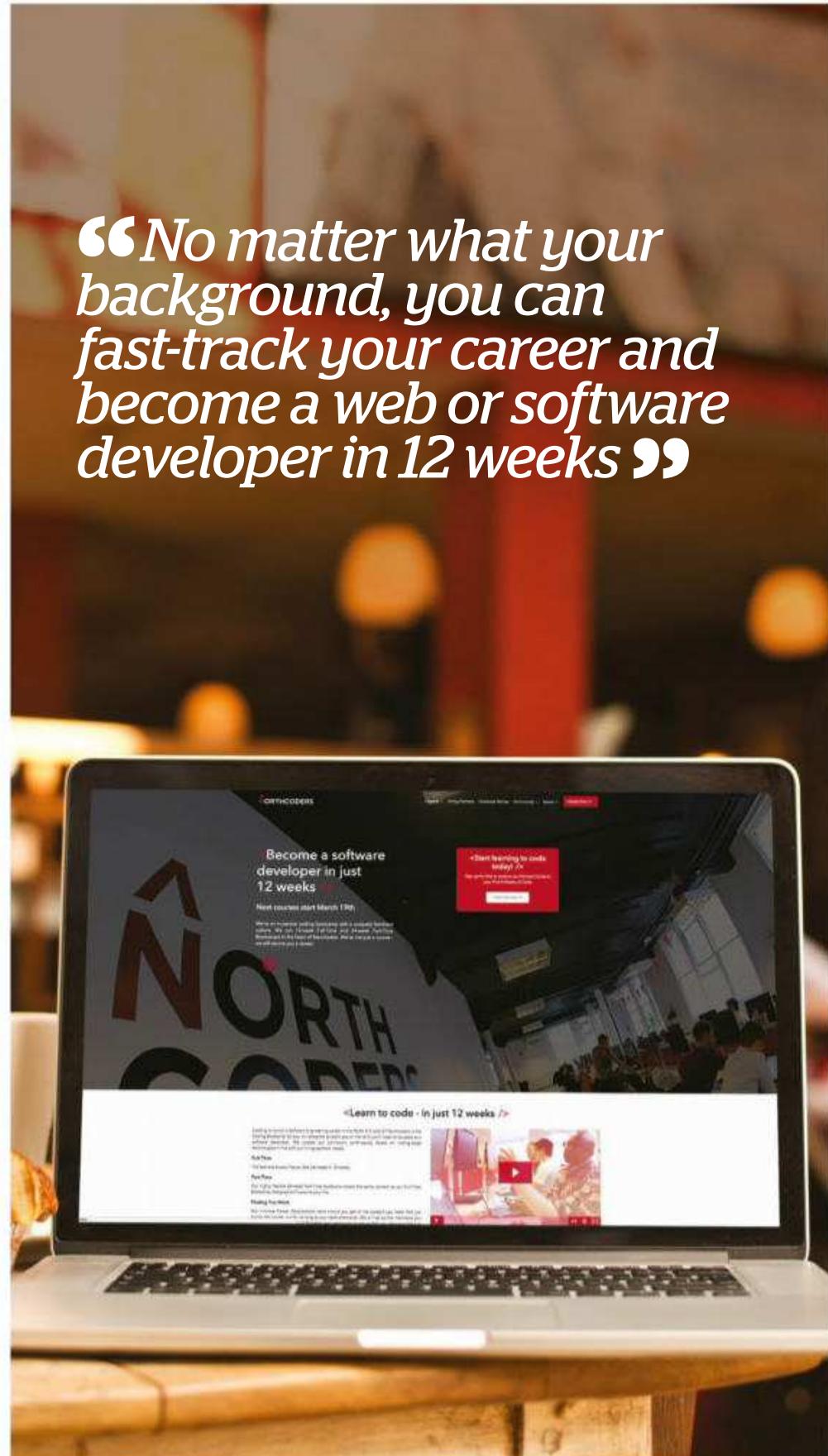
Once you've decided it's right



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



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



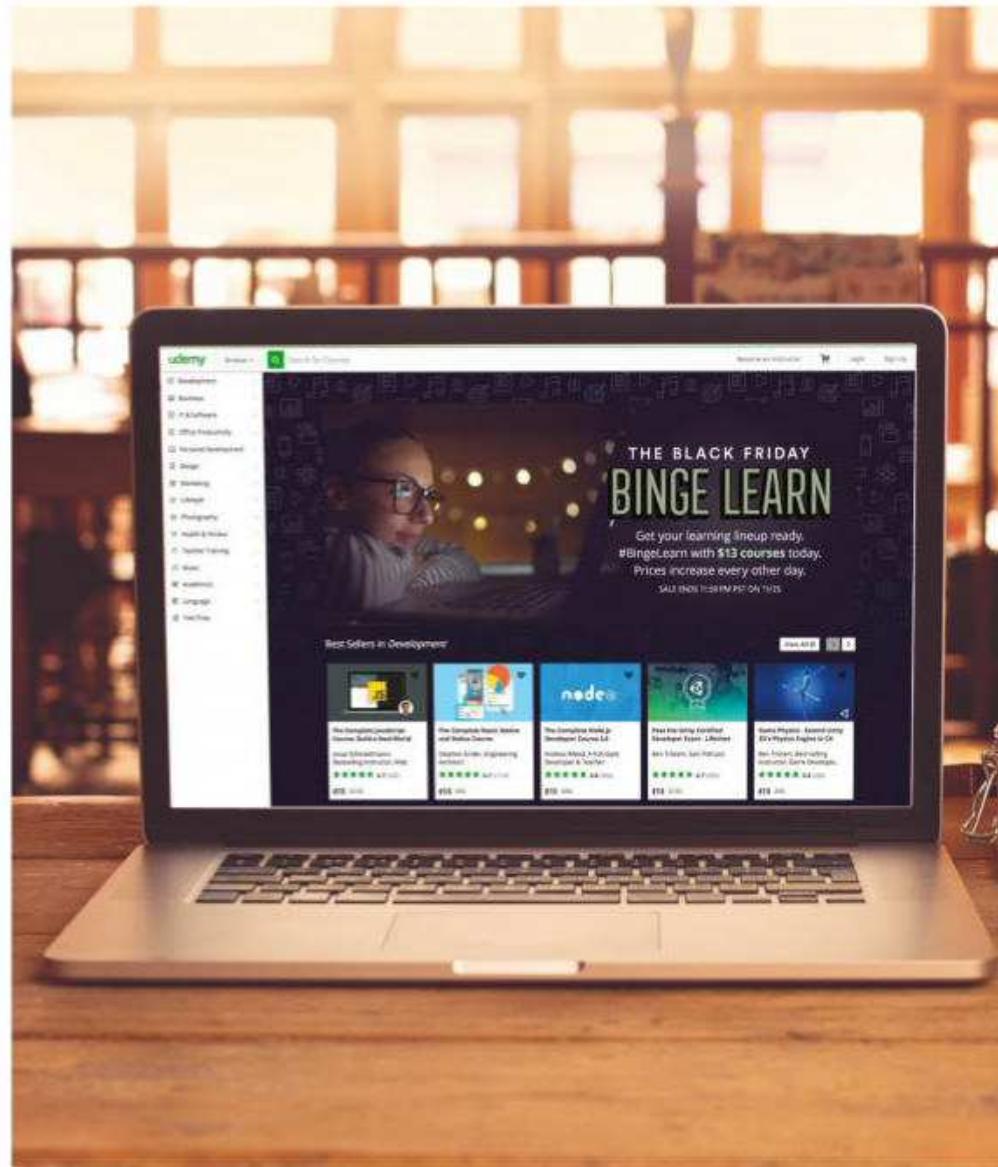
UDEMY

udemy.com

Twitter: [@udemy](https://twitter.com/udemy)
Facebook: [udemy](https://facebook.com/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.



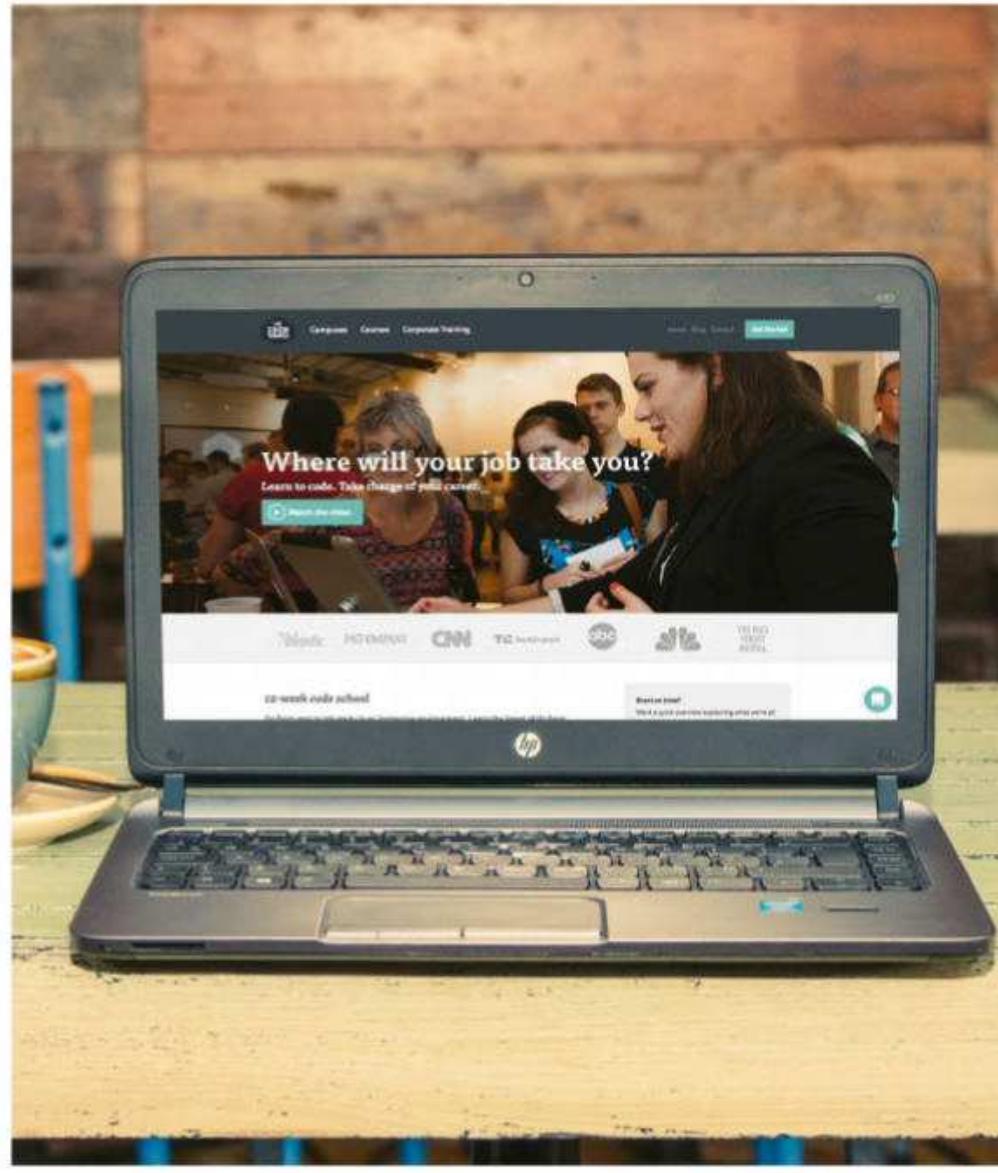
THE IRON YARD

theironyard.com

Twitter: [@TheIronYard](https://twitter.com/TheIronYard)
Facebook: [TheIronYard](https://facebook.com/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 backend engineering, frontend 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



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
- 60 minutes of video from the Complete Freelancer Course by Killersites (shop.killervideostore.com)
- 40 Photoshop travel and portrait filters and 17 Animated banner ads 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 60 issues

Access our entire library of resources with a money-saving subscription to the magazine – that's more than 900 free resources

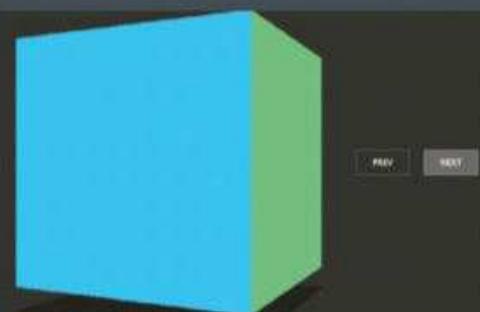
Over 60 hours of video guides

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 32 to subscribe now



Already a print subscriber?

[Here's how to unlock FileSilo today...](#)

Unlock the entire Web Designer FileSilo library with your unique Web ID – the 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

HTML, CSS & JAVASCRIPT

We reveal 20 APIs that every designer and developer needs to be using to build today's web experiences

A DESIGNER'S GUIDE TO SVG

Discover the benefits, how to create them and how to use them in your projects

GET STARTED WITH THREE.JS – PT5

In the final part of the series, learn how to load complex 3D models

WHAT IS THE WEB THING API?

Find out what it is and how to it can build physical devices that connect to the web

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 280 on sale
Tuesday 16th October 2018

SUBSCRIBE TODAY Go to page 32 to learn more

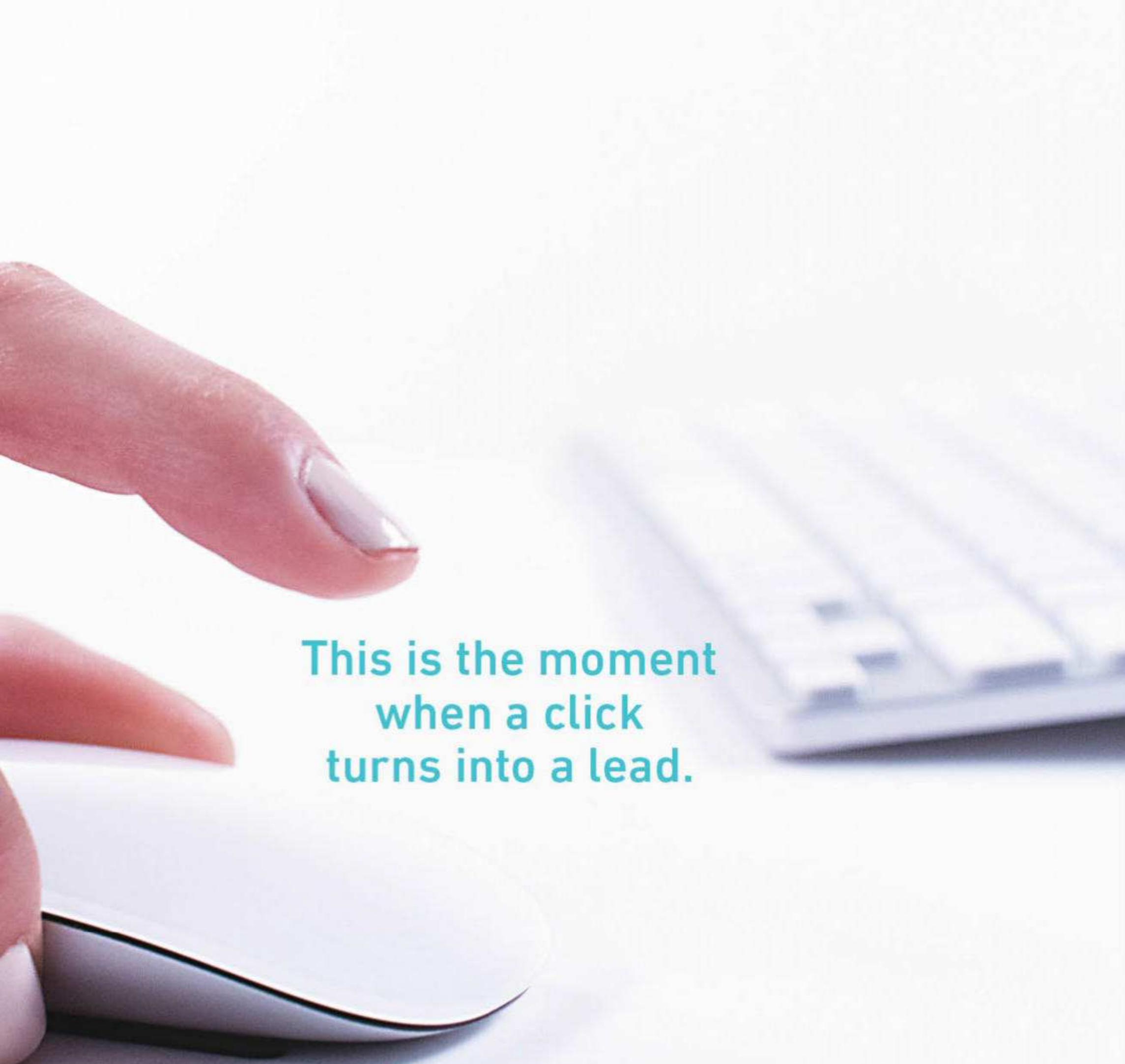
PUT A PAUSE IN YOUR DAY

With so many demands from work, home and family, there never seem to be enough hours in the day for you. Why not press pause once in a while, curl up with your favourite magazine and put a little oasis of 'you' in your day.



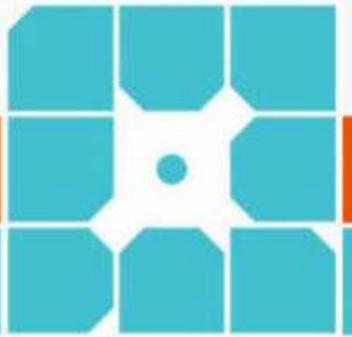
PRESS PAUSE
ENJOY A MAGAZINE MOMENT

To find out more about Press Pause, visit:
pauseyourday.co.uk



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®