

FREE 57 MINUTES OF JAVASCRIPT VIDEOS

designer

TM

Expert tutorials, techniques and inspiration



GET STARTED WITH **THREE.JS**

Add textures for realistic looks and visual effects

CSS GRID

How to power up your web layouts

SAY HELLO TO FLUTTER

Google's SDK to build beautiful native apps

JAVASCRIPT WHAT'S NEW?

REVEALED: THE LATEST UPDATES TO ES2018, REACT, ANGULAR, VUE & NODE

+

WORK WITH WEBRTC

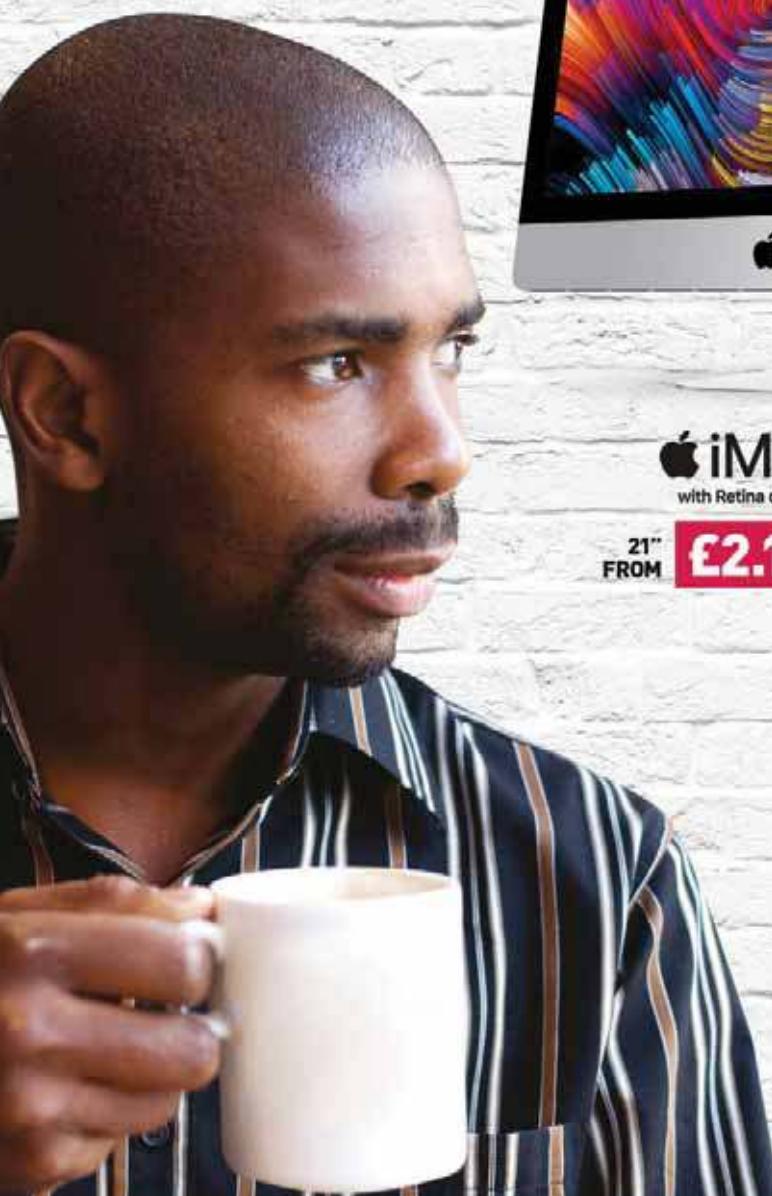
CODE AN INTERACTIVE HEADER

CREATE PATTERNS WITH CSS

Lease an Apple Mac!

Cheaper than your daily coffee

- Including accidental damage cover
- Upgrade when you want
- Own at the end
- 3 years service



iMac
with Retina display

21" FROM **£2.15** +VAT
PER DAY



MacBook Pro

13" FROM **£2.20** +VAT
PER DAY

Discover the benefits of leasing computers for business at
www.hardsoft.co.uk Email info@hardsoft.co.uk

HARDSOFT
Established for over 25 years

 Authorised
Reseller

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



Subscribe today
and get a
FREE GIFT
page 32

Steven Jenkins
Editor

Get ahead of the game



JavaScript is everywhere today. But it wasn't until a new set of standards came along with ES2015 that it could meet the demands that developers had. A new release has happened every year since and in our latest lead feature (page 42) we look at what's new in ES2018.

Alongside JavaScript's modern-day emergence came a host of new libraries and frameworks that now hold a place in developers' and designers' arsenals. We take a look at the popular and most commonly used choices – Angular, React, Node and Vue – and take a closer look at the latest updates for each of them.

ES2018 is all about making complex tasks easier for everyone. This can only highlight JavaScript and the web as a serious platform for app development

Continuing along the "what's new" theme, we take a closer look at Google's SDK, Flutter (page 74). Find out how it is helping create beautiful cross-platform native apps and get a quick guide on how to build your first app.

New fonts are always making an appearance, which inevitably leads to new typography styles. We check out some of the best free and paid-for fonts currently available (page 66), and throw in some essential Photoshop tricks to make sure that your carefully selected fonts stand out in your designs.

We like to think that CSS Grid is still new, and it's about time everyone learned how to get the most from it. Work through our five-page guide (page 86) and discover how to create super-flexible, shape-shifting layouts and a fallback for unsupported browsers. Enjoy the issue.

Highlight

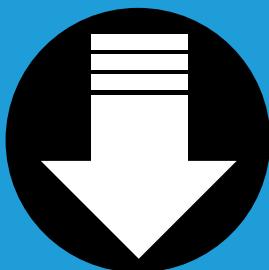


We love how HTML, CSS, and JS are integrating closer to mobile and desktop hardware

Find out how founder Matt Faulk went from BMX to BASIC and created the digital innovators of today. [Page 34](#)

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
422 Designer resources

Video Tuition - Part 5 of Beginner's JavaScript video guides from Killersites (shop.killervideostore.com)

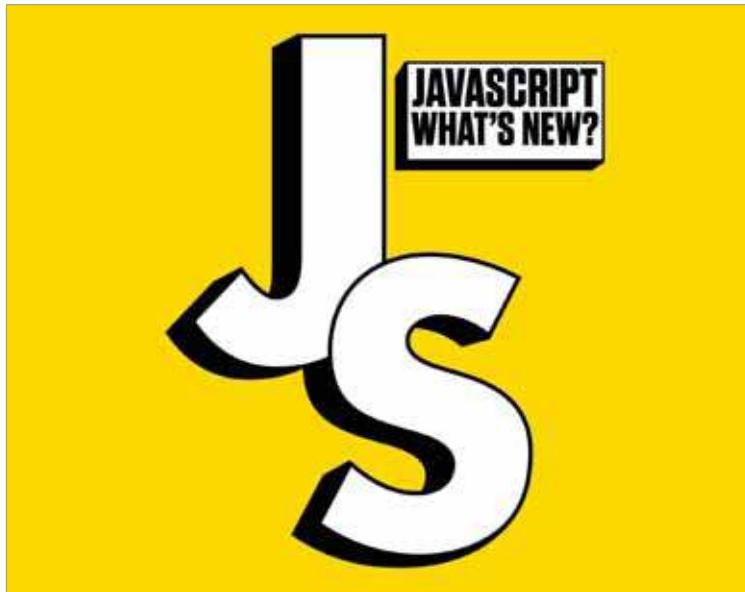
Assets - 21 Pixel art Photoshop actions and 400+ Isometric alphabet tiles from Sparklestock (sparklestock.com)
- Tutorial files and assets



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...



Matt Crouch

Matt is a frontend engineer at Vidsy in London. With the web changing rapidly, it's always important to keep up with the latest updates and try stay ahead of the curve. In this issue, he takes a look at the best new features in JavaScript (ES2018) and its popular frameworks React, Vue, Angular and Node.

Page 42

While the later rounds of standards updates have not been as feature-rich as ES2015, they still hold lots of useful changes

Simon Jones



Simon leads full-stack software engineering teams. Like many, he's found there to be major limitations to previous approaches to cross-platform mobile builds, and is hoping Flutter can set some of these right. **Page 74**

Mark Shufflebottom



Mark is a professor of Interaction Design at Sheridan College. In this issue Mark will be creating a stunning animated web site header and page design, which will be rendered through WebGL using the Three.js library. **Page 60**

Richard Mattka



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

Frank Kagumba



Frank is a frontend developer and technology writer based in Nairobi, Kenya. In this issue he will show you how to create some interesting website patterns using the CSS Doodle web component. **Page 56**

Mark White



Mark has used Photoshop for over a decade and writes tutorials for our sister mag Photoshop Creative. In his spare time he creates free use typefaces for download. This issue he selects some hot fonts and reveals his essential PS tricks. **Page 66**

Ahmed Abuelgasim



Ahmed is a frontend developer working at Potato in London. In this tutorial he will be showing us how to use the CSS Grid layout module to create dynamic website layouts with a fallback for unsupportive browsers. **Page 86**

Tam Hanna



Tam has always had a soft spot for image processing and convolution. Being able to access real-time communication and data streams from a browser is fascinating, and he shows you how this issue with the help of WebRTC. **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**

Future PLC Richm ond House, 33 Richm ond Hill,
Bournem outh, Dorset, BH2 6EZ

Editorial

Editor **Steven Jenkins**
steve.jenkins@futurenet.com
0 1202 586233

Designer **Harriet Knight**

Editor in Chief **Amy Hennessey**
Senior Art Editor **Will Shum**

Contributors

David Redfern, Mark Billen, Leon Brown, David Howell, Mark Shufflebottom, Matt Crouch, Richard Mattka, Frank Kagumba, Mark White, Simon Jones, Tam Hanna, Ahmed Abuelgasim, 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
0 1225 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 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.

Follow us!

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

Twitter: [https://twitter.com/
webdesignermag](https://twitter.com/webdesignermag)

F U T U R E

Connectors.
Creators.
Experience
Makers.

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)225 442 244

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

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 The state of eCommerce

How is the shift to purchasing via mobile changing the online buying landscape?

10 WebKit: The best must-try resources out there

Discover the libraries and frameworks you need

11 Freelancers and IR35

Dean Redfern, Director at DSR Tax Claims, reveals if this tax legislation applies to you

14 Lightbox

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

26 Dutch courage

Designer Mathis Biabiany reveals how he put together his particle-perfect portfolio site

34 Basic experiences

Find out how ex-pro BMX rider Matt Faulk built his very own digital agency, BASIC

42 JavaScript: what's new?

A closer look at the best new features in ES2018 and frameworks React, Vue, Angular and Node

66 Top type

Everyone loves a good font. Here we show top fonts, type and Photoshop techniques.

74 Say hello to Flutter

Discover how Google's new framework offers a new approach to cross-platform native apps.

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



66
FONTS AND PHOTOSHOP
Discover hot fonts and quick PS techniques



74
TIME TO TRY FLUTTER
Build beautiful apps with Google's SDK

FileSilo

96 Get the latest must-have resources and videos

- Part 5 of Beginner's JavaScript video guides from Killersites
- 21 Pixel Art Photoshop actions
- 400+ Isometric Alphabet Tiles



Subscribe today and save 20%

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



If you provide your services to a client through an intermediary, you come under the remit for IR35

COMMENT — David Redfern, p11



34

ProFile: BASIC
From BMX to digital innovation



14

Lightbox: KOOX
Getting fancy with food

Visit the **WEB DESIGNER** online shop at
myfavouritemagazines.co.uk
for the latest issue, back issues and specials

Tutorials

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



52 Get started with Three.js

The latest in the series demonstrates how to add textures for realistic looks and cool visual effects.

56 Create CSS web patterns

Learn how to create interesting and engaging patterns using the CSS Doodle web component.

60 Code a WebGL header

Make your site stand out by creating an interactive animated header using the Three.js library.

Web Workshop

50 Create an SVG text effect

sorrytobotheryou.movie

Learn how to introduce animated GIFs behind SVG text titles.

64 Code a notification toggle

portion.io

Add a simple but smart and smooth animated notification element.

Web Developer

74 Say hello to Flutter

Discover how Google's new mobile SDK helps develop native interfaces on iOS and Android.

80 Harness the power of WebRTC

Discover real-time communication through the browser with access to camera and microphone hardware.

86 Working with CSS Grid

Build shape-shifting layouts, as well as a fallback for unsupported browsers.

Header

The tools, trends and news to inspire your web projects

The state of eCommerce today

Who's buying what and why? Web Designer finds out more

Back at the end 2016 there was a shift in the worldwide desktop vs. mobile market share with the small screen finally overtaking the big screen. Ever since, mobile market share has been slowly moving away from desktop with mobile now standing at 52 percent and desktop around 43 percent. Add in tablets (they're mobile, and the share goes up to 56 percent). This shift in market share means that buyers are slowly moving away from the desktop as the default device for purchasing online. According to the latest eCommerce report from GlobalWebIndex

(globalwebindex.com), 54 percent of internet users purchased on mobile and 44 percent on desktop PC/laptop.

It won't come as any surprise that the under-45s are the key players when it comes to online buying. But head to Europe and North America and PC and laptop are still the preferred choice (that's the over-45s for you). The online purchasing market is not consistent

across continents either. While some form of online commerce activity is typically over 90 percent across most regions, the Middle East and Africa are the exception. With a more cash-driven economy most transactions are completed offline, though there is an appetite to purchase online – they just need to introduce the right technology.

So where do consumers start their purchase journey? For brand discovery,

brand sites and price comparison sites all sharing a chunk of the process. Not far behind are Q&A sites, coupon sites and blogs on products and brands.

So what are people buying/researching online? Personal effects/gifts are the most researched and purchased online, closely followed by Grocery & Household. This category is helped by Amazon introducing food and combining this with its Echo smart speaker. Away

“54% of users purchased on mobile and 44% on PC/laptop”

TV and search engines are neck and neck with word-of-mouth recommendations sitting very close. Women are more likely to hear about brands through personal recommendations, the report says.

The younger consumer is looking to social for info on products, but it's search engines that top the bill for product research. Social networks are a close second with consumer reviews, product/

from physical products it's all about downloads and streaming, with music and movie/TV services being the big winners. News is making small gains, while the once popular e-book is slowly dwindling in popularity. Finally, what's tempting consumers to take the last step and make a purchase? Free delivery is the prime motivator wherever in the world the buyer is – we all like something free.

CB 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/2qLxVl4>

STAT ATTACK

MOBILE VENDORS

Who rules the mobile device market across the globe?

Samsung

30.66%



The makers of the popular Galaxy S series

Apple

18.91%



Popular in North America and Oceania

Xiaomi

7.02%



Brilliant budget phones. Big in Asia

Huawei

7.01%



Increasingly popular in Europe

Unknown

4.12%



Lesser-known brands

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

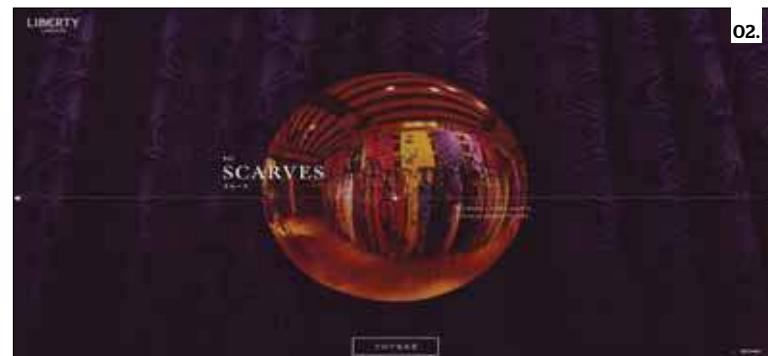
Sites of the month

01.

BRAND EXPERIENCE

Nike SkAir

[VIEW PROJECT](#)



Graphics Manly Matches

bit.ly/2uOAcxa

Who doesn't like retro design? Check out this Swanson tribute from Chet Phillips.



Colour picker ASAP

bit.ly/2LoOtL4

#46A598

#4EBFA1

#5CC7D2

#B2D235

#576DB4

Typesetter DIN Next Decorative

bit.ly/2uPrn6i

An update to the popular DIN family. Now with added textures and new variations.

ABCabc
012345

WordPress Lunatic

quomodothemes.website/lunatic

Need to promote your very own app? This single-page theme has the answers.



webkit

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

Accessibility for Teams

A 'quick-start' guide for embedding accessibility and inclusive design practices into your team's workflow

Everyone who works on government websites has a role to play in making federal resources accessible and inclusive.

Choose the guide that fits your role:



Product



Content



UX



Visual design



Front end

These roles are based on the roles we have at the Technology Transformation Services at GSA.

Accessibility for teams

accessibility.digital.gov

It's always interesting to see which design practices the big corporations and governments are using. Are they doing it right? Take a peek at this US government 'quick-start' guide with sections on UX, Content, Product, Visual design and Frontend.



Spectator

netbasal.gitbook.io/spectator

Need to test your Angular code? Spectator is written on top of the Angular Testing Framework and provides a cleaner API for testing.



Term Sheets

gpoitch.github.io/term-sheets

Use this simple tool to create animated Terminal-style presentations. Choose colours, title, width, height and export as an SVG, animated GIF, or HTML & CSS.



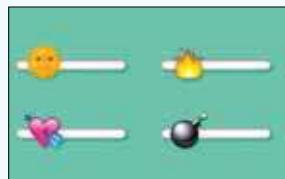
Popbox.js

oncebot.github.io/popbox.js

Popbox is a neat, tiny JavaScript plugin that enables you to create stackable modals. Include the supplied code and start making magic popups.

TOP 5 Codepens to inspire

Head on over to the code playground and find yourself a range of impressive techniques



Responsive Emoji

bit.ly/2LAYXq7

A collection of simple but cute sliders that change from emoji to emoji when activated.



Checkbox challenge

bit.ly/2A4RDym

A smart interactive animation that sees the tick enter the box and status (on the box) change.



Isometric eCommerce

bit.ly/2uLtxKg

Trainers galore in this imaginative presentation of the online shop theme.



Storm

bit.ly/2uYmUOn

Watch as the virtual clouds slowly move across the screen before lighting up as the storm erupts. Just needs sound now.



Always Ascending

bit.ly/2LhQDw5

Mesmerising text effect that constantly repeats the key words. Hard to resist.

Freelancers, contractors and IR35

Does this complex tax legislation apply to you?



David Redfern
Director, DSR Tax Claims
dsrtaxclaims.co.uk

“If you provide your services to a client through an intermediary, you come under the remit for IR35**”**

If you are working as a contractor or freelancer, there is a very good chance that you have heard the term 'IR35' banded around in employment news lately. For such a complex set of tax regulations, it is certainly causing tremors in the contracting sector thanks to a number of high-profile cases featuring freelancers and contractors pitted against Her Majesty's Revenue and Customs (HMRC). But, what is this legislation, and does it apply to you?

IR35 or 'off-payroll working through an intermediary', is a piece of HMRC legislation originally introduced in April 2000 and intended to eliminate tax avoidance by workers who supply their services via an intermediary, otherwise known as "disguised employees" by HMRC. The idea behind it is to crack down on those workers and clients who are evading their full PAYE and Class 1 National Insurance responsibilities by falsely claiming self-employment when they are de facto employees. And as a piece of legislation, it sounds perfectly reasonable – HMRC should have the powers necessary to crack down on those claiming false self-employment for tax purposes. In practice, however, it isn't proving to be so simple, and that's the reason for the multitude of articles on the subject.

If you provide your services to a client through an intermediary, such as a personal service company (PSC), you come under the remit for IR35 which means that you need to be aware of the bounds of your contract with your client to ascertain whether this legislation shows you to be employed or self-employed. And it should be a simple process – HMRC has even designed an online tool to help you to determine whether you are classed as self-employed in the eyes of HMRC. But due to the complex nature of IR35, it is proving to be anything but simple.

Some of the criteria for self-employment used by HMRC and CEST (Check Employment Status for Tax) are easy enough: do you use your own equipment, are you allowed to work for more than one client, do you have the right to send a substitute in your place? Unfortunately, with IR35, HMRC fails to recognise the range and variety of working practises and contracts within our modern, contemporary economy in which an estimated 15 per cent of all workers identify as freelancers and contractors, 200,000 of them through PSCs.

Reforms to the legislation made in April 2017, placed the onus on determining who was to be classed as employed or self-employed to public sector employers, resulting in such organisations making blanket decisions about the contractors within their organisation – with disastrous results for some of those contractors. After all, when HMRC decides to investigate it's the taxpayer who gets investigated, not the organisation who deemed them self-employed. The media have made much out of the case of Christa Ackroyd, a BBC presenter whose tribunal case loss against HMRC is likely to cost her an extra half a million pounds in tax, but there are many other lesser-known contractors who are being forced to battle HMRC through the courts – such as IT contractor Ian Wells, who provided services to the Department of Work and Pensions (DWP) through his own limited company and found himself running up against HMRC, who believed he should have been classified as an employee rather than a contractor. In that case, the tribunal found in favour of Mr. Wells – luckily for him, because HMRC believed him to owe an additional £27,000 in unpaid tax.

Further controversies, surrounding the accuracy of HMRC's CEST tool and its omission of mutuality of obligation (MoO), may give you a clearer idea of why it is so vitally important for all IT contractors to upscale their knowledge and awareness of IR35, and its implications for their own employment status and tax position. With HMRC looking to extend the 2017 reforms to the private sector soon, IR35 will draw more contractors and freelancers into its net. As tempting as it can be to bury one's head in the sand when contemplating taxes, I would urge all freelancers and contractors within the IT industry to obtain clarity on their own employment status without haste. Due to the complexities of the legislation, far greater than I can do justice to here, expert advice may be advantageous to ensure that you aren't involved in a lengthy and costly battle with HMRC.

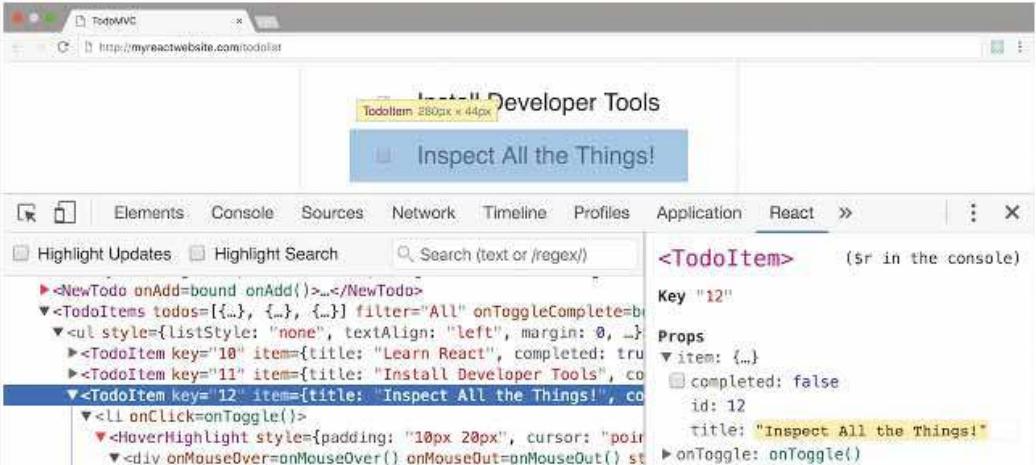
webkit

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

 **React Developer Tools**
offered by Facebook

★★★★★ (1010) | [Developer Tools](#) | 1,118,446 users

OVERVIEW **REVIEWS** **SUPPORT** **RELATED**



Adds React debugging tools to the Chrome Developer Tools.

React Developer Tools is a Chrome DevTools extension for the open-source React JavaScript library. It allows you to inspect the React component hierarchies in the Chrome Developer Tools.

You will get a new tab called React in your Chrome DevTools. This shows you the root React components that were rendered on the page, as well as the subcomponents that they ended up rendering.

By selecting one of the components in the tree, you can inspect and edit its current

React Devtools

bit.ly/2g5FyOk

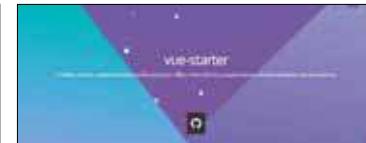
React Developer Tools is a DevTools extension for Chrome and Firefox. Users can get a look at a view of the component tree and the current state and props of each component selected. Check out this brief introduction to learn more: alligator.io/react/react-devtools-intro



Browsh

browsh.sh

Browsh is a modern, text-based web browser. It can render most things that a contemporary browser can – HTML5, CSS3, JS, video and even WebGL.



Vue-starter

vue-starter.herokuapp.com

A flexible, scalable boilerplate for production-ready PWAs. It has a focus on performance, development speed, and best practices.



Sucrase

sucrase.io

Sucrase is an alternative to Babel that allows super-fast development builds. It assumes users are targeting a modern JS runtime, avoiding older ones.

TOP 5 WordPress themes

Need to get a good-looking website up and running quickly? Try one of these themes



Lexio

lexio.famithemes.com

Starts off fullscreen and stylish, before delivering a host of standard sections that make any site essential.



Andé

ande.mikado-themes.com

If food is your business then this modern, elegant theme has plenty of options and styles. Select the one that suits you.



Cosmetista

bit.ly/2NKICg6

Plenty of white space, big images and contemporary fonts ensure that the viewer is always focused on the product.



Sweetness

bit.ly/2v4Jp3A

A single-page site with a host of sections to celebrate the world of cakes. Includes blog, gallery, menu and services.



Kossy

bit.ly/2NJwKuR

A responsive WooCommerce theme with a multitude of home page and standard page layouts to get you selling today.

HEAR FROM THE PROS

SIGN UP TO THE

designerTM

NEWSLETTER TODAY!



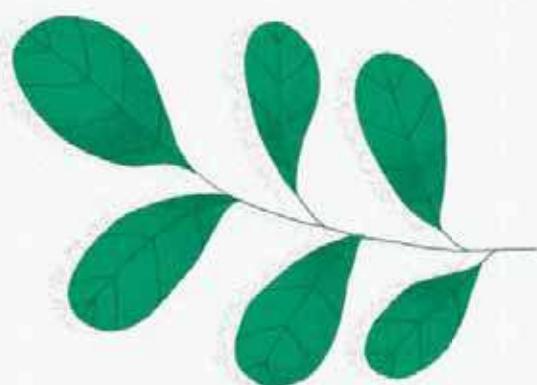
Get weekly news, tips & inspiration

SIGN UP NOW!

bit.ly/2KI5b7Y



Creative Food & Pr Coffee

[Scroll to discover](#)

“A fancy front for fancy fast food, this dynamic menu site tastefully mixes scrolling navigation with delicious transitions”

Order



ve
mium



Colours



#EFF3FO



#215732



#52C0A9



#FAC2A1

Tools

Amazon CloudFront,
WebGL, Node.js, GLSL

Fonts

abcABC
1234567890

Rational TW by Rene Bieder is the typewriter variant of the Rational font family, and is used exclusively across the site in Text Medium style.

Contact



soups

creamy broccoli

£4.75



The image displays two screenshots of the KOOX mobile application. The left screenshot shows a dish called "chicken mushroom" with a price of £6.95. It includes a list of ingredients: rye flour, buckwheat flour, butter, ham, eggs, cheese, chicken, mushrooms, and hollandaise sauce. The right screenshot shows a dessert called "chocolate mousse" with a price of £8.75. Both screens feature a clean design with a white background, dark borders, and a green header bar. A navigation component consisting of a diamond shape with an 'x' and a '+' sign is visible on both screens.

Above
Sumptuous food photography really pops against a simple design that combines a limited, clean colour palette with negative space

Far left
On touchscreen devices, menu navigation benefits from a drag and drop slider component for quicker choice transitions

Left
A responsive design ensures that the KOOX experience looks and behaves almost identically from desktop to mobile

Build a fullscreen expanding navigation menu

Use CSS and JS to create a menu that allows navigation to avoid conflicting with your content layout

1. HTML template

The first step is to initiate the structure of the document. This consists of HTML to describe the document container, which stores the head and body sections. While the head section is used to load the external CSS and JavaScript resources, the body is used to store visible content created in the next step.

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

2. HTML content

The content consists of a title heading, and most importantly, the navigation description. This attribute has a 'data-action' attribute set to 'expand', as well as containing two child 'span' elements. The first span will be crafted into the opening button, while the second span is used to store the navigation links.

```
<h1>Expanding Menu</h1>
<nav data-action="expand">
<span></span>
<span>
    <a href="#">Page 1</a>
    <a href="#">Page 2</a>
    <a href="#">Page 3</a>
</span>
</nav>
```

3. Opening click

Create a new file called 'code.js'. This step firstly places a listener on the browser window, which will trigger a function once the page has loaded. This function searches for the first child of any 'nav' element with the 'data-action' attribute set to 'expand'. These elements have a 'click' listener applied, which will toggle the application of the 'open' class to the parent nav container.

```
window.addEventListener("load",function(){
    var nodes = document.
    querySelectorAll('nav[data-action="expand"]'
    *:first-child');
    for(var i=0; i<nodes.length; i++){
```

```
        nodes[i].addEventListener("click",func
tion(){
    var parent = this.parentNode;
    if(parent.classList.contains("open"))
    == false)parent.classList.add("open");
    else parent.classList.remove("open");
})
}
*** STEP 4 HERE
});
```

4. Appearance delay

The final piece of JavaScript searches for the elements inside the second child of the nav container - ie the links. Each of these elements has a 'transition-delay' attribute applied that is calculated in relation to their index position, plus 0.5 seconds. This will result in each link individually appearing in descending order from top to bottom.

```
var nodes = document.querySelectorAll(
    'nav[data-action="expand"] *:nth-
child(2) > *'
);
for(var i=0; i<nodes.length; i++){
    nodes[i].style.transitionDelay =
(0.5+(i/3))+s";
}
```

5. CSS navigation container

Create a new file called 'styles.css'. This first set of formatting sets the navigation container to appear at the right side of the screen. Fixed positioning and a high z-index is used to guarantee its visibility without obstruction from the page content or page scrolling location. The transparent background allows any underlying colours, content or effects of the main page to remain visible.

```
nav{
    display: block;
    position: fixed;
    box-sizing: border-box;
    font-family: arial;
    top: 0;
    right: 0;
    z-index: 9000;
    padding: 0;
    width: 2em;
    height: 100vh;
    background: transparent;
}
```

6. Open button styling

The open button is made from two types of element. The first child inside the nav container is used for JavaScript click detection, while a virtual 'before' element is used to show the status icon. This step uses the 'content' attribute to show the default hamburger icon, along with setting the size and position of the open/close button.

```
nav::before,
nav > *:first-child{
    content: "\2261";
    display: block;
    position: absolute;
    right: 0;
    top: 48vh;
    z-index: 99999;
    font-size: 2cm;
    width: 0.5em !important;
    height: 1em;
}
```

7. Navigation link container

This step styles the second child inside the navigation used to store the links. Fixed positioning is used to make sure that it is always ready to appear into view from the bottom of the screen regardless of the page scrolling location. Size is set so the container and its content is invisible, while other presentation attributes such as background colour are set for future statuses to inherit.

```
nav > *:nth-child(2){
    position: fixed;
    height: 0;
    width: 100vw;
    bottom: 0;
    left: 0;
    background: red;
    opacity: 0;
    transition: opacity 1s;
    overflow: auto;
}
```



Epicurrence - Summer Revival

epicurrence.com

EPIC

Welcome to the Revival

**Mountain Air.
Inhale. Exhale.
Adventure.
Relax.
Revive.**



*BADGE BY ALE

*ILLUSTRATIONS BY ANASTA

Designers:

Aristide Benoist aristidebenoist.com Zhenya Rynzhuk zhenyary.com Anastasia Rud dribbble.com/anastasiarud



“More fireside chat than typical conference, this web flyer for Yosemite’s latest Epicurrence meetup features beautiful illustrations”

Colours



Tools

HTML5, SVG, JavaScript, ASP.NET

Fonts

abcABC
1234567890
abcABC
1234567890

The Suisse Int'l font, formerly known as Suisse BP International, appears first in Regular and Semi Bold varieties.

abcABC
1234567890
abcABC
1234567890

Suisse Int'l is then also used in its Mono form, with Regular and Bold typefaces found in the site's stylesheet.



00:47 / 00:55



Above

A fullscreen embedded video plays a stunning sequence from previous Epicurrence events on a repeating loop

Far left

Parallax scrolling brings the charming and warmly coloured background illustrations to life, with text and images floated above

Left

Epicurrence on mobile places the Get Ticket link right up front to make attendee registration even more convenient

Introduce a background zoom effect for page scrolling

Present a header section with a background image that resizes in response to user scrolling

1. Document template

The first step is to define the HTML document template. This consists of the HTML document container used to store the head and body sections. While the head section is used to load external CSS and JavaScript resources, the body is a container for storing the visible page content defined in step 2.

```
<!DOCTYPE html>
<html>
<head>
<title>Background Zoom</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 visible HTML content consists of a 'header' section containing a 'h1' heading, along with a container for the main page content. The unique header section will be referenced by JavaScript and CSS to apply the image effect. The use of the main container allows the effect to avoid interfering with the main page content.

```
<header>
    <h1>Some Title</h1>
</header>
<main>
    Content goes here!
</main>
```

3. CSS document initiation

Create a new file called 'styles.css'. This step initiates the default state of the document and body containers. These are set to cover the full screen size, with default colours for background and text. Padding is applied to the bottom so that there is available space to scroll for observation of the effect. This padding can be taken out for your real project.

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

```
padding-bottom: 300%;
```

```
text-shadow: 2px 2px #000
```

4. Header container

The header container is set to its required size - set to match the width and height of the screen for this example. The background image is applied with both horizontal and vertical positioning set to 'center'. The background size is set to match 100% of both width and height of the header element.

```
header{
    display: block;
    position: relative;
    width: 100vw;
    height: 100vh;
    text-align: center;
    overflow: hidden;
    background: url(background.jpg) center
    center;
    background-size: 100% 100%;
}
```

5. Header heading

The 'h1' element inside the header section is set with a colour and text shadow that will make it stand out from the background image. The font size is also set so that its text appears prominently - six times bigger than the font size inherited from the default size for the page.

```
header h1{
    font-size: 6em;
    color: #c00;
```

6. Main content

The main content container is set to appear across half of the page width using a font size that's three times bigger than the default page font. Auto is applied to the side margin to make sure that the main content container is positioned in the middle of the screen.

```
main{
    width: 50%;
    font-size: 3em;
    color: #fff;
    margin: 0 auto 0 auto;
}
```

7. JavaScript scrolling listener

Create a new file called 'code.js'. This step attaches a 'scroll' event listener to the browser window. When page scrolling occurs, this function will be triggered to calculate a new size for the background image based on the new scroll position. The calculation uses the 'speed' variable to define how fast the image should be zoomed in relation to the scrolling. Feel free to experiment by increasing or decreasing the speed variable.

```
window.addEventListener("scroll", function(){
    var speed = 5;
    var size = 100+(window.scrollY/speed);
    document.querySelector("header").style.backgroundSize = size+"%" "+size+"%";});
```



Epicurrence is the original activity-focused non-conference for creatives that

Join the crew in Yosemite National Park as we enjoy a week loaded with hiking, mountain biking, rock climbing, dirt bikes, treasure hunts, unheard of stories from creatives of all types and inspiration everywhere.

Epicurrence is the definition of an adventure conference.

Van Holtz Co

<https://vanholtz.co>

Designer:

Eric Van Holtz <https://vanholtz.co>

BR

2017 / ONIX

2018 / SP

VAN
HOLTZ
CO.

Studio of Eric Van Holtz
Web Developer

Portland Oregon
hello@vanholtz.co

21 about
22 journal

“Specialising in refined digital web experiences with a focus on animated, responsive and interactive content”



Colours



#4801FF

#000000



#fdfdff

Tools

jQuery, Barba.js, Flickity,
Fizzy UI Utils

Fonts

abcABC
1234567890

abcABC
1234567890

The Druk font by Berthon Hasebe for Commercial Type is featured here in Wide Web Medium and Super varieties.

abcABC
1234567890

Akzidenz-Grotesk from Berthold is the second font served to the site via Typekit in Std Regular and Std Med typefaces.

Create an animated 3D outline menu effect

Use some simple CSS transform and transition effects to add some depth to your menu navigation

1. HTML initiation

Initiate the HTML template by creating the page template. This creates the document container, along with its head and body sections. The head's primary purpose is to load the external CSS stylesheet resource, while the body section will be used to present the visible content elements in the next step.

```
<!DOCTYPE html>
<html>
<head>
<title>3D Hover Outline</title>
<link rel="stylesheet" type="text/css"
href="styles.css" />
</head>
<body>
*** STEP 2 HERE
</body>
</html>
```

2. Page content

The visible page elements consist of the navigation container and its child links. Each child link has its own text to display, along with a page or element to navigate towards using the 'href' attribute. All required presentation settings will be applied to these elements in future steps via CSS.

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

3. Sort the CSS

Create a new file called 'styles.css'. The first step for the stylesheet is to initiate the HTML document and body containers with a default font and colours to display throughout the page. While the font is defined to be inherited by the navigation links, the background colour is also related to the text colour set in step 5.

```
html,body{
  display: block;
  font-family: sans-serif;
  background: silver;
  color: #000;
}
```

4. Navigation container

The default font size of text inside the navigation container is set to be four times bigger than the font size inherited by the nav element. With no font size set so far, this will be four times bigger than the default browser

font size. Navigation elements will also use bold text.

```
nav{
  font-size: 4em;
  font-weight: bold;
}
```

5. The outline

The only way to achieve the outline effect with cross browser compatibility using regular HTML/CSS is to create the outline using a text shadow. This requires multiple shadows to be strategically positioned with zero opacity to build the outline. The downside to this technique is that the text can't appear hollow, hence the need to set the text with the same colour as the page background.

```
nav a{
  display: block;
  color: silver;
  text-shadow:
  -1px -1px 0 #000,
  1px -1px 0 #000,
  -1px 1px 0 #000,
  1px 1px 0 #000;
}
```

6. 3D settings

The 3D positioning effect is defined using the transform attribute to adjust the rotation of the links. This rotation is defined to make each link appear with isometric positioning. A transform origin is applied so that all changes are made from the middle left. A transition is also applied so that all attribute changes are animated over a duration of one second.

```
nav a{
  transform: rotateY(-54deg) rotateZ(-10deg);
  transform-origin: 0 40%;
  transition: all 1s; }
```

7. Hover changes

Links being hovered by the mouse cursor are required to be displayed with a new colour and rotation position. With the transition applied to each link in step 6, these changes are introduced as a one second animation. The new rotation swings the text to appear without a slant - fully facing the user.

```
nav a:hover{
  color: #fff;
  transform: rotateY(-14deg)
  rotateZ(0deg); }
```



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   The voice of web design 

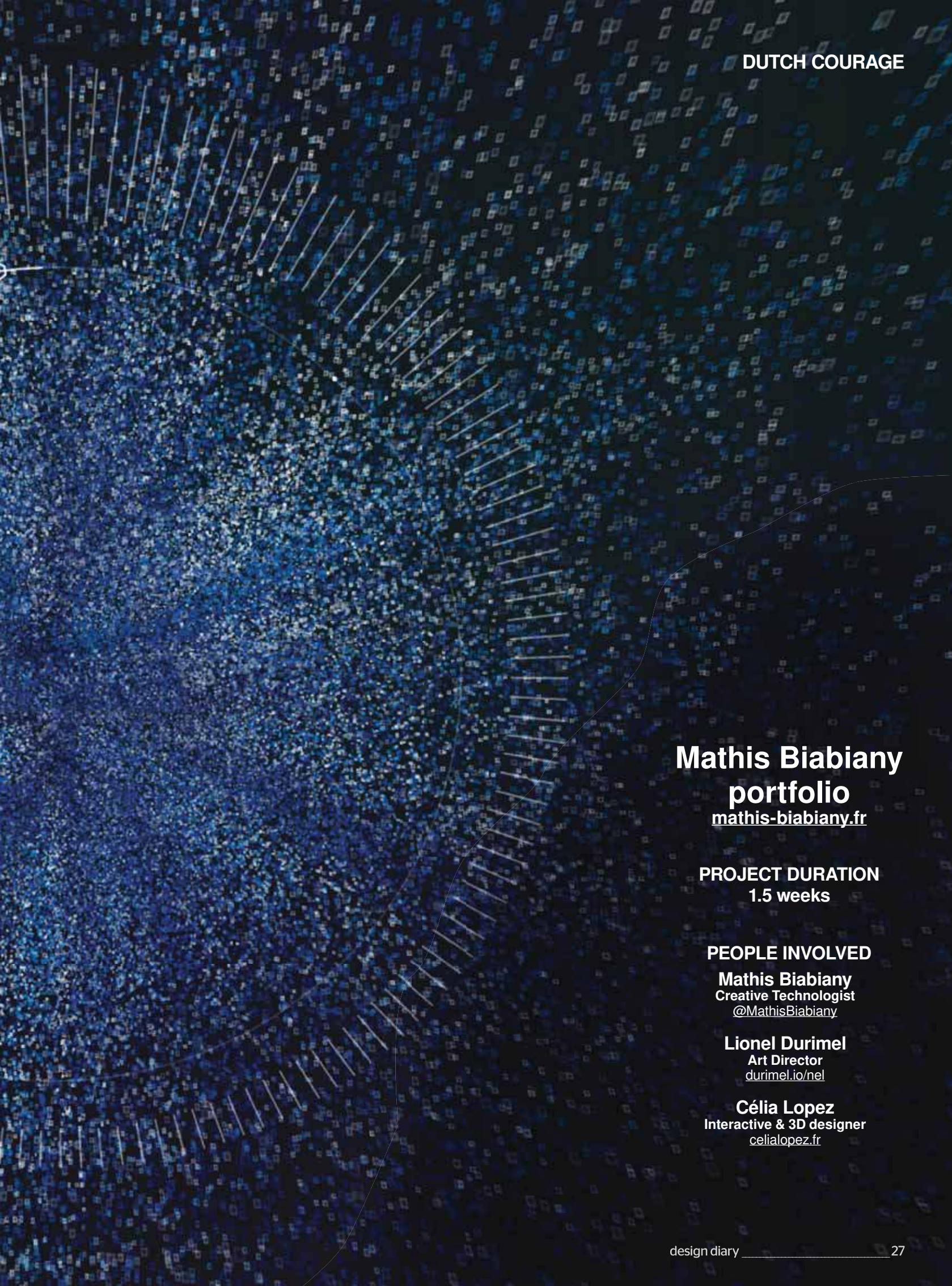
DUTCH COURAGE

HAVING MADE A NAME FOR HIMSELF WITHIN RESN'S AMSTERDAM STABLE OF MAVERICK TALENT, FRENCH DESIGNER MATHIS BIABIANY SET ABOUT CREATING A PARTICLE-PERFECT PORTFOLIO SITE FIT FOR GOING FREELANCE

WHEN YOU THINK OF SOME OF THE MOST ICONIC and inspiring designs that mankind has conjured up, they usually strike an admirable balance between form and function. Sometimes such design aligns much more with artistry, transcending a beautiful surface aesthetic to still deliver a practical use. Web design as a contemporary concept really lends itself beautifully to this strain of design because the advancement in technology and browser standards continues to expand its possibilities. Where 'pages' were once purely passive, modern websites are far deeper and often so dynamic and engaging that they defy convention. This has seen their function change with an ambitious approach to form and presentation that many of the more experimental digital agencies have pursued. One of the most lauded and awarded is Resn, a studio originating out of New Zealand that now also has offices in Shanghai, San Francisco and indeed the Amsterdam office where our latest hero, Mathis Biabiany, cut his teeth. Fans of this startlingly brilliant agency will know that its manifesto to "infect minds with gooey interactive experiences" has not only been hugely influential to a progressive, artistic web design movement but also a furtive plateau for the talent it enlists. Biabiany, a 24-year-old Creative Technologist, joined Resn in 2016 after completing his studies in web development and multimedia direction at the Gobelins School in Paris. "For almost two years now, I've enjoyed

"MY PREVIOUS PORTFOLIO WAS GETTING QUITE OLD, CONSIDERING I MADE IT WHEN I WAS STILL AT SCHOOL, AROUND THREE YEARS AGO"

working on some awesome projects with handsome teammates," begins Mathis. "I've learned a lot from them, but now I'm taking on full-time freelancer status and preparing myself to leave Amsterdam to try new challenges in Japan." So in short, our man needed a new digital portfolio, something he had considered realising after his first year at Resn. "I made couple of projects and experiments and my previous portfolio was getting quite old, considering I made it when I was still at school, around three years ago now." The challenge was to create an experience that would pay homage to his Resn roots, but also demonstrate technical skill and personality to match. ▶



DUTCH COURAGE

Mathis Biabiany

portfolio

mathis-biabiany.fr

PROJECT DURATION
1.5 weeks

PEOPLE INVOLVED

Mathis Biabiany
Creative Technologist
[@MathisBiabiany](https://twitter.com/MathisBiabiany)

Lionel Durimel
Art Director
durimel.io/nel

Célia Lopez
Interactive & 3D designer
celialopez.fr

DUTCH COURAGE

FRIENDS WITH BENEFITS

After Mathis found the inspiration he needed, he had a pretty clear vision of how his own portfolio should be. But given his own value of collaboration and the great colleagues he had around him, there was no reason to shoulder the production alone. "I started to create the concept myself but it's still really helpful when working solo on a project to pick the brains of talented people around you," Biabiany concedes. "This is where Lionel (Durimel) really helped me because we thought together about the look of the timeline element and its operation. Also, Célia (Lopez) made all the images as previews for me and I'm pretty sure I wouldn't have been able to do it in the same amount of time without her help. We took some time to pick the right images from all my projects and she started working on that while we discussed animation ideas for the menu on the home screen too." Such practical assistance certainly works wonders on a project where time is short and much is being done in those spare hours around your full-time agency job. Equally, though, you feel this effort really benefitted from the bouncing of ideas so things didn't get too insular, while never expanding practical input too much and diluting the central purpose.

"In terms of the development, I coded everything by myself and it wouldn't have been terribly valuable to add other people into that process. Primarily because I really wanted to handle the development duties for myself and stay true to that seeing as it is my own portfolio!"



SOLIDIFYING A CONCEPT

"It's, of course, thanks to my wish to become a freelancer that I needed a strong portfolio, able to show all of that stuff effectively while being beautiful and interesting to navigate through," Biabiany explains. "This website would help me to improve my own visibility on the web and hopefully provide a nice place to make first contact with potentials collaborators and clients, etc." Mathis already knew he wanted this 'nice place' to highlight two important areas of work, with the first being his experience in team-based collaborative projects for high-profile clients. Secondly, his collections of personal experiments amassed during a lifetime learning to code. "They always were important for me because I always made them in order to learn something specific which could be useful later." Focus would then shift naturally to seeking a visual inspiration that might inform the presentation of this exciting work. Mathis scoured some of his favourite online sources here such as Behance, Pinterest and **DesignInspiration.net** before happening

across Infilm After Effects Templates, a YouTube channel showcasing animations he could see using for the portfolio's navigation. "The motion was showing particles in a perspective view, which were moving in depth. When they stopped to move, a kind of fade was happening between the particles and a visual. I thought that if I could use the same logic for presenting my projects, I would be able to make something nice and easy to use." This concept would initiate some coding tests that prove the technique to be more difficult than first thought, but gave Mathis enough impetus to consider the kind of UI needed to merge with the animation. The findings here would lead to the enlist of friend and colleague Lionel Durimel, with the two talking at length about what Biabiany hoped to achieve. "His expertise was really useful," beams Mathis. "I believe that every good production starts with a really solid concept so we took a long time to discuss about what I should do and include into the website. As long that the concept is done, all the rest will come way more simply."



TOP LEFT
Grab the cans, wiggle the cursor and wait to see what happens

TOP RIGHT
Resn's Little Helper powered by WebGL and Three.js

BOTTOM LEFT
Tucked away top right is a navigation that is ready to go

BOTTOM RIGHT
Check out the work of friend and designer Célia Lopez

"MY PORTFOLIO DOES NOT INCLUDE A LOT OF BACKEND DEVELOPMENT BECAUSE IT DOES NOT REALLY NEED IT, AND ALSO BECAUSE I'M NOT GOOD AT IT"

PARTICLE THEORIES

The visual concept was, of course, established, but the big focus had to be on making it functionally plausible – adding that design to the art. Mathis knew it was vital to keep the user informed of where they were within this depth particle animation, make it a viable navigational element. "The idea was to use scrolling to control the particles and go through the projects, but I needed something more than just modifying an image to show that the user was switching between them. The user needed to know where they were within the navigation and what the current project is, etc., so with Lionel we thought about a circular timeline which would appear only when modifying the particles positions, which meant, eventually switching projects." This timeline device would display the current project title while indicating the current depth position within this abstract particle world. Users could then jump to specific projects via timeline hotspots, or use a more conventional method to keep things intuitive. "This is why we thought about a menu, listing

all the projects, directly on the main page, which would be easier to access by using a simple hover action." At least Lionel was on hand to help with the page designs, producing wireframe templates that could be applied throughout. His work on the 'About' pages here would carry over to defining various details of the main page UI, deciding on where titles, social network and contact links should be positioned. Similarly Mathis also had the aid of a second helpful friend in the hugely talented Interactive Designer Célia Lopez, primarily enlisted to take care of the featured project previews. "I sent her some visuals to crop in order to have some nice images, which I then used to colourise the particles with all the previews positioned into a spritesheet. Célia was also great in sending me additional ideas about how the menu itself should animate."

A SHADE DIFFICULT

When the subject moves briefly to how any backend coding work was handled, Mathis is the first to admit this was deliberately quashed.

"My portfolio does not include a lot of backend development because it does not really need it, and also because I'm not good at it," Biabiany chuckles self-effacingly. "I always prefer to focus on the front part of a production." So in this instance, in order to quickly add another project or experiment to the site, he simply added a file to fill in order to list them. This file would provide all the information the website would need to work, such as titles, descriptions and assets used, etc. Something slightly more challenging was catering to different platforms of course, a typical conundrum when trying to offer consistency and adequate performance. Visitors to the final live site will know that the portfolio does not look the same on mobile and desktop, largely because the same kind of scroll couldn't be used. So Mathis did two separate builds effectively, using server-side checking to deliver the right version to the target device. "On the other side, I think the most challenging thing I had to do on my portfolio was to keep the performance high. As a website dedicated to show projects that consume lots of ▶

resources, it was mandatory." This was highlighted when an initial approach to give colour information independently to each particle worked to a point, only to show issues on mobile when the shaders Mathis wrote weren't supported by iOS. "This is why I changed my method here and put all my colours data into the spritesheet instead. This way I just had to provide a texture into my shader and going through the images while the user was navigating. By scrolling, the user increments a value in my shader that is then picking the right colour for my texture."

"IF I THINK THAT THE PROJECT HAS ENOUGH VALUE, I MIGHT SUBMIT IT ON THE FWA, AWWWARDS AND CSS DESIGN AWARD"

UNIFORMLY ACCEPTED

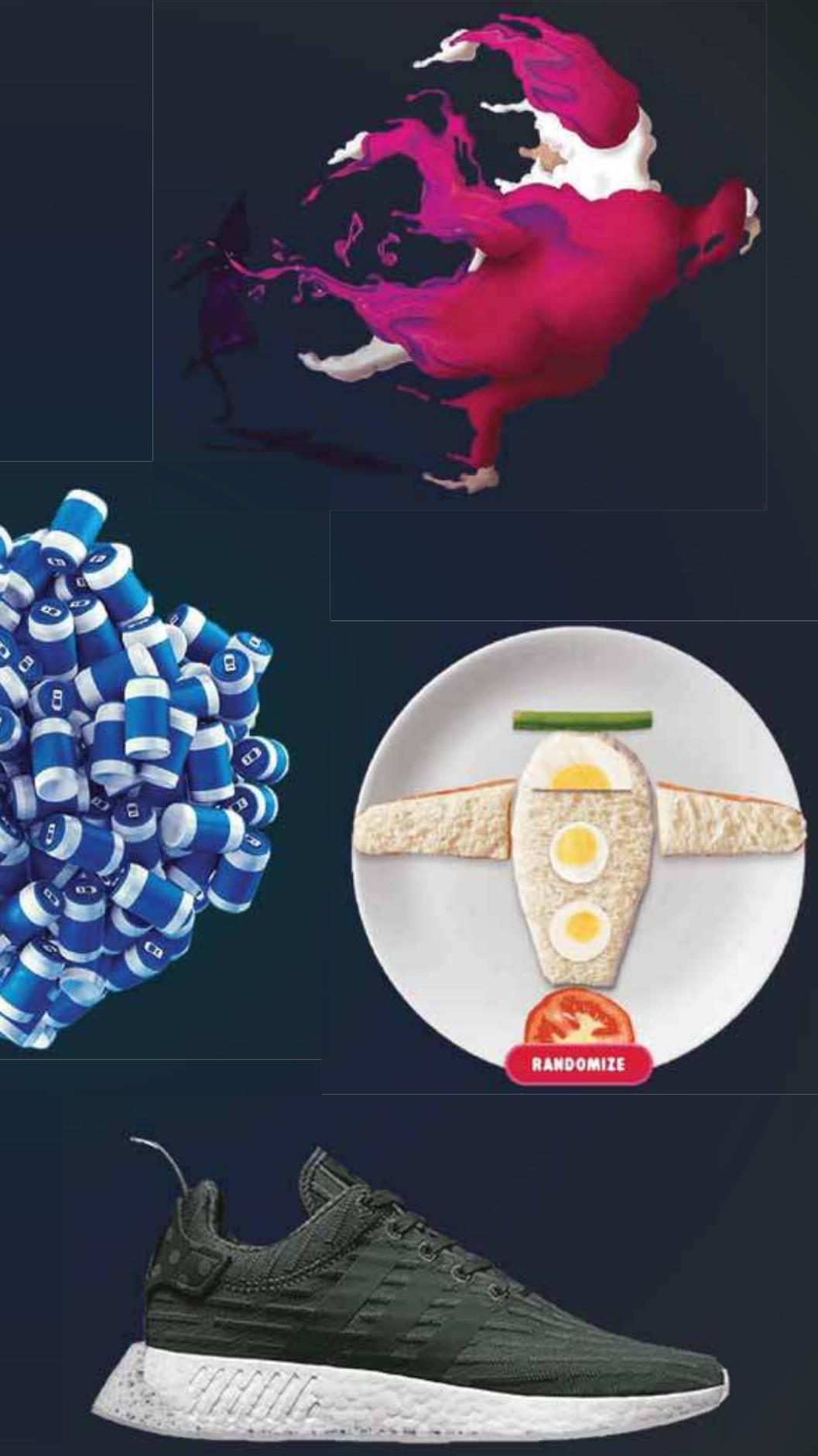
These little challenges fed into the general struggle to make the whole website uniform and coherent. Given its very experimental nature and construct, everything would change and react during the time a user would scroll so it wasn't easy to do in terms of structuring the code. However, he achieved it and all within less than two months of production turnaround in his own private spare time. So what about the launch then Mathis, how does a freshly freelance digital genius unleash his finished portfolio on the world? "OK, well, each time I release a personal production," he laughs coyly. "I tend to share it on Twitter and Facebook as this is, in general, where I get the most visibility. Recently, too, I've started to do the same on Instagram. Then, if I think that the project has enough value, I might submit it on the FWA, Awwwards and CSS Design Award. As long as you maybe get a site of the day on there, you know that many people will potentially have a look into it. Such exposure can be a really good opportunity to make contacts." This was, after all, much of the motivation behind the project in the first place. Far from a vanity exercise started to pass the time between him and two friends, the new Mathis Biabiany portfolio site might have been that, but crucially it would be an engaging platform for selling his considerable talent. "In my opinion, as a final thought, I think my portfolio and the project worked out pretty well," our hero beams. "The original navigation concept I realised and the projects it showcases are what made it a success. Also, it fulfils my goals to bring me new contacts to work with and a nice platform to communicate what I can bring." We couldn't agree more Mathis and wish you every success! ■

SITE HIGHLIGHT

We put Mathis on the spot and ask which feature or aspect of the project turned out to be a particular highlight and why it provides such a lasting impression.

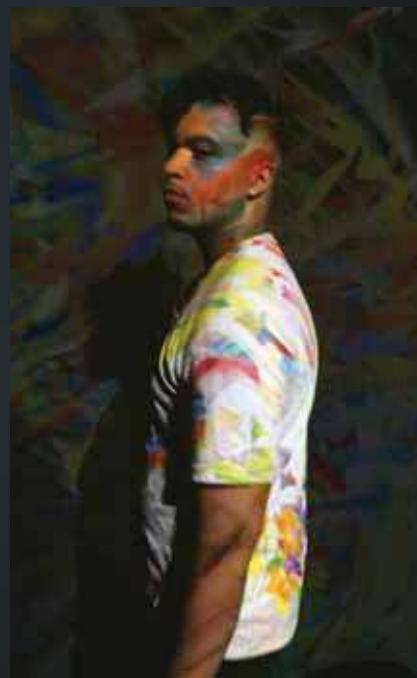
"I think the navigation I built in order to go through my projects is the standout feature. I don't think it has been seen on another digital production and that is what people have remembered the most. I hope it helped to show one more time that we can literally do whatever we want on the web nowadays"





BACK TO THE FUTURE

"I'm planning to use this website as a showroom, especially for my future clients," Mathis reiterates in response to what his future plans for the portfolio might have in store. "The other reason was also, of course, to give me more visibility on the web." Both stand to reason and sum up why an effective digital portfolio and especially a website warrants such an investment of effort for online creative professionals. "Since I released the website, I have indeed received a lot of offers for work and considerably more than before. So within that I have a lot of interest to keep it updated." Cleverly, Biabiany factored this possibility in and conceived it in a way that would make it possible to add subsequent projects or experiments fairly quickly. "I also thought about adding another section for eventually, perhaps, writing some articles about how I'm doing specific things within development. I've always received a lot of help on my path so I would love to give that back and sharing is definitely what I like the most within the work we do. Plus, as a personal portfolio, I might want to do a total revision in a few years but it depends, I think, on how I'll manage my time when my professional situation changes."



SUBSCRIBE

TODAY AND RECEIVE AN

ECOCOFFEE 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 October 2018.

BASIC EXPERIENCES

At the intersection of branding and experience you will find BASIC. Crafting the digital spaces that form daily life, BASIC can see tomorrow. As digital platforms develop, this design agency continually pushes the boundaries with creativity and innovation

Who BASIC

What Brand Development and Identity, Web and Digital Experiences, Advertising and Marketing, Product Design and Packaging, Video Production and Animation, Content Creation and Strategy, eCommerce Design and Development, and UX/UI Design

Where 251 10th Avenue,
San Diego, CA 92101

Web basicagency.com

Key Clients Under Armour, Beats by Dre, Snapchat, Adobe, L'Oréal

THE TRAJECTORY THAT LEAD TO THE FOUNDING OF BASIC IS UNCONVENTIONAL TO SAY THE LEAST

Founder Matt Faulk, a pro-BMX rider, suffered a career-ending injury and was no longer able to compete professionally. In recovery and with a lot of downtime, he considered his next steps.

During his BMX years, he had enjoyed working with brands through sponsorships, events, and advertising. He was inspired by the way strategy and design came together to connect people. From that, he decided to pursue a new-found passion for design, and fell in love with the industry while developing his craft through self-taught instruction and failures (opportunities).

He built a thriving freelance career leveraging his personal contacts and slowly

taking bigger and bigger jobs. Finding freelancing limiting and coming to an understanding that this industry is about connecting to people, he decided he wanted to bring other people together and founded BASIC with co-founder Erich Broesel, who already had a small branding studio.

Naming an agency is a challenging task. You want something memorable, aligned with your values, and something that is ownable within the industry.

Matt explains their approach: "We named our agency BASIC because to us, the best brands are simple ones. Brands thrive on their ability to be understood.

"The name actually came to us while discussing what it was we actually wanted to solve for our clients. The word 'basic' is ▶



Basic Experiences

actually defined as ‘forming an essential foundation or starting point; fundamental.’ This notion ties into exactly what we do as an agency: We help clients establish and create their future.”

As an agency that is creating new digital experiences, their own website has to be a strong indicator of not only their skills, but also their approach to design. “We believe an agency site is very important,” says Matt. “Clients are constantly looking for talented shops who can deliver on their needs. Because of this, agencies need a presence.

“We put a lot of effort into our site and we continuously evolve it. The site provides an opportunity for others to connect with us, see what we are doing, who we are, and how we do things. It is an expression of our culture and it will evolve as we do. While there are outlets like Dribbble and Behance, we feel there isn’t anything better than your own dotcom. It’s done some great things for us and we’re excited to continue to evolve it.”

In the early days of BASIC, Matt leveraged existing connections. Today, BASIC use a multifaceted approach to gaining new clients as Matt explains: “Over the years, we’ve made a lot of friends (clients) through our focus on doing great work for our partners, as well as building our own agency brand. Through this, we’ve earned a strong reputation that has brought us new opportunities.

“What’s unique about us is that we’ve always put the work first. We don’t have an outbound sales program and instead focus our efforts on creating work that gets talked about. By putting the work first, it has made repeat business easier, as well as made new business inquiries

more reliable as they are typically through referral from work we’ve done for others.

“As far as pitching goes, it’s part of the game for some of the

accounts we work on. This is typically part of the process as clients solicit us for potential work. We don’t always pitch, but it is becoming more and more common as we’ve grown.”

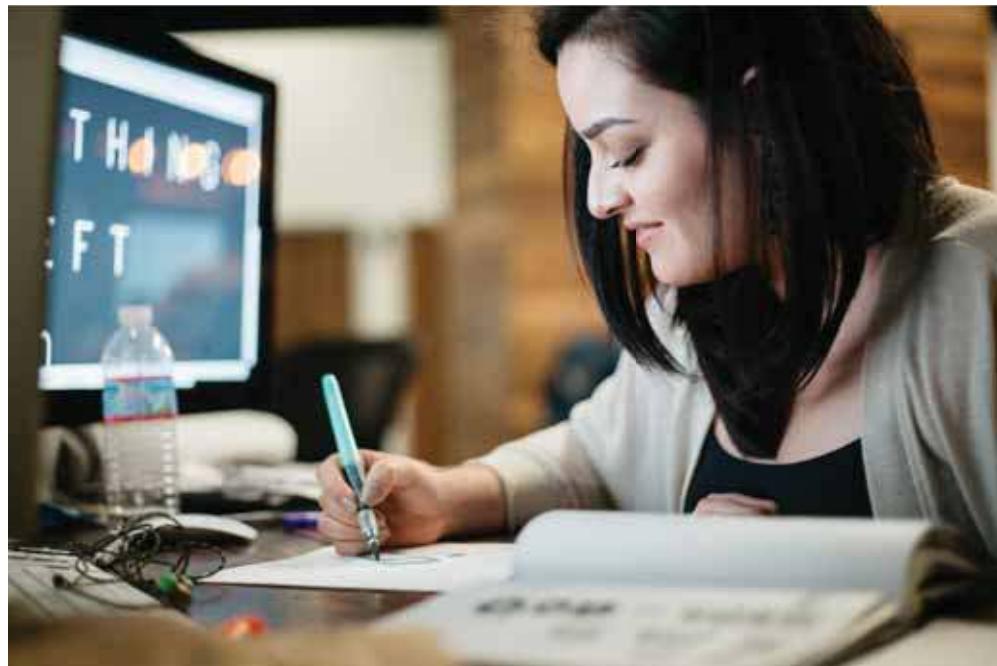
Choosing which clients to work with can be a complex equation that needs to balance the commercial aspects of a project, the brand in question, and the needs of BASIC. Matt outlines their approach: “We’re very selective regarding who we work with and the types of projects we take on. To us, what matters most is fit and alignment. We want to have a shared passion and vision for the work. It has to be a category we’re interested in. We want to be aligned on expectations and what can be done together within budget and the timeline. We want to enjoy the people we’re working with. If we

“We don’t always pitch, but it is becoming more and more common as we’ve grown”



“We love how HTML, CSS, and JS are integrating closer to mobile and desktop hardware, and we expect to see more and more offline device control and device interconnectivity. The line between websites and apps are getting blurrier and blurrier, and it will be interesting to see how web apps versus native apps compete in the market”

Matt Faulk, CEO and co-founder



can be aligned on those fronts, we usually want to take it on.”

Taking time to choose the brands and projects BASIC works on, does this mean BASIC has

become known for a certain type of work that encapsulates their design ethos? “Ultimately the work that comes out of here isn’t defined by one project,” says Matt. “It’s a symbiotic relationship that creates something unique and ownable for the brand we are partnering with. It’s more about how we work. If it was about our process, Beats by Dre is a great example. We went beyond the brief. We try to do not just what we’ve been asked, but what’s right for our partners.”

Crafting great digital experiences takes time. For BASIC a typical project will last between six and 12 months. The scale of the project will also dictate how staff are allocated. Typically this will include a Creative Director, Strategist, Designers, Producer, and Developers. The

particular project will also define who is assigned. “We like to staff our teams with individuals who are native to the category we’re working on, combined with specialists of a specific skill such as interactive design or branding,” Matt explains.

Ashley Reichel, Vice President of Operations also comments: “We scope our projects in design sprints, grouping deliverables within each sprint to create an agile process. It starts with a discovery and look-and-feel sprint to establish the strategic approach and visual aesthetic, from there we begin tackling groupings outlined in the scope. Development and design execution is, typically, the most time-consuming phase, but many engagements can take months for just the strategy and the vision.”

The toolset that any agency uses can be highly varied. For BASIC, the rich development environment that has spawned new tools appearing regularly, just gives this agency a fun playing field to work with. Ashley outlines BASIC’s approach: “From pen to paper, ▶

Culture
Manual

Our Values

The core of our company culture since day one.

01

Our Way

The expectations that fuel our team's success.

02

Our Offices

The spaces where we come together to create.

03

Our People

The diverse group that makes things happen.

04

FAQ

A completely different approach to the employee manual is typical of BASIC that strives to innovate in all they create

Intro



Basic Culture

A guide to being more Basic.



Intro — Important stuff



Our Values

01 — 03

Love what you do.

This is why we do it. Passion. Hustle. Grit. Dedication. Fun. Positivity.

Using a new visual language not only communicates to employees, but also to a wider audience, explaining who BASIC are



BASIC Culture Manual

culture.basicagency.com

As we rapidly grow, it is critically important that our values and vision remain intact. In March, we introduced the BASIC Culture Manual, an interactive digital guidebook that serves as a fresh approach to the employee onboarding process and builds alignment towards our collective future.

The manual strengthens our company culture and attracts world-class talent, acting as an expression of our brand, a guide for our way of doing things, and as a tool for recruiting. The site includes an all-new visual language — typographic framework and adaptable identity — that showcases our history and work we've done for billion-dollar brands.

It flips the script on the traditional PDF manual and features individual Q&As with BASIC team members, and a list of our favourite local spots in each city, all providing a closer look at the people and places that are moving our company culture forward.

Since launching in March, the site has been widely recognised and awarded by the design community including the FWA, Awwwards and CSS Design Awards.

Intro — Our Values



Speaking directly to the design community, BASIC's podcasts take a deep dive into every aspect of the creative digital landscape



Brandbeats

WHAT DOES IT MEAN TO BE A DESIGNER?

Brandbeats Podcast

brandbeats.basicagency.com

We launched the sophomore season of our Brandbeats podcast in March, which attracts a global audience of like-minded creatives, garnering 45,000 listeners per episode, and reaching as far as Ukraine, Japan, Spain and Brazil.

Last year, we released a dozen 30-minute episodes combining music and candid industry discussions with thought leaders covering views on design, technology, art and, you guessed it, culture.

Employees of BASIC sit down with creative minds to break down topics like comparing Nike versus Adidas as culture influencers, diversity in the workplace and the future of user experience with AR/VR, all while expanding into the micro trends and experiences that shift the way consumers think, work and live.

The screenshot shows a dark-themed website for the Brandbeats podcast. On the left, there's a sidebar with a profile picture and the text "Season Two Episodes". The main content area lists five episodes with their titles and small thumbnail images:

- Data Fixation** (Season Two, Episode One)
- Technology in our lives** (Season Two, Episode Two)
- The Power of Politicized Brands** (Season Two, Episode Three)
- Diversity in the Workplace** (Season Two, Episode Four)
- Introduction to Season Two** (Season Two, Episode One)

Covering leaders across the design space, the podcast has become required listening for the entire design community

Post-it notes and white walls to Photoshop, Sketch, Principle and InVision or whatever new prototyping launched that week, we utilise them all."

Ashley continues: "Our typical software stack is the Adobe Creative Suite, Slack, and InVision, for sure. We love them because they're the most powerful in the industry and enable us to work across teams with ease. There's always a lot of debate around software and, to us, that is a waste of time. Our recommendation is to use what works best for your workflow and your people. Great things will come of it."

Matt also explains: "We are excited about the increasing pace of progress in web technology. Until this year, jQuery was part of our regular stack (thanks IE), but now that browsing demographics are in favour of ES6, we're finally able to remove almost all of our third-party core library dependencies.

"ECMAScript (JS), CSS, and HTML have been doing a great job recently at integrating features that previously required competing third-party libraries – like LESS versus SASS versus SCSS, or Backbone versus Babel versus React – and special compilers. The fact that ECMAScript has announced annual releases (no matter how tiny) is awesome, and we hope to see more predictable (and legible) roadmaps for CSS and HTML.

"We love how HTML, CSS, and JS are integrating closer to mobile and desktop hardware, and we expect to see more and more offline device control and device interconnectivity. The line between websites and apps are getting blurrier and blurrier, and

it will be interesting to see how web apps versus native apps compete in the market. We recently worked with Snapchat on an AR project that opened our eyes to a lot of opportunities in that space. We think AR has a ton of potential and we've been seeing more and more beneficial consumer experiences coming to life."

Speaking of mobile, does BASIC adopt a mobile-first approach to their designs? "Users don't care if your site is adaptive or responsive, they care about the experience they are having with your site on the device of their choice," says Matt. "We design equally for mobile, tablet, and desktop as user behaviours differ across devices, and what works on one won't necessarily work on all.

"We believe that websites shouldn't just respond to devices, they should respond to multiple contexts including things like priority, content, business goals and user objectives, as well as expectations. We've designed responsive sites as well as adaptive ones, and the approach should be defined based on project goals and the operational capabilities of the client. Adaptive builds take more time and require more resources. This should always be considered in the approach."

The adaptive approach to design, of course, also extends to the ubiquitous social media networks. "The consumer today doesn't experience a brand through a singular channel, they engage with a brand through their method of choice," Matt explains. "With this in mind, brands must consider this in their efforts and look to create a truly holistic brand experience across every touchpoint. When it comes to ▶

"The consumer today doesn't experience a brand through a singular channel"



Timeline

2010

Matt Faulk and Erich Broesel merge independent design agencies to form BASIC in Encinitas, CA.

Employees: 3 and a labrador named Sam



2011

BASIC moves to a new office in Solana Beach, CA building relationships with brands Hyundai, Sempra Energy & Pepsi.

Employees: 5



2012

BASIC lean into branding, key art, and digital experiences work with Sony PlayStation, History Channel, HTC, and REI.

Employees: 9

2013

BASIC gets busy establishing a name for itself at the intersection between brand and digital. We move offices to downtown San Diego in East Village.

Employees: 20

2015

BASIC carves a niche for itself in brand led eCommerce experiences with NIXON. Campaign work with New Era and digital exploration with Blue Shield begin to take shape.

Employees: 22

2016

BASIC forms strategic partnerships with Beats by Dre, BB Dakota, and L'Oréal across digital, content and brand channels.

Employees: 28

2017

BASIC continues its digital exploration across new industries. Working with new clients including RIOT Games, Volcom, and Snapchat.

Employees: 35

2018

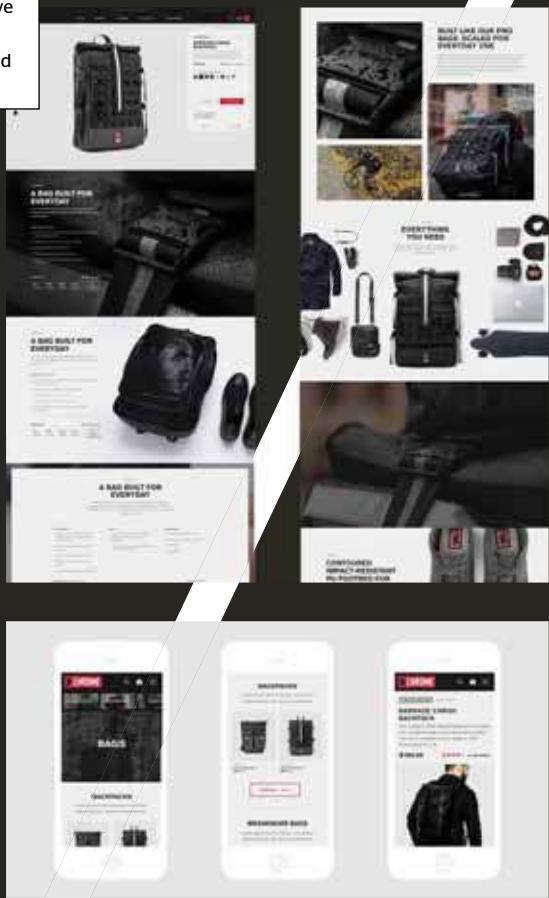
BASIC opens another office in Mountain View, CA

Employees: 70

Basic Experiences

FOR OVER 20 YEARS CHROME HAS CREATED STREET TESTED BAGS, FOOTWEAR & CLOTHING MADE FOR LIVING IN THE CITY.

BASIC's philosophy that the device being used should have little bearing on the user experience is amply illustrated with the Chrome mobile site



Chrome
chromeindustries.com

Chrome, an iconic brand built by bike messengers, for bike messengers. Known for functional products, Chrome was looking to create an online presence that was not solely transactional and to break into new audiences.

BASIC invested the time to understand Chrome's culture, the way it creates products and how it interacts with people in the real world. Chrome is a brand with a purpose. Known for meticulously crafted and resolute gear, perfect for urban life. The brand's consumer lives life on-the-go; they explore brands online via organic searches, social channels, advertising and editorial content.

The new visual and modular structure puts the history of Chrome centre stage, features new collections and the stories of brand ambassadors – all building to a new chapter in the brand's strong heritage. Built on Salesforce Commerce Cloud (Demandware), the brand is able to frequently update the site, create custom landing pages, brand pages, and product pages that are filled with unique and personalised content.

The navigation of the site is meant for exploring and the framework reflects the presumption that users will not always enter the site through the homepage, giving consumers the opportunity to craft their own experience based on their browsing habits and preferences.

The new online experience for Chrome brought the brand to a bigger stage, new audiences, and helped it travel fast into the future of eCommerce – all while bringing the already well-established community for the brand together, online.

The new digital experience from BASIC is suitably 'on brand', turning complex to simple in an eCommerce platform inspired by the bike messenger and urban creative



social, understanding the landscape, maximising reach, and leading conversations to dotcom experiences is core to our strategies and activations."

With a large staff, BASIC have masses of talent and skill to draw upon. What's their advice to anyone with ambitions to work in this industry? "Advice? Create without fear," says Matt. "When we think of our agency, we see ourselves as a multidisciplinary group of creators, thinkers, and doers from around the world," Matt continues. "Hustle comes from within and we have a shared passion for building brands. We aren't big on hierarchy and we all contribute to our future.

"We want everyone here to grow as a professional, as well as an individual. Respect each other. Share and learn from one another. When it comes to the people we hire, we look for those with different perspectives and a deep passion for the craft, clients, and culture. We believe that when people bond through a shared passion and a relentless pursuit of better, great things come."

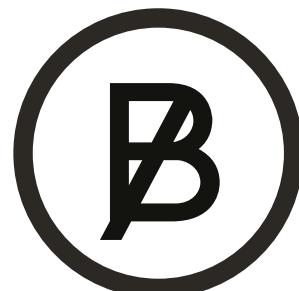
As the digital spaces we all inhabit continue to morph and evolve, BASIC will continue to craft new experiences for their clients'

customers. With an eye on the future, Matt concludes: "In the last year, we have doubled in size and revenue with new clients including Snapchat, L'Oréal, Heath Ceramics, and Under Armour. Along with this, we've hired some of the best thinkers in the industry like Steve Denekas, formerly VP of Creative at Instrument; Cristina Torres, who ran the Apple Program at AKQA, and Michael Madavi, a senior strategist at Apple.

"Our future is bright, and we're excited to continue our progression. We plan on continuing to double down on our core competencies, as well as bringing on world-class talent. What's fun is seeing where our team and our passions take us. We've always been a group that has adapted with the times and the interests of our people, and we envision that remaining the same."

There is no doubt that BASIC have their fingers on the pulse of today's digital design, but it's more than just pixels. The experiences they craft and the journeys they take clients' customers on is more important than using the latest tools. BASIC do have a rich environment to work within, and craft landscapes that are engaging and have an emotional resonance that few agencies can match. □

"When it comes to the people we hire, we look for those with different perspectives"



basicagency.com

Founders

Matt Faulk & Erich Broesel

Year Founded
2010

Current Employees
70

Location
San Diego, CA, Mountain View, CA, & St. Louis, MO

Services
Brand Strategy, Identity and Communications
Digital Experience and Innovation
Content and Campaign Strategy
Video Production and Photography



**Subscribe
today & save**

Go to page 32
to learn more

JAVASCRIPT WHAT'S NEW?

DISCOVER THE LATEST UPDATES
TO THE WEB'S FAVOURITE TOOLS
AND FRAMEWORKS. GET AHEAD
OF THE GAME AND DISCOVER
SOME NEW TIME-SAVING TRICKS

What's new for JS?



ES2018 is all about making complex tasks easier for everyone. This can only highlight JavaScript and the web as a serious platform for app development.

Matt Crouch
Software Engineer at Vidsy
@mattcrouchuk
www.mattcrouch.net

For the longest time, JavaScript stood still. Thinking back just ten years ago, Chrome didn't exist yet and nearly two-thirds of users were using Internet Explorer 6 or 7. The web was stuck in a mix of aged browsers and painfully fragmented standards.

Between then and now, users have begun asking more from their browsers. Over time, browsers have become more capable and the appetite for more complex apps grew all around.

That meant JavaScript had to grow with it. It wasn't until a new set of standards came along with ES2015 that it could meet the demands the developers had of it.

A new release has happened every year since then; each building upon the breakthroughs of ES2015. It's important to have a good grounding of knowledge in those features before checking out the newer ones.

One of ES2015's key new features was the game-changing 'Promise' object. Promises can be used to deal with data that isn't immediately available, such as a file from a server.

```
const myPromise = fetch('data.json'); // Returns a Promise object
myPromise.then(data => console.log(data))
```

The browser is better equipped than ever before to take advantage of the latest advancements

Instead of returning data straight away, asynchronous actions can return a 'Promise' object in its place. By using the 'then' method on that promise, we can act on the eventual data returned.

While the later rounds of standards updates have not been as feature-rich as ES2015, they still hold useful changes. ES2016 only provided two small features:

```
const values = [1, 2, 3];
values.includes(1); // true
values[1] ** values[2] // 8
```

The 'includes' method is a simple check to see if a value exists within an array. In most cases it can be used to replace 'indexOf'.

Exponential math operations can be performed using the '**' operator. With two values, this works the same as the existing 'Math.pow(x, y)', but the updated syntax makes longer equations easier to understand.

ES2017 had 'async' functions, a way to work with promises.

```
async function fetchUserPosts() {
  const users = await getUsers();
  const posts = await getPosts(users);
  return posts;
}
```

The 'async' keyword tells a function to run asynchronously, which can avoid slowing down the rest of the site. Inside, the "await" keyword waits for the value from

any other promise before continuing - similar to the 'then' method. The end result is a function that returns a promise, that can be used as any other promise would.

```
const obj = {
  username: "Matt",
  active: true,
}
Object.values(obj) // ["Matt", true]
Object.entries(obj) // [["username", "Matt"], ["active", true]]
```

ES2017 also brought more object convenience methods. The 'values' method returns all the values for the object it's passed, with 'entries' doing a similar job returning an array of key/value pairs.

```
const val = "Web";
val.padStart(5, "*"); // "***Web"
val.padEnd(5, "*"); // "Web***"
```

A couple of convenience functions were also added for strings. The 'padStart' and 'padEnd' methods will pad a string to the desired length. By default, it's padded with a space character, but this can be anything that's passed through.

All of these features are available in the latest browsers. Older browsers, ie Internet Explorer, have polyfills, or compilers like Babel can transform code into a format that is backwards compatible.

How standards are made

The ES2015 specification added too many new features in one go. It was hard for browsers to ship its features in a timely manner. As a result, the process for adding to the standard changed. From now on, the feature would need to pass the TC39 process. But what is the TC39 process?

The TC39 committee is a group of browser vendors and prominent industry figures that meet every couple of months to discuss ECMAScript along with any proposals to change it. ECMAScript is a general-purpose language standard that JavaScript adheres to.

There are five stages of proposal ranging from 0 to 4, with various criteria they must meet before advancing to the next level.

Stage 0 - or 'strawman' - is the starting point for all proposals. There is no restriction on what can be submitted, but to get any further there must be a viable reason for its addition.

Stage 1 - 'proposal' - requires a strong enough case for the proposals inclusion alongside a working demo with a polyfill. At this point it would be championed by a member of TC39 and have its benefits discussed with the rest of the committee.

Stage 2 - 'draft' - means a proposal is gaining traction. At this point, a roughly complete specification for the feature needs to be provided. Edge cases and any lingering issues are discussed at this point.

Stage 3 - 'candidate' - is where a proposal is all but complete. It stays in this stage until it has acceptance from all those responsible for updating the standard. To move on from here, it needs to have at least two implementations in environments such as browsers or compilers as well as tests to make sure it complies to the standard.

Stage 4 - 'finished' - means the work has been completed and the proposal becomes part of the next year's standard. From here, implementations will be made in all major JavaScript parsers and renderers to keep up-to-date. Then you finally get to use the feature for yourself.

What's new in ES2018?

The updated specification brings lots of useful properties, methods and general improvements ready to use right in the browser

Feature name	Current browser	Compilers/polyfills										Desktop browsers										Mobile	
		Tracer	Babel 6.2+ core-js	Closure 2018.06	Type- Script 2.8+ es6-sh	ES2018 shim	IE 11	Edge 18	Firefox 52	FF 52 ES6	FF 60 ES6	FF 61	Ch. 66 OP 53	Ch. 67 OP 54	SE 11	SE 12	IE 11.1	IE 11.2					
2017 features																							
Object static methods	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
String padding	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Validating commas in function syntax	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Arrow functions	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Shared memory and atomics	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
2017 misc	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Proxy 'ownKeys' handler, duck-typing keys for iterable targets (ES2017 Semantics)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SameEnv 'v' flag, case folding arguments, reader removed	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
2017 annex B	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object iterator/getter/setter methods	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Proxy Interval-Call getter/setter methods assignments allowed in for-in head in non-strict mode	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
2018 features	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object rest/spread properties	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Promise prototype finally	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Abort API flag for regular expressions	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Set/Cap named promise errors	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
RegExp Lockshered Assertions	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
String template Properties, Escapes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Asynchronous Iterators	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
2018 misc	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

In January, the final list of features for ES2018 was confirmed. While this means they will form part of the latest standard, they will typically be implemented by browser vendors at different times.

Here are some of the highlights of this year's updates.

Spreading objects

ES2015 brought 'rest' parameters to JavaScript. By using the '...' notation on the last parameter of a function it can capture any further arguments. This can be useful for open-ended functions that can combine any number of passed values.

The same notation can be used to expand array values into other places with a technique called 'spreading'. It often gets used to copy or combine arrays together.

```
const original = { x: 4, y: 5 };
const copy = { ...original }
```

In the latest specification, the same can be applied to objects to make copying shallow objects easier. This comes in handy when striving for immutability in frameworks like Redux.

This feature is available in all the latest browsers apart from Edge.

Asynchronous iteration

Another feature introduced in ES2015 was the concept of an iterator. At a higher level, iterators are any object with a 'next' method

that returns both the next value in a sequence as well as a 'done' value to note when there are no more values to iterate.

```
for await (const data of
  getIterableData()) {
  console.log(data);
}
```

The new asynchronous iterators work in a similar way, but instead of returning an object with the value and a 'done' flag, it returns a promise that resolves with those values.

By combining this with the 'await' keyword from ES2017 it's now possible to loop over asynchronous iterable objects. This can be especially useful when dealing with large remote files, such as videos.

Right now, it's supported in the latest versions of Chrome, Firefox and desktop Safari.

Clean up promises

Promises that succeed can be chained by using the 'then' method. If any issues arise it jumps to the next use of the 'catch' method. To execute the same code in both cases, it needs to be duplicated in each block.

In ES2018, the 'finally' method provides a place to add such logic.

```
showSpinner();
fetch("data.json")
  .then(/*process json*/)
  .error(/*process error*/)
```

```
.finally(() => hideSpinner());
```

By placing logic that always has to run inside 'finally' it will run regardless of the state of the promise. This keeps the other blocks clean and well and truly focused on one job.

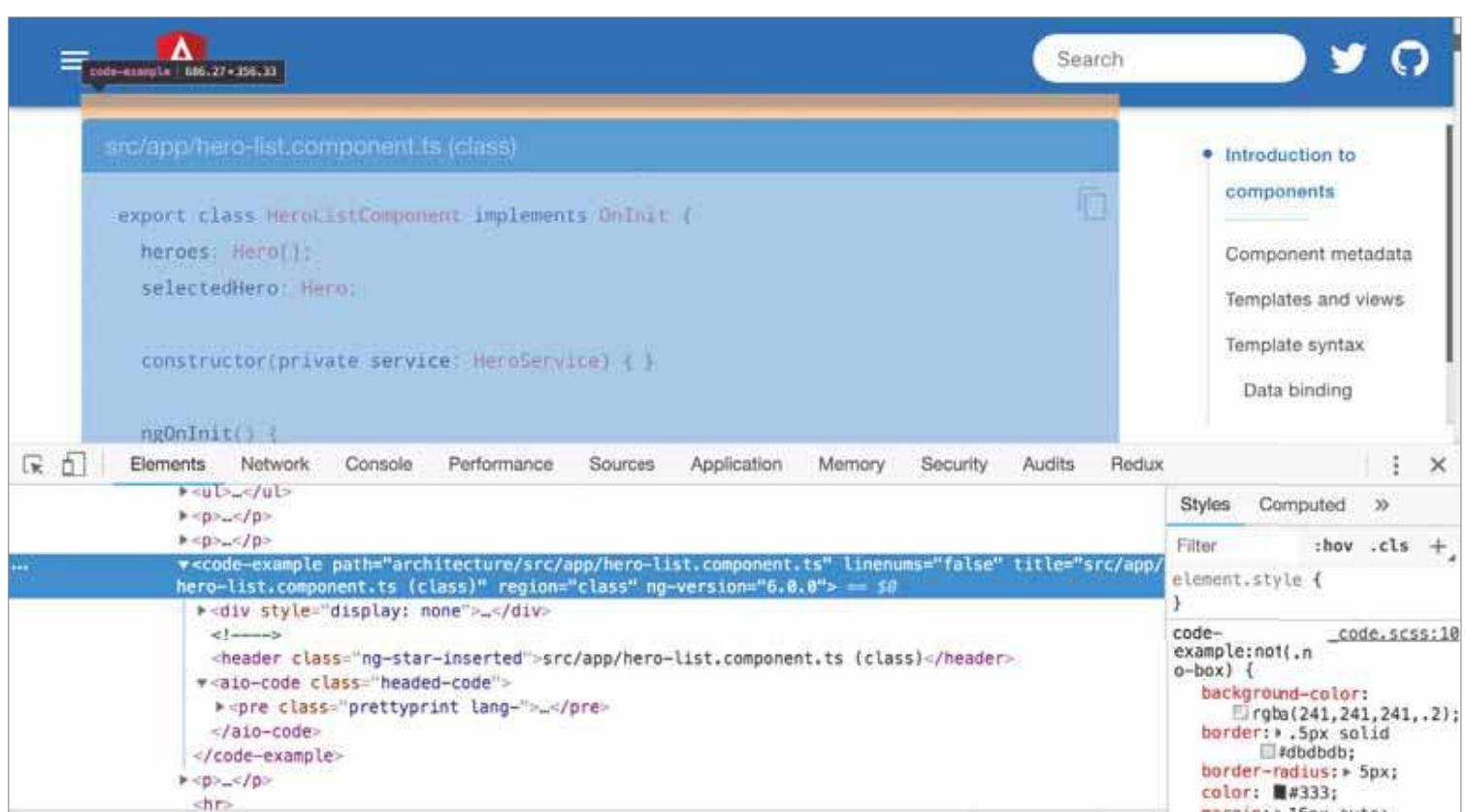
This feature has landed in the latest versions of all browsers except Edge, where it will be released soon.

Use Babel

Even before ES2015 was finalised, Babel has been a tool to help developers use the latest features. Babel compiles the new syntax down into more traditional language that more browsers can understand. Today Babel is still helping to convert ES2018 code to be suitable for users in all browsers depending on the plugins used. The most common plugin preset - 'babel-preset-env' - will transform the latest syntax to a version compatible with the browsers the project targets. Other plugin presets target features in various stages of the standardisation process. Find out more at <http://babeljs.io/docs/en/>.



The new asynchronous iterators work in a similar way, but instead of returning an object with the value and a 'done' flag, it returns a promise that resolves with those values



What's new in Angular?

The web's most popular framework focuses on tooling

Version 6 targets an improved developer experience over code changes. In fact, projects can be updated to the latest version with no issues thanks to the improvements in tooling.

Update in one go

To update an existing project to version 6, the updated Angular CLI can help.

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

When ran against a project, Angular checks the other dependencies. The package being updated is also aware of other updates that relate to it and will pull those in as well.

Once those packages have installed, the update command will then go through their code looking for hints called 'schematics' that can help the migration process.

For example, this command will install 'rxjs-compat' to make sure all RxJS code works with the latest version. If a third-party package includes the appropriate

schematics, they too can make use of this technology. Find out more about schematics and the way they work at <https://blog.angular.io/dc1dfbc2a2b2>.

Adding with schematics

Adding a package with schematics can also make use of this functionality.

```
ng add @angular/material
```

When 'ng add' runs, an installation script can then update config, install polyfills and even provide application-specific boilerplate code. The above example will not only install @angular/material, but also register starter components to use with 'ng generate'.

Angular Elements

It can be useful to have a component work free of the Angular life cycle. For example, a widget made for WordPress posts becomes much easier to implement if it can be dropped in like any other HTML element.

Angular Elements is a project that converts Angular components into web components. As of version 6 it's not quite ready to use just yet, but can be used wherever Angular is available. In cases where Angular serves dynamic content from a server it's definitely a perfect fit.

```
import { createCustomElement } from '@angular/elements';
[...]
const MyElement = createCustomElement(MyComponent,
{ injector: this.injector });
customElements.define('my-element', MyElement);
```

By creating a custom element during bootstrap, it becomes available to the browser to register and use as needed. Any HTML that uses that component can now mount successfully.

This is just the start for Angular Elements. Find out more about what's possible and their future at <angular.io/guide/elements>.

Coming soon - Ivy

Angular bundles are too big. Its internal structure makes it difficult for Webpack to filter out the parts an application it isn't using. With Ivy, that could soon be changing.

Ivy is the codename for the next-generation rendering engine. Instead of pre-calculating what needs to be bundled, Ivy takes things one file at a time and skips this step entirely. This enables faster incremental builds for development and smaller bundles for production.

It's currently being tested on Angular projects within Google to make sure all other applications can use it without any issue.

You can find out more about Ivy and how to try it out on a project today by heading to [is-angular-ivy-ready.firebaseioapp.com](firebaseapp.com)

What's new for Node?

Speed things up and bring the newest features server-side in version 10



With the introduction of ECMAScript modules in ES2015, Node has been trying to let the two work together despite their slight differences

The latest major update came in April this year and brought with it lots of small but very useful features to make the development of server applications easier and more secure.

The most prominent update is to do with modules. Node has been using CommonJS modules for a long time. With the introduction of ECMAScript modules in ES2015, Node has been trying to let the two work together despite their slight differences. They are now available when used with the '--experimental-modules' flag. An in-depth explainer for this by Gil Tayar can be found at <https://medium.com/@giltayar/ee5ea3001f71>.

Errors have also been significantly improved, with standardised codes being introduced where there were previously only string descriptions. Whilst before any errors would have to match against that generic string, version 10 introduces standardised error codes that return alongside the string description. A slight change to the description of an error will no longer break an application. A new set of experimental asynchronous file

system methods have also been added to the "fs" module. Instead of using callbacks, these use promises to communicate with the system in a more readable and efficient manner.

Updated V8
10.7 ships with an update to version 6.7 of Chrome's V8 engine. It brings with it lots of new JavaScript features including 'BigInts', 'async' generators and new 'string' methods along with a 30% speed increase.

Finally, version 10.7 also includes version 6.1 of npm. This release focuses mainly on performance and security improvements, including the new 'npm audit' command, which can check if any dependant packages have known security vulnerabilities.

```

JS index.js
1 const { promises } = require("fs");
2
3 async function fetchData() {
4   let data;
5   try {
6     data = await promises.readFile("data.json");
7     data = JSON.parse(data.toString());
8   } catch {
9     data = false;
10 }
11 return data;
12
13
14 console.log("Fetching Data...");
15 const data = fetchData().then(data => console.log(data));
16

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: bash

```

matts-mbp-2:node-fs-promises matt$ node index.js
Fetching Data...
(node:64782) ExperimentalWarning: The fs.promises API is experimental
{ success: true,
  user:
   { id: 1206,
     name: 'Matt Crouch',
     position: 'Software Engineer',
     skills: [ 'css', 'html', 'javascript' ] } }
matts-mbp-2:node-fs-promises matt$ █

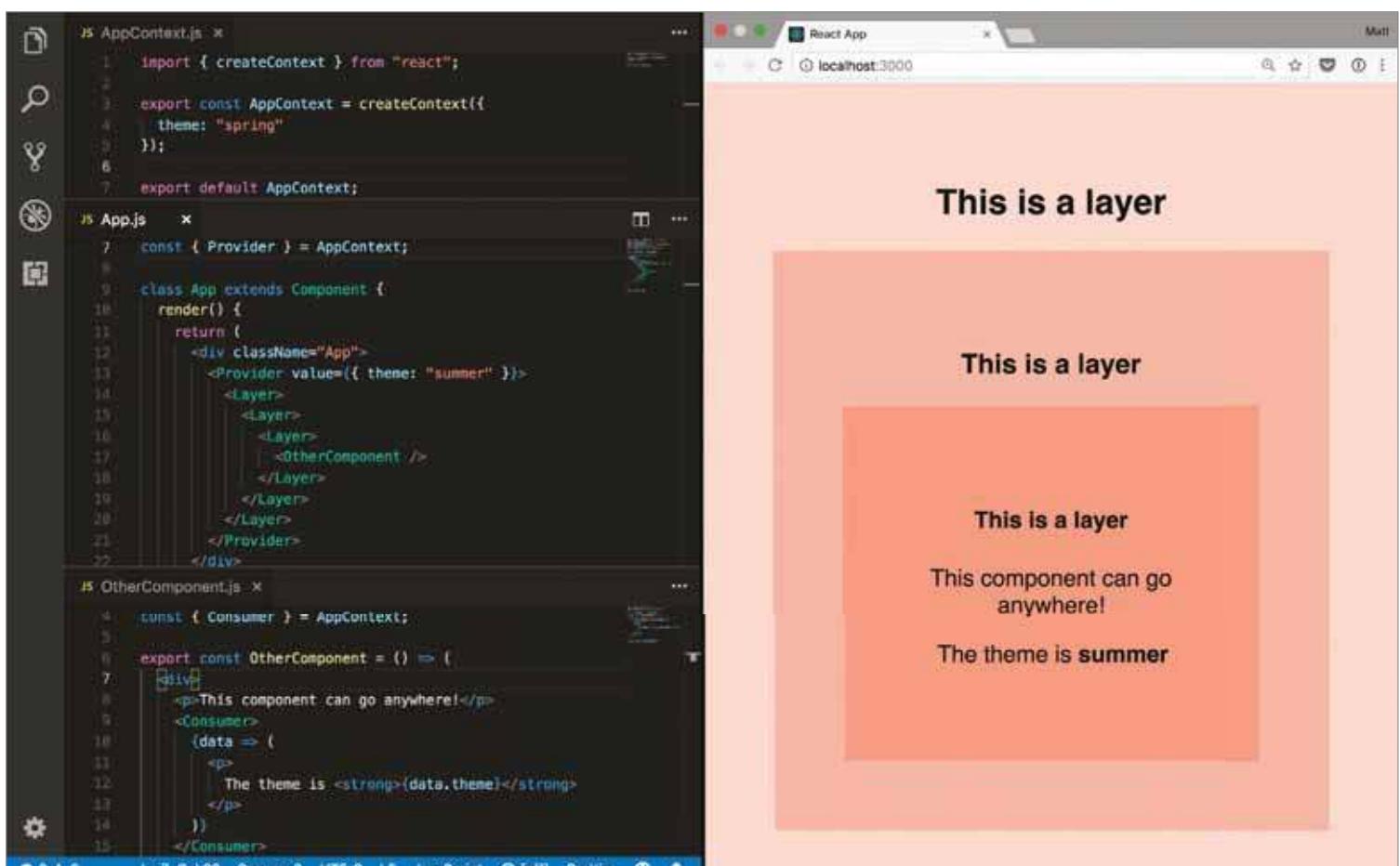
```

Import module
The new promise-based methods can be imported from the 'fs' module itself.

Async functions
Node 8 brought support for 'async' functions - a new, cleaner way to work with promises more linearly.

Optional catch binding
Node already supports catch blocks without binding an error parameter. This is currently part of the ES2019 standard, which is coming shortly.

Promise-based return
'Async' functions will always return a promise, regardless of the contents of the function. Be sure to resolve them at the first stage.



What's new in React?

Handle data with ease across an application with a revamped Context API and lots more

Component life cycle changes

React's life cycle methods have been around for a long time. Specifically, 'componentWillMount', 'componentWillReceiveProps', and 'componentWillUpdate' are often the easiest places to perform component-based tasks like adding events or updating state.

Future updates to the framework open up ways to asynchronously render components. This makes applications perform better and provide a more enjoyable experience for users.

Unfortunately, the work often performed in these life cycle methods makes this new type of rendering trickier. As a result, they are being deprecated and being replaced with more suitable methods - 'getDerivedStateFromProps' and

'getSnapshotBeforeUpdate' - which cover most of the same use-cases.

Updated context API

The context API allows information to be passed to deeply-nested child components without having to pass it through as props at every level. It's used heavily in frameworks like Redux and can help manage app-wide configurations such as theming and internationalisation.

In previous versions, the API was undocumented and its use was discouraged. In 16.3, it was revamped and made more efficient and easier to use.

```
export const AppContext = React.createContext({
  theme: "spring"
});
```

When creating a context, it returns a provider and consumer component. Anything inside the

provider receives the context values, while anything inside the component can receive it.

Easier refs

Refs are a great way to get a reference to a rendered DOM node. This can be useful when trying to accomplish something React itself isn't capable of, such as managing input focus.

They can be defined either using a plain string or a callback function to get a reference to the element within a component class. But React now includes a new way - the 'createRef' API.

```
this.inputRef = React.createRef();
<input ref={this.inputRef} />
```

For DOM nodes nested inside parents, the new 'forwardRef' API can also pass refs to the relevant child nodes in a similar manner.

Pointer events

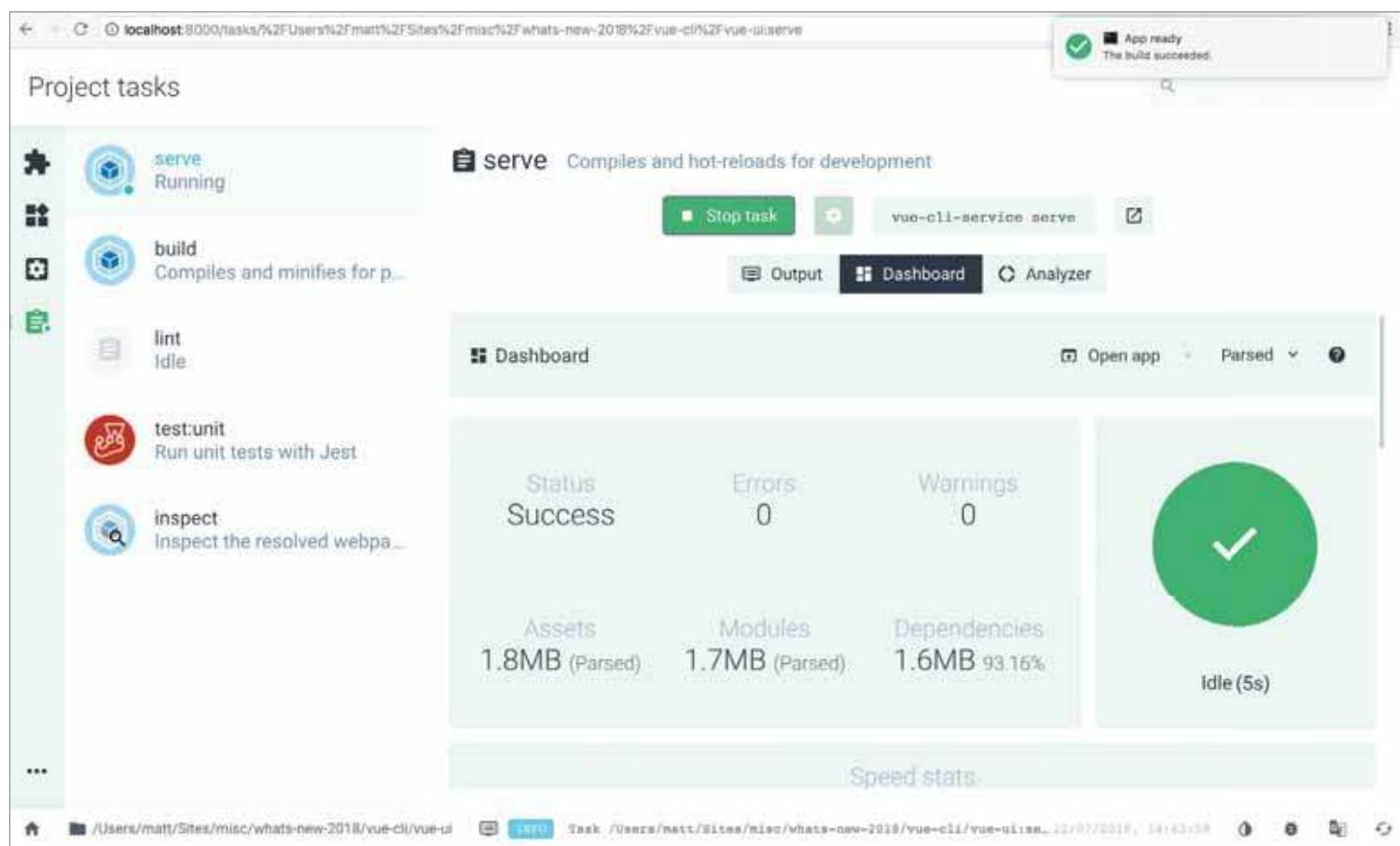
Applications that support both mouse and touchscreen input can be tricky to implement. At a basic level it can involve setting up two event listeners, but can get difficult when dealing with things like pressure values.

Pointer events were created to make developing that experience easier. Events such as 'pointerdown' and 'pointermove' capture all coordinate-based input types equally.

React DOM 16.4 supports pointer events as events directly on the component:

```
<div onPointerDown={this.addPointer} />
```

Browsers that do not support pointer events still need a polyfill to support this behaviour. However, any polyfill will work rather than one made specifically for React itself.



What's new in Vue?

New releases focus on tools and extras to make the development experience a breeze

The last sizeable update to the core Vue platform - 2.5 - was back in October last year. Being a minor release, it only brought with it slight changes to the platform.

Improved error handling

In previous versions, error handling was a problem whenever an unexpected error occurred. Developers could use the provided 'Vue.config.errorHandler' global method and apply it across the whole application, or 'renderError' inside a component. A more flexible solution was required:

```
export default {
  ...
  errorCaptured (err, vm, info) {
    this.error = true
  }
}
```

In 2.5 Vue introduces the 'errorCaptured' hook that can live inside a parent component. This hook captures any error that

occurs inside a child, much like React's error boundary concept.

The 'errorHandler' global method can still be used with this new approach. It makes an ideal place to report errors to third-party loggers to keep an eye on errors thrown by the application.

Version 3 coming soon

A larger update is already in the works. It will drop support for browsers that do not support ES2015 features, such as Internet Explorer 10.

While its release date has not yet been revealed, the development team have agreed to a six-month lead time between announcement and release to be sure all issues are ironed out.

Vue CLI update

In the meantime, emphasis has been placed to improve the tooling surrounding Vue. At the centre of

this is a major update to the command line interface (CLI) used to stamp out starter projects.

Previous versions used the CLI to clone template starter projects. While this made it easy to get started, any updates to those templates did not benefit projects already generated.

The new CLI instead uses plugins that work independently of each other. Features such as Vuex, TypeScript and ESLint can be added with the press of a button. When one of these plugins gets updated, it gets pulled into that project like any other dependency.

It hasn't yet been officially released, but stable versions are being provided to try it out. Find out more about Vue CLI 3 at [cli.vuejs.org](https://v2.vuejs.org).

A CLI UI?

Alongside the new CLI, a separate interface in the browser has been

developed to help build and maintain projects as they develop.

When building a new project, it works as a graphical interface for the CLI. It shows which plugins are available, what they do and why they may be useful.

Once a project is built, it can be maintained inside the new UI. Separate tabs show out-of-date dependencies, tasks that can be performed, and other plugins the project might benefit from.

By running the 'build' task, it can give instant feedback of the progress, the size of the bundles and even where improvements can be made. Developers can create interfaces for their own plugins also.

The idea behind the new interface is to help those not confident on the command line get all the benefits it provides, as well as presenting the data in a more informative way.

What next for JavaScript?

Take a look at what's in the pipeline for ES2019 and beyond

1

Optional catch binding

Up until now, try/catch blocks require the assignment of an exception identifier even if that value never gets used. This change allows that value to be omitted entirely.

```
try {
  JSON.parse(json);
  return true;
} catch {
  return false;}
```

While capturing and interpreting the error is usually a good idea, sometimes it isn't needed. In this example, it only checks if a string is valid JSON and no extra information is needed.

This change has passed all the stages of the TC39 process and will form part of the official ES2019 standard, although Chrome and Firefox have already implemented it.

2

Dynamic imports

ES2015 brought the 'import' statement, which allowed developers to compose scripts that lived in other modules. These are static declarations – they only allow fixed strings to define which modules to import in order for JavaScript to know upfront which files to fetch. Dynamic imports are a promise-led approach to importing modules dictated at runtime.

```
import("/module.js")
  .then(module => {
    module.default();
 });
```

These are currently at stage 3, but are available to play around with in Chrome for now.

3

Private class fields

As it stands, all properties in ES2015 classes are public. They can be read and modified by any code that accesses it. Fields that are supposed to be private typically start with an underscore, for example 'this._value = 'myPrivateValue''. This new proposal provides the hash symbol for a similar job, but one that's enforced through the language:

```
class Magazine {
  #internalId = "353010";
}
```

This is currently at stage 3, alongside a similar proposal for class methods also.

4

Decorators

The higher-order function pattern is used to provide additional behaviours to other functions. Decorators are a new, neater syntax to perform a similar job on classes, methods and fields.

```
@trackUsage(true)
Class Magazine {
  @deprecated
  fetchArticles() {...}}
```

In this example, a decorator called 'trackUsage' is being used to log how often the class is used, while 'deprecated' will warn the developer that this method should be avoided in future. As long as it returns a function, it can be a decorator.

This is currently a stage 2 proposal, but it can already be used in the likes of TypeScript and other compilers.

5

The 'global' keyword

JavaScript can be used in many situations – not just on webpages but in web workers and on servers as well. On a webpage, referring to the global object uses 'window', which isn't available in those other environments. There is no one approach that works in each case. The new 'global' keyword provides a single approach that simplifies code that needs access to it. It's currently at stage 3, awaiting a better name to avoid backward compatibility issues. Keep your eyes peeled.

The new 'global' keyword provides a single approach that simplifies code that needs access to it. It's currently at stage 3, awaiting a better name to avoid backward compatibility issues.

web workshop

Create a rollover text effect with SVG

Inspired by <http://sorrytobotheryou.movie>

SVG clip path

Using SVG's 'clipPath', the shape of the SVG is used to clip the image that is sitting behind to give a really stand out effect with the large typography.

Main menu

The menu is positioned at the top of the page and scrolls users down to the appropriate place on the page.



Create a rollover text effect with SVG

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

EXPERT ADVICE

Communicating the film through design

The comedy elements of the film are easily translated into design for this site. The bright use of vivid colour and the typography selected all communicate that this isn't to be taken too seriously. Add in the animated elements and the whole site demonstrates the quirky originality of the film.



<comment>

What our experts think of the site

Design and motion align

Taking the comedy values of the film and creating a vibrant colour scheme helps to make this website a very effective one-page site. The movement from section to section, which is not always straight down as it moves in diagonal directions, adds another interesting dimension to the overall user interface.

Mark Shufflebottom, Professor of interaction design

Technique

1. Adding the clipped image

Making the image have a clipping path involves using an SVG. Add your image and bare bones SVG to the body of your page as shown in the code. Later we'll add a real SVG shape inside the 'clipPath'.

```

<svg>
  <defs>
    <clipPath id="svgPath">
      </clipPath>
    </defs>
  </svg>
```

2. Making it work

In order to make the shape in the SVG clip the image, a little CSS is required, so just add this CSS to either a style tag or to a separate style sheet. This tells the image which clip path to use.

```
.svg-clipped {
  width: 640px;
  -webkit-clip-path: url(#svgPath);
  clip-path: url(#svgPath);}
```

3. Get the SVG

In Adobe Illustrator, grab your shape and choose File>Export>Export As>SVG. In the pop-up window, make sure your settings match those shown. Click Show Code and grab the path code and copy and paste it into the 'clipPath' of step 1. You can preview this in the browser.

4. Rollover clip

Just after the first clipPath in the SVG, add this second path element. Again, inside this element we will get the hover effect that we want. Just cut and paste from Illustrator's SVG export feature.

```
<clipPath id="svgHover">
</clipPath>
```

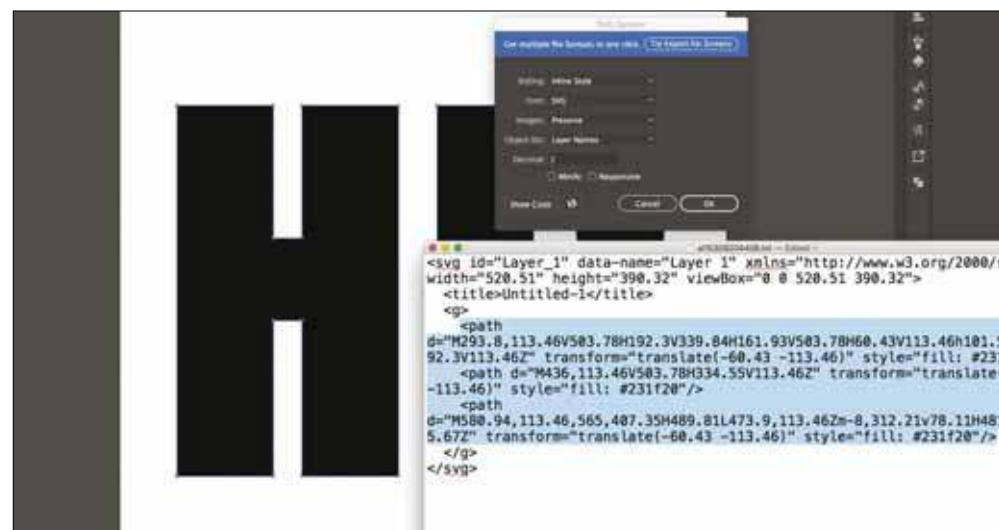
5. Make it a rollover

To make it a rollover, in the CSS add a hover pseudo class. Notice that on the hover we are targeting the new clip path that was added in the previous step.

```
.svg-clipped:hover {
  -webkit-clip-path: url(#svgHover);
  clip-path: url(#svgHover);
  width: 640px;}
```

6. Add the SVG rollover

Back in Illustrator, make the changes to your shapes and choose File>Export>Export As>SVG. In the window match the settings again. Click Show Code and grab the paths, copy and paste them into the clip path that was added in step 4. The rollover effect is complete.



Get started with Three.js - part 3

In this third tutorial, you'll learn about WebGL 3D using Three.js to add textures for realistic looks and cool visual effects



**Subscribe
today**

Go to page 32
to learn more



early universal browser and device support for WebGL makes it a perfect approach for real-time rendering. No plug-ins are required and you can start learning these technologies right away. WebGL 3D enables browser-based experiences, based on the powerful OpenGL language. It taps into the graphics pipeline, for highly optimised, interactive experiences.

Complex models with high levels of detail, reflections, environment maps and shadows can all be generated in real-time. You can give users access to beautiful visual experiences on their desktops or in the palm of their hand via a smartphone or tablet.

You'll be using the popular 3D library Three.js to dive into creating scenes and animating objects. It's free and open source, lightweight and boasts countless award-winning websites that have used it. Facebook 3D objects are also now powered by this 3D library.

Continuing from the last tutorial, you will move onto learning about materials and textures. You'll learn how to load external files to create realistic textures, apply lighting and environment maps for great results. 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 set up external CSS and JavaScript files or include 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>
  </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 here: <https://github.com/mrdoob/Three.js>. Note: The code in this tutorial has been tested on the latest release of Three.js v91.

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

3. Add basic CSS

In your style.css or between your style tags, set up a couple of basic styles. These are simple style rules to keep your canvas full screen, removing any margins or padding. We'll handle sizing of the renderer later on. In previous tutorials you also saw how to create a window resize handler.

```
html, body {
```

```
margin: 0;
padding: 0;
}
canvas {
  width: 100%;
  height: 100%
}
```

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 of some form. What is in that stage and in view of the camera is what the user will see. Add the following code to add a scene:

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

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 and far clipping plane distances, which define what is to be visible to the camera. Also push the camera back in Z space a little to see things easier.

```
var camera = new THREE.PerspectiveCamera( 75,
  window.innerWidth/window.innerHeight, 0.1,
  1000
);
camera.position.set(0,0,4.0);
```

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. 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 a renderer
var renderer = new THREE.
WebGLRenderer({antiali
as:true});
renderer.setSize( window.innerWidth, window.
innerHeight );
document.body.appendChild( renderer.domElement
);
```

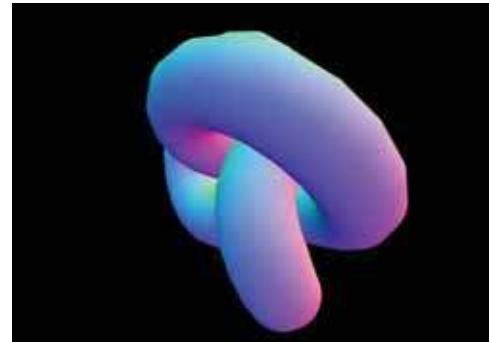
7. Create object geometry

Three.js includes a number of geometries. Try out a Torus Knot using the 'TorusKnotBufferGeometry'. You will swap out another geometry after to see how easy that is as well. Try this code first:

```
// create a 3D object
var geometry = new THREE.
TorusKnotBufferGeometry( 1,.4,60,60 );
```

8. Create a normal material

Next, you need a material. Yes, finally materials! That's



Initially you can use Normal Materials to see your object in 3D space without having to set up lights. With the basics set, you can add lights and shadows

why you've stuck it out this long! Three.js includes a range of materials, including physical shaders, lambert, phong and others. You can set textures to video or images as well. Use a 'MeshNormalMaterial' to start. It's a shader that takes in the geometry of the object and returns colour. It creates a nice colour range without needing an image texture or lights to see it.

```
var material = new THREE.MeshNormalMaterial();
```

9. Create a mesh and add it to the scene

To create an object 'mesh', you combine the geometry and material you just defined. 'Physical' objects in 3D require a geometry that defines the faces, vertices and drawing of the shape. They also require a material or skin to cover that object so we can see it. Create the mesh object and add it to the scene, like this:

```
var object = new THREE.Mesh( geometry,
material
);
scene.add( object );
```

10. Render the scene each requestAnimationFrame

Next, you call the renderer's 'render' function inside a loop bound to the 'requestAnimationFrame' function. To animate scenes smoothly you need to render at least 24 frames per second (ideally 60 fps), which this loop does optimally. You also add a little rotation animation to the object to see it move. Add this new code and run your scene after:

```
// render the scene
var animate = function () {
  requestAnimationFrame( animate );
  object.rotation.x += 0.01;
  object.rotation.y += 0.03;
  renderer.render( scene, camera );
```

Refresher: Defining a 3D object mesh

3D object meshes are much like physical objects. They are comprised of geometry and materials. Geometry defines the shape through vertices, faces etc, and 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 3

```
});  
animate();
```

11. Load a texture from an external file

Now that you can see your animated object using a Normal Material, you can explore various materials and how they work. First, you need to load an external material. This is done by using the TextureLoader class. For better code and load management, you can also listen for progress and completion as well on this class. In its simplest form use this code, replacing the filename with your own texture:

```
// load texture  
var texture = new THREE.TextureLoader().  
load("assets/rock_01_diffusion.jpg");
```

12. Applying a loaded texture to a material

Once loaded, a texture file can be used by a material. These are typically applied a type of 'map'. These include normal maps, bump maps, ambient occlusion and more. The main colour map of a material is simply called 'map'. It's what you would think of typically as an image texture. Replace your MeshNormalMaterial line of code with this one:

```
// load texture  
var material = new THREE.  
MeshBasicMaterial({map:texture});
```

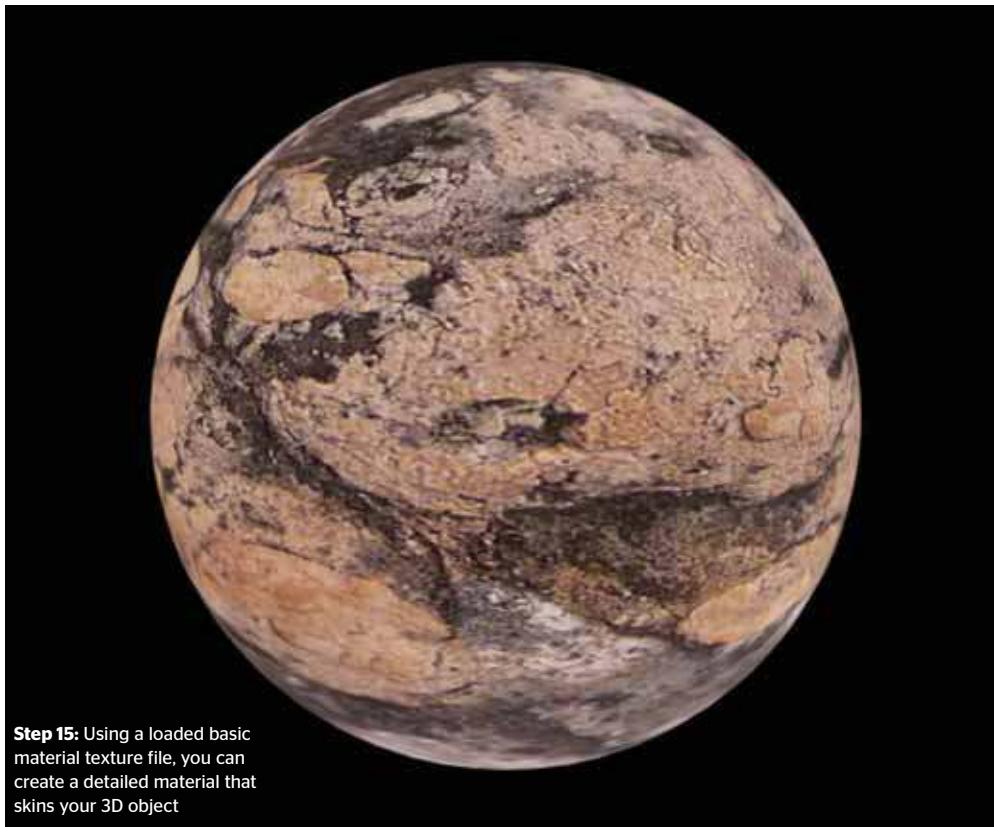
13. Setting texture properties

Textures have a number of useful properties. These include how a texture will wrap or repeat around an object, offsetting the start position, centring and more. To see how it works, try this code to adjust the wrapping of your texture. It repeats four times in the horizontal and one time in the vertical.

```
// load texture  
var texture = new THREE.TextureLoader().  
load("assets/rock_01_diffusion.jpg");  
texture.wrapS = THREE.RepeatWrapping;  
texture.wrapT = THREE.RepeatWrapping;  
texture.repeat.set( 4,1);
```

14. Add an ambient light

Basic and normal materials did not require lights, since they do not react to them. More realistic materials do react to light. Three.js lights include directional, spot lights, point lights, ambient lights and many others. Ambient lights will cast a general light on the entire scene, with no direction. Add one to your scene, setting a colour and an intensity like this:



Step 15: Using a loaded basic material texture file, you can create a detailed material that skins your 3D object

```
// create an ambient light  
var light = new THREE.AmbientLight(  
0xffccff,.4  
);  
scene.add( light );
```

15. Add a directional light

Directional lights cast an even light from a specific direction without any falloff. These emulate light from a large source like the sun. They can also have colour and intensity to light your scene. Notice you can also set the light position. Add a light like this:

```
//Create a Directional Light  
var light = new THREE.DirectionalLight(  
0xffff99,1.9 );  
scene.add( light );  
light.position.set(-10,10,0);
```

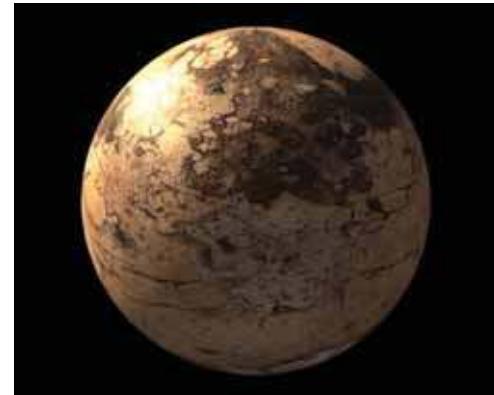


Step 16: Using a lambert material you can have lighting affect your material to give much more realism to your scene

16. Use lambert material

Lambert shader materials give you more options than basic materials and react to light. However, they are best for non-shiny surfaces, without specular highlighting. They're good for stone or untreated wood, for example. They are efficient due to their simplicity. Replace your basic material with this code:

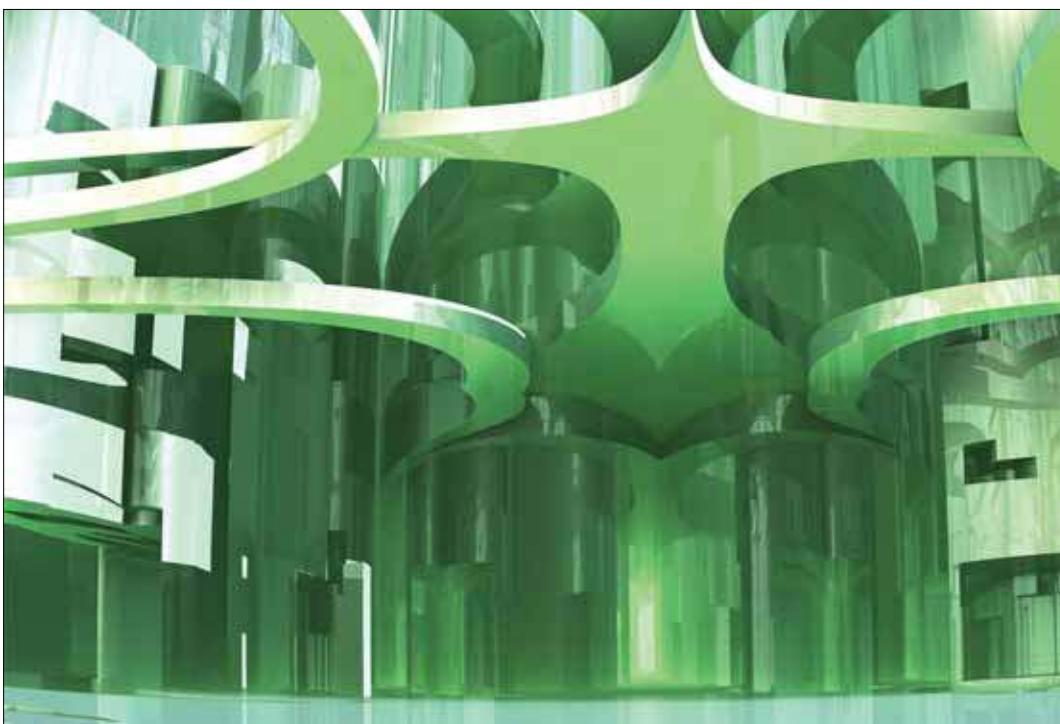
```
var material = new THREE.  
MeshLambertMaterial({map:texture});
```



Step 17: Using a phong shader is more accurate than lambert materials, allowing more detailed specular and shininess detail to surface

Lighting in 3D

Lighting is critical to creating a great-looking scene. Three.js includes a number of lights such as ambient, directional, spot lights and more. Combining these lights, their intensity, angle and colour will make your materials look more realistic and create a variety of moods and effects.



Materials for 3D

Three.js supports a variety of material types:

- MeshBasicMaterial: no lights required, great for baked-in lighting and backgrounds, low overhead.
- MeshLambertMaterial: low overhead, light effects it, but no specular or shininess, no detail on surfaces.
- MeshPhongMaterial: More reflective, shiny and supports specular colour and detail.
- MeshPhysicalMaterial: Most computation heavy, but realistic and detailed. Used in engines like Unreal Engine.
- ShaderMaterial: A custom shader for your textures. Build on existing ones or create your own.

Properties include maps (files that affect colour values in various ways) such as bump maps, normal maps, specular maps and environment maps. You can also adjust metalness, shines, reflectivity, opacity and many other attributes.

best for shiny surfaces, with specular highlighting. They are more computationally expensive than lambert, but still less so than physical shaded materials. Replace your material with this code:

```
var material = new THREE.MeshPhongMaterial({map:texture, specular: 0x333333, reflectivity:1.0});
```

18. Use a physical material

The most accurate shading material is the physical material. Its approach is used in the Unreal engine among others. It requires more computation per frame, but gives great realistic results. Swap out your 'TorusKnot' for a 'Sphere' and remove the wrapping code we applied. Try out this code to replace your previous entry:

```
var geometry = new THREE.SphereBufferGeometry(2,60,60);
var material = new THREE.MeshPhysicalMaterial({ map:texture });
var object = new THREE.Mesh(geometry, material);
scene.add( object );
```

19. Load an environment map

Physical materials work best with environment maps applied. These maps are sky boxes that surround the object so it 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 here: <http://www.humus.name/index>.

php?page=Textures Try adding this code to load a map.

```
var envMap = new THREE.CubeTextureLoader()
.setPath( 'assets/' )
.load( [ 'posx.jpg', 'negx.jpg', 'posy.jpg',
```

```
'negy.jpg', 'posz.jpg', 'negz.jpg' ] );
```

20. Apply environment map and more settings

Envmaps impact the colour and intensity of the colour on the surface texture, from all around it. How much can be set by their roughness, reflectivity and metalness. Update your material line to this one:

```
var material = new THREE.MeshPhysicalMaterial(
{ map:texture, envMap:envMap,
metalness:1.0,roughness:0.2 });
```

21. Apply environment map to scene

To make the scene really come alive, you can also apply the environment map as a cube map to the scene. Keep experimenting with these materials, combinations and settings to see what else you can accomplish. Bump maps and normal maps will give great detail to your textures as well! Finally, try adding this code after declaring your scene:

```
// apply envMap to scene
scene.background = envMap;
```



Step 21: Adding an environment cube map to your scene, combined with physical materials, can create realistic results

Create web patterns with CSS Doodle

In this tutorial, you will learn how to create interesting website patterns using the CSS Doodle web component





Patterns are everywhere. Chances are that you have stumbled upon them either in fashion, interior design, food packaging or even in nature. In addition, patterns have also been adopted in frontend web design. While they are applied in different sections of websites, they have been a popular choice for web backgrounds given the unique appeal they create. However, in the past, patterns could only be added to websites using images. The downside of relying on image-based patterns is that file sizes increase with dimensions thereby lowering their suitability in responsive and mobile-first designs. Furthermore, where frontend designers lack illustration skills, updating the website can take a significantly long period of time.

With CSS Doodle, the two challenges can be solved easily by simply relying on CSS to create complex yet appealing patterns. Its usage is preferred to reliance on JPEG-based images due to its lightweight nature. The web component generates a grid of 'divs' by the rules (plain CSS) inside it. As such, you can easily manipulate those cells using CSS to come up with a graphic pattern or an animated graph. The only limit is the limit of CSS itself. In addition, backgrounds can be changed easily with only a few lines of code, and interactivity enhanced by adding JavaScript.

1. Getting started

Start off by creating a folder named 'css_doodle' on your desktop to store the tutorial files. Then create two additional folders within it: 'css' to store the styling files, and 'js' to store the JavaScript files. HTML files will be stored in the root folder.

2. Create the HTML page structure

Open your code editor and create an 'index.html' document to contain markup for the main web page. Begin by creating the basic structure and give a suitable title to the page.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>CSS Doodle </title>
  </head>
  <body> </body>
</html>
```

3. Add content to the HTML structure

Since the focus is creating a pattern in the body of the page, we will add the CSS code in this section. Begin by adding the HTML element 'css-doodle', which will contain all the CSS code that dictates the nature of the patterns created. The grid size is proportional to the size of patterns created, as such, higher values translate to more grids while the inverse is true. A grid size of '8' is adequate for the tutorial.

```
<body>
  < css-doodle grid ="8">
</css-doodle>
</body>
```



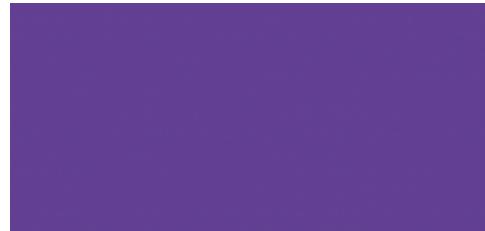
4. Style the web page – linking the CSS file

Link the CSS stylesheet to the HTML document. The styles are useful for creating both the body and pattern styles. 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" >
```

```
<body>
  < css-doodle grid ="8">
  :doodle {
    size: 100vmax;
    background: #673AB7;
    overflow: hidden;
  }
</css-doodle>
</body>
```

The page should render as follows:



5. Style the web page – style the body

Begin by setting the margin to '0' for all elements in CSS. This ensures the design remains consistent throughout. Next, add an overflow with the hidden parameter. This cuts off any content that breaks out of its bounds. Further, set the page height to '100vh'.

```
*<
  margin:0;
  overflow: hidden;
  height: 100vh;
}
```

6. Linking the CSS Doodle script

In order to utilise the component, it is important to include it in the body section. To do this, either download it from Github and manually link it to the HTML file. Alternatively, add it from CDNJS by the following code. At this point, also link the JavaScript file:

```
<body>
  < css-doodle grid ="8">
</css-doodle>
<script src='https://s3-us-west-2.amazonaws.com/s.cdpn.io/52982/css-doodle.min.js'>
</script>
<script src="js/index.js"></script>
</body>
```

Since we are yet to add any patterns in the css-doodle element, your page should be blank

7. Setting up the pattern

To start setting up the pattern, first specify its viewport dimensions using VMAX. Next, add a purple background using 'background' and specify hidden overflow to cut off content that breaks out of its bound.

8. Adding pattern shapes

The 'clip-path' CSS property creates a clipping region that defines what part of an element is to be displayed. On the other hand, '@shape' creates a shape pattern on the canvas such as a rectangle, rhombus, circle, clover, bud, etc. Thus, the two properties are used in creating a five-sided clover using the code below:

```
:doodle {
  clip-path: @shape
  (clover, 5);
}
```

Though the shape has been added, it is currently not visible as we are yet to specify its colour properties. Your page should remain purple.

So much you can do with css-doodle

The resource, css-doodle.com provides numerous examples of patterns created using the web component. Constantly refresh the page to observe diverse patterns and the code used in implementing them. Simply review the code to gain insight on creating patterns.

Tutorials

Create web patterns with CSS Doodle

9. Adding the pattern colour

Next, change the colour of the pattern to a light shade using the HSLA property. Hue, Saturation, Lightness Alpha (HSLA) defines colour based on these four properties. Hue defines the degree of colour (0 to 360) with 0 and 360 as red. Saturation defines its saturation (0% shade of grey, 100% full colour) while Lightness defines the lightness (0% black, 100% white). Further, Alpha defines a colour's opacity (0.0 fully transparent, 1.0 fully opaque).

```
//Copy the code below for the color//  
background: hsla (  
    calc  
(300 + 3 * @index()), 75%, 70%, @rand(.8)  
);
```

'@calc' is a CSS Doodle function that evaluates calculations while '@index' exposes the index of each grid cell to the specified colour. '@rand' is a CSS random generator function that randomises the opacity of the shapes used. As such, the background colour results from the addition of '300' to the product of each index by '3' (simply computes a unique saturation colour). Render the page.



10. Adding transition

The 'transition' property enables property elements to change value over a specified period of time, animating the property changes rather than having them occur immediately. In this particular case, specify a 0.2-second ease to occur randomly.

```
transition: .2s ease @rand(300ms);
```

Refresh the page and observe the transition change in the occurrence of the pattern shapes.

SETTING UP THE PATTERN

11. Scale them to different sizes

The current pattern shapes are all sized equally. However, to add a unique appeal, transform them into diverse sizes by using the 'transform' property. Begin by specifying the different sizes of the patterns using the 'scale()' function.

```
transform: scale (@rand (.2, 1.5))
```

12. Translate its position

Next, translate the pattern using the 'translate3d()' function, which positions elements in 3D space. Note: only the 'x' and 'y' positions are of interest in this tutorial. These are randomly generated by the '@rand' function thereby creating random positions on the canvas.

```
translate3d (  
    @rand (-50%, 50%),  
    @rand (-50%, 50%),  
    0  
);
```

Render your page. Then refresh it a few times to observe the simultaneous change in position, colour and size of the pattern.

13. Adding animation

Next, let's define how the pattern should animate. Though having different-sized patterns creates a visual appeal, enhance it further by adding animation to each individual grid cell. Add WebKit animation '10s', and at an opacity of '0.2', to each cell class in the stylesheet.

```
.cell {  
    -webkit-animation: rotate 10s  
    infinite;  
    animation: rotate 10s  
    infinite;  
    opacity: 0.2;  
}
```

Further, specify the rotation of the keyframes in the stylesheet. Add the code below:

```
.cell {  
    -webkit-keyframes rotate {  
        0% {  
            -webkit-transform: rotate(0deg);  
            transform: rotate(0deg)  
        }  
        100% {  
            -webkit-transform: rotate(360deg);  
            transform: rotate(360deg);  
        }  
    }  
}
```

```
transform: rotate(360deg)  
}  
}  
@keyframes rotate {  
0% {  
    -webkit-transform: rotate(0deg);  
    transform: rotate(0deg)  
}  
100% {  
    -webkit-transform: rotate(360deg);  
    transform: rotate(360deg)  
}  
}
```

Now render the page.

INTERACTING WITH THE COMPLETE PATTERN

14. Adding JavaScript interactivity

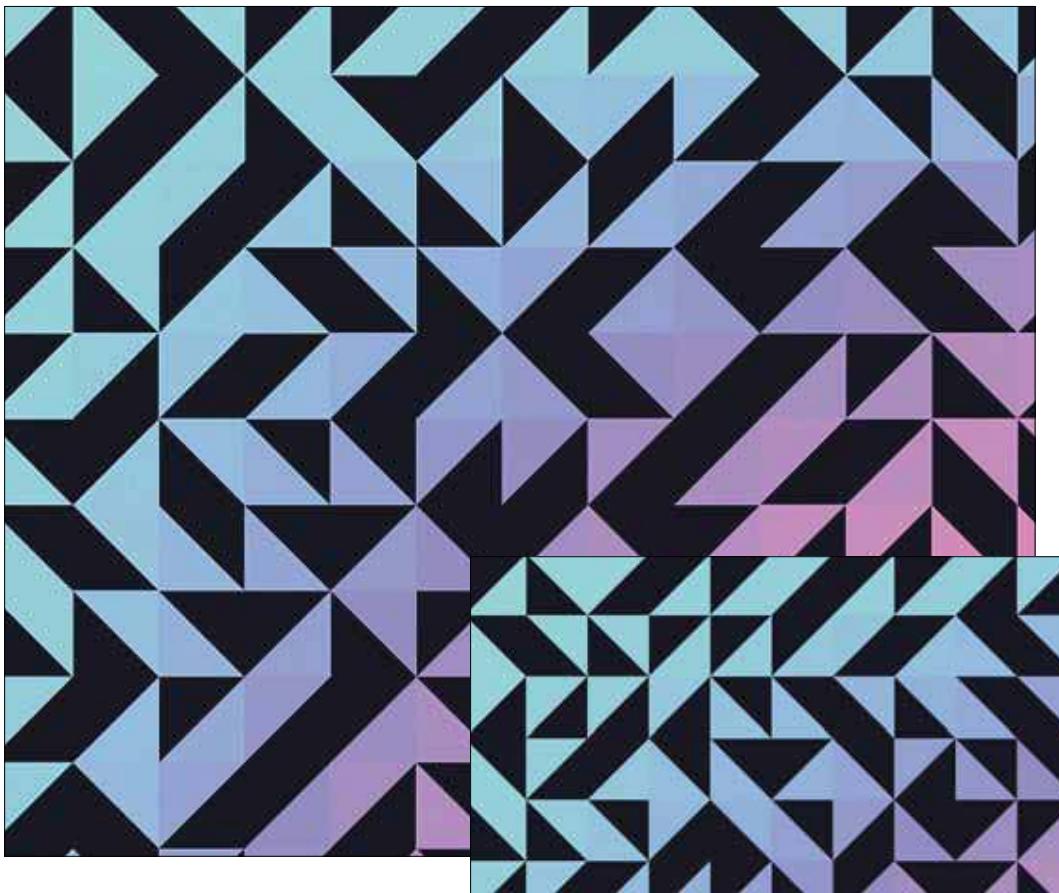
So far, we are yet to add content to the JavaScript file created earlier. This is because, JavaScript functionality in the tutorial is only for extra enhancements, as opposed to creating the actual patterns. Nonetheless, add a JavaScript function that refreshes the pattern each time the page is loaded. Add the code below:

```
document.body.addEventListener ('mousemove',  
function(e) {  
    if (e.target.matches('css-doodle')) {  
        e.target.refresh ();  
    }  
})
```

The code refreshes the pattern once the page is reloaded. Try out and observe.

Learning from examples

The best way to learn how to interact with CSS Doodle is by interacting with as many examples as possible. That way, your creativity will be enhanced greatly and so will creation time. Head over to css-doodle.com where there are numerous patterns made using the web component.



Creating a simple pattern background

As an illustration of the simplicity of working with CSS Doodle, navigate to the website where a pattern is shown. With your current page structure, copy the following code from the website to the 'css-doodle' section:

```
<css-doodle>
:shape {
@grid: 18 / 100vmax;
background: #0a0c27;
}
--hue: calc (180 + 1.5 * @row () * @col ());
background: hsla (@hue, 50%, 70%);
margin: -.5px;
transition: @r(.5s) ease;
clip-path: polygon (@
pick (
'0 0, 100% 0, 100% 100%,
'0 0, 100% 0, 0 100%,
'0 0, 100% 100%, 0 100%,
'100% 0, 100% 100%, 0
100%');
));
</css-doodle>
```

Render the page, then observe how the unique shape pattern results from these few simple lines of code.

As an example, try changing the grid size to '10'. Observe the increased pattern sizes.

15. Changing shapes

Now that the pattern is complete and rendered onto the page, change some of its settings in order to understand how to improve it further. First, change the shape to a three-sided bud. Simply amend the code specified by the 'shape' property as below:

```
clip-path: @shape (bud, 3);
```

Results should appear as shown.



16. Changing grid size

Next, increase the grid size from the current '8' to, say, '13'. Observe the change that occurs in relation to the pattern. Making such changes can help you decide on the most appropriate grid size to use for your project.

```
grid="13"
```

Results should appear as shown.

17. Changing background color

Finally, experiment and change the background colour and that of the pattern. Ensure observable differences can be noted in both instances.



Create an interactive WebGL header

Make your site stand out from others by creating an interactive and animated site header in WebGL using the Three.js library



Looking at the kind of sites that have been winning 'site of the day' on judging panels such as Awwwards recently, you will have noticed a trend emerging. Where once a large image or slider gallery would sit at the top of the page, there now resides animated content that is interactive with the user and shows off a unique aspect of what the site has to offer.

This trend has been growing over the past couple of years and it's almost always the minimum requirement to win a 'site of the day'.

In this tutorial, an animated header will be created that will be rendered onto the page using WebGL with the very popular Three.js library. A webpage of content will also be displayed so that you can see how WebGL fits with a page design. Over the top will be the usual header and menu, which are found at the top of most pages; further down you will find column layouts. The idea here is that the animated content at the top is used to grab the user's attention, making them want to explore more. To be a real contender for a 'site of the day' award, the rest of the page should continue to have animations and interactions for the user, but that's beyond the scope of what we can present here. Instead you will create a visually stunning 3D scene that the user can interact with and will animate on the screen.

1. On your marks

Open the 'start' folder as the project in your code editor and open the 'index.html' file. The page has been created already (you can look at it in the browser) but it's missing the dynamic elements. Scroll down to the 'script' tag and add the links to the libraries and some extra helpers just above that 'script' tag.

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

2. Inside the script

There are a couple of event listeners in the script already, which will make the mouse reactive. Above that add in the variables as shown below. They are the global variables used throughout the project to make all of the elements work together.

```
var container;
var camera, scene, renderer, group,
cubeCamera,
texture, texture2, boxes, model;
var load = 0, sliders = [];
var mouseX = 0, mouseY = 0;
var windowHalfX = window.innerWidth / 2;
var windowHalfY = window.innerHeight / 2;
```

3. Loading elements

The first part to getting things working is to load in relevant textures and models that will be used in the project. First up is the reflection map for the triangles in the scene. The second image will be used in the background and placed onto a 'plane' so that it can move with the scene.

```
var textureLoader = new THREE.TextureLoader();
texture = textureLoader.load("assets/space.
```



Step 01: The page content will be set up using the Three.js library available from threejs.org. The site contains all of the documentation to get you started



Step 02: The initial page design has been put in place and the tutorial will focus on creating the interactive, animated header section

```
jpg",
function() {
    texture.mapping = THREE.EquirectangularReflectionMapping;
    loaded();
});
texture2 = textureLoader.load("assets/space.png",
function() {
    loaded();
});
```

4. Load a model

The model that is being loaded in is actually a 3D model of text so that a texture can be applied to the front of the text geometry. If this was an image, then another texture would not have been able to be applied to it, as that alpha mask is going to be animated when finished.

```
var loader = new THREE.ColladaLoader();
loader.options.convertUpAxis = true;
loader.load('assets/text.dae',
function(collada) {
    model = collada.scene;
    model.scale.x = model.scale.y = model.scale.z = 0.5;
    model.updateMatrix();
    loaded();
});
```

5. Checking the load

The last image texture is loaded and you will notice that after each load the 'loaded' function is called. Here you can see that All it is doing is increasing the number of elements that have been loaded. If those elements are equal to four then everything has loaded so the 'init' and 'animate' functions are called.

```
var alphaMap = textureLoader.load('assets/clouds.jpg', function() {
    loaded();
});
function loaded() {
    load++;
    if (load >= 4) {
        init();
        animate();
    }
}
```

6. Setting the scene

The 'init' function is just short for 'initialise'. This adds all the elements into the scene so that it looks correct when displaying on the screen. Here the scene is set up with the right background colour and then a camera is added to the scene to view the content that will be added.

```
function init() {
    scene = new THREE.Scene();
    scene.background = new THREE.Color(0x687576);
    camera = new THREE.PerspectiveCamera(50, window.innerWidth / window.innerHeight, 1, 3000);
    camera.position.set(0, 0, 500);
    scene.add(camera);
```

7. Lighting up

In order to see the scene a light needs to be added and this is made from a child of the camera so that there is always a light pointing at the content from the position of the camera. Geometry is created with a yellow material; these will be animated lines moving diagonally when finished.

```
var light = new THREE.PointLight(0xffffff, 0.8);
camera.add(light);
var geometry = new THREE.PlaneGeometry(300, 5, 1, 1);
var material = new THREE.MeshBasicMaterial({
    color: 0xffffc0, });
```

8. Yellow planes

From the material and geometry in the previous step, two planes are created and placed on the screen at different positions. These lines will not stay on the screen in these places, but will be animated moving diagonally in line with their angle.

```
var plane = new THREE.Mesh(geometry, material);
```

Making sure content is loaded

The 'init' function is only called once all the assets are loaded. This ensures that no errors occur in the code, and so that the user is presented with the whole scene at once.

Tutorials

Create an interactive WebGL header

```
plane.position.set(180, 200, 80);
plane.rotation.z = -2.2;
sliders.push(plane);
scene.add(plane);
plane = new THREE.Mesh(geometry, material);
plane.position.set(580, 600, 80);
plane.rotation.z = -2.2;
sliders.push(plane);
scene.add(plane);
```

9. Adding the background

The background is created now by making a new plane. This is quite large as it's going to be placed behind all other elements. The material for this is created here. It uses the image of 'texture2', which is a transparent PNG of stars against a painted background.

```
geometry = new THREE.PlaneGeometry(1600, 1000,
1, 1);
material = new THREE.MeshBasicMaterial({
map: texture2,
opacity: 0.8,
transparent: true
});
```

10. Building the scene

The geometry and material are placed into the mesh so that it creates the plane. This is placed backwards on the z axis so it sits behind all other elements. Then a new group is created. This will have all of the triangular shapes added so that they can be positioned all together within the scene more easily.

```
plane = new THREE.Mesh(geometry, material);
plane.position.z = -500;
scene.add(plane);
group = new THREE.Group();
scene.add(group);
```

11. Creating the triangle shape

There are several models that are made using the triangular shape here. These are created by making the outer triangle and the inner triangle, which is the hole to make it a compound path. These co-ordinates for the points have just been lifted from the positions of the points in Adobe Illustrator.

```
var TRIShape = new THREE.Shape();
TRIShape.moveTo(387.6, 502.2);
TRIShape.lineTo(0.0, 0.0);
TRIShape.lineTo(775.2, 0.0);
TRIShape.lineTo(387.6, 502.2);
var TRIPath = new THREE.Path();
TRIPath.moveTo(115.5, 56.8);
TRIPath.lineTo(387.6, 409.3);
TRIPath.lineTo(659.7, 56.8);
TRIPath.lineTo(115.5, 56.8);
```

Using 'for' loops

Computers are very good at repetitive processes, so it's just as easy to control 300 boxes as it is to control one. The key to this is using a 'for' loop to update all copies at the same time.

12. Extruding the shape

The settings here are for the extruded triangle shape. It sets various properties that will give it the right look, such as how deep the extrusion will be, the size and depth of the bevelled front. These will be applied in the next step to create two triangles on the screen.

```
TRIShape.holes.push(TRIPath);
var extrudeSettings = {
depth: 8,
bevelEnabled: true,
bevelSegments: 2,
steps: 2,
bevelSize: 1,
bevelThickness: 1
};
```



Step 10: The background image is loaded and placed onto a flat plane. As this is a transparent PNG the image appears with transparent edges on the plane



13. Interlocking triangles

Here, two triangles are created and positioned on the screen. They are created with a cut out hole in the centre and slightly rotated so that they interlock with each other to make it look more interesting. One is placed slightly higher on the screen so that they are both clearly visible.

```
addShape(TRIShape, extrudeSettings, 0x3e2b43,
300, 240, -50, 0, 0.1, Math.PI, 0.8);
for (var i = 0; i < TRIShape.holes.length; i +=
1) {
    addLineShape(TRIShape.holes[i], 0x3e2b43,
300, 240, -50, 0, 0.1, Math.PI, 0.8);
}
addShape(TRIShape, extrudeSettings, 0x624562,
300, 110, 0, 0, -0.1, Math.PI, 0.8);
for (var i = 0; i < TRIShape.holes.length; i +=
1) {
    addLineShape(TRIShape.holes[i], 0x624562,
300, 110, 0, 0, -0.1, Math.PI, 0.8);
}
```

14. Box particles

After the triangles have been added some boxes will be created on the screen. These will act like particles and spin outwards while fading out before respawning in the centre of the screen. A group is created to put them in and then the basic geometry is created. A 'for' loop makes 300 boxes each with their own unique material, containing a random opacity.

```
boxes = new THREE.Group();
scene.add(boxes);
geometry = new THREE.BoxBufferGeometry(25, 25,
25);
for (var i = 0; i < 300; i++) {
    var object = new THREE.Mesh(geometry, new
THREE.MeshBasicMaterial({
color: 0x000000,
transparent: true,
opacity: Math.random()
}));
```

15. Adding randomness

The boxes are to move around randomly so they are positioned at random locations, then given random rotation and velocity so that they appear unique in their

Step 13: The triangle shapes have been created. There are several shapes for each triangle, and they are interlocked in their position and placed in the scene

movement. Because of the different axis to randomise, the code appears quite long, but it is very simple.

```
object.position.x = Math.random() * 300 - 150;
object.position.y = Math.random() * 300 - 150;
object.position.z = -250;
object.directionX = Math.random() * 2 - 1;
object.directionY = Math.random() * 2 - 1;
object.rotation.x = Math.random() * 2 * Math.PI;
object.rotation.y = Math.random() * 2 * Math.PI;
object.rotation.z = Math.random() * 2 * Math.PI;
```

16. Random scaling

The final part of the randomising is to give each box a slightly different scale value so that they appear to be different. Once this is completed each box is added to the box group. The text model is added to the screen and its material is stored in a variable to manipulate its alpha mapping.

```
object.scale.x = Math.random() * 2 + 0.5;
object.scale.y = Math.random() * 2 + 0.5;
object.scale.z = Math.random() * 2 + 0.5;
boxes.add(object); }
scene.add(model);
model.position.set(0, 0, 100);
var material = model.children[0].children[0].material;
material.alphaMap = alphaMap;
material.alphaMap.magFilter = THREE.NearestFilter
```

17. Text alpha map

Because the text is geometry the alpha map is repeated so that it can be animated later. The movement of the



Step 17: The text is placed in the scene and the alpha map is added to give a soft edge to the text and allow content behind to be seen through.

alpha map across the text geometry will give it a wispy semi-transparent look instead of the hard edges of the geometry. The renderer is set up and added into the document; this appears as a HTML5 Canvas element on the page.

```
material.alphaMap.wrapS = THREE.RepeatWrapping;
material.alphaMap.wrapT = THREE.RepeatWrapping;
material.alphaMap.repeat.x = 1;
material.alphaMap.repeat.y = 1;
material.transparent = true;
material.alphaTest = 0.1;
renderer = new THREE.WebGLRenderer({
  antialias: true });
renderer.setPixelRatio(window.devicePixelRatio);
renderer.setSize(window.innerWidth, window.innerHeight);
document.body.appendChild(renderer.domElement);
```

18. Finishing the 'init' function

The last parts of the 'init' function are added, which registers all of the event listeners to register touch or mouse movement on the screen so that the graphics on screen can react according to the user input. A final listener checks if the screen is resized. The 'animate' function calls the screen to render.

```
document.addEventListener('mousemove',
```

```
onDocumentMouseMove, false);
document.addEventListener('touchstart',
onDocumentTouchStart, false);
document.addEventListener('touchmove',
onDocumentTouchMove, false);
window.addEventListener('resize',
onWindowResize, false); }
function animate() {
  requestAnimationFrame(animate);
  render(); }
```

19. Rendering each frame

The animation occurs by updating the screen each frame. Here the render function takes the position of the mouse or touch and moves the camera around while looking at the centre of the screen. Then a 'for' loop updates the 300 boxes on the screen by moving their position and reducing the opacity each frame.

```
function render() {
  camera.position.x += ((mouseX / 4) - camera.position.x) * 0.05;
  camera.position.y += ((mouseY / 4) - camera.position.y) * 0.05;
  camera.lookAt(scene.position);
  for (var i = 0; i < 300; i++) {
    var object = boxes.children[i];
    object.material.opacity -= 0.001;
    object.position.x += object.directionX;
    object.position.y += object.directionY;
    object.rotation.z += Math.random() * 0.1;
```

20. Resetting boxes

When the boxes fade out with their opacity at zero, they are reset in the centre of the screen. The alpha map on the text is updated so that it moves diagonally up and to the right each frame. This is updated as an offset on the material.

```
if (object.material.opacity <= 0) {
  object.material.opacity = 0.7;
  object.position.x = 0;
  object.position.y = 0;
}
```

Creating the triangles

The triangles are created through code using a procedural process that takes the points of the triangle and turns them into a 3D model. Several iterations of the triangle are created. First in the background is just a flat triangle with a black surface colour to create the background. Then there is another flat version, but this time using the purple colour. In front of that is an extruded triangle with a bevelled edge and a reflection map or an environment map. Finally there are three line versions at the front including a dashed line. You can see the process by opening the 'shapes.js' file. The points to make the triangle were taken from the points of the triangle in Illustrator, taking the x and y location of each point.

```
}
```

```
var materialBg = model.children[0].children[0];
material.alphaMap;
var time = Date.now() * 0.00025;
var ox = (time * -0.1 * materialBg.repeat.x) % 1;
var oy = (time * -0.1 * materialBg.repeat.y) % 1;
materialBg.offset.set(ox, oy);
```

21. End of the code

Now the yellow sliders are moved diagonally across the screen. These are reset when they are far enough off screen. Save the page and test this on a server so that all of the materials and models load. It won't work as just a locally opened file on your computer.

```
for (var i = 0; i < sliders.length; i++) {
  sliders[i].position.x -= 0.9;
  sliders[i].position.y -= 1.2;
  if (sliders[i].position.y < -400) {
    sliders[i].position.y = 400;
    sliders[i].position.x = Math.random() * 600 - 50;
  }
}
renderer.render(scene, camera); }
```



Step 21: Everything is in place – the particles are animated, the yellow lines are moving and the alpha map on the text is animated. The whole scene responds to the user's mouse movement

web workshop

Code an animated notification toggle icon

Inspired by <http://portion.io>

Menu toggle

The toggle feature is also used for access to the menu, allowing users to be presented with a full screen menu when required.

Fixed positioning

Navigation and notifications use fixed positioning to remain visible regardless of page scrolling, providing easy user access.

Animation features

The animation within the content image can be fully appreciated by the user without obstruction from the notification.

You're one of a kind.

• Art is too.

Main content title

The main content title is presented to be fully visible without any risk of being obstructed by the notification content that appears.

Notification icon

The closed notification can be opened by clicking on the bell icon, providing users with choice for when to read the content.





comment
What our experts think of the site

Extended functionality

While notifications tend to be simple text, consider how offering access to functionality can benefit the user. From providing an automatic calculation tool, through to allowing submission of a question, integrating functionality can extend notifications to become a helpful feature for the user.

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

Technique

1. Document initiation

The first step is to define the structure of the HTML document. This consists of the document container, which stores the head and body sections. While the head sections is used to load the external CSS and JavaScript resources, the body is used to contain the content elements created in step 2.

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

2. Content definition

The page content consists of an article container that stores two child elements. The first child will be formatted to become a toggle button, while the second 'section' element is used to contain the notification content. Take note of how the 'data-toggle' attribute is used on the article container.

```
<article data-toggle="closed">
    <span></span>
    <section>
        <h3>Notification</h3>
        <p>Message.</p>
    </section>
</article>
```

3. Toggle search

Create a new file called 'code.js'. The first part of the JavaScript waits for the page to load. Once loaded, a function is activated to search for any first child inside an element with the 'data-toggle' attribute - ie the span element from step 2. A loop is used to reference each item found.

```
window.addEventListener("load",function(){
    var nodes = document.
querySelectorAll('[data-toggle] *:first-
child');
    for(var i=0; i<nodes.length; i++){
        *** STEP 4 HERE
    }
});
```

4. Apply toggle

Each item referenced by the loop in step 3 has a 'click' event listener applied to them. This will trigger a function to toggle the value of the parent 'data-toggle' attribute between 'on' and 'off'. In this case, the parent is the 'article' container defined in step 2.

```
nodes[i].addEventListener("click",function(){
    var parent = this.parentNode;
    if(parent.getAttribute("data-toggle")
    == "on")
        parent.setAttribute("data-
toggle","off");
    else
        parent.setAttribute("data-
toggle","on");
});
```

5. Toggle container

Create a new file called 'styles.css'. This step sets the formatting for the toggle container - ie the element using the 'data-toggle' attribute. The default settings for this element is to appear as a block in the bottom-right corner using a high z-index for visibility above all page content.

```
[data-toggle]{
    display: block;
    position: fixed;
    box-sizing: border-box;
    font-family: arial;
    bottom: 0;
    right: 0;
    z-index: 9000;
    padding: 0;
    width: 2em;
    height: 2em;
    background: transparent;
    transition: height .2s;
}
```

6. Toggle button space

The first child of the toggle container is used to define the interactive toggle button. Firstly, this must be defined with the correct default positioning and size settings so that click interactions can be detected by the JavaScript from steps 3 and 4.

```
[data-toggle] > *:first-child{
    display: block;
    position: absolute;
    right: 0;
    bottom: 0;
```

```
z-index: 99999;
font-size: 2cm;
width: 1em !important;
height: 1em;
transition: bottom 1s;
}
```

7. Toggle button icon

The icon shown is made using the 'before' virtual element placed over the toggle button element. This allows the presentation of the icon to be fully controlled from CSS. This step sets the defaults for the icon, which is to display as a '?' character that covers the full space of the button.

```
[data-toggle] > *:first-child::before{
    content: "?";
    display: block;
    width: 100%;
    height: 100%;
    background: silver;
    color: #000;
    text-align: center;
}
```

8. Notification container

The notice container is the second child inside the toggle container. Default settings are set for background colour and positioning. Placed at the bottom of the screen using fixed positioning to avoid being affected by the scrolling location, the element is set with zero height to be invisible.

```
[data-toggle] > *:nth-child(2){
    position: fixed;
    height: 0;
    width: 100vw;
    bottom: 0;
    left: 0;
    background: red;
    transition: height 1s;
    overflow: auto;
}
```

9. Toggle visibility

Visibility of the notification container and its associated toggle icon need to be changed in response to the 'on' value defined in step 4. Content becomes visible due to the height of the notification container. Changes appear animated due to the applied transition setting set in step 8.

```
[data-toggle="on"]{
    width: 100%;
    height: 100%;
}
[data-toggle="on"] > *:nth-child(2){
    height: 40%;
}
[data-toggle="on"] > *:first-child{
    bottom: 40%;
}
[data-toggle="on"] > *:first-child::before{
    content: "\00d7"; }
```

HOT TOP



The best fonts, for me, are the ones that are cohesive and professional-looking with just the right amount of quirkiness. Though of course, every typeface can have a purpose

Mark White,
Techniques Editor
@markwhlte

TYPE TRENDS FOR 2018

Here we reveal a small but smart selection of fonts that you need in your collection. Plus, find a few Photoshop tricks to make them work in your designs

Free fonts for 2018

The best of what's about on the web – for absolutely nothing!

**THE FIVE BOXING
WIZARDS JUMP
QUICKLY**

POGO

bit.ly/2NxP5sQ

Pogo is a long, sharp, good-looking font that looks superb in headings; it's both curved and crisp, making it very versatile.

AMAZINGLY FEW
DISCOTHEQUES
PROVIDE
JUKEBOXES

FACON

bit.ly/2OsMN10

A strong, capitalised font that looks dynamic and has strong movement. It's cohesive as a group of characters. Great on posters.

ABCDEFGHIJ

1234567890

KOLIKO

bit.ly/1Oyiofu

Comes in three weights and has both upper and lower-case letters, making it good for paragraphs and text as well as headings.



PEACE SANS

bit.ly/21FIHST

This bold font works both big and small for your projects and is versatile enough to use in a all kinds of work.



LUCY ROSE

bit.ly/2v3mZR1

With tons of great variations, this serif typeface is perfect for classy spreads and can be paired with strong sans serif fonts, too.



KUNG FONT

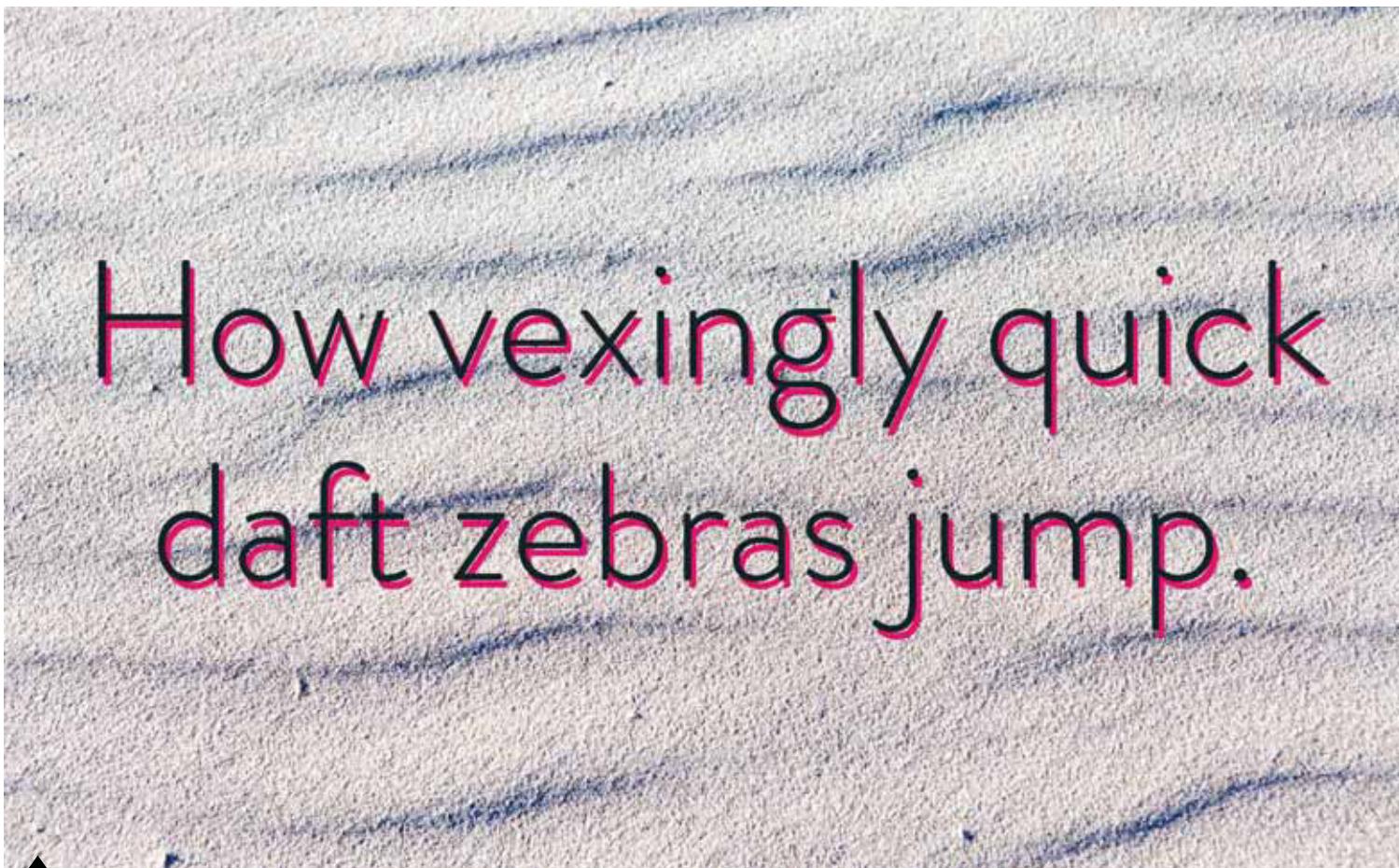
bit.ly/2M579fa

Kung Font is one of the best brush fonts out there; it works at any size, has great texture and the characters are all cohesive.



Paid fonts for 2018

You will need to untie the purse strings for these, but they are well worth it



BRANDON GROTESQUE bit.ly/1dnxm70

A font that can be used for paragraphs on your website, Brandon Grotesque is a popular serif font that feels sleek and current.

ABCDEFGHIJ 1234567890

AUTHENTIA

bit.ly/2NZOn9H

A typeface that looks like it's been sketched with a marker pen, Authentia is good for giving a more personal touch to posters.



OBJECTIV adobe.ly/2Ke7ZV9

Objectiv has a modern feel to it, but it's also appropriate in classic designs. It's pretty in all weights, and doesn't feel too quirky either.

ABCDEFGHIJ 1234567890

Embedding Google Fonts

Embedding typefaces via Google Fonts is often the simplest way to get a great typeface on your site. This is simpler to code and more reliable in many ways, and you only have to search for the font on

fonts.google.com to locate it before you embed it. You can contact Google to have your own fonts added to the database; to contribute, get in touch via GitHub, at bit.ly/2LKzJq7

Contribute to Google Fonts

Feedback

If you have any feedback on Google Fonts API, delivery, or the fonts themselves, please let us know.

New Families

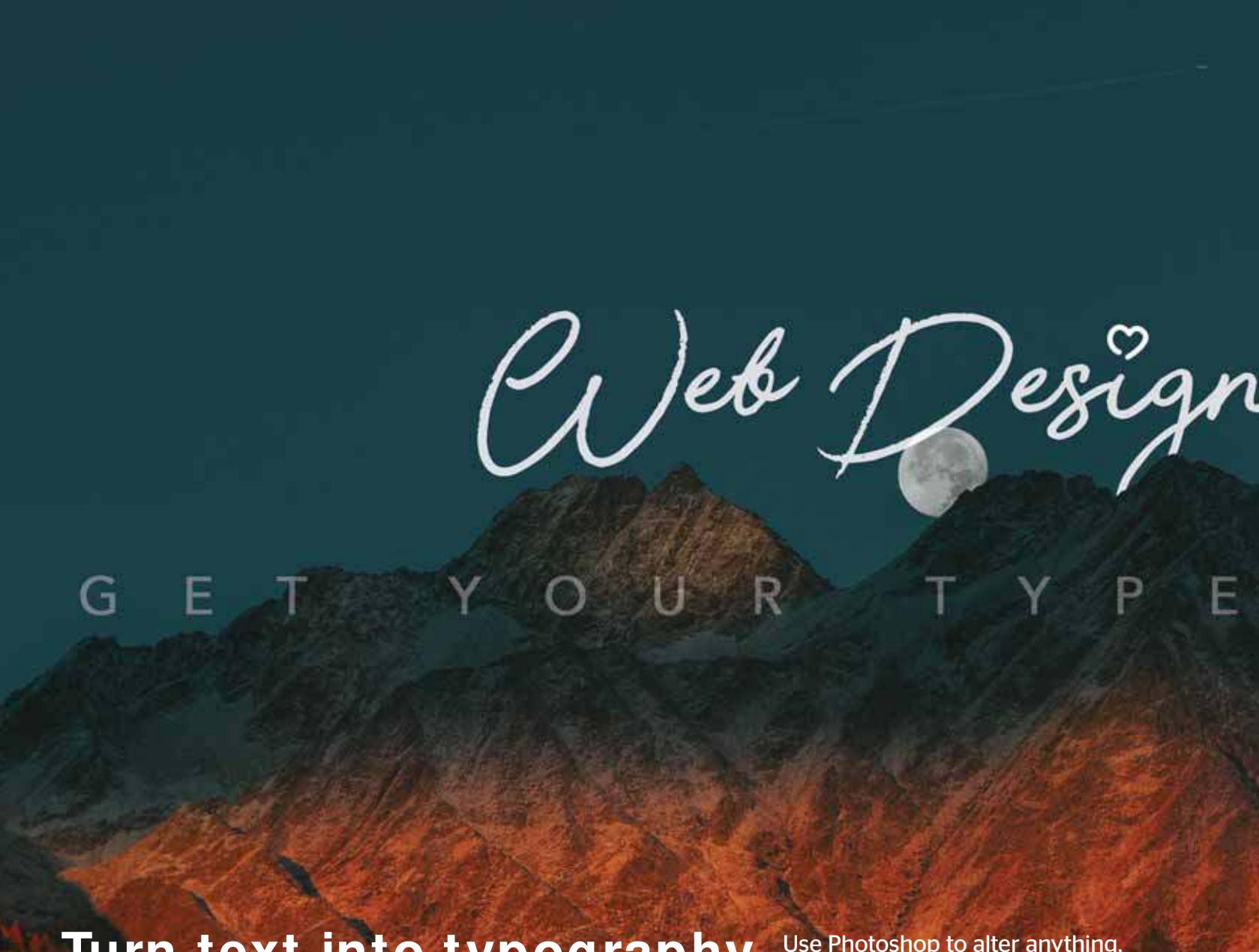
Want to add a new font family to the Google Fonts collection? Let us know by filling in the following form:

- The proposed design is original, or it represents a subset of the public domain, and of good quality. The Google Design team curates the overall Google Fonts collection and reserves the right to use or not use any font family based on its own judgment. We encourage you to keep your designs as open as possible to allow for further development.
- The project is actively developed under the MIT License (or equivalent), and there are no copyright restrictions on the font files.

LUMIERE bit.ly/2LBUELY

The Lumiere font family has a whopping 14 typefaces included for all occasions, and looks fantastic for vintage settings or stylish headings.

A B C D E F G H I J 1 2 3 4 5 6 7 8 9 0



Turn text into typography

Use Photoshop to alter anything, from the spacing to the style

Type is something that can vary from project to project; sometimes you're going to need to mask it, other times you'll need to set it to bold or italic, and sometimes you can just leave it all together. Use this guide, though, almost as a checklist for what you need to do to get the best from your text project. All steps are optional!

01. Align your text

You may be used to selecting the Type tool in Photoshop and simply clicking on the page wherever you'd like to create text. If you drag the tool however, you'll have more control for where you'd like to place that text; drag from one side to the other to leave the text directly in the centre. Use the alignment icons, too, to set your text's placement.

02. Mask your text

Sometimes, you may want your

text to interact with the background you're creating it on. This is where the Mask tool can come in. Select Mask on your Type layer, reduce the opacity so you can see the background through the text, and use a brush to mask elements out.

03. Pairing fonts

There are certain rules on which fonts look good together, but ultimately, it's all about what looks good to the naked eye. Search online for popular font combos, and play around until you have two fonts that really look good together as well as apart.

04. Caps, bold, etc.

Click on the Character icon on the right on Photoshop, and if it's not there, find it under the Window menu. Here, you can adjust text size, and choose whether you want to make your text bold, italic, all





caps, small caps, have an underline or a strikethrough and much more.

05. Leading

Use the 'AV' icon to alter the spacing of your text. This will shift individual letters exactly the same number of pixels away from each other. It's great for subheadings and it

can also help to fit more characters together on a line, as well as spacing them apart.

06. Opacity and blend mode

The opacity and blend mode of your layer will tell you how your layer reacts to the layers below.



Lower the opacity to blend it further into the image and experiment with the blend mode to discover cool effects.

07. Tracking

The tracking of your font refers to the space from one line to the next, and can be found on the dropdown menu that's just above the 'AV' icon. This can help you to shift your text closer together, or space it out as appropriate.

08. Moving text

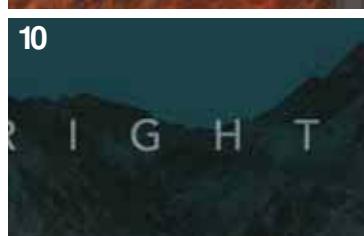
If you want to move your text independently from the mask – so the mask stays in the same place and the text moves – click the lock symbol between the type layer and its mask to remove it.

09. Edit the text itself

You can always mask out characters or parts of characters and draw your own text. Here, we've got rid of the dot over the 'i' with a mask and on a new layer, drawn a heart with the Pen tool, before stroking it with a white, hard brush.

10. Manual kerning

Sometimes, you may have to edit your characters manually. To do this, Ctrl/right-click a layer and choose to Rasterize it. This will allow you to move specific letters however you please.



There are certain rules on which fonts look good together, but ultimately, it's all about what looks good to the naked eye

Kerning explained

Kerning is the spacing between the characters of your font. Kerning adjusts the spacing between individual characters; tracking on the other hand is what you use to alter the spacing between letters in a uniform way.

In Photoshop, the easiest way to kern is to Rasterize your layer by Ctrl/right-clicking it and manually nudging the characters either left or right, depending on where you'd like them to go. This does, however, mean that once you've rasterized, you can't edit the text itself.

In font creation programs such as FontLab and High Logic Font Creator, you can edit the kerning of a font and auto-kern every single letter when creating your own typefaces.

THE BEAUTIFUL



Make text stand out

The key to any type poster is to make it striking. When it comes to the web, it's important to make it legible, too. Here, we're going to be making a cool, sports-themed typography header in Photoshop that feels pretty and embellished while also standing out against its background.

01. Spell out your text

Drag a Type box from one side to the other in your document, and write your title. We've supplied the font family, Tiki Taka, on FileSilo, so you'll be able to use the regular font (Mediocentro) to spell out your text.

02. Alter characters

Swap out some of the characters for some other characters in this step. There's another version of

Tiki Taka, named Ramdeuter, that's included, and this has lots of football-related iconography within the characters for you to swap into this title.

03. Add type embellishments

Where the 'T' and 'H' meet, we're going to create a new layer and add a full stop above the text layer, just as a quick embellishment. You can draw with the Pen tool, too; just make sure to group your layers together when you do this.

04. Add a drop shadow

Ctrl/right-click your layer and go to Layer Style. There, you'll find the Drop Shadow style. As you can see, this gives your text a bit of space between itself and the background. Experiment with

Create a text heading that will stand out against its background on a website

the values until you find something you're happy with.

05. Gradient overlay

Layer Style also enables you to recolour your text layers. We're going to go for a subtle white to grey gradient here, but you can use bright and outlandish colours, if you wish, too.

06. Inner shadow

We're going to insert an inner shadow to the type to give it even more shape. Photoshop will automatically use Global Light, meaning that the light direction and intensity will remain the same for the drop shadow and inner shadow, but you can change this by checking the option for that.

07. Paste layer style

One quick trick that's useful to

know is that you can make any layer have exactly the same Layer Style as another. To do that, simply Ctrl/right-click a layer and choose Copy Layer Style, before going onto another layer and Ctrl/right-clicking before clicking Paste Layer Style.

08. Pen tool magic

Now let's create some Pen tool embellishments for inside the text; here, we've done some tactical arrows and dots to continue the theme. Again, go to Layer Styles, and add a subtle drop shadow to these, too.

09. Mask

Cmd/Ctrl+click on the text layer's preview window to select all of the pixels on that layer. When you see the marching ants to signify that this space is selected, mask the

GAME



Pen tool embellishments so that they only appear over the text.

10. Saving for web

Now we're going to save the heading to be used as a PNG image on our site. Hide the background and hit Cmd/Ctrl+Alt/Opt+Shift+S to Save For Web. If you want this image to remain static while the background behind it moves, use the CSS code as follows: position:fixed;



When choosing fonts for your projects, it's important to ask yourself what kind of typeface you want to be looking at

Bold, regular, italic, thin or black?

So many font families have an array of weights when it comes to different versions of fonts. The difference between the Light and Black versions, too, can even look like completely different typefaces; at least though, that makes them easy to pair together.

When choosing fonts for your projects, it's important to ask yourself what kind of typeface you want to be looking at. Bold or black style fonts can often make the most obvious impact, but sometimes going with a lighter weight of font can be subtler and thus more striking. With Layer Styles in Photoshop, there are ways to make any weight stand out.

Carefully decide before you commit. It's worth spending a little while experimenting with font weights to work out what makes what kind of impact in your work.

FLUTTER

YOUR WAY TO NATIVE MOBILE APPS

Google's new framework offers a fresh approach to developing cross-platform native apps



Until now, cross-platform mobile development has been a game of trade-offs with no perfect solution.

It's just possible that Flutter might be able to change that

Simon Jones
Software Engineering Director
@j_a_nomis
<http://simonjones.tech>

Approaches to mobile development

Navigate the minefield of native, hybrid, web apps and much more

If you're building for mobile these days, then almost without exception you'll want to cater for both iOS and Android (the market share of other mobile operating systems is negligible). A native experience is usually best for users, but traditionally this meant building two separate apps and maintaining two code bases - typically using Swift for iOS and Java for Android.

The problem of overhead is obvious. Many companies maintain two separate development teams, and for individual app developers this can mean picking one platform or another.

Fortunately, there are many cross-platform frameworks out there. The earliest of these were 'hybrid' app frameworks that used WebViews - a way of embedding HTML and JavaScript within an app shell. This came with an impact to performance, and didn't offer a true native user experience. Ionic is one of the better-known of these frameworks.

Facebook's React Native took the mobile world by storm on its release, as it seemed to solve this problem by allowing apps written in JavaScript (with React syntax) to use native device widgets. This took away a major limitation of hybrid apps, but still brought performance limitations due to the need for a 'bridge' between JavaScript and native code.

Recently it's become increasingly viable to dispense with native build completely and build 'progressive web apps'. These are fully web-based, and use modern APIs such as service workers, to provide an app-like experience. While still a relatively new concept, as it matures this could reduce or eliminate the need for native mobile build.

As it stands today, if you're going to build a mobile app, there's no easy answer, and each approach has trade-offs. Many company meetings have been spent debating the merits of native versus hybrid development. Perhaps Flutter can change this.

What is Flutter?

Flutter is shaping up to offer all of the pros of cross-platform development with very few of the cons

Flutter is a new cross-platform mobile software development kit from Google. Approaching a 1.0 release at the time of writing, what sets it apart from other cross-platform frameworks is that it compiles to pure native code for both platforms. There's no JavaScript bridge or WebView in there at all. This means that while you code once, what you get is in essence two completely native mobile apps. A single UI widget you use in your Flutter app will be built natively for each device. Flutter development is done in the Dart programming language, another Google creation. Dart is object-oriented, with C-style (curly bracket) syntax, not dissimilar to Java. App development with Flutter takes a lot of influence from React, and if you've used React or React Native in the past, it will feel very familiar once you learn your way around Dart's syntax.



The Flutter website homepage features a prominent 'Flutter' logo at the top left. To its right are links for 'Docs', 'Showcase', 'GitHub', 'Packages', and 'Support'. A search bar is located on the far right. Below the header, a banner displays the text 'Latest release: Flutter Release Preview 1'. The main headline reads 'Build beautiful native apps in record time'. A descriptive paragraph explains that Flutter is Google's mobile app SDK for crafting high-quality native interfaces on iOS and Android in record time. It highlights that Flutter works with existing code, is used by developers and organizations around the world, and is free and open source. A blue 'GET STARTED' button is positioned below the paragraph. To the right of the text, there is a 3D-style illustration of a smartphone and a tablet displaying their respective interfaces.

4 reasons to use Flutter

Developing in Flutter feels very refined once you get going. Here's why it's worth trying

COMPILING TO NATIVE

The ability to generate native apps is very powerful. By going fully native, you ensure you can take advantage of the full range of device features, and give your users a better experience than with a WebView solution (or a mobile webpage). The downside though is Flutter will build widgets which look consistent on both platforms, but come out of the box styled more suitably for Android, so the native widgets Flutter builds don't quite look like iOS.

You'll also get faster app startup times and improved in-app performance compared to a JavaScript-based framework, since there's no need to load hefty runtime framework code, nor pass back and forth across a JavaScript 'bridge' (which the likes of Ionic and React Native use to allow JS to interface with native code). Flutter builds using the Android SDK and Xcode, and so you can use their debugging tools and device emulators too.

THE DART LANGUAGE

Programming languages have come a long way over the years. Going back to Objective-C can feel a little dated and cumbersome after Swift, and similarly, while Java has aged relatively well, it's hard to find the motivation to go back for it once you've indulged in Kotlin or Scala.

Dart feels like a modern language, a language for today, and offers some compelling evolution over earlier C-style languages. Yet it doesn't stray far enough that it will be completely unfamiliar to users of Java or even JavaScript. Dart is object-oriented, supports strong static typing, uses class definitions with support for inheritance, interfaces and abstract classes, and various other attributes that you've probably seen before.

Dart isn't only used for Flutter, and can also be used to build web applications and server-side code. It's also possible to compile Dart to JavaScript.

WIDGET-BASED APP STRUCTURE

It's quite possible that React is the most popular way to develop complex web applications today. There are a few reasons - one of them is its component structure with data handled through a combination of props, state and events. It's no accident that Flutter adopts a similar paradigm to React. Everything in Flutter is a 'widget', which is conceptually very similar to a React component. Widgets' presentation is controlled through a 'build()' method, similar to React's 'render()' function. Your app is simply a hierarchy of widgets, each with clearly encapsulated behaviour, which can be re-used as needed. Widgets can be customised at creation by passing them 'props', and some widgets can hold state. Flutter applies some clever techniques to optimise redrawing the page when widgets change, providing performance optimisation similar to React's virtual DOM.

MATURE DEVELOPMENT ENVIRONMENT

While Flutter is very new, it's clear that a huge amount of thought has gone into the development experience. While the setup is understandably complex, Flutter does a great job guiding you through it.

While Flutter can be run at the command-line, you also get a choice of IDE integration, and at the moment can choose to work within IntelliJ, Android Studio or VSCode. One of Flutter's best features is support for hot reload, which will update a running version of your app on a device emulator as code changes. You're probably used to this with most web frameworks by now, and Flutter brings it to mobile too. Finally, Flutter ships with a great selection of widgets out-of-the-box, and is particularly geared toward Google's Material Design framework. Meaning respectable-looking apps are on the cards.

Your first Flutter app

The path to cross-platform goodness

1. INSTALLATION

It can take a bit of time to get Flutter set up correctly. Make sure you've installed:

Xcode
Android Studio
Android SDK
VSCode

(Flutter works with other IDEs, but this is a good choice). Run all of these and install any additional components they prompt you to. Now you're ready to download Flutter from <https://flutter.io> and unzip it wherever you'd like it installed. You'll also want to add it to your path variable, instructions for which can be found on the Flutter website, to allow it to be invoked from anywhere.

2. CALLING THE DOCTOR

Flutter ships with a handy 'doctor' tool which you can use to check it's installed correctly. If you run:

`flutter doctor`

It will tell you what's set up correctly and what isn't. Take close note of the instructions, until it's happy that Flutter, the iOS Toolchain, Android Toolchain and Android Studio are all set up nice and correctly.

3. IDE SETUP

VSCode has a very good Flutter extension which will integrate Flutter into the IDE for you. In the command palette, search for 'Extensions: Install Extension', then on the Extensions menu, search for 'Flutter' and install the plugin. It will also install the Dart plugin for you. Restart VSCode, and you're ready to go. You can select 'Flutter: New Project' through the command palette to generate a starter project.

4. RUNNING THE APP

The easiest way to run the app is using an iOS simulator or Android emulator on the desktop. If you've already set these up (which can be done using Xcode and Android Studio respectively - more details on how on the Flutter website), then you will find that when you

click 'No Devices' in the bottom right of VSCode, you see them to select. Choose one, then if you hit Debug>Start Debugging, VSCode will invoke Flutter to compile the starter project and boot it up on a virtual device for you.

5. STARTING TO CUSTOMISE

We'll create a simple log-on page for what could be a banking app. The app's code is contained within 'main.dart'. Start by changing the page title:

```
home: new MyHomePage(title: 'Superbank'),
```

You'll see the hot reload as soon as you save the change. Let's also get rid of the existing content by removing the body and 'floatingActionButton' parameters from the Scaffold in the 'MyHomePageState' class.

6. ADDING OUR OWN WIDGETS

Now let's add some widgets of our own. We want a username and password field. In Flutter terms, this is a Form widget containing 'TextFormField' widgets. To create a username field:

```
body: new Form (
    child: new Center (
        child: new Column (
            mainAxisAlignment: MainAxisAlignment.spaceAround,
            children: <Widget>[
                new Container(height: 210.0,
                    padding: const EdgeInsets.symmetric(horizontal: 16.0),
                    child: new ListView (
                        // ...
                    )
                ),
            ],
        )
    ),
),
```

7. COMPLETING OUR FORM

We also need a password field and a button. Add some further children to the 'ListView' within the Form like below.

```
new TextFormField(
    decoration: const InputDecoration(
```

```
icon: const Icon(Icons.lock),
hintText: 'Enter your password',
labelText: 'Password',
obscureText: true,
),
new RaisedButton(
child: const Text('Submit'),
onPressed: () {
}),
)
```

8. IMPROVING THE LAYOUT

It doesn't look great right now, bunched up at the top of the page. Let's apply some padding and centre it vertically, which we do by creating a Center widget, containing a Column widget, and wrapping the 'ListView' in a container. This might seem slightly confusing, and it's worth reading up online about how Flutter's layout widgets work.

```
body: new Form (
    child: new Center (
        child: new Column (
            mainAxisAlignment: MainAxisAlignment.spaceAround,
            children: <Widget>[
                new Container(height: 210.0,
                    padding: const EdgeInsets.symmetric(horizontal: 16.0),
                    child: new ListView (
                        // ...
                    )
                ),
            ],
        )
    ),
),
```

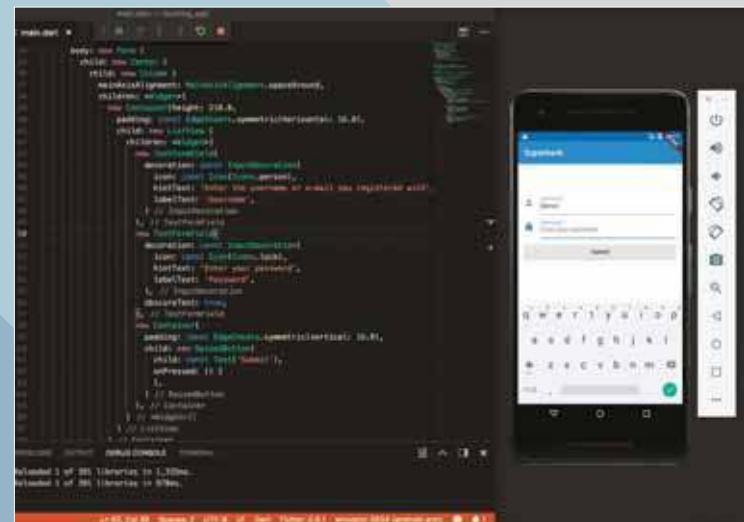
9. MORE LAYOUT

Layout is one of the toughest things to get right in Flutter, but hopefully you're beginning to get a feel for how widgets come together to construct apps, as well as how functional the out-of-the-box widgets are. Let's also add a bit of padding to the button. To do this, we wrap it within a container too.

```
new Container(
padding: const EdgeInsets.all(16.0),
child: new RaisedButton(
child: const Text('Submit'),
onPressed: () {
}),
),
```

10. NEXT STEPS

There's more we need to do here. We need to add some sort of validation to our form, which is easy to do in Flutter. We also need to extend the 'onPressed()' method to do something. It would probably also be appropriate to split out the log-in form into a new custom widget. But before doing that, why not take a look at it in both Android and iOS, on different devices, and at different screen rotations? It's impressive.



Where's Flutter being used?

We'll need to wait for a stable release before it starts to become mainstream, but there are already a few apps (big and small) out there using Flutter. As well as Google themselves, Alibaba is one big name which is already using Flutter, and Groupon are beginning to work with it. A complete list of apps built with it can be found here: <https://itsallwidgets.com>

Other cross-platform frameworks to check out

REACT NATIVE

<https://facebook.github.io/react-native>

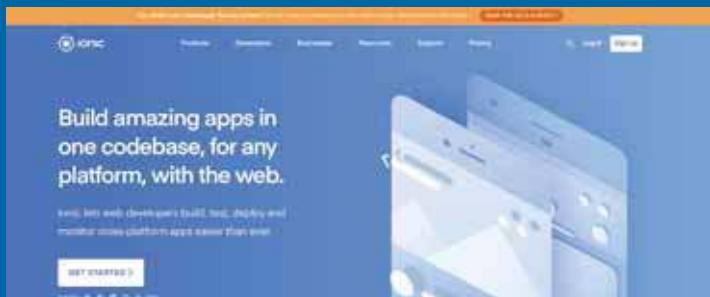
This has a lot of similarities with Flutter, and has been in use for a few years. If you like developing with JS, and React syntax in particular, you'll feel at home with it. It uses native widgets controlled by JS code, so delivers superior performance and experience to hybrid apps.



IONIC

<https://ionicframework.com>

Ionic is designed for the creation of hybrid apps. It has a large community, good level of enterprise use, so the ecosystem surrounding it is very good. The downside is, you're not going native.



XAMARIN

<https://xamarin.com>

Xamarin, owned by Microsoft, throws C# into the mix for cross-platform development. Xamarin compiles native apps from within Visual Studio. It's popular with .NET developers, but requires a license for commercial use.



CORDOVA

<https://cordova.apache.org>

Cordova is about bringing apps built in HTML, CSS and JS to mobile. Arguably the simplest way to build a cross-platform app, and probably the most familiar for web devs. But you won't get the performance.



Flutter resources

There's a wealth of information online to help you with Flutter

FLUTTER DOCUMENTATION

<https://flutter.io/docs>

Under the Documentation section you'll find information about how to get set up, and guides to common tasks.

DART LANGUAGE

www.dartlang.org

Dart isn't that difficult to get to grips with, but for the nuances of the language, check out its excellent documentation online.

FLUTTER PACKAGES

<https://bit.ly/2v0aqpO>

It's early days yet, but there are already quite a few modules out there built for Flutter to save you building things yourself.

GOOGLE CODELABS

<http://bit.ly/2vk6hfB>

Codelabs provides guided, hands-on tutorials to help you learn development, and there's content for Flutter already available.

CREATE THE IMPOSSIBLE

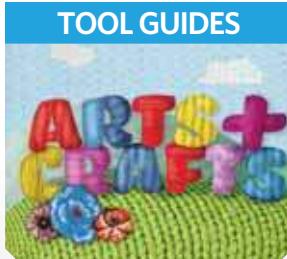
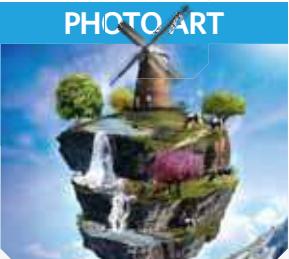
www.photoshopcreative.co.uk



Photoshop® creative

ON SALE NOW

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



Available
from all good
newsagents and
supermarkets

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

Developer tutorials

Harness the power of WebRTC

JavaScript runtimes are getting faster every day. WebRTC now takes a first step into real-time communication – find out how...





WebRTC – short for Web Real-Time Communication – is a by-product of Google's acquisition of GIPS.

Google open-sourced most of the intellectual property held by the video conferencing vendor, thereby motivating standardisation committees to accept the technology.

WebRTC is a bit of a beast due to the breadth of issues handled by the specification. The standard does not limit itself to the provisioning of a high-speed low-latency connection system, but also provides methods to access camera and microphone hardware in a convenient fashion. The difficulties involved in the real-time processing of audio and video data led to the creation of RTCPeerConnection – a class dedicated to combining the two above-mentioned APIs.

Due to the complexity of real-time communication, creating one from scratch is infeasible for all but the most experienced of web developers. The following story uses a set of open-source examples, which we will discuss in a piecemeal fashion. This knowledge lets you create an actual system by combining one or more of the systems.

As for technology, an AMD OctaCore workstation is being used. It runs Ubuntu 14.04 and a recent version of Firefox, and the AMD machine uses a BlackBerry PRIV as a webcam substitute.

1. Collect pictures

Our first example, hosted at <https://webrtc.github.io/samples/src/content/getusermedia/canvas>, contains a live WebCam preview and a canvas tag intended to hold static image information. In HTML markup, the three controls shown next to this step are relevant.

```
<video playsinline="" autoplay=""></video>
<button>Take snapshot</button>
<canvas width="480" height="360"></canvas>
```

2. 'getUserMedia' gets media information

'getUserMedia' is one of the three core methods of the WebRTC standard. It queries the media input devices available to the browser in an asynchronous fashion while keeping the constraints object in mind – data can

only be harvested from the promise element and returned after the mention.

```
const constraints = {
  audio: false,
  video: true
};

navigator.mediaDevices.
getUserMedia(constraints).then(handleSuccess).
catch(handleError);
```

3. Connect video and data source

If a media device satisfying the demands of the constraint object is found, it will be returned to the success event handler registered in the promise. In the case of our project, we simply pass the stream into the video tags 'srcObject' property:

```
function handleSuccess(stream) {
  window.stream = stream; // make stream
available to browser console
  video.srcObject = stream;
}
```

4. Take out information comfortably

Video stream objects change their content permanently. Our sample stands out from the rest in that it also contains a canvas tag, whose elements get populated with a single mention of 'drawImage'. This bitmap could then be convoluted or post-processed in any other way you fancy.

```
button.onclick = function() {
  canvas.width = video.videoWidth;
  canvas.height = video.videoHeight;
  canvas.getContext('2d').drawImage(video, 0,
0, canvas.width, canvas.height);
};
```

5. Beware of legacy code

During the protracted development process of WebRTC, the 'getUserMedia' method originally was not housed inside of the 'mediaDevices' enum. Legacy code is likely to contain snippets similar to the one accompanying this step – they should be restructured.

WebRTC samples captureStream(): video to video



3

WebRTC samples [getUserMedia => canvas](#)

4

WebRTC samples [getUserMedia => canvas](#)

Beware of slow clients

Handling live video is a computationally intensive task – especially when run in the efficiency-constrained environment that makes up a modern browser's JVM. Always keep this in mind when proposing WebRTC-based solutions to friends and customers.

Tutorials

Harness the power of WebRTC

16

WebRTC samples Peer connection



Start

Call

Turn on video

Hang Up

WebRTC samples Transmit text

Send:

Receive:

View the console to see logging.

The RTCPeerConnection objects localConnection and remoteConnection are in global scope, so you can inspect them in the console as well.

For more information about RTCDatChannel, see [Data Channel API](#).

```
navigator.getUserMedia(constraints, successCallback, errorCallback);
```

6. Use advanced constraints

While chat applications tend to be able to make do with just about any bargain-basement webcam, facial identification and similar jobs require more. In this case, the 'constraints' object can be provided with additional parameters limiting webcam selection.

```
{
  audio: true,
  video: {
    width: { min: 1024, ideal: 1280, max: 1920 },
    height: { min: 776, ideal: 720, max: 1080 }
  }
}
```

WebRTC Internals - Google Chrome

Caller origin: https://webrtc.github.io

Caller process id: 24672

Audio Constraints

Video Constraints

GetUserMedia

Create Dump

7. Source and sink

One interesting aspect of the WebRTC GUI integration is the capability to make one video tag mirror the content of another. Sadly, two different methods exist for this operation at the time of writing.

```
if (leftVideo.captureStream) {
  stream = leftVideo.captureStream();
  rightVideo.srcObject = stream;
  console.log('Captured stream from leftVideo with captureStream', stream);
} else if (leftVideo.mozCaptureStream) {
  stream = leftVideo.mozCaptureStream();
  rightVideo.srcObject = stream;
  console.log('Captured stream from
```

```
leftVideo with mozCaptureStream()', stream);
}
```

8. Transmit control data

While it's obvious that sending video is the name of the game, let us start out by transferring control data via a low latency link. This is best handled via 'RTCPeerConnection' – think of the class as a WebSocket replacement working with WebRTC-derived technologies.

9. Create a connection

While RTCPeerConnections can – in theory – transfer tremendously large data streams, they are usually used to transmitting control information. The example at <https://webrtc.github.io/samples/src/content/datchannel/basic> uses WebRTC to create some kind of chat client that transmits data between two chat windows.

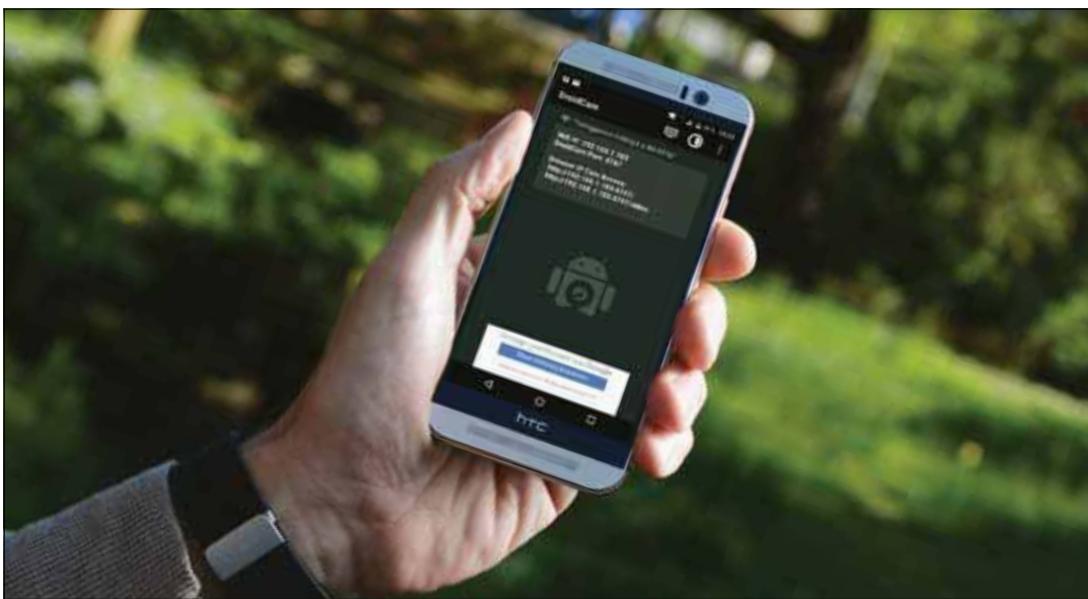
10. Link up

RTCPeerConnections do their magic by tunnelling through network hardware. This means that connection establishments can be slightly complex. In particular, one or more instances of the 'RTCIceCandidate' connection parameter objects spawn during connection establishment.

```
window.localConnection = localConnection =
new RTCPeerConnection(servers);
sendChannel = localConnection.createDataChannel('sendDataChannel');
localConnection.onicecandidate = e => {
  onIceCandidate(localConnection, e);
};
sendChannel.onopen =
```

Further reading

Should you feel like additional samples, simply visit <https://webrtc.github.io/samples> in a browser of your choice. The repository contains literally dozens of different open source examples waiting to be integrated into your systems.



Webcams are pricey

Those of a paranoid nature tend to reject the idea of connecting a camera to their workstation. Owners of Android smartphones can visit <http://www.dev47apps.com/droidcam> to download a temporary camera replacement. The product consists of a smartphone app and a desktop component that is available for both Windows and Linux systems. Once installed, start the DroidCam client on your workstation. We prefer the Wi-Fi webcam mode to direct USB connection due to the high power consumption of live video streaming. Either way, Firefox and other browsers should be able to detect the webcam once the connect button was clicked in the desktop client.

```
onSendChannelStateChange;
  sendChannel.onclose =
onSendChannelStateChange;
```

11. Communicate asynchronously

WebRTC's communication API is modelled on the programming model that you might have used in WebSockets. Messages arrive in an event handler, which you must register before use. Sadly, this can not happen directly – you must instead wait until the `WebRTCCConnection` establishes itself.

```
function receiveChannelCallback(event) {
  trace('Receive Channel Callback');
  receiveChannel = event.channel;
  receiveChannel.onmessage =
onReceiveMessageCallback;
  receiveChannel.onopen =
onReceiveChannelStateChange;
  receiveChannel.onclose =
onReceiveChannelStateChange;
}
function onReceiveMessageCallback(event) {
  trace('Received Message');
  dataChannelReceive.value = event.data;
}
```

12. Go even deeper

At this point, it is time to analyse a complete communication program. Head over to [https://webrtc.github.io/samples/src/content/peerconnection/upgrade](https://webrtc.github.io/samples/src/content/peerconnection/) and feast your eyes on two windows. They can, furthermore, be connected to one another via a simple group of buttons.

13. Get audio data

The example project starts out by creating a stream object using the `'getUserMedia'` function we discussed previously. This time, the `'constraints'` element is set up to limit itself to the provision of audio data. `'gotStream()'`

keeps the stream object in a variable called `localStream`.

```
navigator.mediaDevices
  .getUserMedia({
    audio: true,
    video: false
  })
  .then(gotStream)
```

14. Fire up the 'RTCPeerConnection'

As discussed in the introduction, the `'RTCPeerConnection'` class is not limited to transferring textual data. Developers can add media tracks to an established connection; the underlying implementation takes care of marshalling and encoding.

```
const servers = null;
pc1 = new RTCPeerConnection(servers);
pc2 = new RTCPeerConnection(servers);
localStream.getTracks().forEach(track =>
pc1.addTrack(track, localStream));
```

```
pc1.createOffer(offerOptions)
then(onCreateOfferSuccess, onCreateSessionDescriptionError);
```

15. Accept the incoming track

On the side of the data sink, implement a stream receiver. The operating system provides it with a reference object, which contains one or more data streams. We pass the first one into the `'remoteVideo'` element, while setting `'srcObject'` to null before writing the new content avoids some bugs in Edge.

```
function gotRemoteStream(e) {
  remoteVideo.srcObject = null;
  remoteVideo.srcObject = e.streams[0];
}
```

16. Promote connection I

If an audio connection is established, it can be promoted to also carry video information. This is accomplished by invoking `'getUserMedia'` once again. The resulting stream

19 Build Pending

WebRTC adapter

adapter.js is a shim to insulate apps from spec changes and prefix differences. In fact, the standards and protocols used for WebRTC implementations are highly stable, and there are only a few prefixed names. For full interop information, see webrtc.org/web-apis/interop.

Install

NPM

```
npm install webrtc-adapter
```

Bower

```
bower install webrtc-adapter
```

Yarn

```
yarn add webrtc-adapter
```

Tutorials

Harness the power of WebRTC



can then be added to the existing media object:

```
navigator.mediaDevices  
  .getUserMedia({video: true})  
  .then(stream => {  
    const videoTracks = stream.  
getVideoTracks();  
    localStream.addTrack(videoTracks[0]);  
  }...)
```

17. Promote connection II

In the next act, the new stream must also be registered with the stream that transfers content between the two ends. This should also be accomplished in a two-step process, with writing null into 'srcObject' as a bug fix.

```
localVideo.srcObject = null;  
localVideo.srcObject = localStream;  
pc1.addTrack(videoTracks[0],  
localStream);  
return pc1.createOffer();  
})
```

18. Clean up when done

Handling real-time media is computationally intensive. Due to this, stopping the process as soon as it is no longer needed is beneficial. The code accompanying this step outlines the process of closing a connection.

```
const videoTracks = localStream.  
getVideoTracks();  
videoTracks.forEach(videoTrack => {  
  videoTrack.stop();  
  localStream.removeTrack(videoTrack);  
});  
localVideo.srcObject = null;
```

```
localVideo.srcObject = localStream;
```

19. Work around implementation details

In theory, WebRTC should work across all browsers. Sadly, practical browsers tend to bring along idiosyncrasies of their own. <https://github.com/webrtc/adapter> provides a wrapper library – include it in a website or a NPM package to work around nuisances.

20. How to handle connection establishment

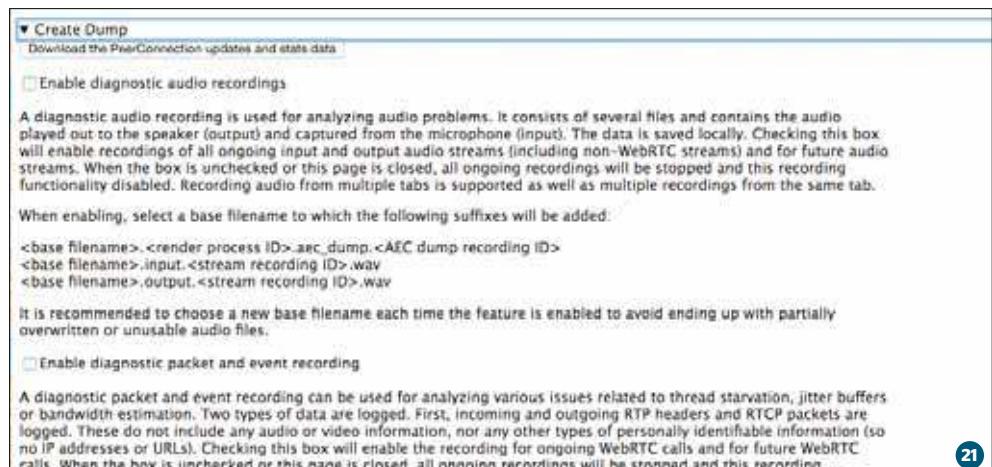
Good communication clients distinguish themselves from their lesser competition by the use of smart connection establishment services. For a web-based system, we would recommend the use of a central

server based on WebRTC. Clients connect to it and ask for peer information, which is then simply used for the actual call setup.

21. Collect connection protocol data

Traditional network sniffers have not adapted to WebRTC yet. Browser vendors combat this trend by adding debugger and statistics utilities to the trap-door pages of their browsers. Simply open them and feast your eyes on the diagrams. Finally, be prepared to switch browsers when hunting down difficult issues – not all analytics suites are effective.

```
chrome://webrtc-internals  
opera://webrtc-internals  
about:webrtc
```



Special offer for readers in *North America*



6 issues FREE

When you subscribe*

FREE
resource
downloads
every issue

FREE 10 MINUTES OF JAVASCRIPT VIDEOS
Expert tutorials, techniques and inspiration

web designer

TOP TYPE
Hot fonts and Photoshop tricks

GET STARTED WITH THREEJS
Add textures for realistic looks and visual effects

CSS GRID
How to power up your web layouts

SAY HELLO TO FLUTTER
Google's SDK to build beautiful native apps

JAVASCRIPT WHAT'S NEW?
REVEALED: THE LATEST UPDATES TO ES2018, REACT, ANGULAR, VUE & NODE

WORK WITH WEBRTC

CODE AN INTERACTIVE HEADER

CREATE PATTERNS WITH CSS

"The only magazine you need to design and develop stunning websites"



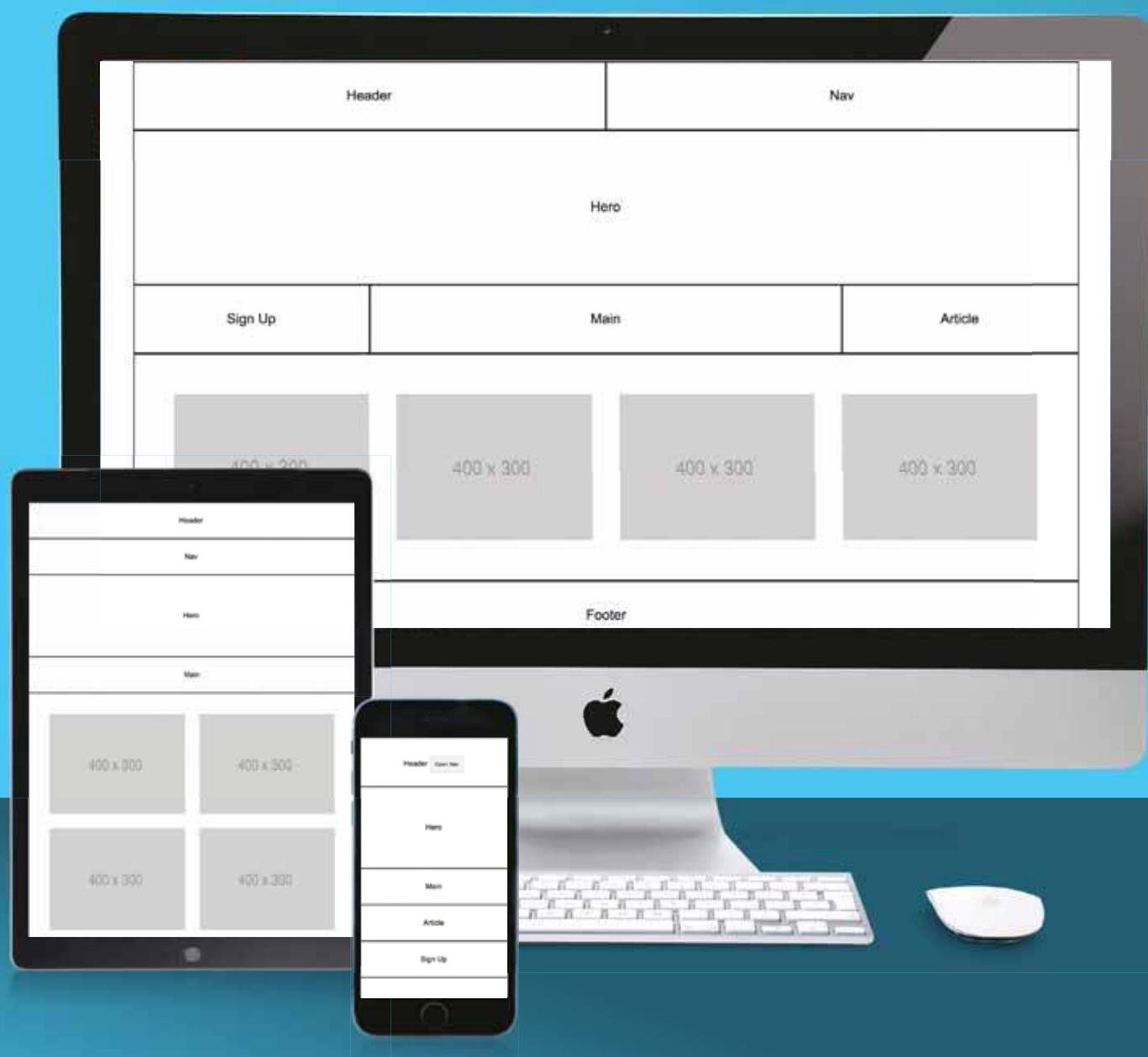
Order hotline +44 (0) 344 848 2852
Online at myfavouritemagazines.co.uk/WEDPQ17

***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 October 31 2018.

Expires
31 Oct
2018

Build better layouts with CSS Grid

Use the CSS Grid layout module to create shape-shifting layouts as well as a fallback for unsupportive browsers





The CSS Grid layout module is a groundbreaking new way to lay out blocks of content on a web page.

Elements can be placed anywhere on a page, and their position can be changed using CSS media queries to create completely different layouts for different-sized devices. A content block can, for example, be placed at the top of a mobile layout, on the left-hand side for tablets, and in the middle of desktop layouts – all without changing the HTML markup or resorting to absolute positioning. This enables elements to be ordered in the markup based on importance as opposed to visual position, a practice particularly useful for users relying on screen readers, thereby leading to improved accessibility. As CSS Grid is now supported by the latest two versions of all major browsers it's the perfect time to start introducing it into your projects. In this tutorial, we will cover how to design and implement a responsive layout using CSS Grid, as well as a fallback layout for browsers that do not support it.

1. Determine layouts and responsiveness

The first step will be to determine how many different layouts we want and for which device sizes, thereby determining responsiveness. These layouts will be implemented using CSS media queries that apply styles based on certain criteria, in this case screen width. For this tutorial we will design separate layouts for mobile devices with screen widths up to 720px, tablets with screen widths between 721px and 1000px, and desktops with widths greater than 1001px.

For all devices we will have a header used for website branding, a navigation or 'nav bar' for links to other pages, a large hero image, a main and an article section, a sign-up form block, a photo gallery and a footer. For the sake of simplicity, we will keep the blocks empty with the exception of the gallery, which we will populate with placeholder images.

2. Design the mobile layout

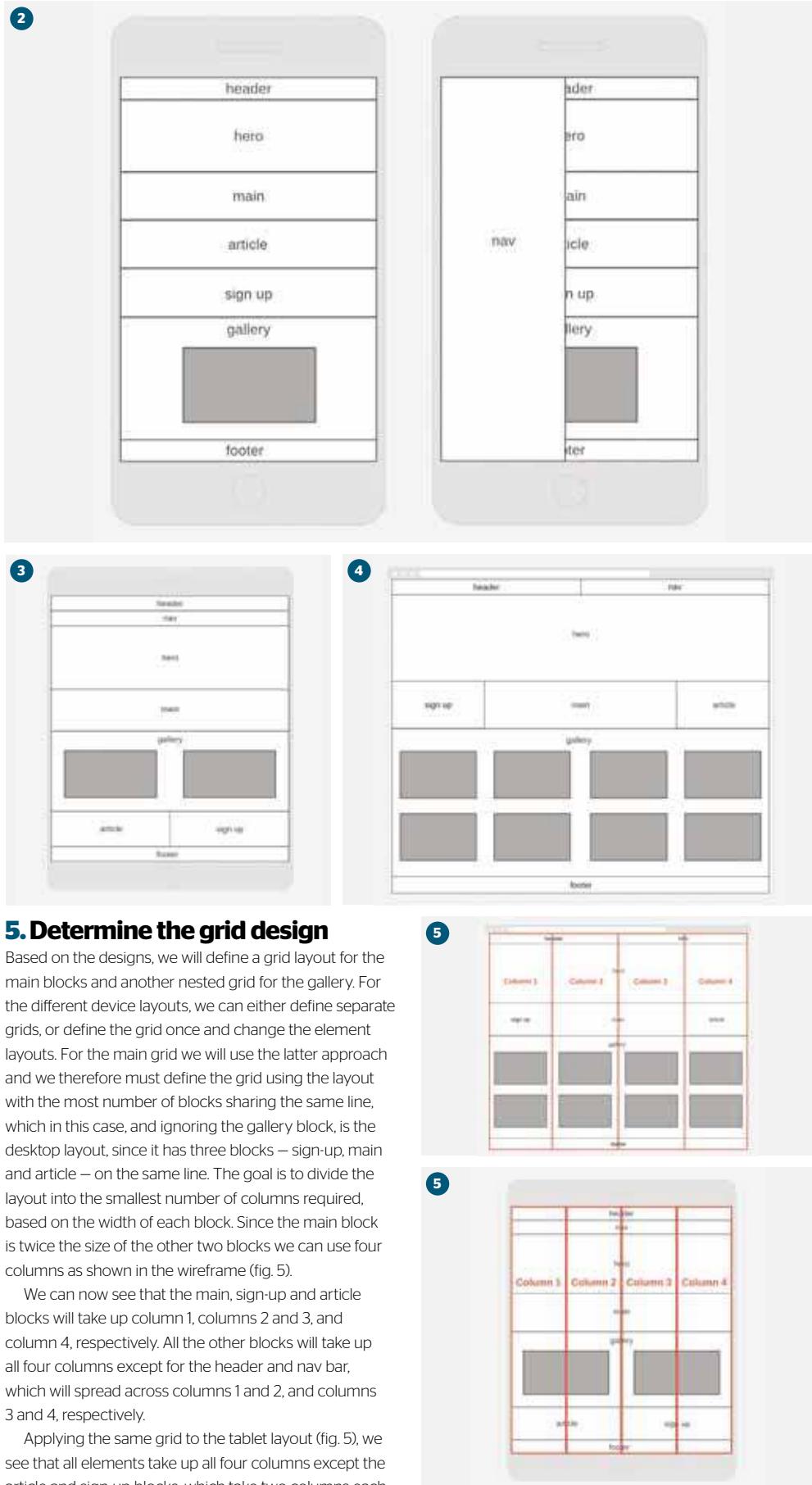
Before any development can begin we of course need to design our layouts. For mobile devices, we will use a vertically stacked layout, with a hidden nav bar that is opened and closed with buttons in the header and nav bar respectively. The nav bar will also take up the full screen height and half the screen width when opened, as shown in the wireframe (fig. 2).

3. Design the tablet layout

For the tablet layout, we will first move the gallery up to be below the main block and show two images per line, as well as place the article and sign-up blocks side by side. We will also place the nav bar under the header and add a small gap between all blocks, as can be seen in the wireframe (fig. 3).

4. Design the desktop layout

For desktop devices, we will make the header and nav bar share the same line, move the sign-up and article blocks to either side of the main block, and show four images per line in the gallery, as shown in fig. 4.



Tutorials

Build better layouts with CSS Grid

6. Set up the development environment

With our layouts and grid determined, we can start setting up the project and actually write some code. To begin with we will create a directory and add an HTML file 'index.html'. In that file we will define our base HTML code and link it to an external stylesheet, 'styles.css', and JavaScript file, 'script.js', in the same directory, in which we'll define our CSS styles and JavaScript for controlling the nav bar, respectively. The HTML file should – at a minimum – contain the following code. Note: we must add the viewport 'meta' tag for CSS media queries to function properly. You can read more about this at MDN (mzl.la/JtxYJF):

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>CSS Grid Tutorial</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" type="text/css" href="styles.css">
  </head>
  <body>
    <script type="text/javascript" src="script.js"></script>
  </body>
</html>
```

7. Define HTML markup

With our project structure set up we can now add the HTML for our blocks to the body section above the '<script>' tag. We will use as many semantic elements as possible and lay them out in order of importance, which is also how they will appear on mobile. We will also add buttons for opening and closing the nav bar on mobile, and wrap everything in a container with ID 'grid' that will be used later to define the grid.

```
<div id="grid">
  <header>Header
    <button id="nav-open">Open Nav</button>
  </header>
  <nav id="nav">Nav
    <button id="nav-close">Close Nav</button>
  </nav>
  <section id="hero">Hero</section>
  <main>Main</main>
  <article>Article</article>
  <section id="sign-up">Sign Up</section>
  <section id="gallery">Gallery</section>
  <footer>Footer</footer>
</div>
```

8. Add general CSS styles

At this point we can add our general CSS styles to 'styles.css'. Firstly, we should limit the width of the 'grid' container to 1100px and centre it so that the elements don't stretch across very wide screens. We should also limit the width of the gallery images to prevent them from overflowing outside the gallery, as well as centre them and separate them from one another using padding. We

can see the CSS code for this below, with other optional, styles used mainly for improved visuals in this tutorial, omitted for brevity:

```
/*...other optional styles omitted for brevity...*/
#grid {
  max-width: 1100px;
  margin: 0 auto;
}
#gallery {
  text-align: center;
}
#gallery img {
  padding: 16px;
  max-width: 100%;
```

9. Add mobile-specific styles

We can now add some styles to make the nav bar take up the left half of the screen on mobile devices using absolute positioning and a media query, and hide it by default. We will toggle it using 'nav-open' and 'nav-close' buttons and JavaScript in the next step. We will also hide these nav buttons for tablets and desktops since the nav bar will always be visible on these devices.

```
@media (max-width: 720px) {
  #nav {
    display: none;
    position: absolute;
    left: 0;
    top: 0;
    bottom: 0;
    right: 50%;
  }
}
@media (min-width: 721px) {
  #nav-open,
  #nav-close {
    display: none;
  }
}
```

10. Add mobile nav bar JavaScript

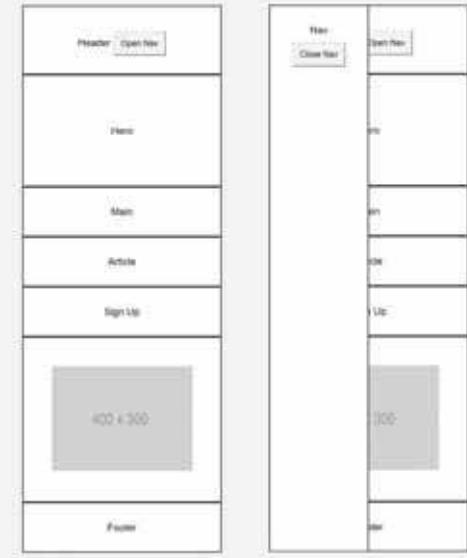
We can now add some basic JavaScript to 'script.js' for showing and hiding the nav bar when clicking Open nav and Close nav, respectively.

```
var nav = document.getElementById('nav');
document.getElementById('nav-open').addEventListener('click', function() {
  nav.style.display = 'block';
});
document.getElementById('nav-close').addEventListener('click', function() {
  nav.style.display = 'none';
});
```

11. Design fallback layout

To account for browsers that do not yet support CSS Grid we will design a simple fallback layout for tablets and larger-screen devices. This layout will make use of the element DOM order determined by the HTML defined earlier. We will also use CSS properties 'float'

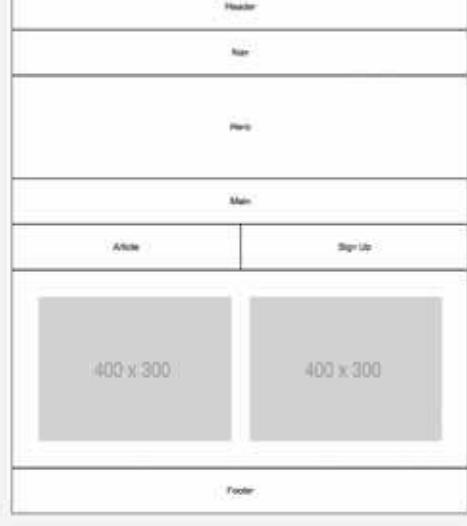
9



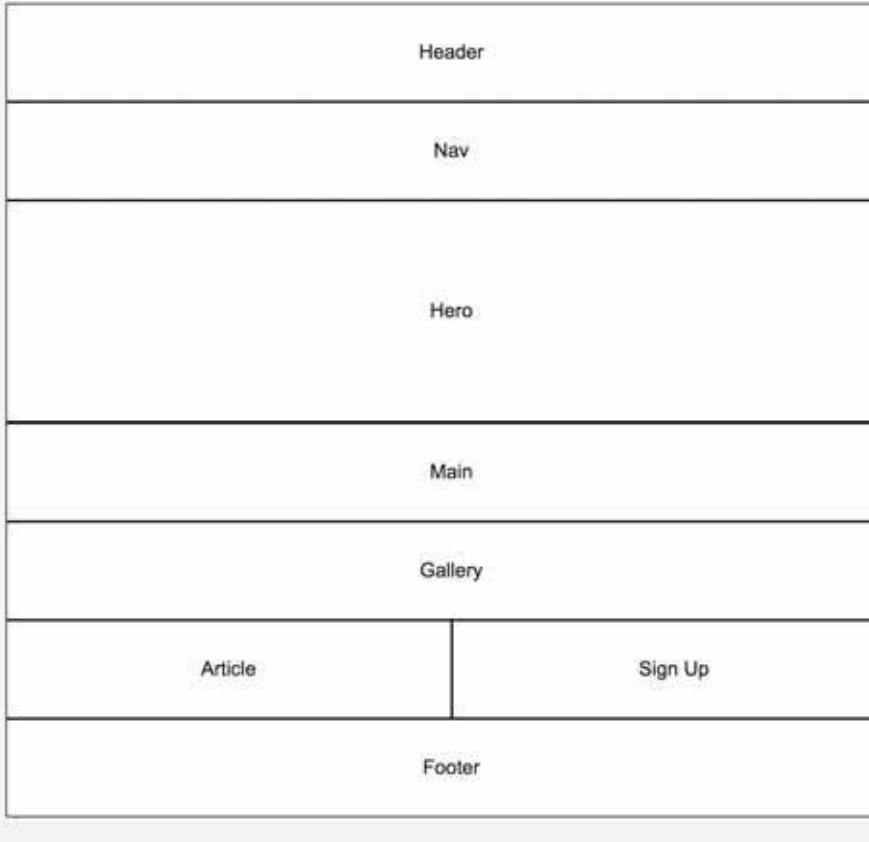
10



11



16



and 'width', which are supported by almost all browsers, to enable the article and sign-up blocks to take up the same line, as well as allow two images per line in the gallery, as can be seen in the wireframe (fig. 11).

12. Add fallback styles

We will set the width of the article and sign-up blocks to half their container's width and float them so that they lie on the same row. We will also do the same to the gallery images so that they lie two per line. We must also float blocks that precede these floated article and sign-up blocks – namely the gallery and footer – to prevent the former from overlapping the latter. Finally, to only apply the fallback styles for tablet-sized devices and larger, we can wrap them in a media query. By adding two placeholder images in the gallery from [placeholder.com](#), we can see that the following CSS code leads to the desired effect:

```
@media (min-width: 721px) {
  article,
  #sign-up,
  #gallery img {
    float: left;
    width: 50%;
  }
  #gallery,
  footer {
    float: left;
    width: 100%;
  }
}
```

13. Override fallback for browsers supporting CSS Grid

For browsers that support CSS Grid we want to override the fallback styles, removing the floats and resetting the widths to their default value. To achieve this, we can make use of the '@supports' CSS query, which applies styles based on support for a specific CSS feature. In this case we will wrap the override styles, as well as all the grid-specific CSS to follow – in a '@supports (display: grid)' block – which will only be applied in browsers that support CSS Grid. Ironically, not all browsers support the '@supports' query, but almost all of these browsers don't support CSS Grid either and will just ignore the whole block of code. Luckily, this works to our advantage even though the code is ignored for the wrong reason.

```
@supports (display: grid) {
  article,
  #sign-up,
  #gallery,
  #gallery img,
  footer {
    float: none;
    width: 100%;
  }
}
```

14. Define CSS grid areas

There are several ways to place items in a grid layout but we will be using named areas. This works by naming each element to be placed in the grid using the CSS property 'grid-area', then using these names in the 'grid-template-

areas' property to determine where they lay in the grid, which will be described later. We will keep things simple and just use the following obvious names:

```
@supports (display: grid) {
  header { grid-area: header; }
  nav { grid-area: nav; }
  #hero { grid-area: hero; }
  main { grid-area: main; }
  #gallery { grid-area: gallery; }
  article { grid-area: article; }
  #sign-up { grid-area: sign-up; }
  footer { grid-area: footer; }
}
```

15. Define the grid

Now it's finally time to set our CSS Grid properties. Inside our '@supports' block we can set the 'grid' container to a CSS grid using 'display: grid' for tablets and desktop devices, and define our columns as per our designs. CSS Grid makes use of a 'fractional' unit, 'fr', that causes available space to be distributed to columns or rows based on the ratios of these units. For example, 'grid-template-columns: 1fr 2fr' creates two dynamic columns with the second column twice the size of the first. In our case using 'grid-template-columns: repeat(4,1fr)' will define four equal width, dynamic columns.

```
@supports (display: grid) {
  @media (min-width: 721px) {
    #grid {
      display: grid;
      grid-template-columns: repeat(4, 1fr);
    }
  }
}
```

16. Define grid position or elements on tablets

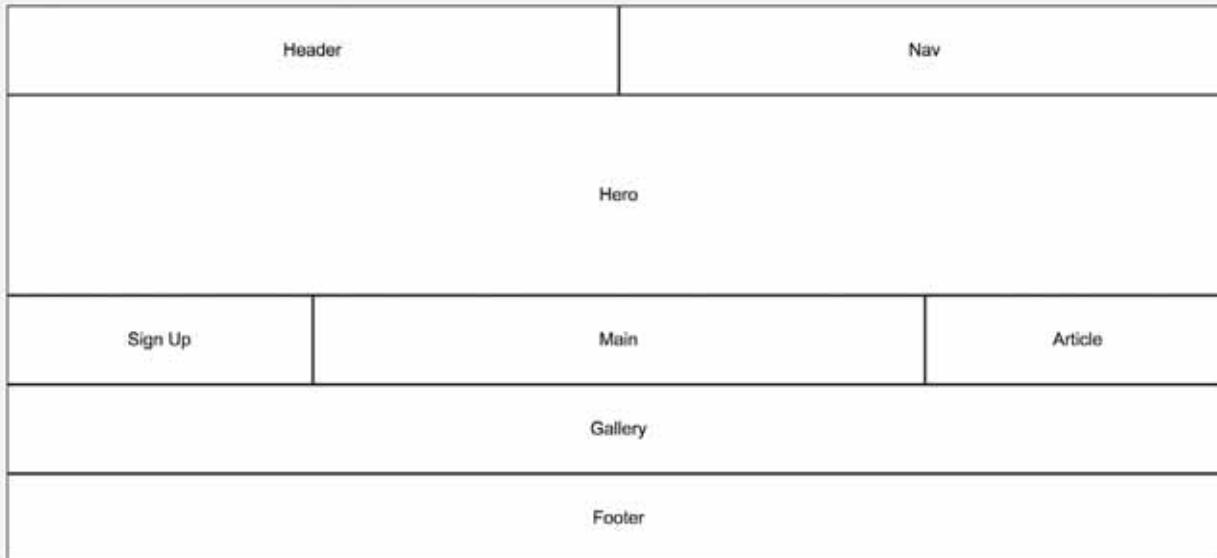
With the grid areas and columns defined, it is now possible to set 'grid-template-areas' using a visual 'map' where rows are enclosed in quotation marks, and the contents of each grid cell are represented by the 'grid-area' names of the elements. Empty cells can be symbolised by a full-stop '.', while spaces signify vertical grid lines. The rows can also be placed on new lines to provide a visual representation of the grid. For tablets, we can place the elements in the grid to match our designs using the following CSS:

```
@supports (display: grid) {
  @media (min-width: 721px) and (max-width: 1000px) {
    #grid {
      grid-template-areas:
        "header header header header"
        "nav nav nav nav"
        "hero hero hero hero"
        "main main main main"
        "gallery gallery gallery gallery"
        "article article sign-up sign-up"
        "footer footer footer footer";
    }
  }
}
```

Tutorials

Build better layouts with CSS Grid

17



17. Define grid for element position on desktops

We will now set the element positions for desktop devices by redefining the 'grid-template-areas' property in another media query, as follows:

```
@supports (display: grid) {  
  @media (min-width: 1001px) {  
    #grid {  
      grid-template-areas:  
        "header header nav nav"  
        "hero hero hero hero"  
        "sign-up main main article"  
        "gallery gallery gallery gallery"  
        "footer footer footer footer";  
    } } }
```

18. Define the nested grid in the gallery

The final task will be to define the nested grid inside the gallery to show one, two and four images per line on mobile, tablet and desktop respectively.

Although not yet implemented by any browsers at the time of writing, one way of defining a nested grid is to set 'display: subgrid' in the gallery block, which enables the images to use the grid columns of the 'grid' container and act as if the gallery block is not present. One workaround for the current lack of support for 'subgrid' would be to use 'display: contents', which also enables an element's children to appear as if they were direct children of that element's parent. According to [caniuse.com](#), however, current support for this appears to be very limited.

For now, we will try to target as many browsers as possible and, therefore, define 'gallery' as a grid container. In this case, we will change the column definitions for the different devices instead of redefining the 'grid-template-areas' as we have done for 'grid'. Also, we will not set the position of each image and instead rely on CSS Grid's default implicit behaviour that will make each image occupy a single grid cell regardless of how many

18



there are. Using the following CSS – as well as a gallery containing four images – we can see that we get the desired layouts:

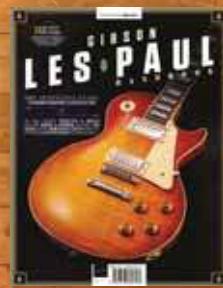
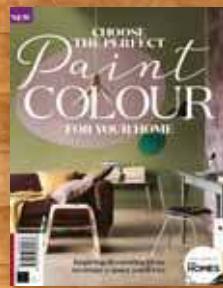
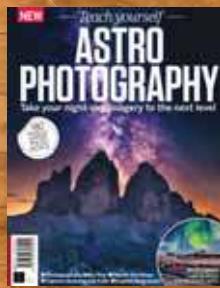
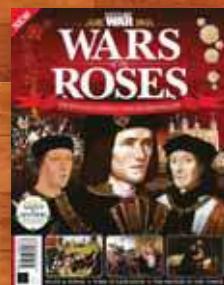
```
@supports (display: grid) {  
  @media (min-width: 721px) {  
    #gallery {  
      display: grid;  
      grid-template-columns: repeat(2,1fr);  
    }  
  }  
  @media (min-width: 1001px) {  
    #gallery {  
      grid-template-columns: repeat(4,1fr);  
    }  
  } }
```

19



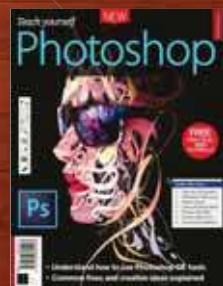
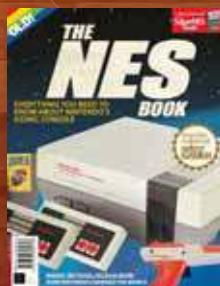
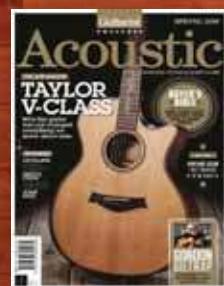
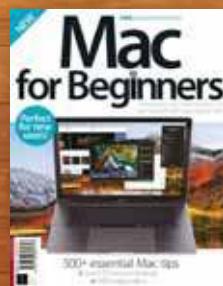
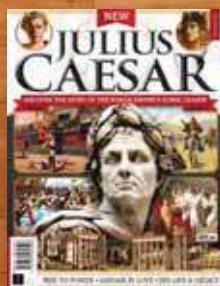
19. Altogether now

Putting together everything we have covered in the previous steps, we can see that with the help of CSS Grid we have created a fully responsive, dynamic design composed of three distinct layouts, all using the same HTML markup and without the need for JavaScript. We've also created a gallery with a nested grid for its images, and implemented a hidden nav bar on mobile, activated using buttons and a few lines of JavaScript. We even managed to add a fallback layout for browsers that don't support CSS Grid. Not bad for seven lines of JavaScript, less than 40 lines of HTML, and 160 lines of CSS!



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.

knowledgeable staff available 24/7 to provide you with assistance when you need it most. Our people make sure you are happy and your problems are resolved as quickly as possible.

2. Secure and dependable

Netcetera prides itself on offering its clients a secure environment. It is accredited with ISO 27001 for security along with the options of configurable secure rackspace available in various configurations.

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
0845 5279 345
Cyber Host Pro are committed to providing the best cloud server hosting in the UK; they are obsessed with automation. If you're looking for a hosting provider who will provide you with the quality you need to help your business grow, then look no further than Cyber Host Pro.

- Cloud VPS servers
- Reseller hosting
- Dedicated servers

Cluster web hosting



fasthosts.co.uk
0808 1686 777
UK-based and operating 24/7 from dedicated UK data centres. Fasthosts keep over one million domains running smoothly and safely each day. Services can be self-managed through the Fasthosts Control Panel.

- Dedicated servers
- Cloud servers
- Hosted email



Budget hosting



hetzner.com
+49 (0)9831 505-0

Hetzner Online is a professional web hosting provider and experienced data centre operator. Since 1997, the company has provided private and business clients

with high-performance hosting products as well as the infrastructure for the efficient operation of sites. A combination of stable technology, attractive pricing, flexible support and services has enabled Hetzner Online to strengthen its market position nationally and internationally.

- Dedicated/shared hosting
- Colocation racks
- SSL certificates



All-inclusive hosting



1and1.co.uk
0333 336 5509

1&1 Internet is a leading hosting provider that enables businesses, developers and IT pros to succeed online. Established in 1988, 1&1 now

operates across ten countries. With a comprehensive range of high-performance and affordable products, 1&1 offers everything from simple domain registration to award-winning website building tools, eCommerce packages and powerful cloud servers.

- Easy domain registration
- Professional eShops
- High-performance servers

SSD web hosting



bargainhost.co.uk
0843 289 2681

Since 2001, Bargain Host have campaigned to offer the lowest possible priced hosting in the UK. They have achieved this goal successfully and built up a large client database, which includes many repeat customers. They have also won several awards for providing an outstanding hosting service.

- Shared hosting
- Cloud servers
- Domain names

Value Linux hosting



patchman-hosting.co.uk
01642 424 237

Linux hosting is a great solution for home users, business users and web designers looking for cost-effective and powerful hosting. Whether you are building a single-page portfolio, or you are running a database-driven eCommerce website, there is a Linux hosting solution for you.

- Student hosting deals
- Site designer
- Domain names

Flexible cloud servers



elastichosts.co.uk
020 7183 8250

ElasticHosts offer simple, flexible and cost-effective cloud services with high performance, availability and scalability for businesses worldwide. Their team of engineers provide excellent support 24/7 over the phone, 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.

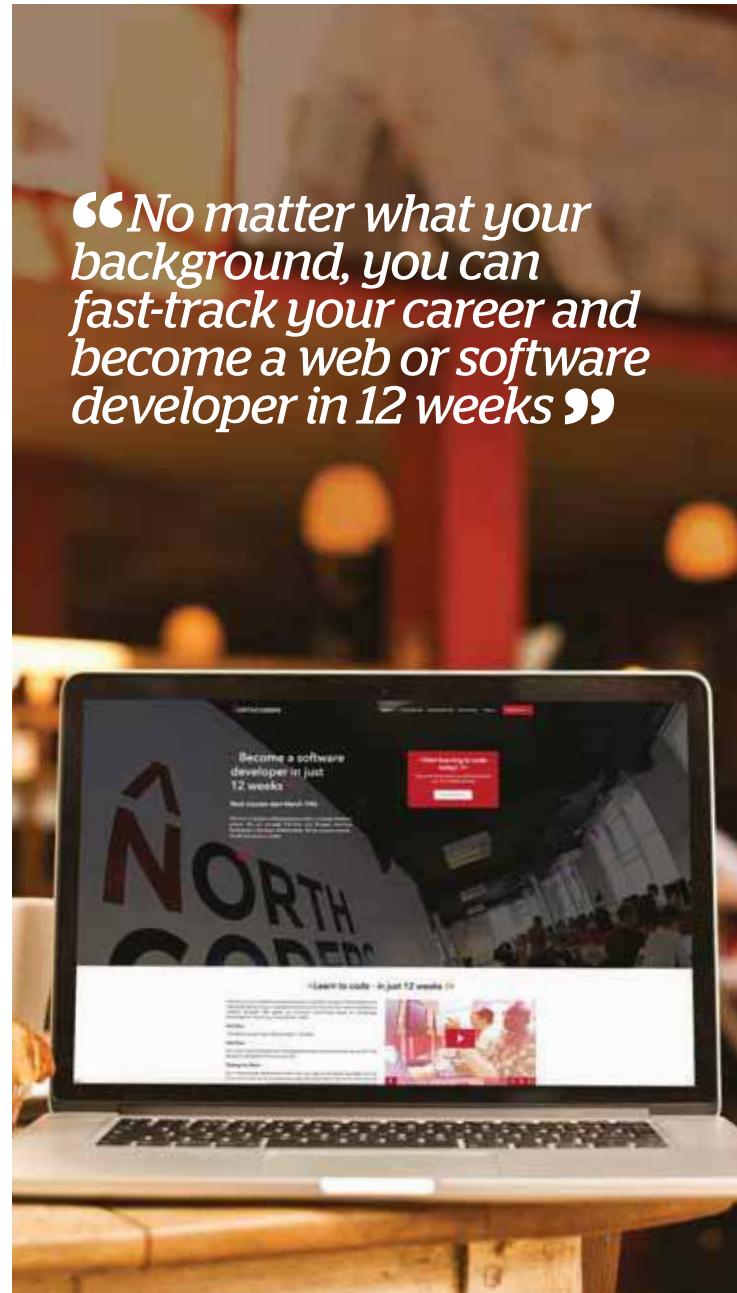
3. Throw yourself in

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](#)

Facebook: [udemy](#)

The inspiration for Udemy began in a small village in Turkey, where founder Eren Bali grew up frustrated by the limitations of being taught in a one-room school house. Realising the potential of learning on the internet he set out to make quality education more accessible. Udemy is now a global marketplace for learning and teaching online. Students can master new skills by choosing from an extensive library of over 40,000 courses including HTML, CSS, UX, JavaScript and web development.

40,000+ courses: There is a course for every designer and dev.

Self-paced learning: Learn how to code at your own pace.



THE IRON YARD

theironyard.com

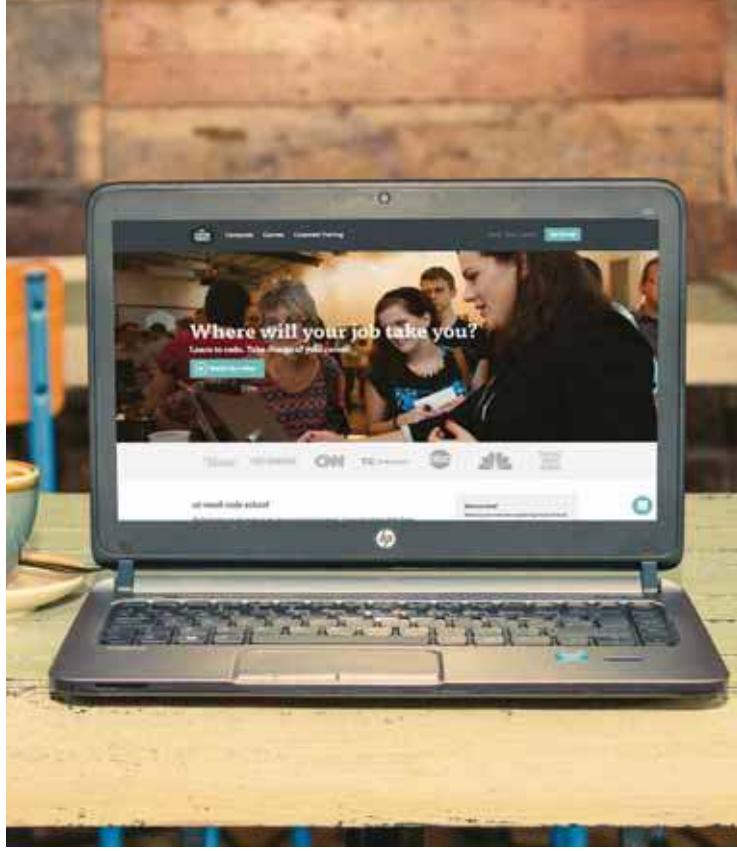
Twitter: [@TheIronYard](#)

Facebook: [TheIronYard](#)

The Iron Yard is one of the world's largest and fastest-growing in-person code schools. It offers full-time and part-time programs in 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



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

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 & PHP



Tutorial project files

All the assets you'll need
to follow our tutorials



Plus, all of this is yours too...

- All-new tutorial files to help you master this issue's HTML, CSS and JavaScript techniques
- Two more chapters from the Beginner's JavaScript video series from Killersites (shop.killervideostore.com)
- 21 Pixel Art Photoshop actions and 400+ Isometric Alphabet Tiles 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 image shows a yellow 'Web Designer' magazine cover on the left, featuring headlines like 'FREE 57 MINUTES OF JAVASCRIPT VIDEOS', 'TOP TYPE', and 'JAVASCRIPT WHAT'S NEW?'. To the right is a laptop displaying the 'FileSilo.co.uk' website, specifically the 'Web Designer' section. A white mug containing coffee is positioned between the magazine and the laptop. The overall theme is digital content creation and web design.



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

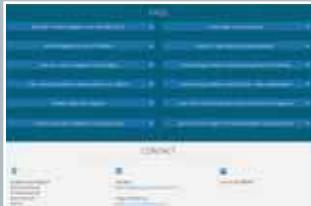
Everything you need to know about accessing your FileSilo account



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

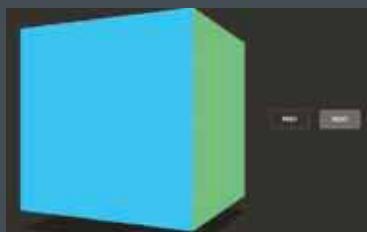
Over 60 hours of video guides

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

WEB ANIMATION

WITH CSS & JS

Get a closer look at the current state of animation on the web, expert insight and practical techniques to improve your projects

THE POWER OF VARIABLE FONTS

Discover what they are, where to get them, and how to use them in your projects

GET STARTED WITH THREE.JS – PT4

In the fourth part of the series, learn how to interact with objects in 3D space

A CLOSER LOOK AT NODEJS 10

What does the latest update have to offer? Find out how to use the most recent additions

Visit the **WEB DESIGNER** online shop at



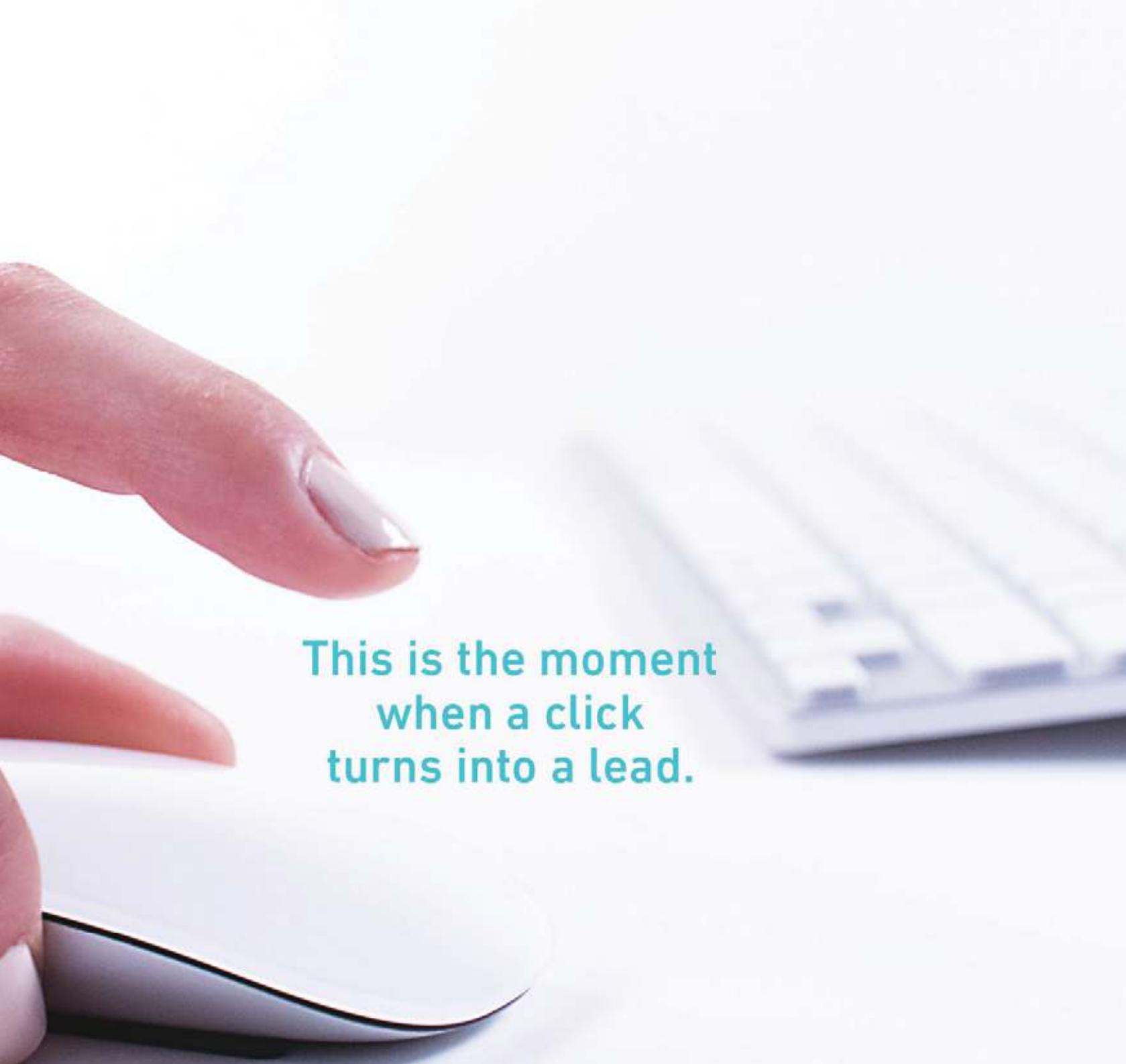
myfavouritemagazines

myfavouritemagazines.co.uk

for the latest issue, back issues and specials

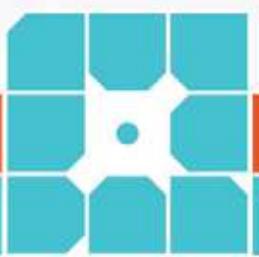
ALL IN YOUR NEXT
WEB DESIGNER
Issue 279 on sale
Tuesday 18th September 2018

SUBSCRIBE TODAY Go to [page 32](#) to learn more



This is the moment
when a click
turns into a lead.

PRESS AHEAD



WP Engine's digital experience platform drives your business forward faster. wpengine.co.uk

WP engine*

STORM

FASTER EVERYTHING

Introducing **STORM**, a hosting platform that helps your agency and websites run smoother. 80% of our customers have seen a 25% improvement in site speeds.

SUPER CHARGE YOUR HOSTING

Call, email or visit us at

0203 308 2886

sales@nimbushosting.co.uk

nim.host/storm

 **Nimbus Hosting**