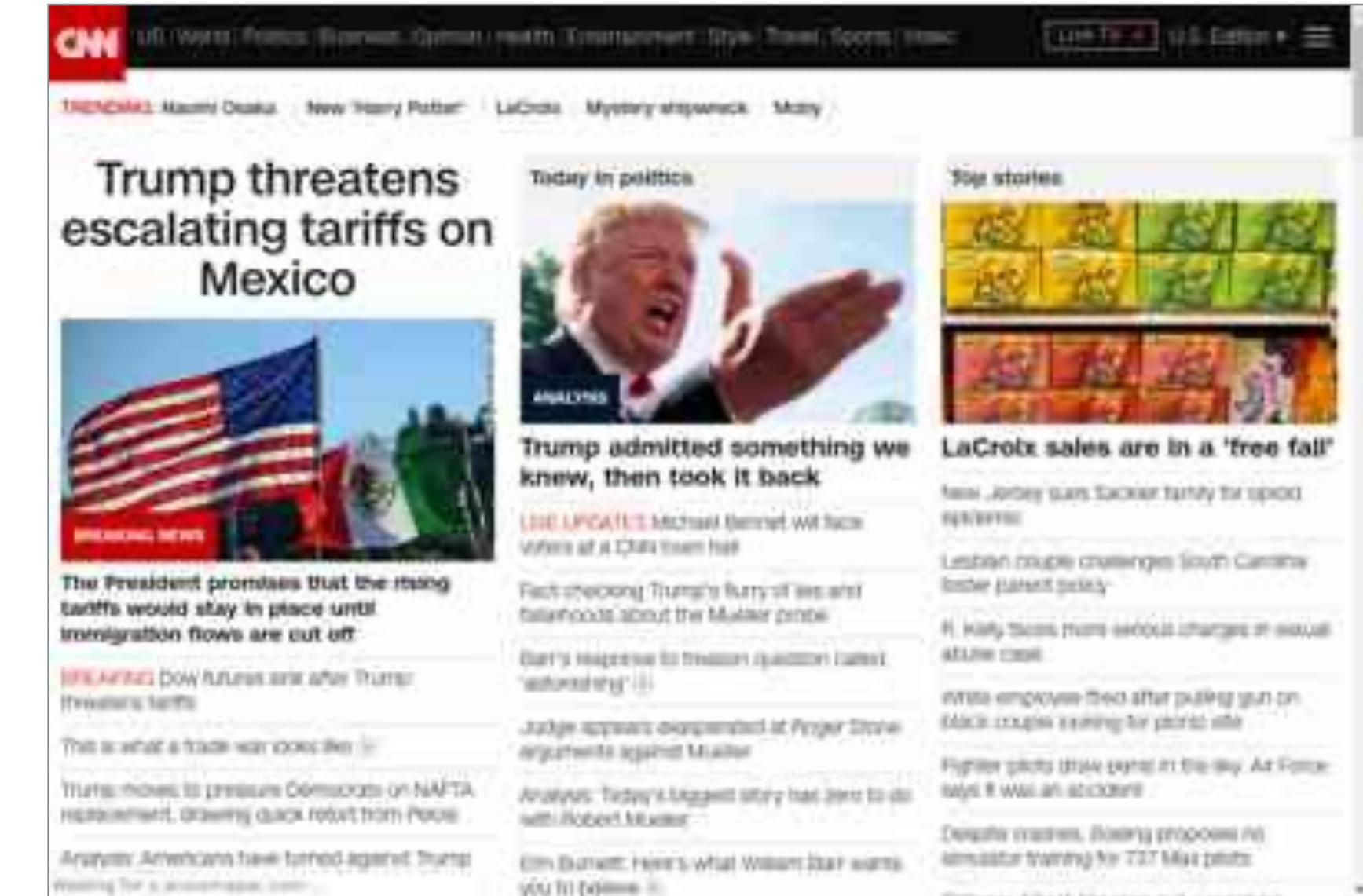
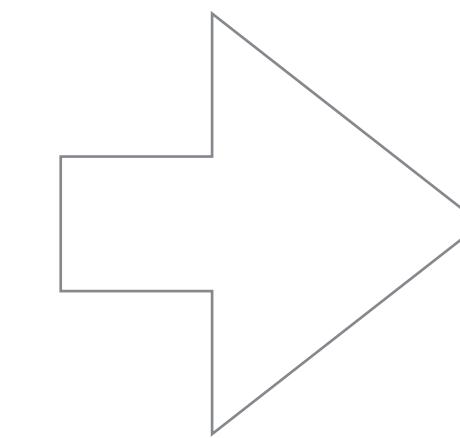
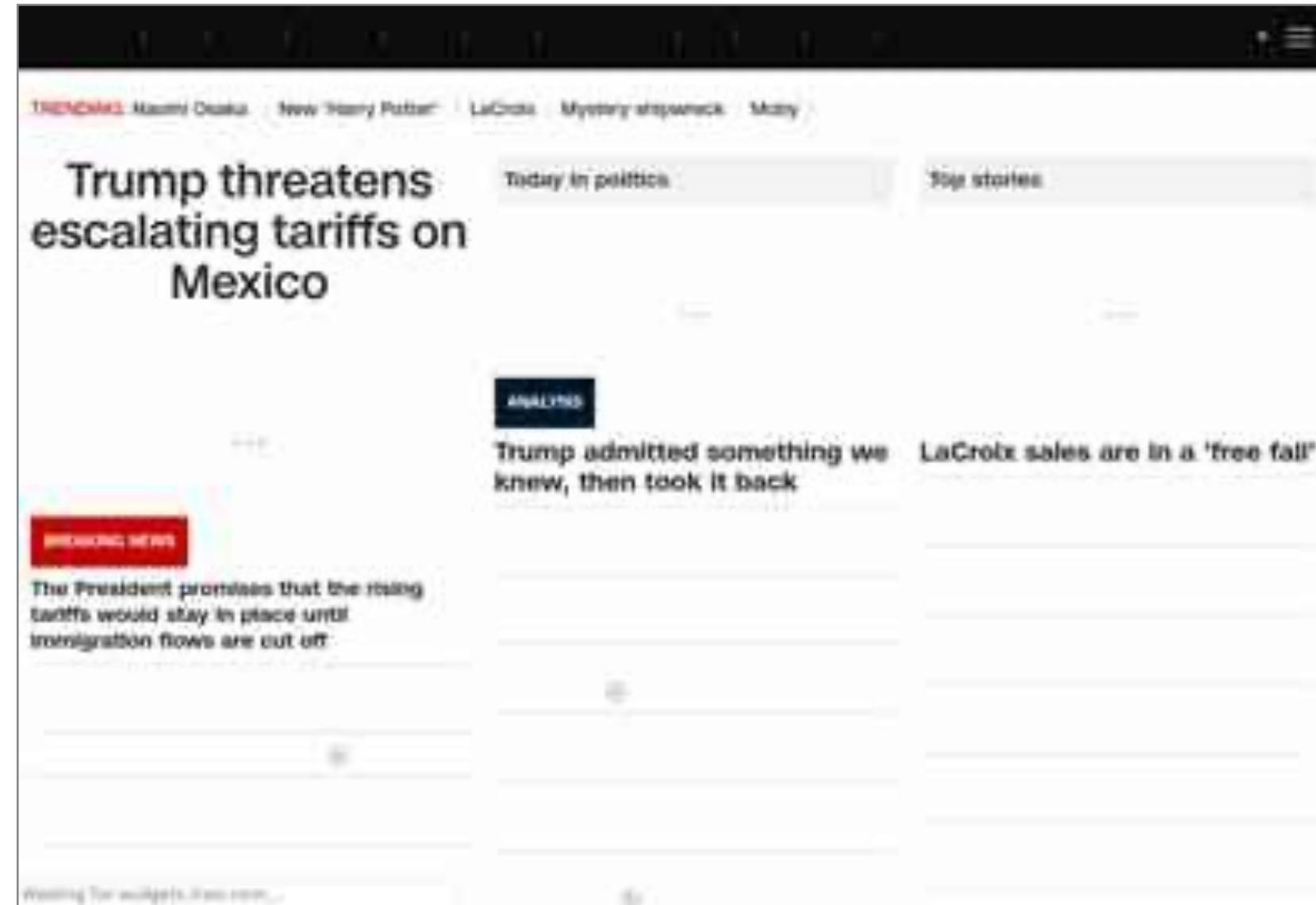


LCP

# Largest Contentful Paint

---

The render time of the largest content element  
visible in the viewport.



First Contentful Paint

~Largest~ Contentful Paint

# LCP players

- (pre)fetching important assets early
- Images - sizing and targeted appropriately
- Fonts: progressive loading strategies
- JS: enhancing content with little jank
- 3rd party JS: loading without blocking content

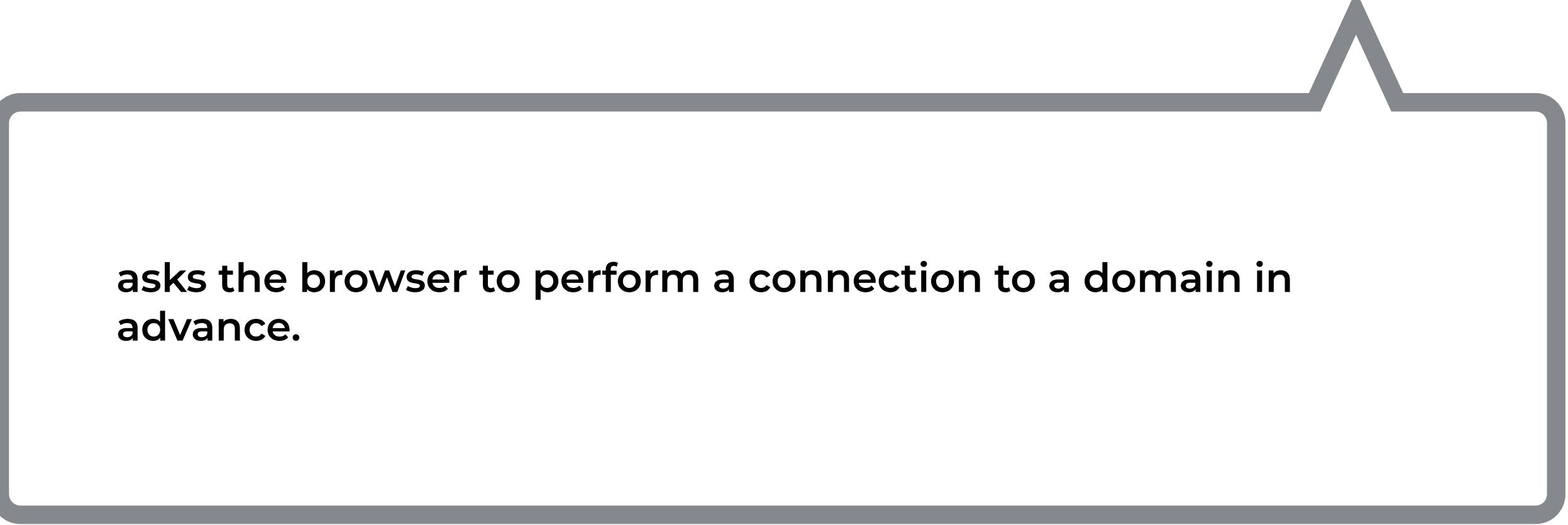
Fetching things sooner...

# The many pre's

---

# Preconnect

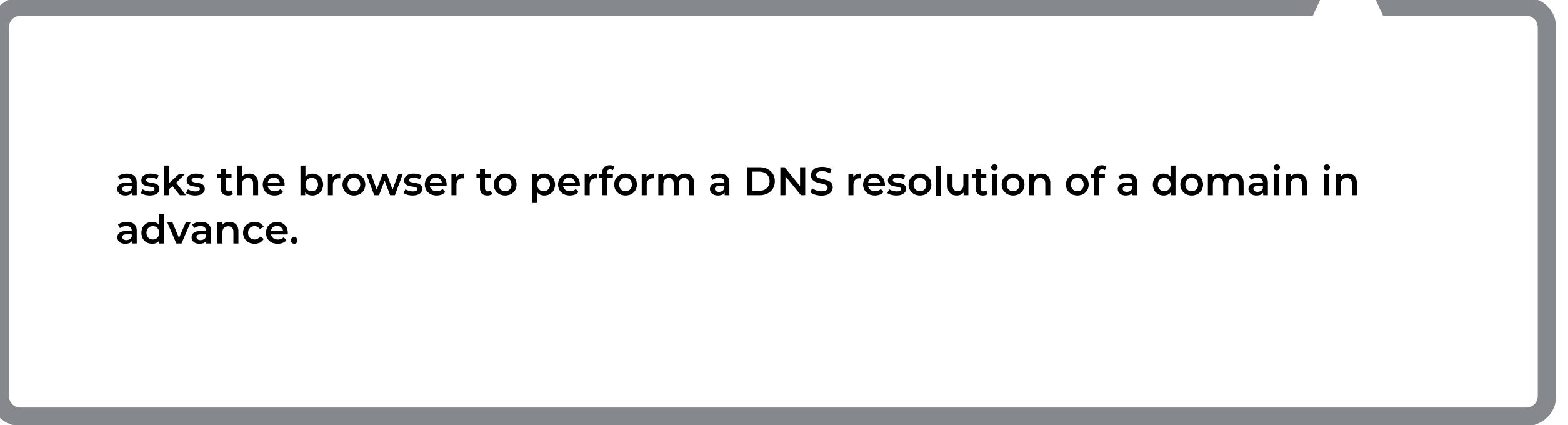
```
<link rel="preconnect" href="https://example.com" />
```



asks the browser to perform a connection to a domain in advance.

# DNS Prefetch

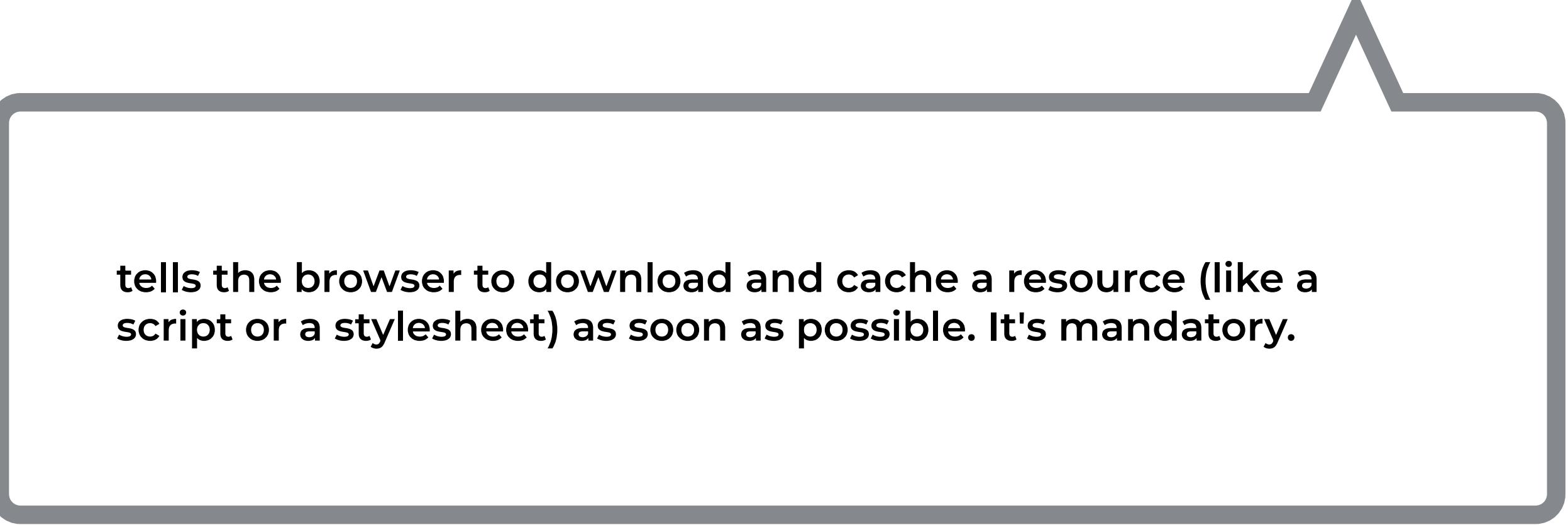
```
<link rel="dns-prefetch" href="https://example.com" />
```



asks the browser to perform a DNS resolution of a domain in advance.

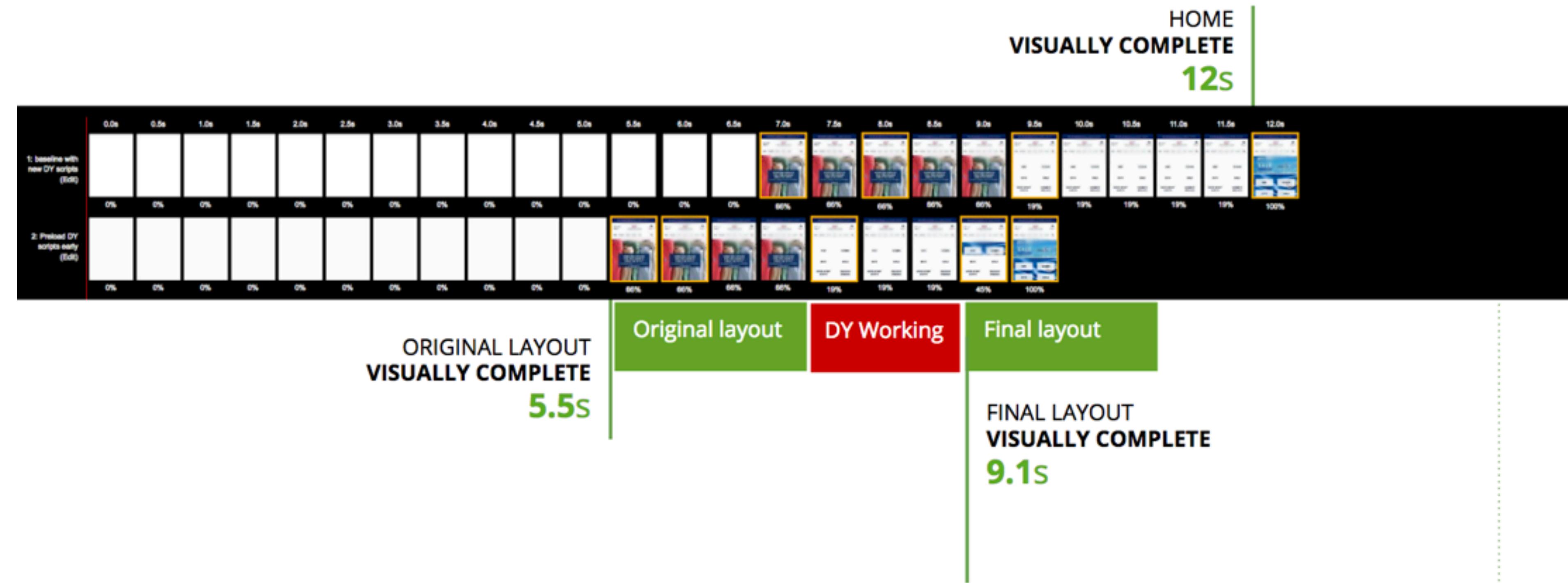
# Preload

```
<link rel="preload" href="/scripts.js" as="script" />
```



tells the browser to download and cache a resource (like a script or a stylesheet) as soon as possible. It's mandatory.

# Potential impact of preloading



# Possible "as" values

**audio**: Audio file, as typically used in <audio>.

**document**: An HTML document intended to be embedded by a <frame> or <iframe>.

**embed**: A resource to be embedded inside an <embed> element.

**fetch**: Resource to be accessed by a fetch or XHR request, such as an ArrayBuffer or JSON file.

**font**: Font file.

**image**: Image file.

**object**: A resource to be embedded inside an <object> element.

**script**: JavaScript file.

**style**: CSS stylesheet.

**track**: WebVTT file.

**worker**: A JavaScript web worker or shared worker.

**video**: Video file, as typically used in <video>.

# Delivering **Images**

Images generally load unobtrusively

```
>
```

# Data URIs: often an antipattern

```
>
```

Don't just optimize images,  
**make them responsive**

# Responsive images

```

```

```
<picture>  
  <source srcset="large.png" media="(min-width: 800px)">  
  <source srcset="medium.jpg" media="(min-width: 400px)">  
    

```

# The srcset syntax

Sml.jpg is 300px wide, and  
lrg.jpg is 800px wide

```
>
```

# The srcset syntax

```

```

This image will be 100% viewport width up until 500px  
viewport size, then it is 50% viewport width after that.

# Swapping Images with the Sizes Attribute

---

POSTED BY  
**SCOTT JEHL**  
03/08/2018

Over the years we've designed components that enable users to magnify, swipe, and pan around images in a wide variety of ways. Early on, it was tricky to build these components in ways that used image sizes responsibly, and we'd sometimes end up delivering larger images than we'd prefer—ensuring that the image would at least be sharp across all devices. Thankfully, as “responsive” image standards have gained support we've been able to offload much of that source selection logic to the browser, and in that regard, one particular trick has served us really well that we wanted to share.



<https://www.filamentgroup.com/lab/sizes-swap/>

# The Picture element syntax

```
<picture>
  <source srcset="large.png" media="(min-width: 800px)">
  <source srcset="medium.jpg" media="(min-width: 400px)">
  
</picture>
```

The first one of these sources with a matching media query wins!  
If none, just use the img src as-is.



<http://usecases.responsiveimages.org/>

# Also good for image types

```
<picture>
  <source srcset="awesome.webp" type="image/webp">
    
</picture>
```

# Also good for image types

```
<picture>
  <source srcset="awesome.avif" type="image/avif">
    
</picture>
```

Great for art directed imagery

# Best bet: use both!

```
<picture>
  <source srcset="small.jpg 400w, medium.jpg 800w"
          sizes="(max-width: 400px) 100vw">

  <source srcset="medium.jpg 800w, medium.jpg 1200w"
          sizes="(max-width: 800px) 50vw">

  
</picture>
```

# Responsive Image builder

SRC 320PX



SRC 640PX



SRC 1024PX



ALT TEXT



Your HTML:

```
<img src="" srcset=" 320w, 640w,  
1024w" alt="">
```



Vue.js

Loading SVGs  
**Performantly**

# External SVG... so many ways



or... <svg width="100" height="100">  
      <use href="skate.svg"></use>  
  </svg>

or... <object data="skate.svg"></object>

or... background-image: url("skate.svg");



Inspector

Console

Debugger

Style Editor

Performance

Memory

»

Search HTML

```
> <p class="note">...</p>
> <p>...</p>
> <p>...</p>
> <h2 id="a-quick-demo%3A-including-svg">...</h2>
> <p>...</p>
-<svg xmlns="http://www.w3.org/2000/svg" width="100" height="100" viewBox="0 0 100 100">
  <path fill="#324D5B" d="M50 0c27.6 0 50 22.4 50 50s-22.4 50-50 50S0 77.6 0 50 22.4 0 50
  0z"></path>
  <path fill="#475F6C" d="M95.2 71.5c-12-5.3-27.6-8.5-44.7-8.5C33 63 17.1 66.3 5 71.8 13.1
  88.5 30.2 100 50 100c19.9 0 37.1-11.7 45.2-28.5z"></path>
  <path fill="#ccc" d="M70.9 66.8l-17.4-10L63 51l-13-8-13 8 9.8 6-17 9.7c-1 .6-1
  1...9-5.3zm9.3 26.8L33.1 68l16.5-9.3.4.3.8-.5L67.7 68l-17.3 9.8z"></path>
```

# External SVG with control

```
<svg style="--wheelcolor: red;">  
  <use href="icons.svg#skateboard"></use>  
</svg>
```

**Inside icons.svg:**

```
<svg>  
<symbol id="skateboard">  
  <path style="fill: var(--wheelcolor, blue);">
```

# *frontstuff*

Front-end web developer  
with a taste for design and typography.



January 29, 2018

## **Multi-Colored SVG Symbol Icons with CSS Variables**

Long gone are the days of using images and CSS sprites to make icons for the web. With the explosion of web fonts, icon fonts have become the number one solution for displaying icons in your web projects.



<https://frontstuff.io/multi-colored-svg-symbol-icons-with-css-variables>

Preloading RWD  
Images?

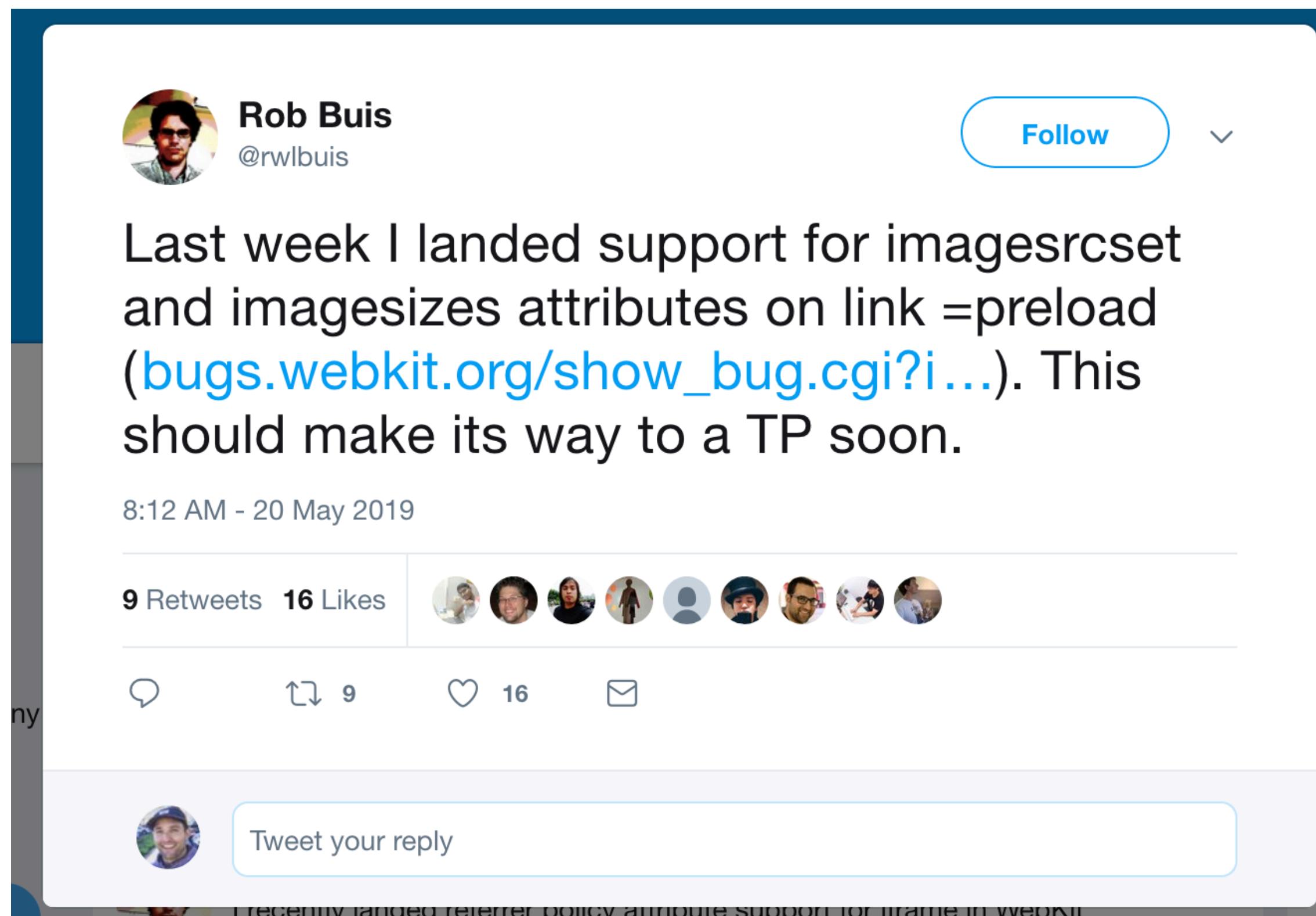
# Preload a <picture> hero image

```
<link rel="preload" href="small.jpg" as="image"  
media="(max-width: 399px)">  
<link rel="preload" href="medium.jpg" as="image"  
media="(min-width: 400px) and (max-width: 799px)">  
<link rel="preload" href="large.jpg" as="image"  
media="(min-width: 800px)">  
  
<picture>  
  <source srcset="large.jpg" media="(min-width: 800px)">  
  <source srcset="medium.jpg" media="(min-width: 400px)">  
      
</picture>
```

# Preload an image[srcset]...?

```

```



# How that would look...

```
<link rel="preload" imagesrcset="sml.jpg 300w, lrg.jpg  
800w" imagesizes="100vw" as="image">
```

```

```

Most images are  
***not that important...***

Previous

BUY ONLINE PICK UP IN STORE [Learn More](#)



POTTERY BARN

PB/APARTMENT

POTTERY BARN KIDS

PBTEEN

Williams Sonoma

WS Home

west elm

Rejuvenation

Mark and Graham

| SHIP TO:

[SIGN IN](#) [KEY REWARDS](#) [FIND A STORE](#) [TRACK YOUR ORDER](#)

SEARCH



P O T T E R Y B A R N

NEW

SHOP BY STYLE

SHOP ROOMS

DESIGN SERVICES

ROOM PLANNER

REGISTRY



CART

Furniture

Outdoor

Bedding

Bath

Rugs

Windows

Lighting

Pillows & Decor

Tabletop & Bar

Organization

**Sale** | *Holiday*

*Gifts*

MEGA SALE  
IN STORES & ONLINE

UP TO 70% OFF + FREE SHIPPING\*

[SHOP NOW](#)

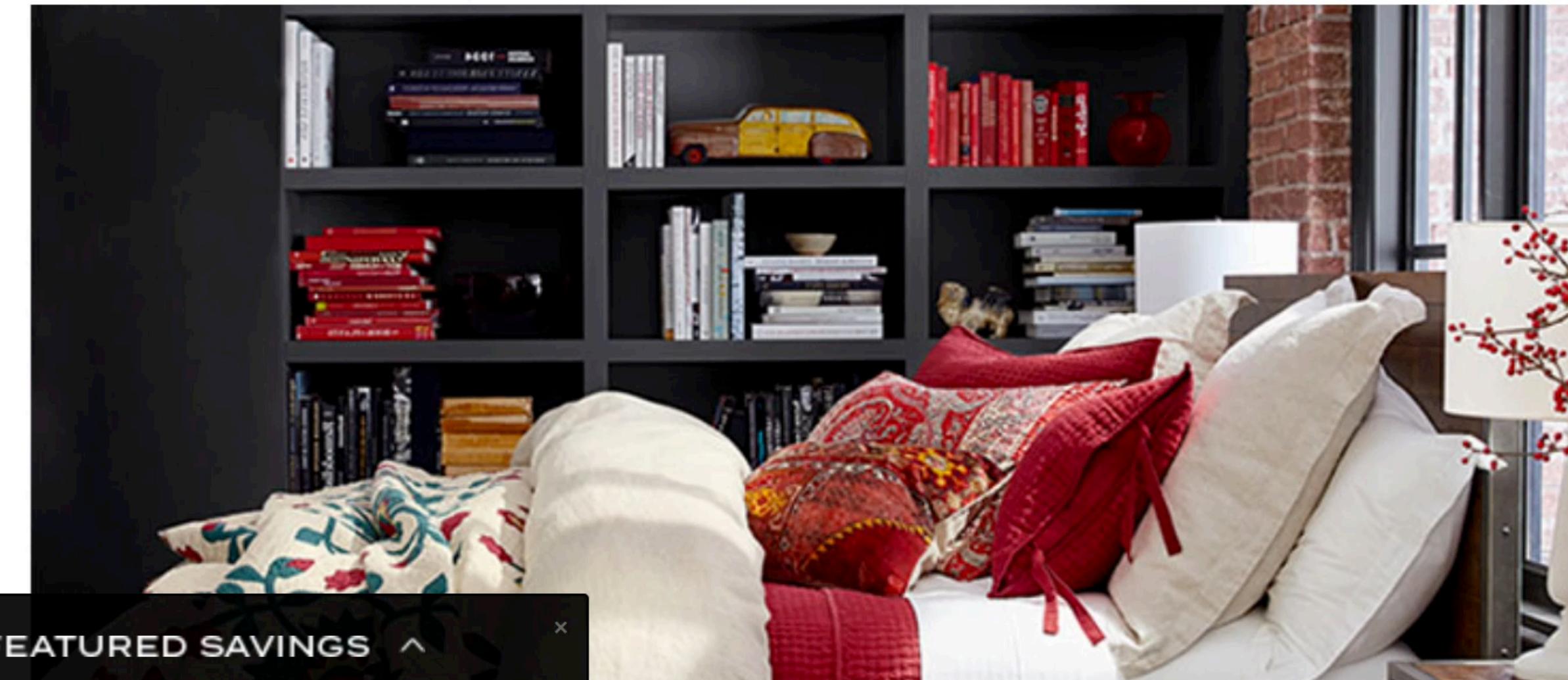
USE CODE: TREAT

\*Exclusions Apply

RED WITH  
A TWIST

Mix statement-making red with dramatic black and rich brown for a sophisticated take on a festive color.

[SHOP OUR FAVORITES](#)



# NEWS

[Home](#) | [Video](#) | [World](#) | [US & Canada](#) | [UK](#) | [Business](#) | [Tech](#) | [Science](#) | [Stories](#) | [Entertainment & Arts](#) | [Health](#) | More ▾

## Saudi state blamed for Khashoggi murder

Turkey's president accuses the Saudi government directly of murdering journalist Jamal Khashoggi.

⌚ 6h | Middle East

- Is the Saudi crown prince finished?



- Khashoggi's fiancee on his death ► Khashoggi's impact on Yemen's war



### US to reinstate all Iran sanctions

The White House describes it as "the toughest sanctions regime ever imposed on Iran".

⌚ 8h | US & Canada



Pakistan deal to end blasphemy case unrest



New Caledonia set for independence vote



Clocks go back: Why do we do it?

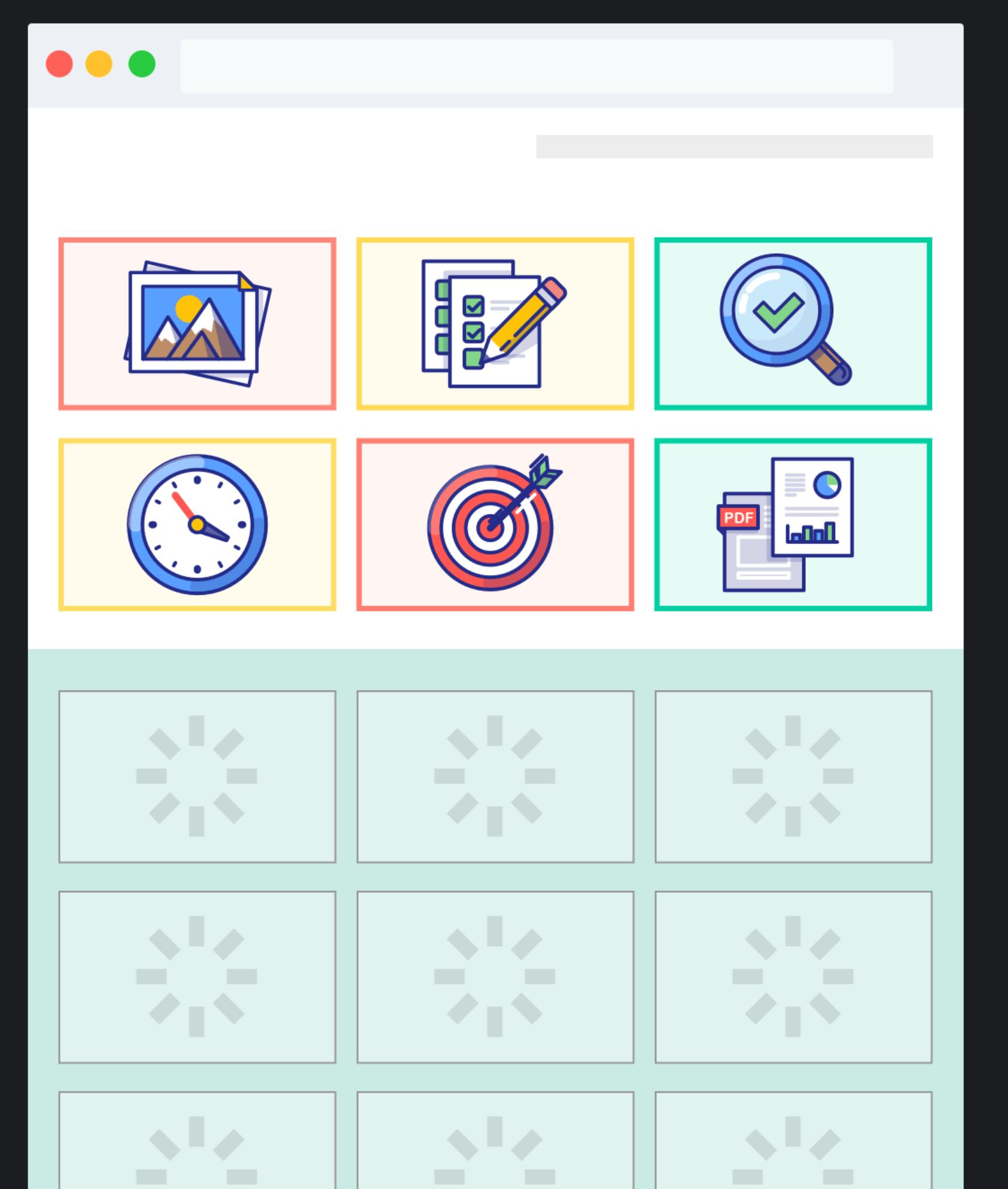


Dissident's HK show axed over 'threats'



# THE `LOADING` ATTRIBUTE

```
<img loading=lazy>  
<iframe loading=lazy>
```



# So Lazy!

```

```

```
<iframe src="ads.html" loading="lazy"></iframe>
```

Loading Video  
**Performantly**

# Video - like picture syntax!

```
<video controls playsinline poster="somevideo.jpg"  
preload="none">  
  <source src="somevideo.webm" type="video/webm">  
  <source src="somevideo.mp4" type="video/mp4">  
</video>
```

# Video - make it responsive too.

```
<video controls playsinline poster="somevideo.jpg">
  <source src="somevideo-sml.webm" media="(max-width: 400px)"
  type="video/webm">
  <source src="somevideo-sml.mp4" media="(max-width: 400px)"
  type="video/mp4">
  <source src="somevideo-lrg.webm" type="video/webm">
  <source src="somevideo-lrg.mp4" type="video/mp4">
</video>
```

# HTML Video Sources Should Be Responsive

---

POSTED BY  
**SCOTT JEHL**  
01/12/2021

Recently, I was adding a video to a website. Intuitively, I started to mark up the HTML `<video>`'s `<source>` elements with `media` attributes to specify the viewport sizes that each of my video source files would best serve. As I was working, I vaguely recalled discussions years ago that made me question: "wait, is media support available for video?" A quick check over at MDN confirmed that indeed it was... NOT. Support was removed from browsers and the spec despite having no HTML-based alternatives to take its place (thankfully, [Webkit browsers still support it](#), so at least *some* browser support does exist, but WebKit alone is not enough).

Removing media support from HTML video was a mistake. It means that for every video we embed in HTML, we're stuck with the choice of serving source

# Font rendering



## Diagnostics — More information about the performance of your application.

### ▲ Ensure text remains visible during webfont load

Leverage the font-display CSS feature to ensure text is user-visible while webfonts are loading.

[Learn more.](#)

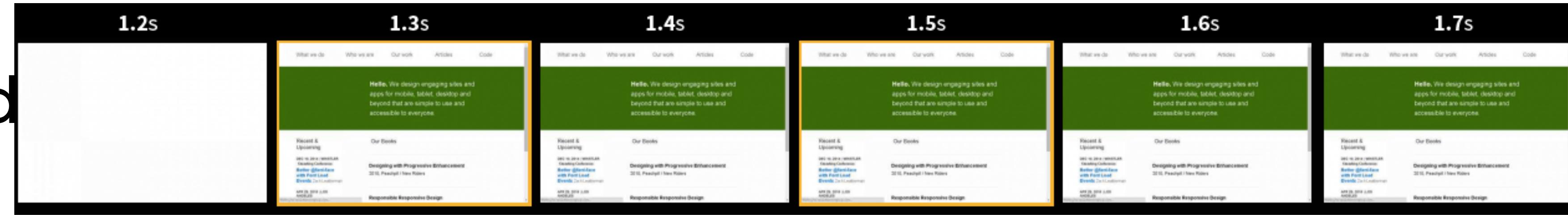
URL	Potential Savings
...type/Lato-Light-s1.woff2 (203-origin-fgwebsite.fgview.com)	120 ms
...type/Lato-Regular-s1.woff2 (203-origin-fgwebsite.fgview.com)	120 ms
...type/Lato-Bold-s1.woff2 (203-origin-fgwebsite.fgview.com)	120 ms
...type/Lato-Black-s1.woff2 (203-origin-fgwebsite.fgview.com)	120 ms

# Progressive font rendering

Standard



Optimized



# No More JS: CSS font-display

```
@font-face {  
    font-family: "Nice Serif";  
    src: url(niceserif.woff2);  
    font-display: swap;  
}  
  
h1 {  
    font-family: "Nice Serif", "Georgia", serif;  
}
```

 **Anna Migas**  
@szynszyliszys

Following ▾

You will not have to self-host Google Fonts any more to get font-display: swap; 🙌 #io19



© 2019 Twitter About Privacy policy



**S**  
Follows you  
-end Dev at @Lunar\_Logic.  
ert. Speaker. Proud member



<https://twitter.com/szynszyliszys/status/1126261253280649216>

# Can I use `font-display` ?

1 result found

## CSS font-rendering controls

Usage % of all users Global 87.68%

@font-face descriptor (currently defined as `font-display`) that allows control over how a downloadable font renders before it is fully loaded.

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Opera Mobile	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ
		2-45	4-48		10-35								4	
	12-18	2 46-57	1 49-59	3.1-11	1 36-46	3.2-11.2							1 5-7.4	
6-10	79-80	58-74	60-80	11.1-13	47-67	11.3-13.3		2.1-4.4.4	12-12.1				8.2-10.1	
11	81	75	81	13.1	68	13.4	all	81	1 46	81	68	12.12	11.1	1
	76-77	83-85	TP											



# Nearby



Rai Eingpu



3 reviews

🐎 0.7 mi

THB 396

# Should I Use JavaScript to Load My Web Fonts?

Posted by [Zach](#) 11/29/2018

When they were first introduced, web fonts were primarily a CSS feature, and for many web developers they've always loaded them using CSS (and nothing more). But in the last decade, default web font loading behavior in many browsers made CSS-only methods a gamble with our page's text render, and thoughtful developers switched to safer JavaScript methods. Recently, browser support for new and safer CSS-only strategies have left some developers wondering: are JavaScript methods to load web fonts necessary? Are they useful? Let's dig in.

## The Epochs of Responsible Web Font Loading

### 1997: CSS-only web fonts were good

Internet Explorer was the only browser that supported web fonts and did so in a



<https://www.filamentgroup.com/lab/js-web-fonts.html>

# Grouped Repaints

```
var font = new FontFace( ... );
var fontBold = new FontFace( ... );
Promise.all([
    font.load(),
    fontBold.load()
]).then(function.loadedFonts) {
    loadedFonts.forEach(function(font) {
        document.fonts.add(font);
    });
});
```

}

# Adapt to User Preferences

```
function loadFonts() {  
    /* NOTE: Reuse the Group Repaints code snippet above,  
here */  
}  
  
if( navigator.connection &&  
    navigator.connection.saveData ) {  
} else {  
    loadFonts();  
}
```

# Adapt to User Context

```
function loadFonts() {  
    /* NOTE: Reuse the Group Repaints code snippet above,  
here */  
}  
  
if( navigator.connection &&  
    ( navigator.connection.effectiveType === "slow-2g" ||  
      navigator.connection.effectiveType === "2g" ) ) {  
    // do nothing  
} else {  
    loadFonts();  
}
```

# A COMPREHENSIVE GUIDE TO FONT LOADING STRATEGIES

July 12, 2016 [#3 Popular](#)

*This guide is not intended for use with font icons, which have different loading priorities and use cases. Also, SVG is probably a better long term choice.*

Updated July 27, 2017 with new information on [font-display](#).

Updated August 14, 2017 with a link to a [glossary page](#).



ZACH  
LEATHERMAN



[HOME](#) [ARCHIVES](#) [ABOUT](#)

[WEB FONTS](#) [ELEVENTY](#)



[NOTES](#) [SPEAKING](#) [PROJECTS](#)

[RESEARCH](#)

## LATEST

The Crushing Weight of the Facepile

## POPULAR 🔥 ➔

Google Fonts is Adding font-display

32 days

Render Snarky Comments in Comic Sans

3 days

A Comprehensive Guide to Font Loading Strategies

2.9 years



<https://www.zachleat.com/web/comprehensive-webfonts/>

# Identifying Problems

[ABOUT](#)[WORK](#)[BLOG](#)[CODE](#)[HIRE US](#)

## FEATURED PROJECTS



Tables						
#	NAME	PARTNER	PRIORITY	CATEGORY	SLA	LAST UPDATED
1	Primary Identifier	ACME Incorporated	Urgent	Inbound Chat	24/7	16 min
2	Primary Identifier	ACME Incorporated	Medium	Outbound Call	DAY	2 days
3	Primary Identifier	ACME Incorporated	Medium	Inbound Email	EVE	1 month

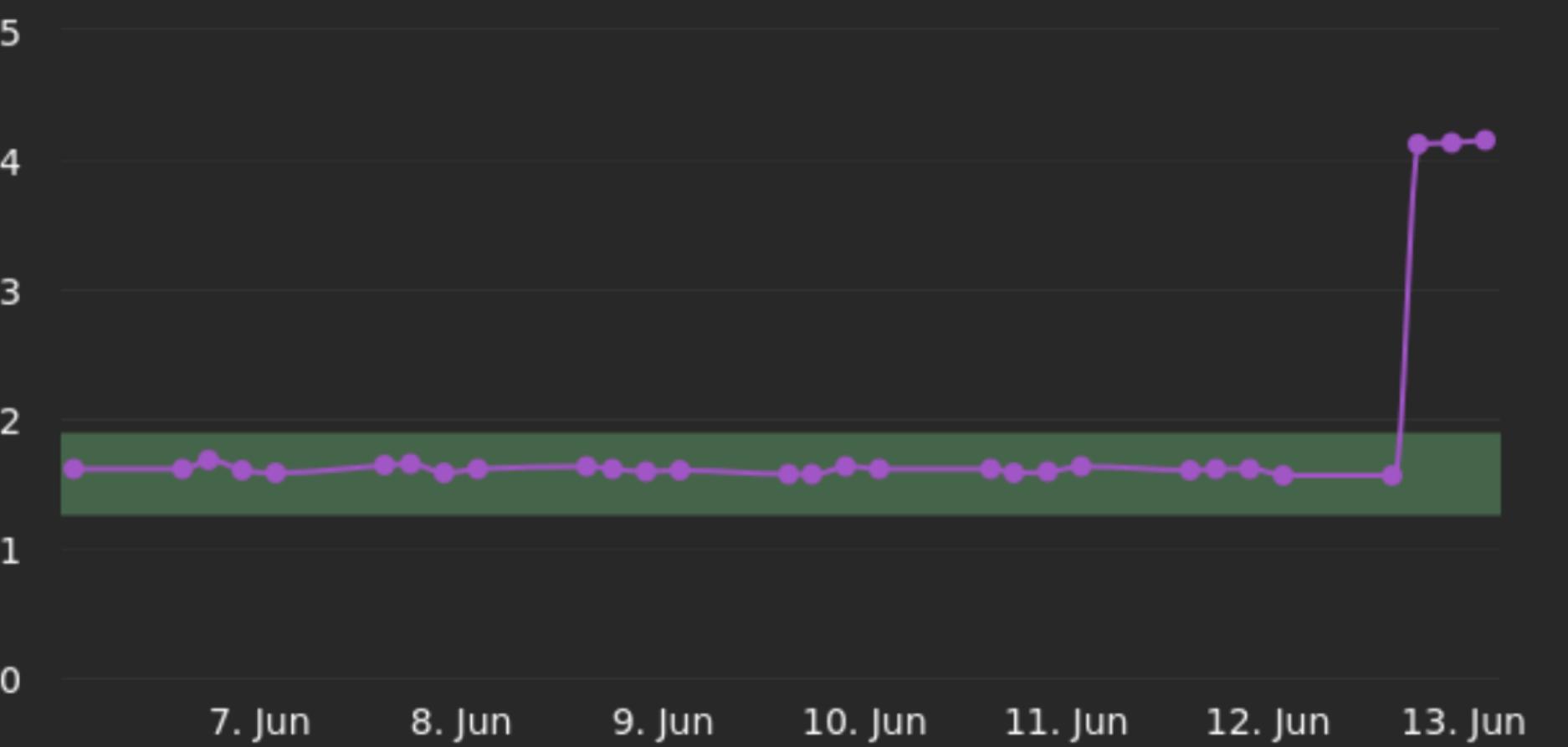


<https://filamentgroup.com>

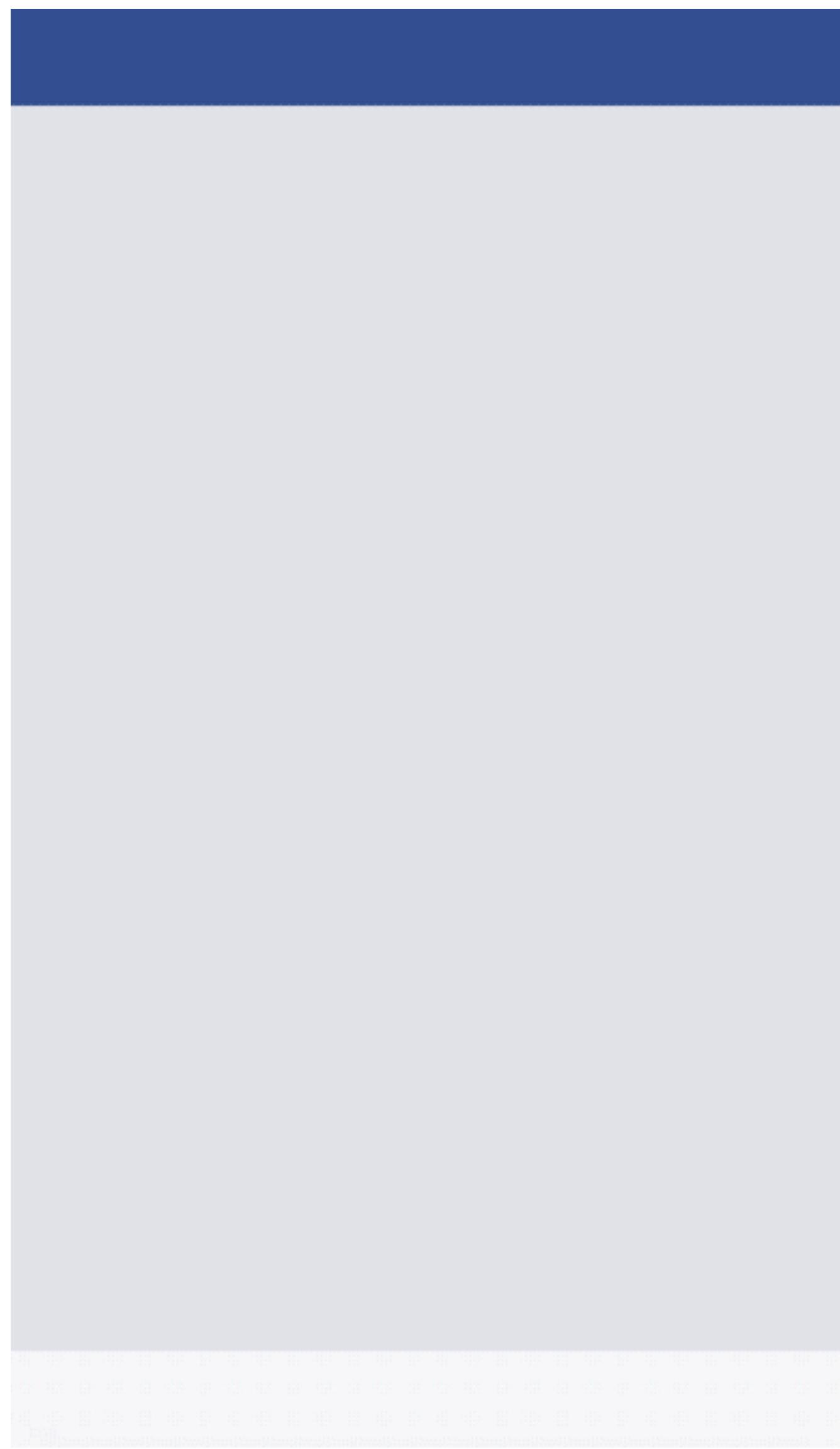


**Last Painted Hero for Filamentgroup has increased by 163%**

Last Painted Hero (Syn), Filamentgroup has increased by 163% and is now 4.15 seconds, crossing over the 20% rate of change threshold.



Over-reliance on JavaScript  
**for Content**



# Empty Body = Fragile/Slow!

```
<body>
  <div id="app"></div>
</body>
```

```
<script>
  createApp( "#app" , data );
</script>
```

```
import express from 'express';
import fs from 'fs';
import path from 'path';
import React from 'react';
import ReactDOMServer from 'react-dom/server';
import Hello from './Hello.js';

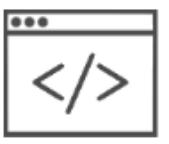
function handleRender(req, res) {
  // Renders our Hello component into an HTML string
  const html = ReactDOMServer.renderToString(<Hello />);

  // Load contents of index.html
  fs.readFile('./index.html', 'utf8', function (err, data) {
    if (err) throw err;

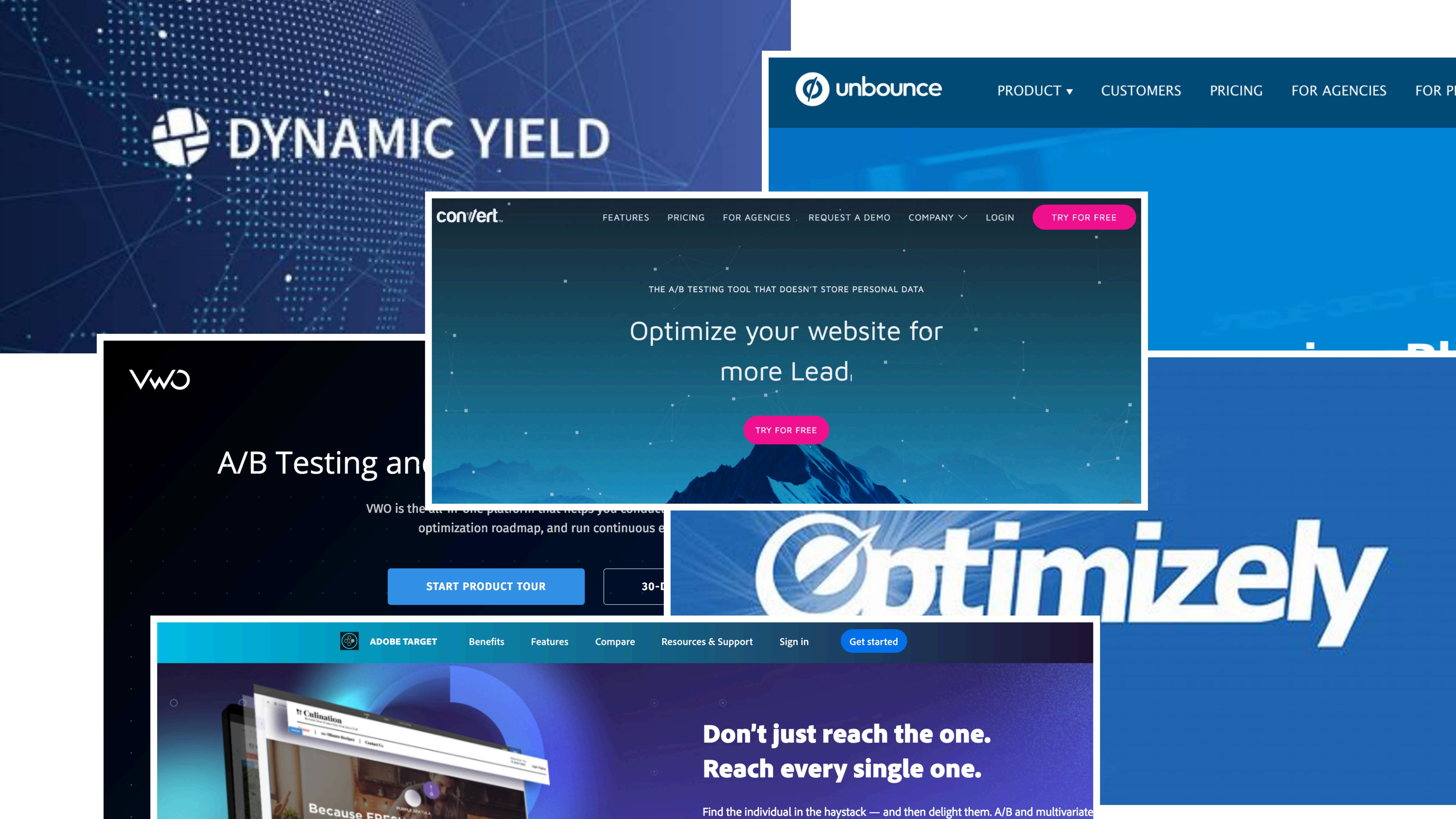
    // Inserts the rendered React HTML into our main div
    const document = data.replace(<div id="app"></div>, `<div id="app">${html}</div>`);

    // Sends the response back to the client
    res.send(document);
  });
}
```



		Server Rendering	“Static SSR”	SSR with (Re)hydration	CSR with Prerendering		Full CSR
Overview:	An application where input is navigation requests and the output is HTML in response to them.	Built as a Single Page App, but all pages prerendered to static HTML as a build step, and the JS is <b>removed</b> .	Built as a Single Page App. The server prerenders pages, but the full app is also booted on the client.	A Single Page App, where the initial shell/skeleton is prerendered to static HTML at build time.	A Single Page App. All logic, rendering and booting is done on the client. HTML is essentially just script & style tags.		
Authoring:	Entirely server-side <small>(request-response, HTML)</small>	Built as if client-side <small>(components, DOM*, fetch)</small>	Built as client-side	Client-side	Client-side		
Rendering:	Dynamic HTML	Static HTML	Dynamic HTML <b>and</b> JS/DOM	Partial static HTML, then JS/DOM	Entirely JS/DOM		
Server role:	Controls all aspects. <small>(thin client)</small>	Delivers static HTML	Renders pages <small>(navigation requests)</small>	Delivers static HTML	Delivers static HTML		
Pros:	👍 TTI = FCP 👍 Fully streaming	👍 Fast TTFB 👍 TTI = FCP 👍 Fully streaming	👍 Flexible	👍 Flexible 👍 Fast TTFB	👍 Flexible 👍 Fast TTFB		
Cons:	👎 Slow TTFB 👎 Inflexible	👎 Inflexible 👎 Leads to hydration	👎 Slow TTFB 👎 TTI >> FCP 👎 Usually buffered	👎 TTI > FCP 👎 Limited streaming	👎 TTI >> FCP 👎 No streaming		





DYNAMIC YIELD

VWO

A/B Testing and more

VWO is the all-in-one platform that helps you conduct A/B testing, multivariate testing, personalization, optimization roadmap, and run continuous experiments.

[START PRODUCT TOUR](#)

convert.

FEATURES PRICING FOR AGENCIES REQUEST A DEMO COMPANY ▾ LOGIN TRY FOR FREE

THE A/B TESTING TOOL THAT DOESN'T STORE PERSONAL DATA

Optimize your website for more Leads.

[TRY FOR FREE](#)

unbounce

PRODUCT ▾ CUSTOMERS PRICING FOR AGENCIES FOR PROFESSIONALS

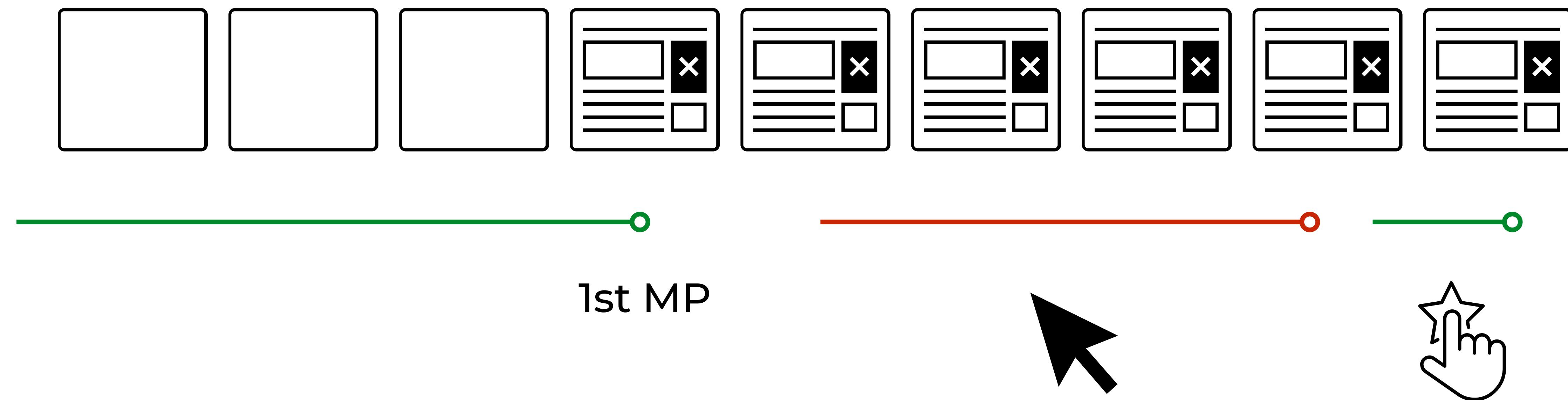
# Optimizely

ADOBETARGET Benefits Features Compare Resources & Support Sign in Get started

Don't just reach the one.  
Reach every single one.

Find the individual in the haystack — and then delight them. A/B and multivariate

# Tag Managers and the Double Delay



“

When we started this process we had a collection of very old scripts and couldn't track the original requester. We removed those on the premise that, if they were important, people would get back in touch—**no one did.**

The Telegraph Engineering

**The Telegraph**



[Overview](#)[Getting Started](#)[Managing Resources](#)[Publishing](#)[Client-side Information](#)[Asynchronous Deployment](#)[Launch Object Reference](#)[Deploy JavaScript tags to opt in to Launch](#)[Administration](#)[Upgrade From a Previous Version](#)[Extension Reference](#)[Tutorials](#)

# Asynchronous Deployment

[Edit on GitHub](#)

Last updated 4 days ago

Performance and non-blocking deployment of the JavaScript libraries required by our products is increasingly important to Adobe Experience Cloud users. Tools like [Google PageSpeed](#) recommend that users change they way they deploy The Adobe libraries on their site. This article explains how to use the Adobe JavaScript libraries in an asynchronous fashion.

## Synchronous vs asynchronous ↗

### Synchronous deployment

Often, libraries are loaded synchronously in the `<head>` tag of a page. For example:

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



By default, the browser parses the document and reaches this line, then starts to fetch the JavaScript file from the server. The browser waits until the file is returned, then it parses and executes the JavaScript file. Finally, it continues parsing the rest of the HTML document.

If the parser comes across the `<script>` tag before rendering visible content, content display is delayed.

If the JavaScript file being loaded is not absolutely necessary to show content to your users, you are unnecessarily requiring your visitors to wait for content. For this reason, website performance benchmark

## CONTENTS

[Synchronous vs asynchronous](#)[Synchronous deployment](#)[Asynchronous deployment](#)[Drawbacks to asynchronous deployment](#)[Timing](#)[Page Bottom event type](#)[Loading the Launch embed code asynchronously](#)[Adobe Target Consideration](#)

# Good options; tough sells

- Don't vary content for first-time visits
- Async the scripts, only change content far down the page
- Preconnect the scripts involved.
- Move the content variation to the server-side

At very least... preconnect the inevitable.

```
<link rel="preconnect" href="https://example.com/">  
  
<script src="https://example.com/thirdparty-2.js"  
async defer>
```



## Cloudflare Workers

The Network is the Computer™

Build serverless applications on Cloudflare's global cloud network of 154 data centers. Cloudflare Workers provides a lightweight JavaScript execution environment that allows developers to augment existing applications or create entirely new ones without configuring or maintaining infrastructure.

[Sign up](#)

[Contact Sales](#)

Already a customer? [Activate Today](#)



“



<https://www.cloudflare.com/products/cloudflare-workers/>

# A/B Testing at the Edge with Servers Workers

---

POSTED BY

**SCOTT JEHL**

11/26/2018

In [Second Meaningful Content: the Worst Performance Metric](#), I noted the terrible impact on page loading performance that can occur when sites introduce personalized content and A/B test logic on the client-side, using JavaScript. To recap, the delivery and application of this pattern tends to produce a “wait and switch” effect; delaying both initial page rendering, and the subsequent re-rendering of portions of the page, as illustrated in this visual timeline:



<https://www.filamentgroup.com/lab/servers-workers.html>