Better performance
=== greater accessibility

Aaron Gustafson
@AaronGustafson
slideshare.net/AaronGustafson

/dəˈzīn/

JEFF VEEN

"I've been amazed at how often those outside the discipline of design assume that what designers do is decoration—likely because so much bad design simply is decoration. Good design isn't. **Good design is problem solving.**"

/ˈfrikSH(ə)n/

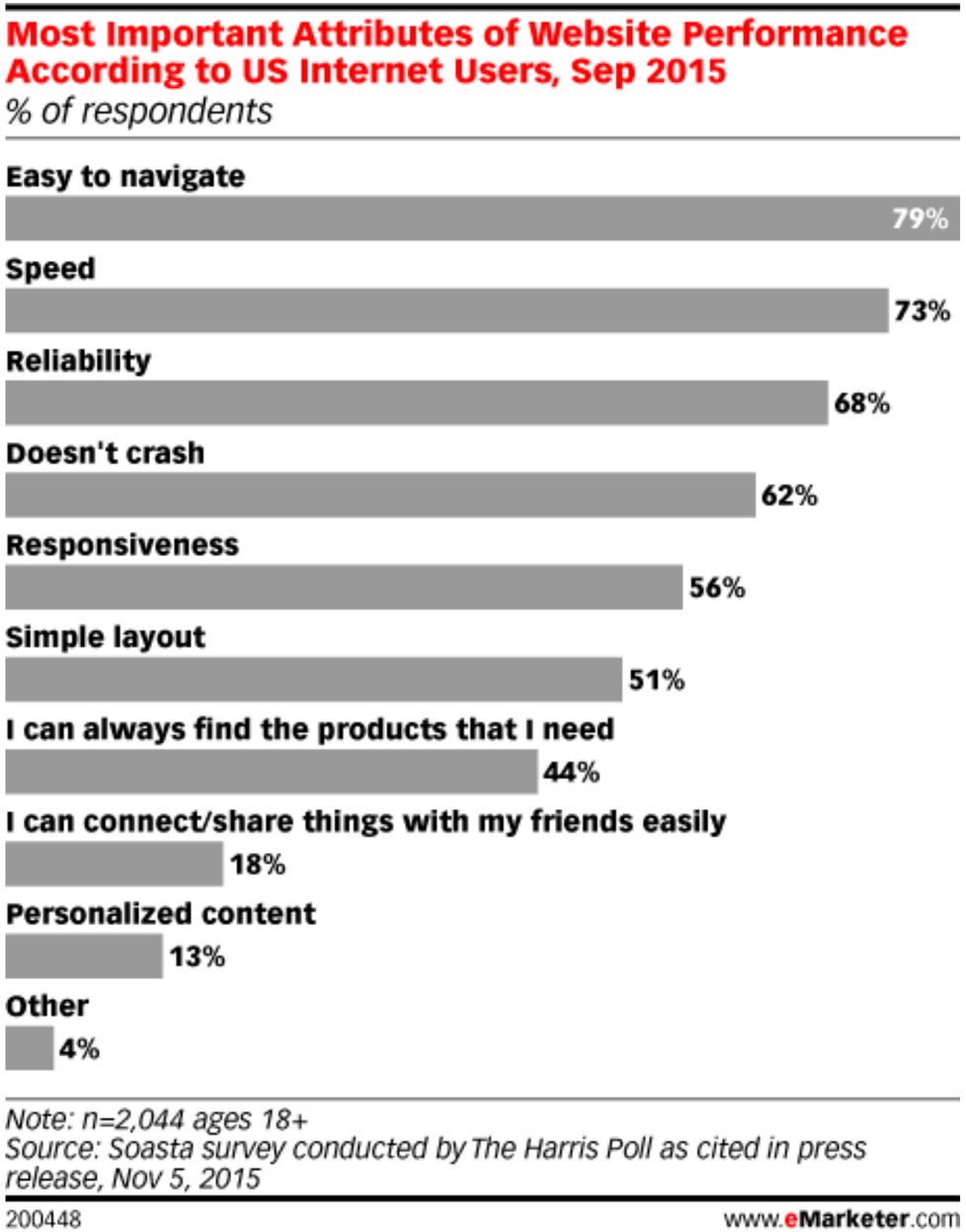
Hallmarks of Good UX

- Streamlined flow
- Clear, concise copy
- Low cognitive load
- Fast performance

Resolutionary

Introducing the new iPad. With the stunning Retina display.
5MP iSight camera. And ultrafast 4G LTE.
Starting at $499.

Poor performance is friction



Source: eMarketer

A 1s delay in page load can reduce conversions by 7%

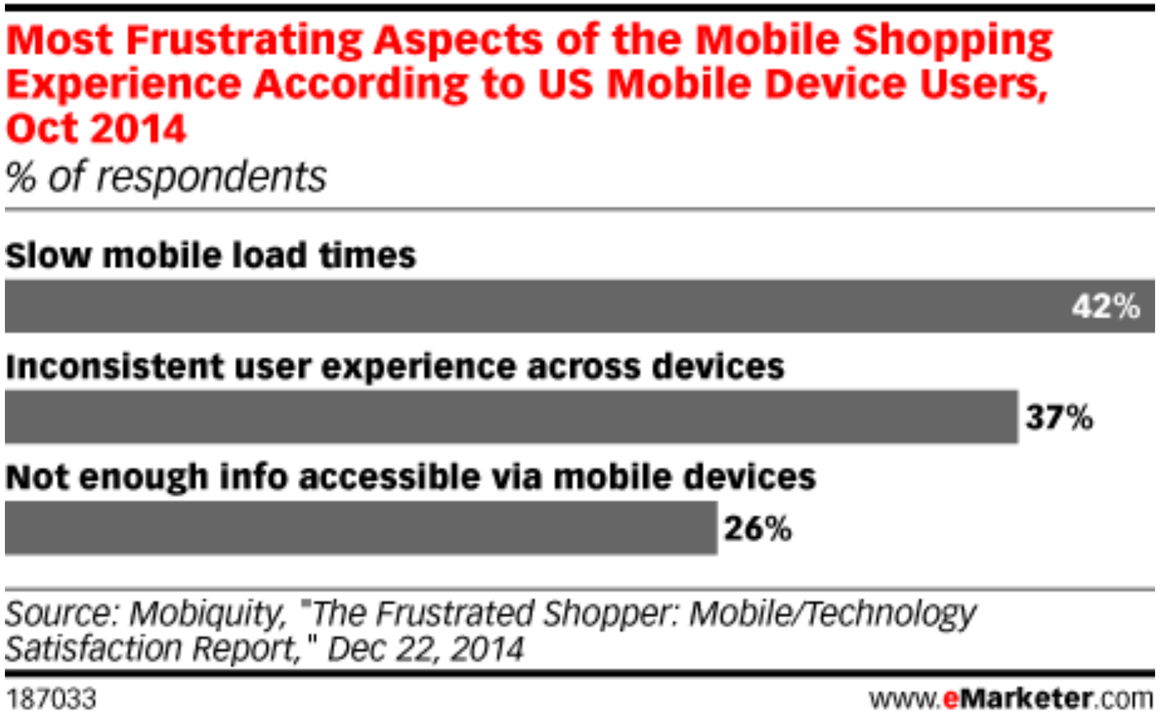
Source: Kissmetrics

For an online shop earning $100k/day, that's about $2.5m in lost sales

Source: Kissmetrics

For Amazon, 1s is worth about $1.6b in sales

Source: HubSpot



Source: eMarketer

53% abandon websites that take more than 3s to load

Source: Google

By shaving 7s off load, Edmunds increased page views by 17% & ad revenue by 3%

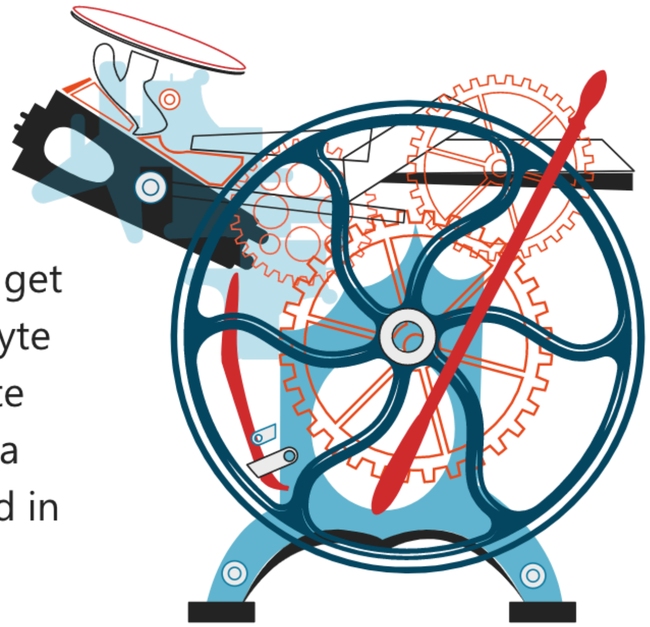
Source: HubSpot

Mozilla reduced page load by 2.2s and saw a 15.4% increase in downloads

Source: HubSpot

Performance matters

Inspiring the Web with Just 10k



With so much of an emphasis on front-end frameworks and JavaScript runtimes, it's time to get back to basics—back to optimizing every little byte like your life depends on it and ensuring your site can work, no matter what. The Challenge? Build a compelling web experience that can be delivered in 10kB and works without JavaScript.

View the Winners | Check Out the Gallery

And the winners are...

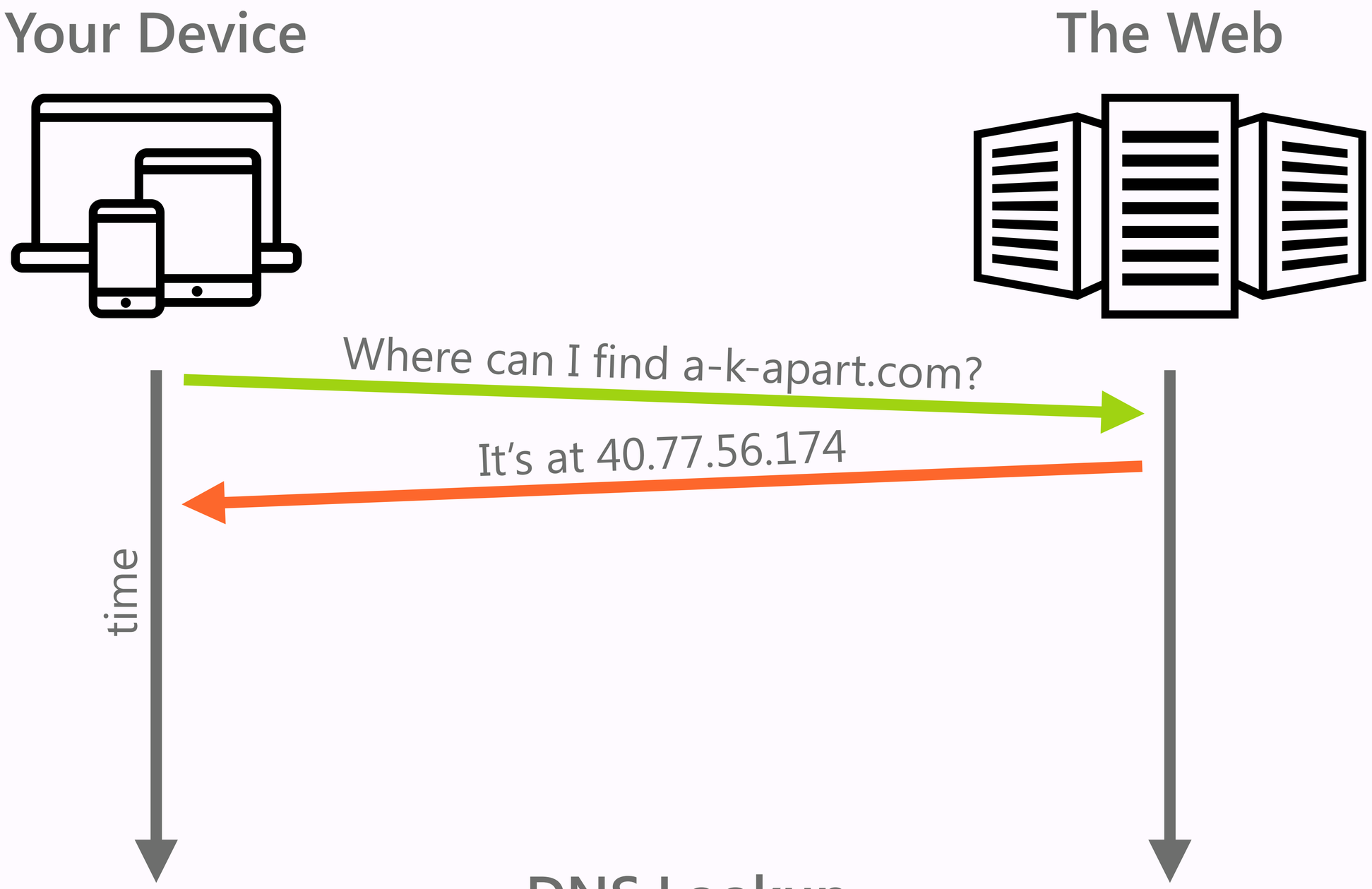
Best Overall:

What we were looking for

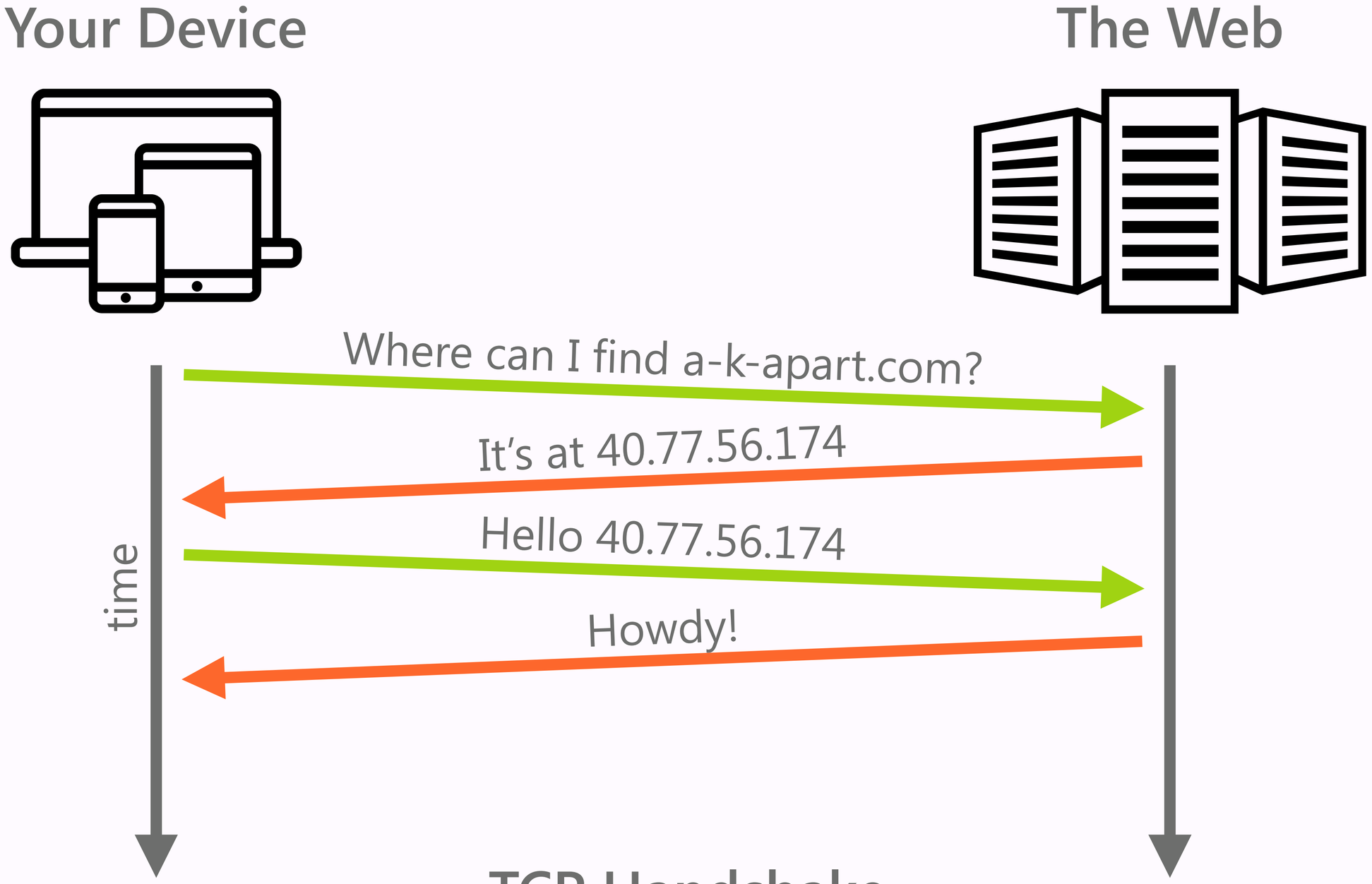
- Usable pages are delivered in 10kB or less
- Pages are accessible via screen readers, the keyboard, etc.
- Entries follow the philosophy of progressive enhancement
- Users can do what they need to without JavaScript

/dôgˈfo͞odiNG/

Let’s talk (briefly) about page load

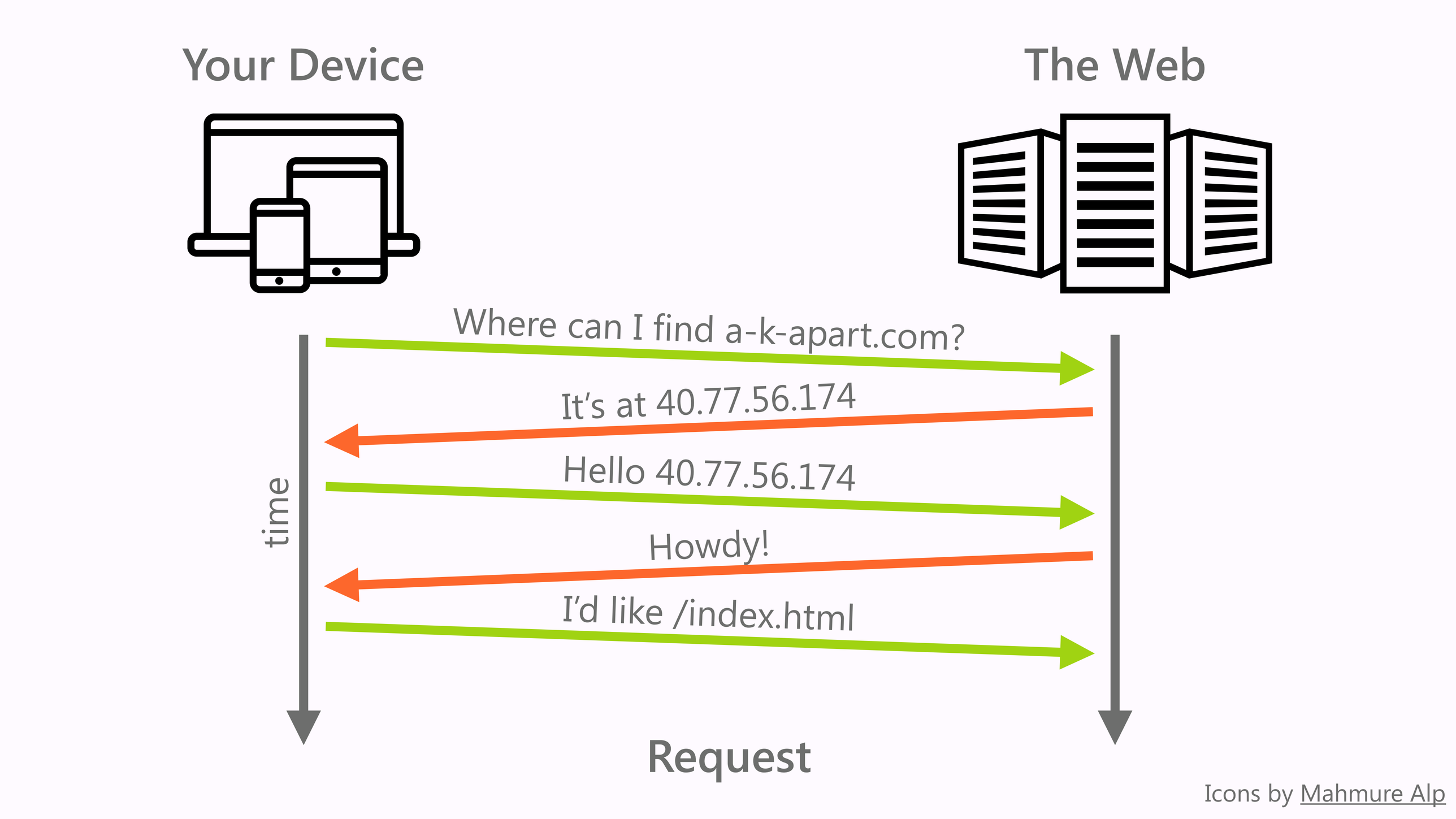


Icons by Mahmure Alp



TCP Handshake

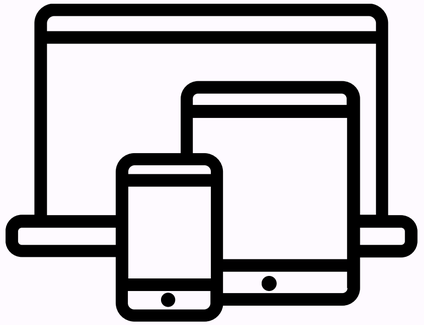
Icons by Mahmure Alp

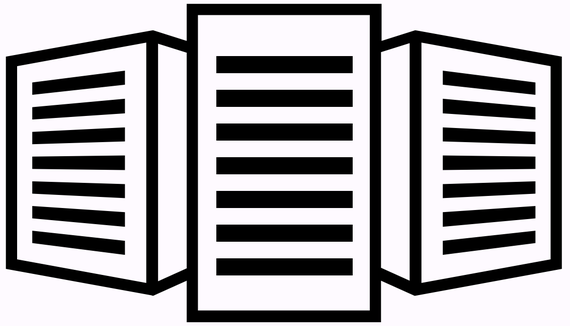


Icons by Mahmure Alp

Your Device

The Web





Where can I find a-k-apart.com?

It's at 40.77.56.174

Hello 40.77.56.174

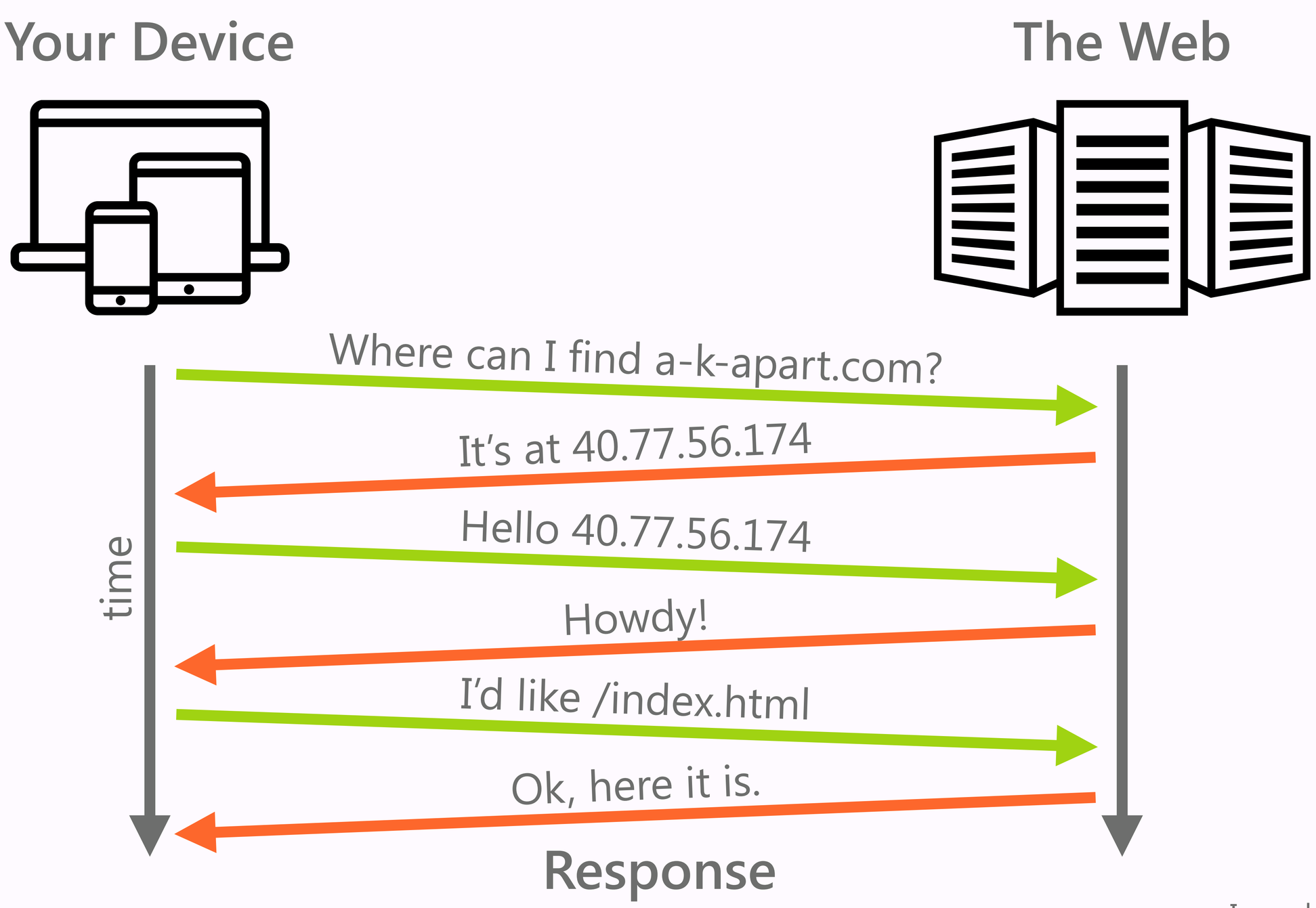
Howdy!

I'd like /index.html

time

Server Processing

Icons by Mahmure Alp



Icons by Mahmure Alp

Icons by Mahmure Alp

CSS & JavaScript can
delay rendering or cause
these to be run again

Response

DOM

CSS

Render Tree

Layout

Paint

Load

Icons by Mahmure Alp

DOM Parsing

```
<html>
 <head>
  <title>Silly example</title>
  <script src="head.js"></script>
  <link rel="stylesheet" href="main.css">
  <style>h2 { font-weight: bold; }</style>
  <script>console.log('hi');</script>
 </head>
 <body>
    <img src="my.png" alt="">
    <script src="body.js"></script>
 </body>
</html>
```

DOM Parsing

```
<html>
 <head>
  <title>Silly example</title>
  <script src="head.js"></script>
  <link rel="stylesheet" href="main.css">
  <style>h2 { font-weight: bold; }</style>
  <script>console.log('hi');</script>
 </head>
 <body>
    <img src="my.png" alt="">
    <script src="body.js"></script>
 </body>
</html>
```

DOM Parsing

```
<html>
 <head>
  <title>Silly example</title>
  <script src="head.js"></script>
  <link rel="stylesheet" href="main.css">
  <style>h2 { font-weight: bold; }</style>
  <script>console.log('hi');</script>
 </head>
 <body>
    <img src="my.png" alt="">
    <script src="body.js"></script>
 </body>
</html>
```

DOM Parsing

```
<html>
 <head>
  <title>Silly example</title>
  <script src="head.js"></script>
  <link rel="stylesheet" href="main.css">
  <style>h2 { font-weight: bold; }</style>
  <script>console.log('hi');</script>
 </head>
 <body>
    <img src="my.png" alt="">
    <script src="body.js"></script>
 </body>
</html>
```

DOM Parsing

```
<html>
 <head>
  <title>Silly example</title>
  <script src="head.js"></script>
  <link rel="stylesheet" href="main.css">
  <style>h2 { font-weight: bold; }</style>
  <script>console.log('hi');</script>
 </head>
 <body>
    <img src="my.png" alt="">
    <script src="body.js"></script>
 </body>
</html>
```

DOM Parsing

```
<html>
 <head>
  <title>Silly example</title>
  <script src="head.js"></script>
  <link rel="stylesheet" href="main.css">
  <style>h2 { font-weight: bold; }</style>
  <script>console.log('hi');</script>
 </head>
 <body>
    <img src="my.png" alt="">
    <script src="body.js"></script>
 </body>
</html>
```

Wait while the browser fetches & parses the script

DOM Parsing

```
<html>
 <head>
  <title>Silly example</title>
  <script src="head.js"></script>
  <link rel="stylesheet" href="main.css">
  <style>h2 { font-weight: bold; }</style>
  <script>console.log('hi');</script>
 </head>
 <body>
    <img src="my.png" alt="">
    <script src="body.js"></script>
 </body>
</html>
```

DOM Parsing

```
<html>
 <head>
  <title>Silly example</title>
  <script src="head.js"></script>
  <link rel="stylesheet" href="main.css">
  <style>h2 { font-weight: bold; }</style>
  <script>console.log('hi');</script>
 </head>
 <body>
    <img src="my.png" alt="">
    <script src="body.js"></script>
 </body>
</html>
```

Wait while the browser fetches & parses the CSS

DOM Parsing

```
<html>
 <head>
  <title>Silly example</title>
  <script src="head.js"></script>
  <link rel="stylesheet" href="main.css">
  <style>h2 { font-weight: bold; }</style>
  <script>console.log('hi');</script>
 </head>
 <body>
    <img src="my.png" alt="">
    <script src="body.js"></script>
 </body>
</html>
```

DOM Parsing

```
<html>
 <head>
  <title>Silly example</title>
  <script src="head.js"></script>
  <link rel="stylesheet" href="main.css">
  <style>h2 { font-weight: bold; }</style>
  <script>console.log('hi');</script>
 </head>
 <body>
    <img src="my.png" alt="">
    <script src="body.js"></script>
 </body>
</html>
```

DOM Parsing

```
<html>
 <head>
  <title>Silly example</title>
  <script src="head.js"></script>
  <link rel="stylesheet" href="main.css">
  <style>h2 { font-weight: bold; }</style>
  <script>console.log('hi');</script>
 </head>
 <body>
    <img src="my.png" alt="">
    <script src="body.js"></script>
 </body>
</html>
```

DOM Parsing

```
<html>
 <head>
  <title>Silly example</title>
  <script src="head.js"></script>
  <link rel="stylesheet" href="main.css">
  <style>h2 { font-weight: bold; }</style>
  <script>console.log('hi');</script>
 </head>
 <body>
    <img src="my.png" alt="">
    <script src="body.js"></script>
 </body>
</html>
```

DOM Parsing

```
<html>
 <head>
  <title>Silly example</title>
  <script src="head.js"></script>
  <link rel="stylesheet" href="main.css">
  <style>h2 { font-weight: bold; }</style>
  <script>console.log('hi');</script>
 </head>
 <body>
    <img src="my.png" alt="">
    <script src="body.js"></script>
 </body>
</html>
```

DOM Parsing

```
<html>
 <head>
  <title>Silly example</title>
  <script src="head.js"></script>
  <link rel="stylesheet" href="main.css">
  <style>h2 { font-weight: bold; }</style>
  <script>console.log('hi');</script>
 </head>
 <body>
    <img src="my.png" alt="">
    <script src="body.js"></script>
 </body>
</html>
```

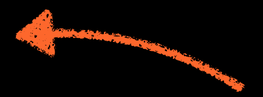
kicks off downloading the image

DOM Parsing

```
<html>
 <head>
  <title>Silly example</title>
  <script src="head.js"></script>
  <link rel="stylesheet" href="main.css">
  <style>h2 { font-weight: bold; }</style>
  <script>console.log('hi');</script>
 </head>
 <body>
    <img src="my.png" alt="">
    <script src="body.js"></script>
 </body>
</html>
```

DOM Parsing

```
<html>
 <head>
  <title>Silly example</title>
  <script src="head.js"></script>
  <link rel="stylesheet" href="main.css">
  <style>h2 { font-weight: bold; }</style>
  <script>console.log('hi');</script>
 </head>
 <body>
    <img src="my.png" alt="">
    <script src="body.js"></script>
 </body>
</html>
```



wait while the browser fetches & parses the script

DOM Parsing

```
<html>
 <head>
  <title>Silly example</title>
  <script src="head.js"></script>
  <link rel="stylesheet" href="main.css">
  <style>h2 { font-weight: bold; }</style>
  <script>console.log('hi');</script>
 </head>
 <body>
    <img src="my.png" alt="">
    <script src="body.js"></script>
 </body>
</html>
```

DOM Parsing

```
<html>
 <head>
  <title>Silly example</title>
  <script src="head.js"></script>
  <link rel="stylesheet" href="main.css">
  <style>h2 { font-weight: bold; }</style>
  <script>console.log('hi');</script>
 </head>
 <body>
    <img src="my.png" alt="">
    <script src="body.js"></script>
 </body>
</html>
```

Steps for better performance

1. Use native features whenever possible
2. Only include assets you actually need
3. Optimize everything
4. Think about when you load assets
5. Consider how you load assets
6. Only load assets when they add value

Step 1:
Use native features
whenever possible

(they're effectively free)

Por exemplo

```
<header>
  Header content…
</header>
```

not only shorter than , but semantic too

depending on its location in the document, could also provide a landmark for navigation

Por exemplo

```
<input id="n" name="n"
       required aria-required="true"
       autocorrect="off"
       autocapitalize="words"
       placeholder="Sir Tim Berners Lee"
       autocomplete="name"
       >
```

browser can inform assistive tech that the field is required

modern browsers require users to enter content

don't let the browser try to correct someone's name

if the browser already knows the user's name, by all means, let it fill it in

auto-disappearing suggestion without JavaScript

Por exemplo

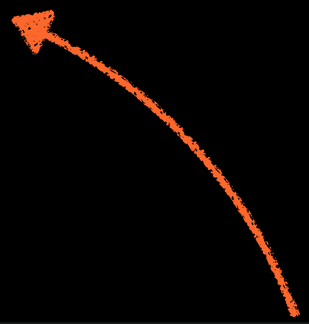
```
<input type="email"
       id="e" name="e"
       required aria-required="true"
       autocorrect="off"
       autocapitalize="off"
       autocomplete="email"
       placeholder="you@yourdomain.tld"
       >
```

modern browsers validate the email address and may supply a custom keyboard layout

let the browser suggest an email address if it has one

Por exemplo

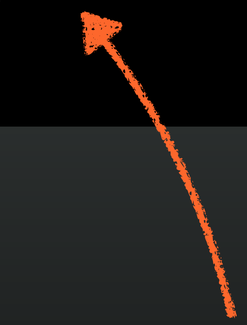
```
@media (min-width:600px) {
   .gridded {
     display: grid;
     grid-template-columns: 1fr 300px;
     grid-gap: 20px;
   }
}
```



So much better than floats

Por exemplo

```
var $radios = document.querySelectorAll(
                'input[type=radio]'
              );
```



CSS selector-based
DOM traversal
without a JS library

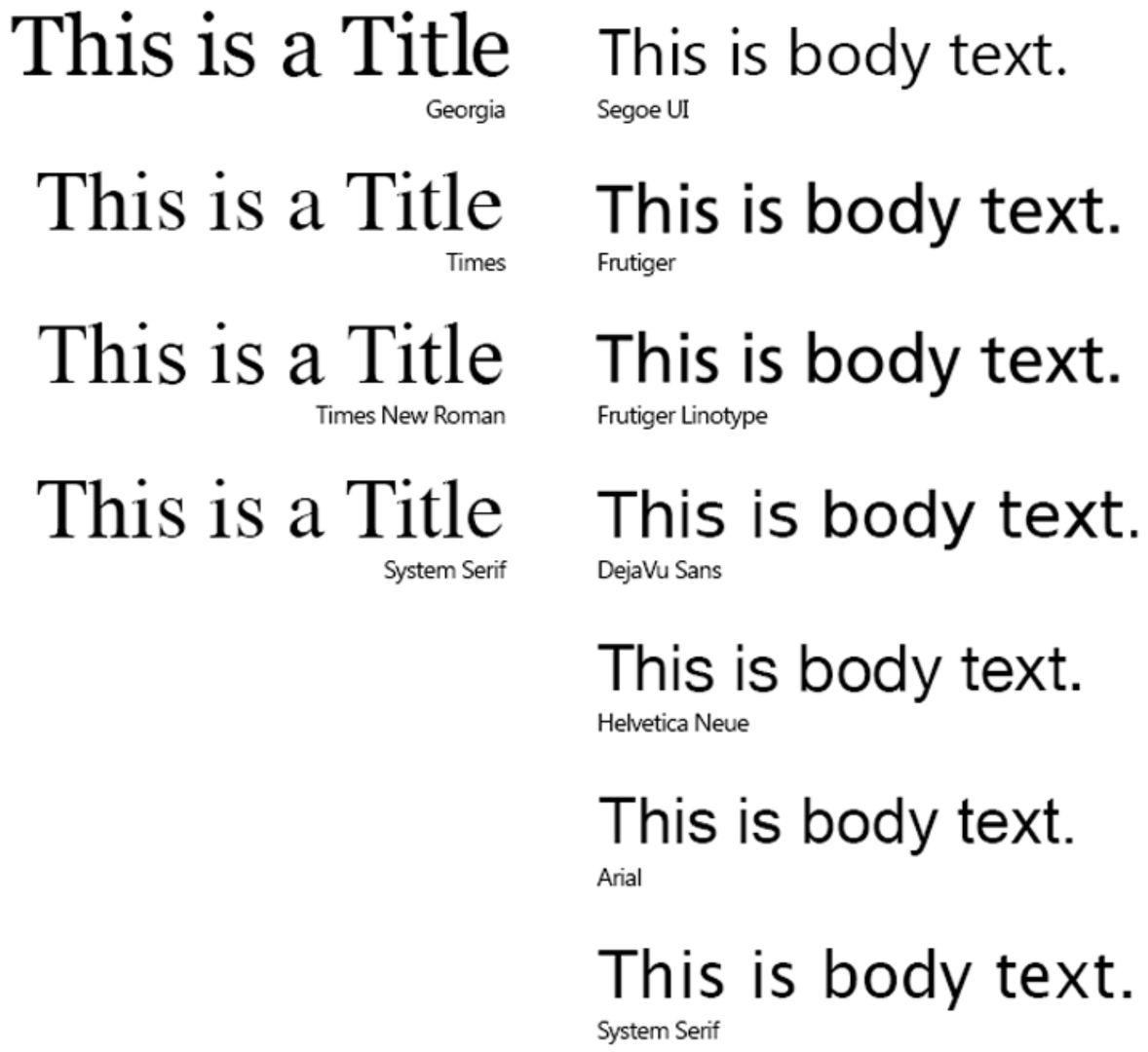
Por exemplo

```
font-family: Georgia, Times,
             "Times New Roman", serif
```

Serif

```
font-family: "Segoe UI", Frutiger,
             "Frutiger Linotype",
             "Dejavu Sans", "Helvetica Neue",
             Arial, sans-serif;
```

Sans Serif

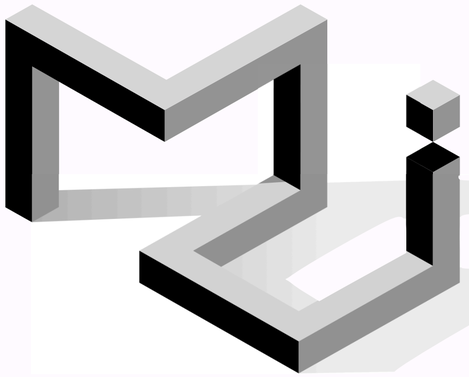

This is a Title
Georgia
This is body text.
Segoe UI
This is a Title
Times
This is body text.
Frutiger
This is a Title
Times New Roman
This is body text.
Frutiger Linotype
This is a Title
System Serif
This is body text.
DejaVu Sans
This is body text.
Helvetica Neue
This is body text.
Arial
This is body text.
System Serif

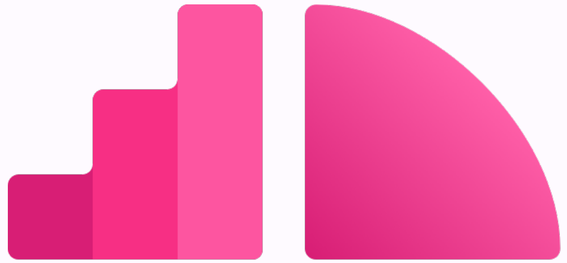
If you need a custom font: subset, subset, subset

Proceed with caution

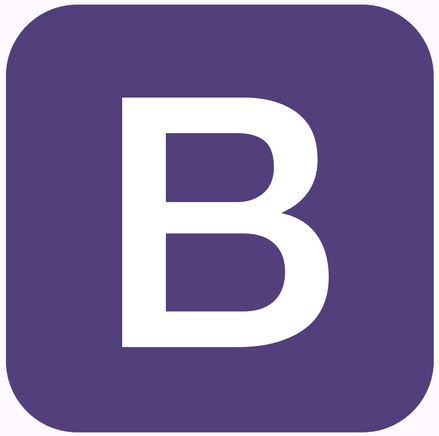
Source: Bran Stein

Step 2:
Only include assets
you actually need





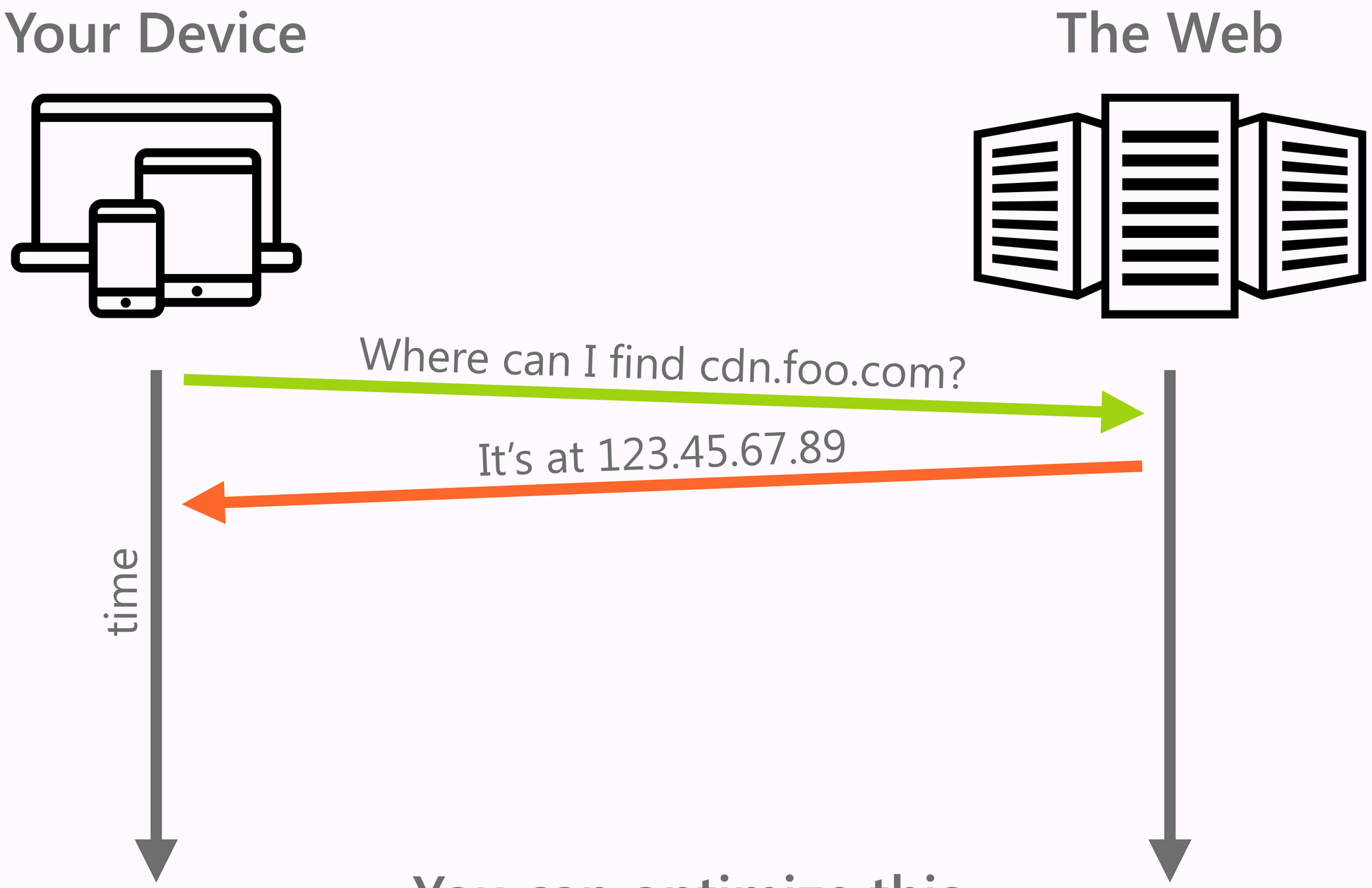
Great tools, possibly overkill



Every tool has a cost

Framework	Size (Compressed)
Bootstrap 2	149 kB
Bootstrap 3	103 kB
Angular 1.4	51 kB
Ember 2.2.0	111 kB
Foundation 4	266 kB
jQuery	32 kB
UI Kit	86 kB
React 16 + React DOM	35 kB
Vue 2.4.2	20 kB

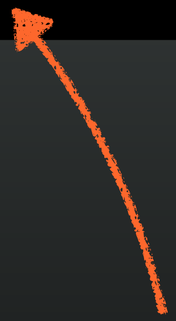
Chances are, your library of choice is on a CDN



Icons by Mahmure Alp

Find the server early

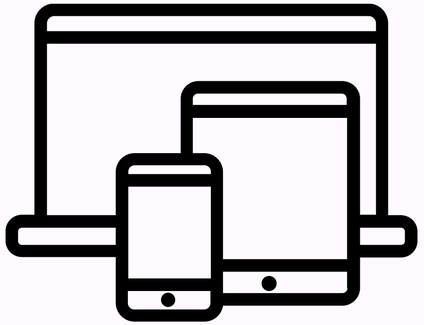
```
<link rel="dns-prefetch"
      href="https://cdn.foo.com">
```

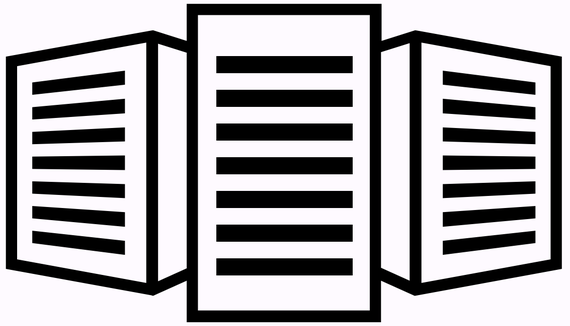


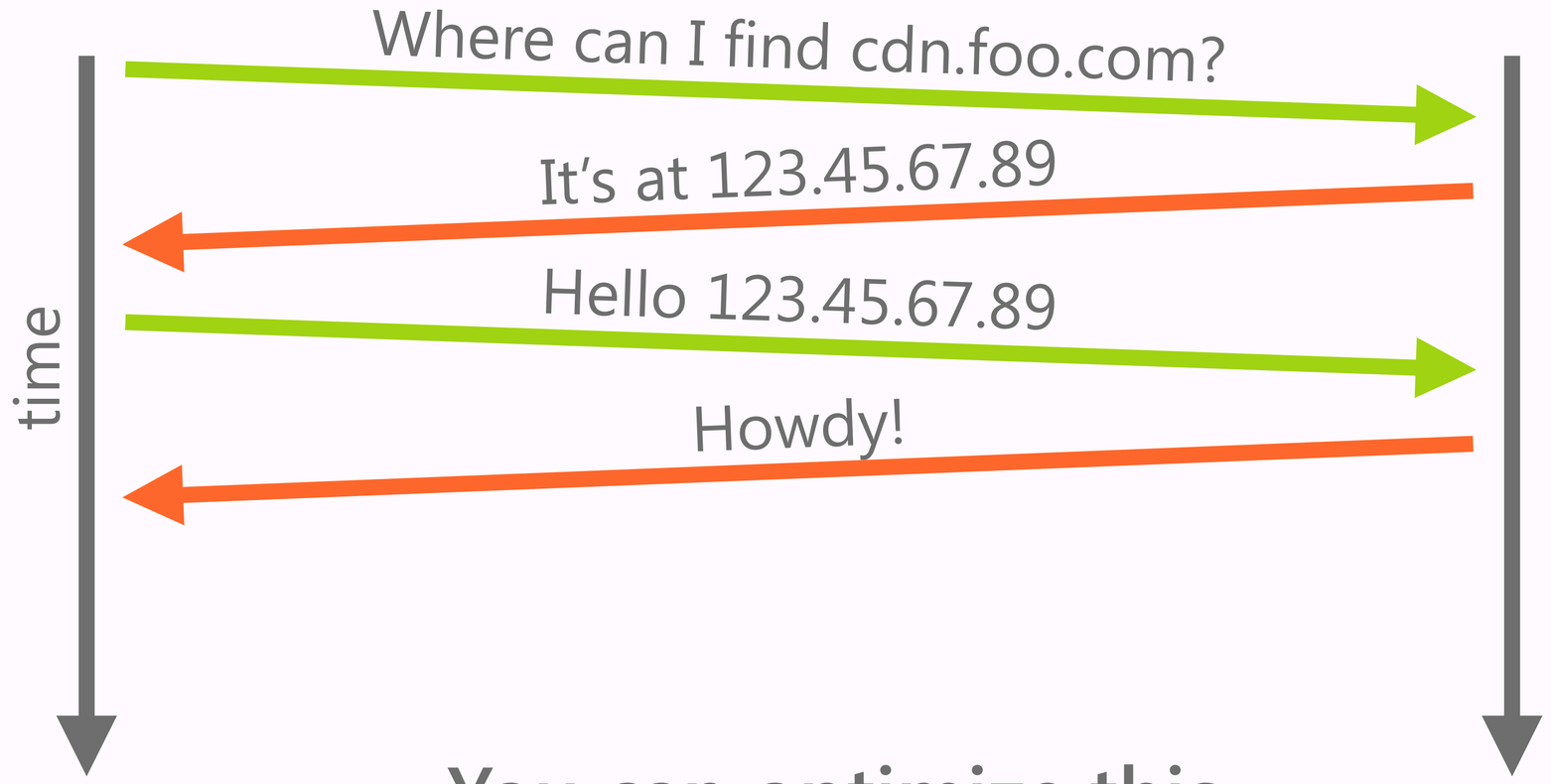
Drop this in the head of your pages

Your Device

The Web





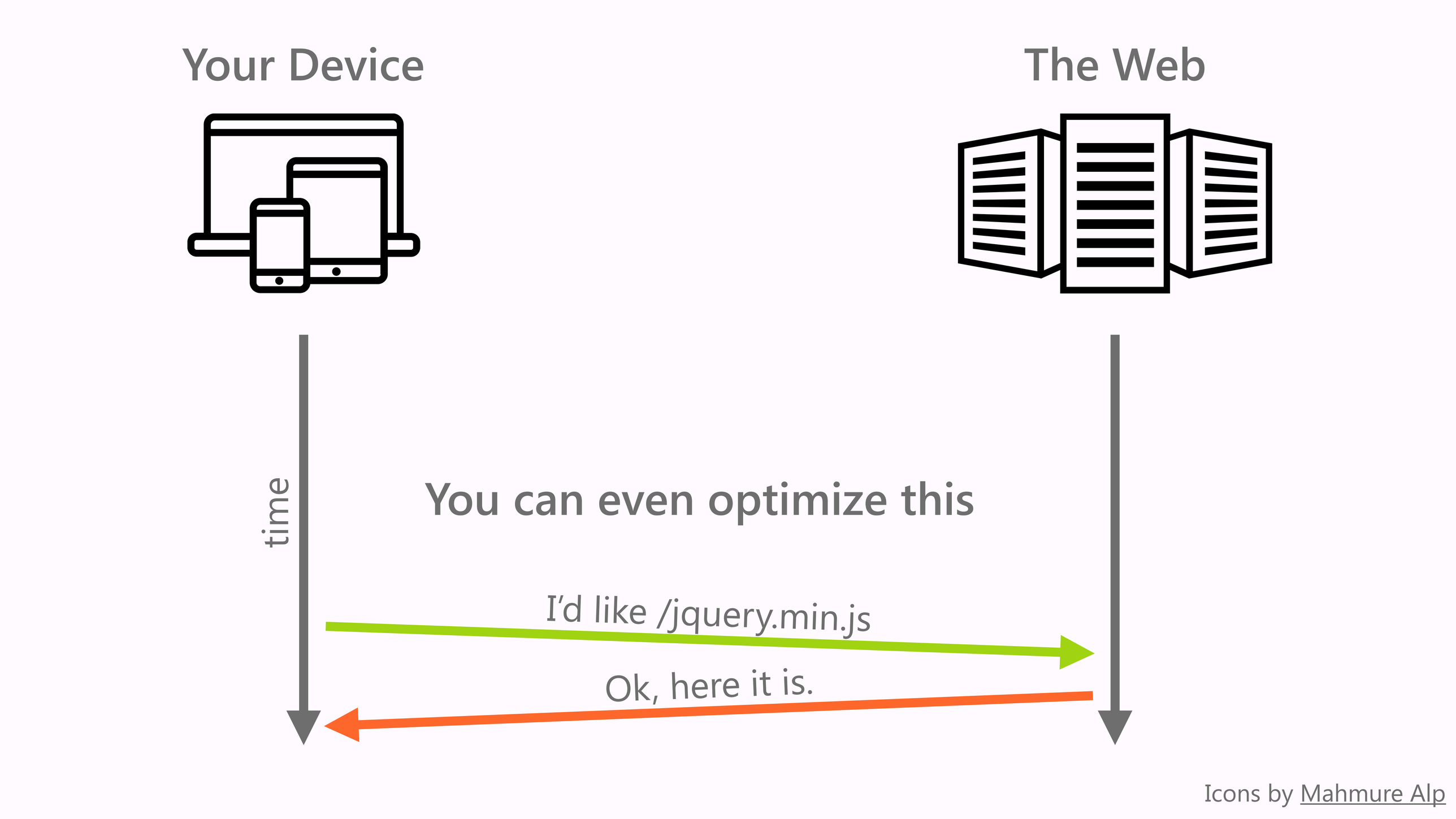


You can optimize this

Icons by Mahmure Alp

Go for the handshake

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

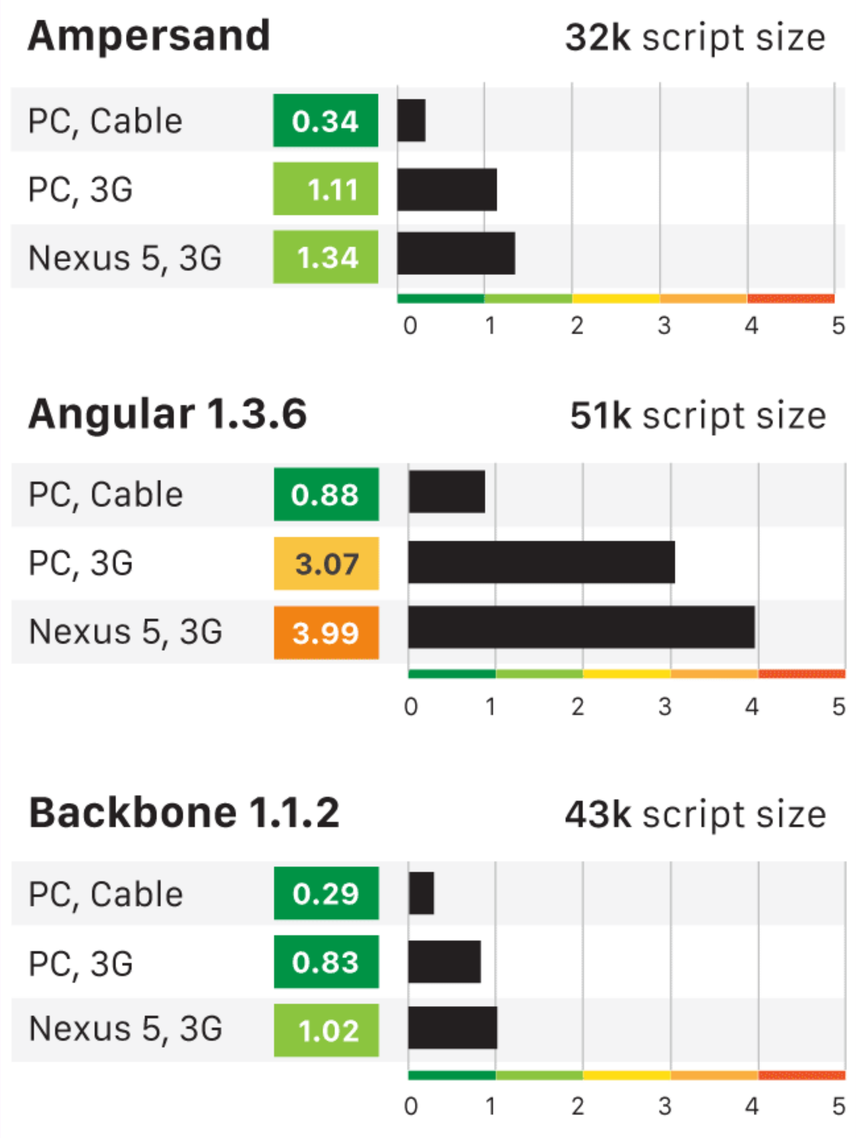


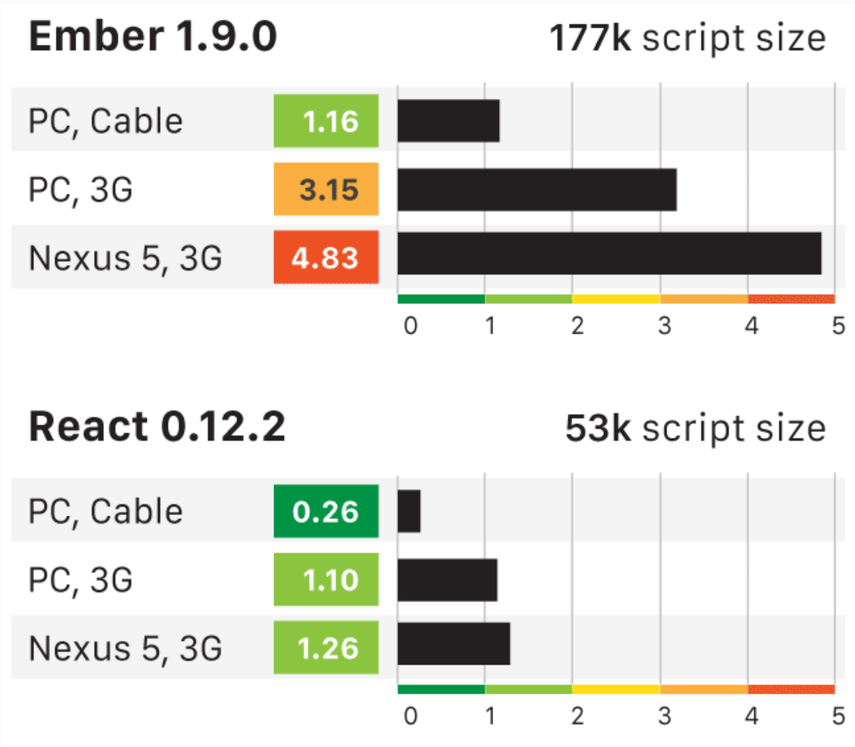
Icons by Mahmure Alp

Grab that resource

```
<link rel="preload"
      href="https://cdn.foo.com/jquery.min.js"
      as="script">
```

Download isn't everything





Source: The Filament Group

Download isn’t everything

Framework	Method/function	operations/s
Vanilla JS	`document.getElementById()`	12,137,211
Dojo	`dojo.byId();`	5,443,343
Prototype	`$()`	2,940,734
Ext JS	`Ext.get()`	997,562
jQuery	`$()`	350,557
YUI	`YAHOO.util.Dom.get()`	326,534
MooTools	`document.id()`	78,802

Source: VanillaJS

10k APART

Inspiring the Web with Just 10k

Read the Rules | View the Gallery | Meet the Judges

Brought to you by Microsoft and An Event Apart

Inspiring the Web with Just 10k

With so much of an emphasis on front-end frameworks and JavaScript runtimes, it… back to basics—back t… like yo…

100% library free

View the Winners | Check Out the Gallery

And the winners are…

Best Overall:

Best Design:

We used some hints though

```
<link rel="preconnect"
      href="//10kapart.blob.core.windows.net">
<link rel="preconnect"
      href="//cdnjs.cloudflare.com">
<link rel="preconnect"
      href="//www.google-analytics.com">
```

Step 3:
Optimize
everything

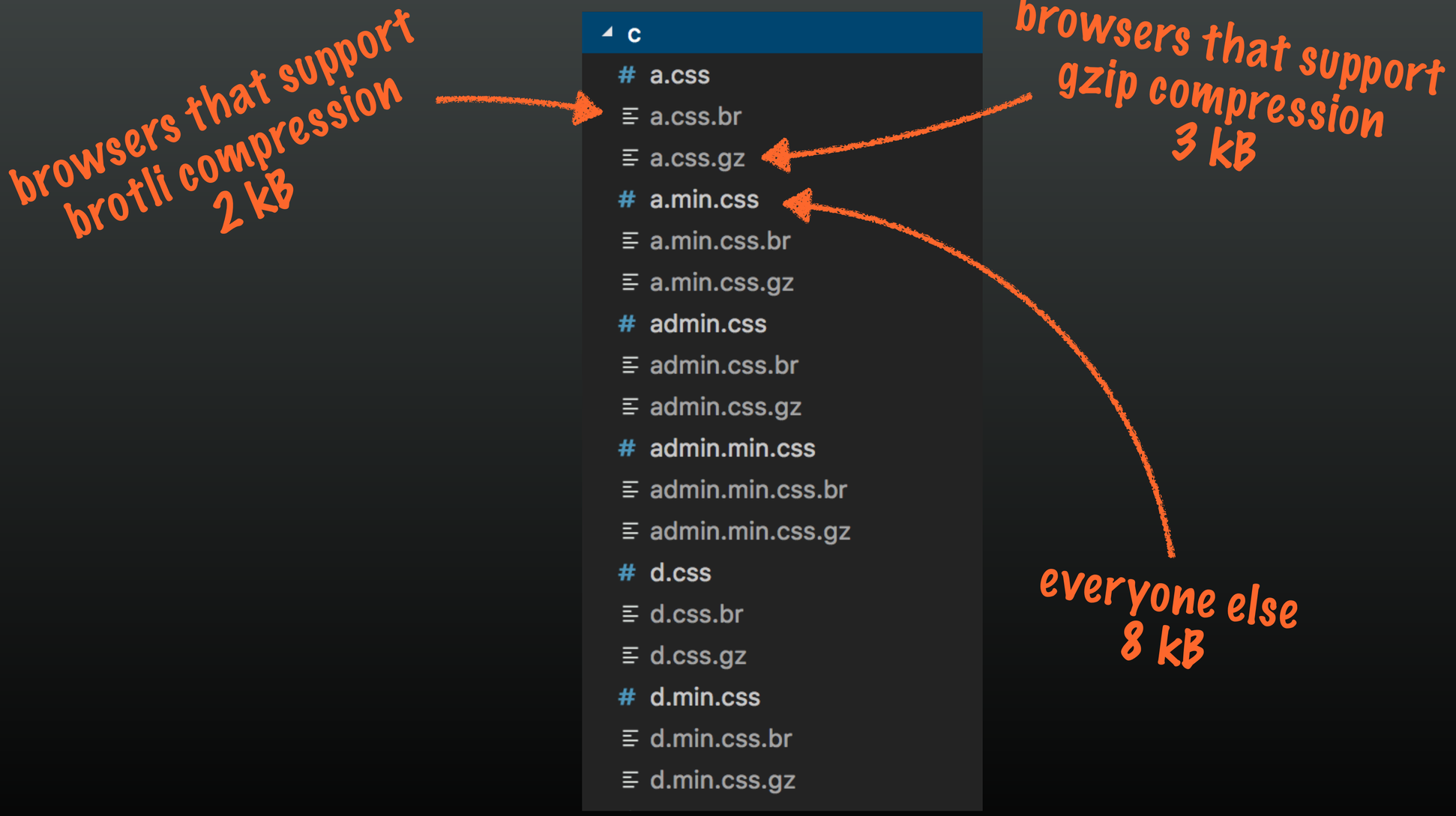
Our approach to CSS (Gulp)

1. Write modular CSS in Sass (+ Breakup for MQ management)
2. Compile CSS with a precision of 4 decimal places (gulp-sass)
3. Fallbacks for the last 2 browser versions (gulp-autoprefixer)
4. CSS shorthand declarations if possible (gulp-shorthand)
5. Remove any unused declarations/rule sets (gulp-uncss)
6. Optimize the files (gulp-csso)
7. Minify (gulp-clean-css)
8. Gzip (gulp-zopfli)
9. Brotli (gulp-brotli)

Before

- scss
  - base
    - __base.scss
    - _blocks.scss
    - _body.scss
    - _forms.scss
    - _headings.scss
    - _links.scss
    - _lists.scss
    - _media.scss
    - _text.scss
  - helpers
    - __helpers.scss
    - _config.scss
    - _functions.scss
    - _mixins.scss
    - _placeholders.scss
    - _variables.scss
- components
  - __components.scss
  - _boxed.scss
  - _cookie-banner.scss
  - _count.scss
  - _cta.scss
  - _faq.scss
  - _footer.scss
  - _gallery.scss
  - _header.scss
  - _hero.scss
  - _judges.scss
  - _main.scss
  - _mq-watcher.scss
  - _nav.scss
  - _pagination.scss
  - _project.scss
  - _target.scss
  - _vote.scss
  - _winners.scss
- pages
  - __pages.scss
  - _home.scss
  - vendors
    - breakup
  - __vendors.scss
- a.scss
- admin.scss
- d.scss

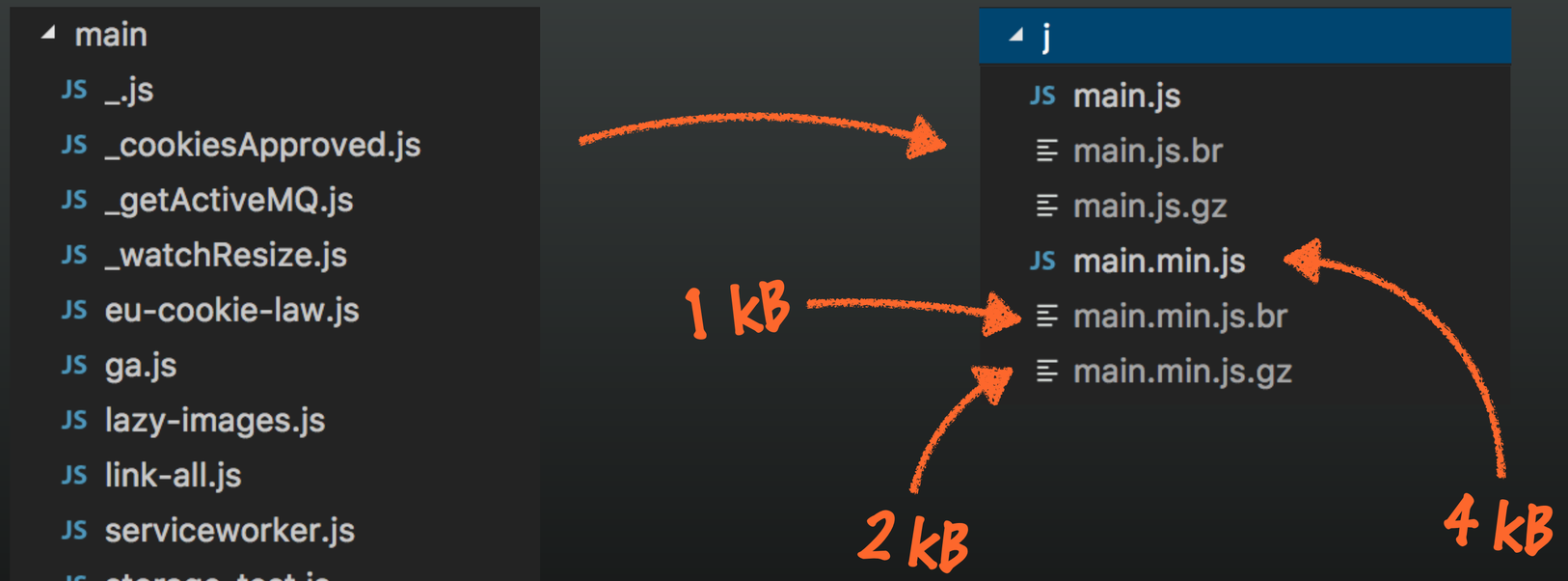
After


browsers that support brotli compression 2 kB
c
a.css
a.css.br
a.css.gz
a.min.css
a.min.css.br
a.min.css.gz
admin.css
admin.css.br
admin.css.gz
admin.min.css
admin.min.css.br
admin.min.css.gz
d.css
d.css.br
d.css.gz
d.min.css
d.min.css.br
d.min.css.gz
browsers that support gzip compression 3 kB
everyone else 8 kB

Our approach to JS (Gulp)

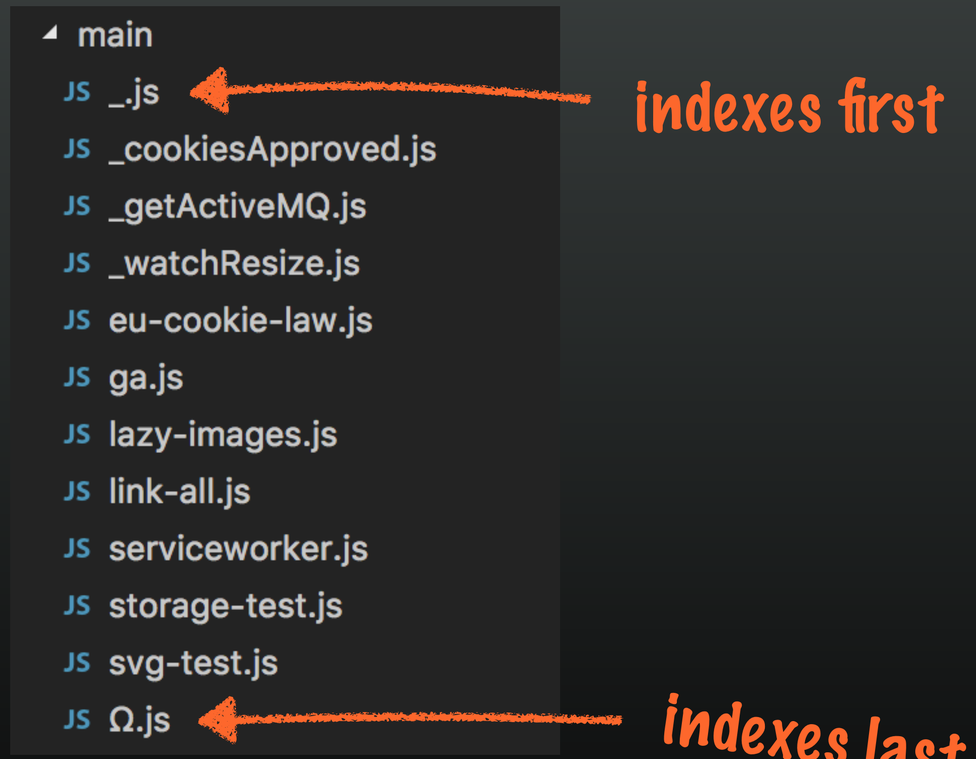
1. Write modular JavaScript, grouped as appropriate
2. Combine files based on folder structure (gulp-folders, gulp-concat)
3. Create an wrapping closure to isolate from other JS (gulp-insert)
4. Minify (gulp-uglify)
5. Gzip (gulp-zopfli)
6. Brotli (gulp-brotli)

Results



about 8 kB all-up

Interesting side note



We also minified & pre-compressed our HTML

Step 4:
Think about when you load assets

We had 10 JS files

- **Global**
  - main.js - the site's library
  - serviceworker.js - The site's service worker
- **Browser-specific**
  - html5shiv.js - local copy of the HTML5 Shiv for < IE9

We had 10 JS files

- **Page-specific**
  - enter.js - Entry form-related code
  - form-saver.js - Used to save form entries locally until submitted
  - hero.js - Runs the SVG animation on the homepage
  - home.js - Handles homepage-specific tasks
  - project.js - Used on project pages during voting
  - update.js - Handles the winner update form

Per the common wisdom

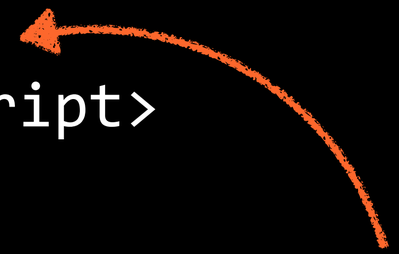
```
    <script src="/j/main.min.js"></script>
  </body>
</html>
```

No need to run immediately

```
    <script src="/j/main.min.js"></script>
    <script src="/j/home.min.js"
            defer
            ></script>
  </body>
</html>
```

Run whenever you can

```
    <script src="/j/main.min.js"></script>
    <script src="/j/home.min.js"
            async
            ></script>
  </body>
</html>
```



run whenever it becomes available, but don't delay page load

Consider dependencies

```
<script src="/j/main.min.js"></script>
<script src="/j/home.min.js" async></script>
```

Consider dependencies

```
<script src="/j/main.min.js" async></script>
<script src="/j/home.min.js" async></script>
```

what if this calls a function in main.min.js?

“race condition”

Avoid race conditions

```
<script src="/j/main.min.js"></script>
<script src="/j/home.min.js" async></script>
```

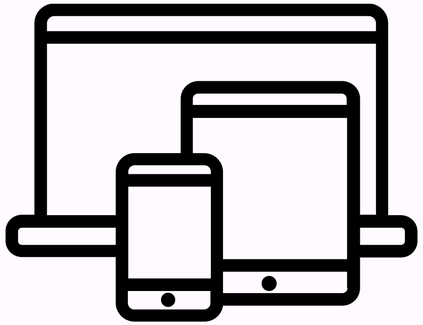
Why so many separate files?

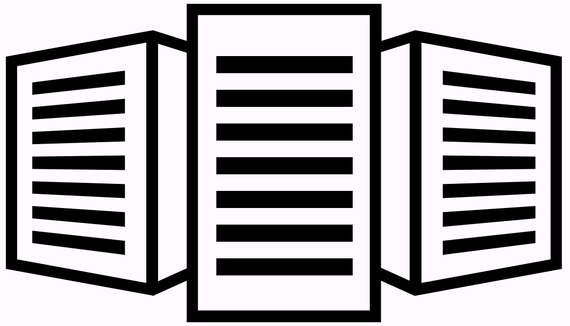
Connections in HTTP/1.1

Browser	Per host	Overall
IE 9	6	35
IE 10	8	17
IE 11	13	17
Firefox 4+	6	17
Opera 11+	6	user defined
Chrome 4+	6	10
Safari 7+	6	17

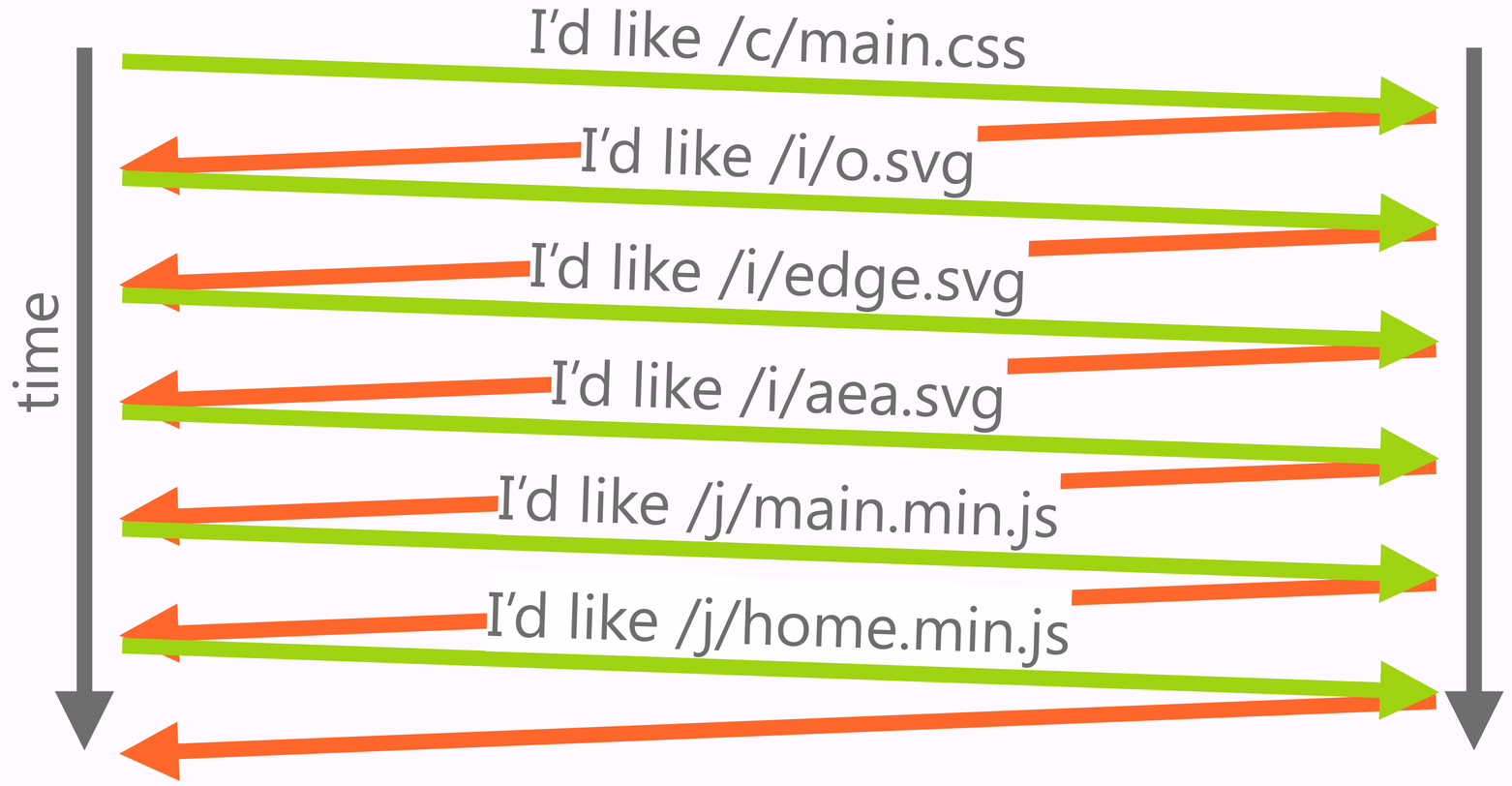
Your Device

The Web



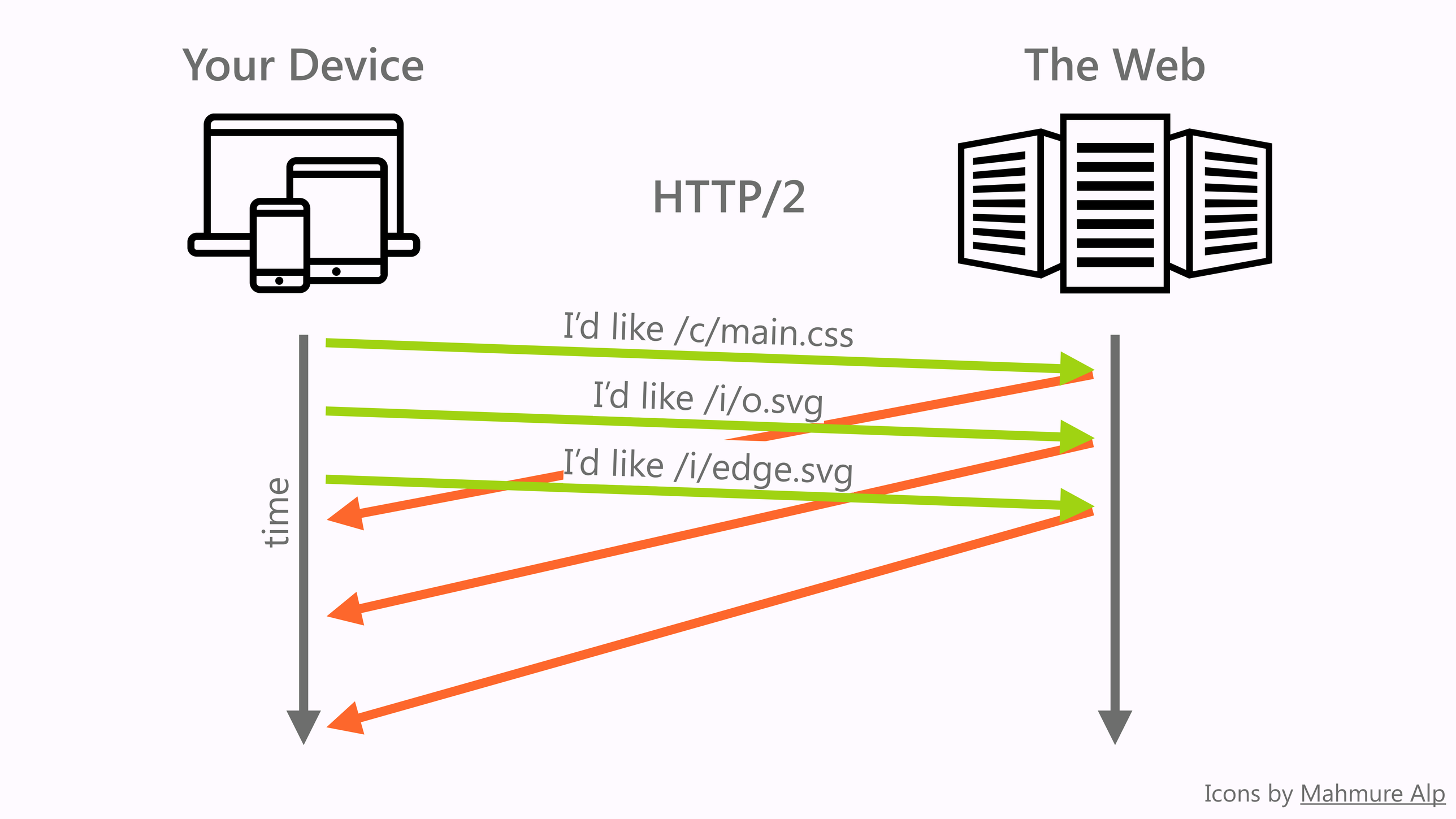


HTTP/1.1

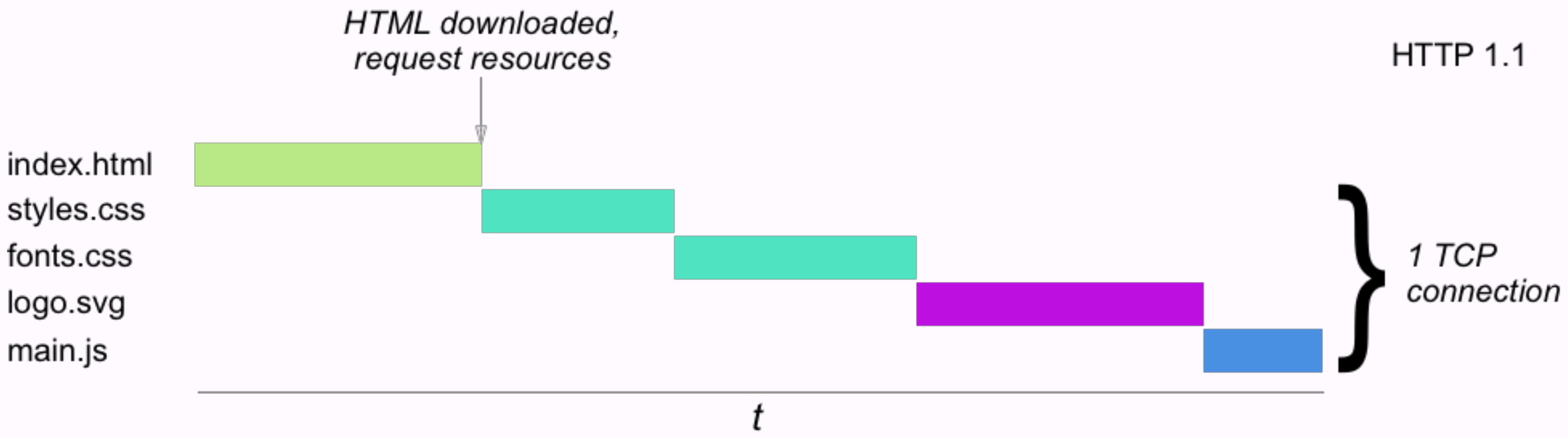


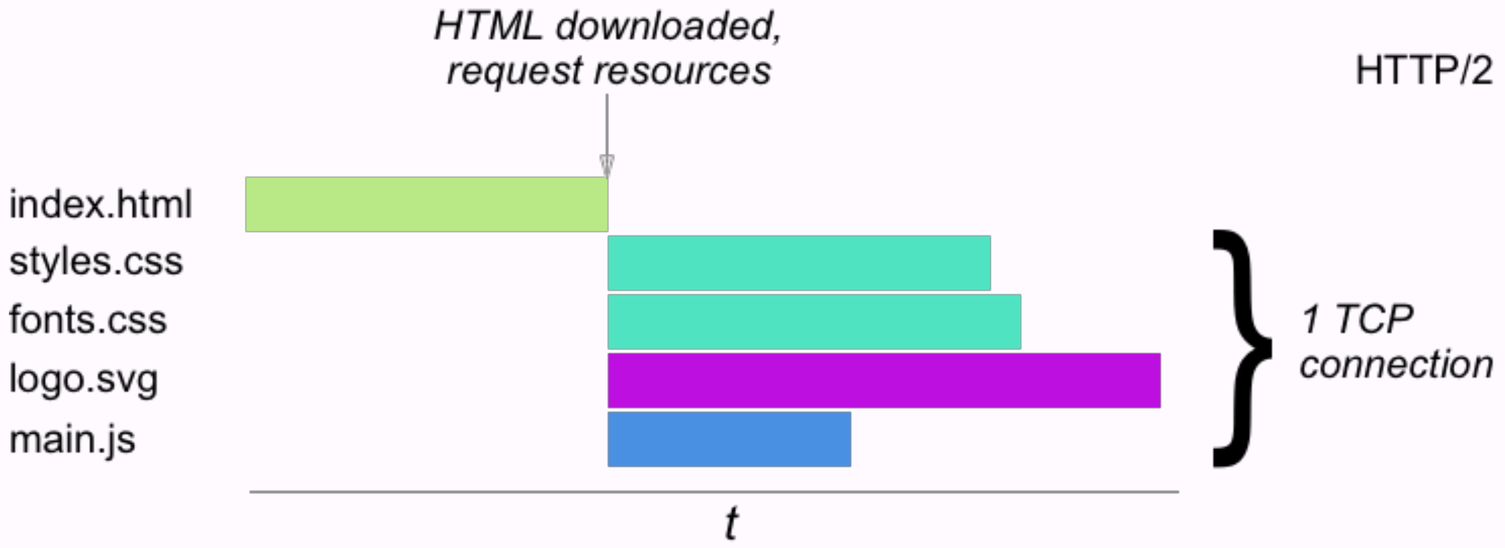
Icons by Mahmure Alp

HTTP/2 creates a single connection and contents stream in



Icons by Mahmure Alp





Source: A List Apart

A LIST APART

Illustration by Dougal MacPherson

The Best Request Is No Request, Revisited

by **Stefan Baumgartner** · November 28, 2017

Published in Browsers, The Server Side

Over the last decade, web performance optimization has been controlled by one

Source: *A List Apart*

Step 5:
Consider how you load assets

Start simple

```
<link rel="stylesheet" href="/c/d.min.css">
<link rel="stylesheet" href="/c/a.min.css"
      media="only screen">
```

default styles (all browsers)

advanced styles (modern browsers)

Fault tolerance FTW!

```
<link rel="stylesheet" href="/c/d.min.css">
<link rel="stylesheet" href="/c/a.min.css"
      media="only screen">
```

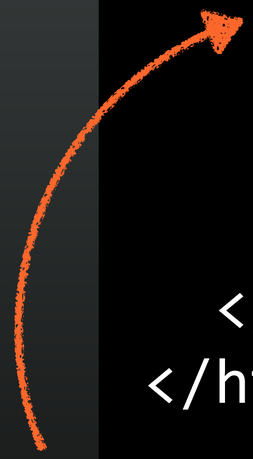
browsers that don't grok
media queries ignore this file

Conditional scripting

```
<!--[if lt IE 9]>
  <script src="/j/html5shiv.min.js"></script>
<![endif]-->
```

Conditional scripting

```
    <!--[if gt IE 8]><!-->
      <script src="/j/main.min.js"></script>
      <script src="/j/home.min.js" async
              ></script>
    <!--<![endif]-->
  </body>
</html>
```



hides scripts from <= IE8
(no need to test!)

Conditional imagery

Brought to you by Microsoft Edge, in association with An Event Apart.

©2007–16 Microsoft | Legalese | Privacy Statement & Cookies | Contact Us
Find out how this site was made

Conditional images

```
@media (min-width: 36.375em) {
  .presented-by [href*=microsoft]::before {
    background-image: url(/i/c/edge.png);
    background-image: url(/i/c/edge.svg),
                      none;
    …
  }
}
```

Conditional images

```
@media (min-width: 36.375em) {
  .presented-by [href*=microsoft]::before {
    background-image: url(/i/c/edge.png);
    background-image: url(/i/c/edge.svg),
                      none;
    …
  }
}
```

Conditional images

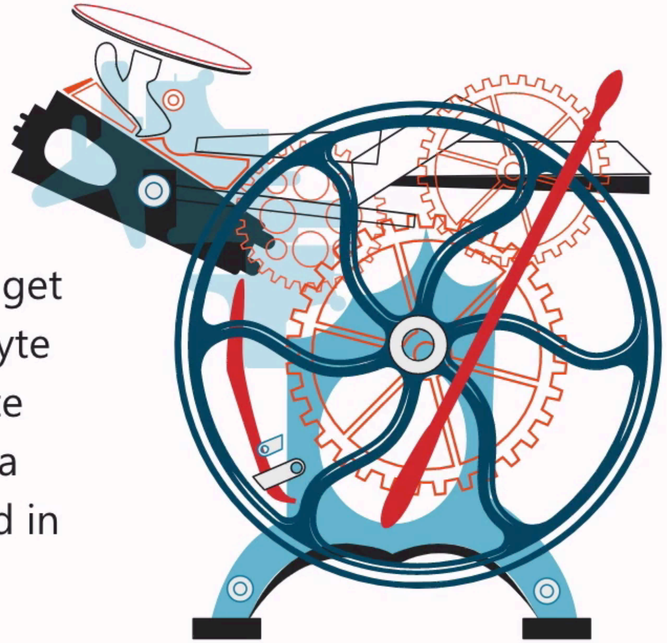
```
@media (min-width: 36.375em) {
   .presented-by [href*=microsoft]::before {
      background-image: url(/i/c/edge.png);
      background-image: url(/i/c/edge.svg),
                        none;

      …
   }
}
```

browsers that support multiple backgrounds also support SVG

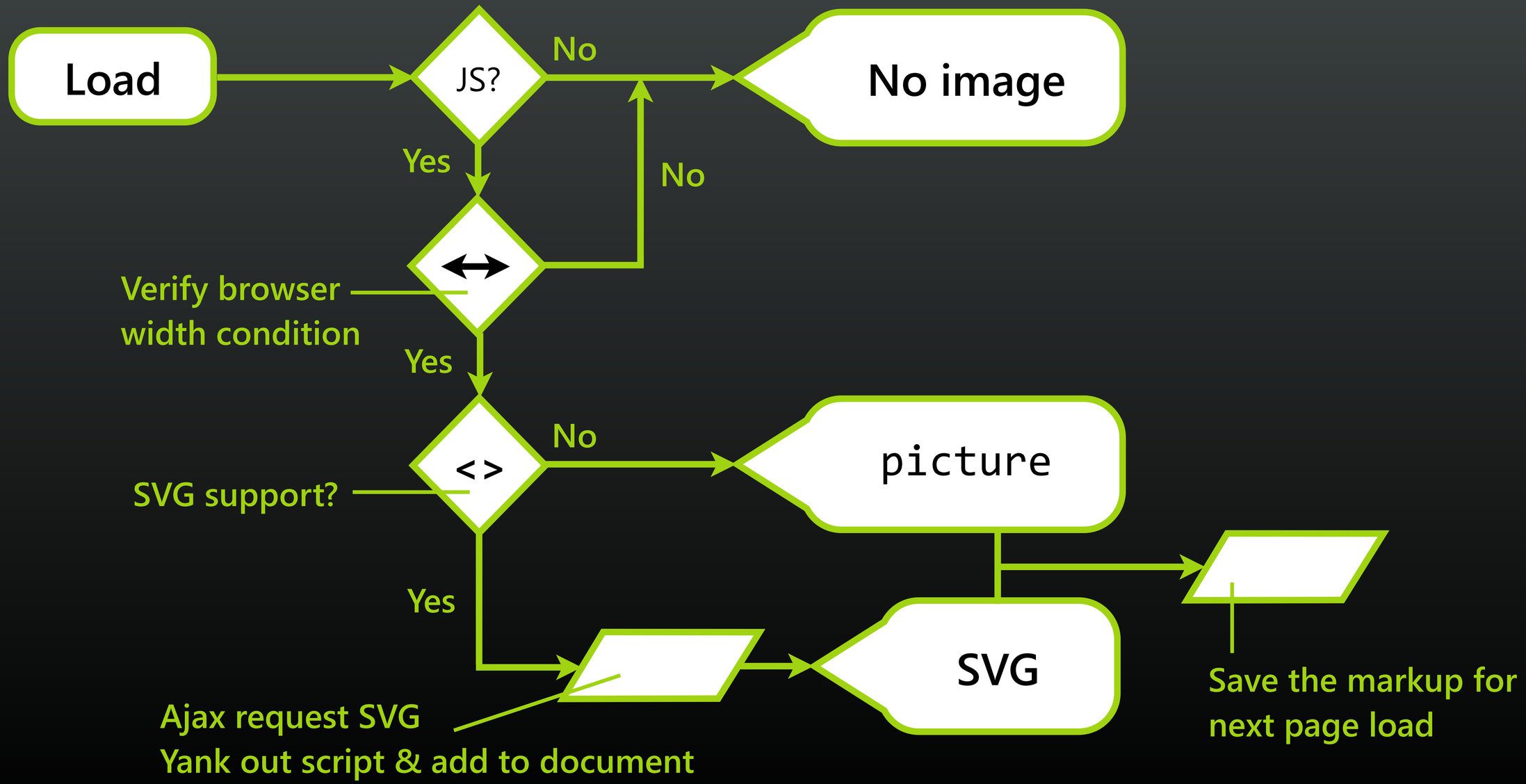
Conditional animation

Inspiring the Web with Just 10k

With so much of an emphasis on front-end frameworks and JavaScript runtimes, it's time to get back to basics—back to optimizing every little byte like your life depends on it and ensuring your site can work, no matter what. The Challenge? Build a compelling web experience that can be delivered in 10kB and works without JavaScript.



How do we get there?



Step 6:
Only load assets
when they add value

G.O.P. TAX PLAN

Adverse Report Threatened Tax Bill, So G.O.P. Went on Attack

- A Republican requirement that Congress consider the full cost of major legislation threatened to derail the party's $1.5 trillion tax rewrite last week.
- Lawmakers went on the offensive to discredit the agency performing the analysis.

21m ago

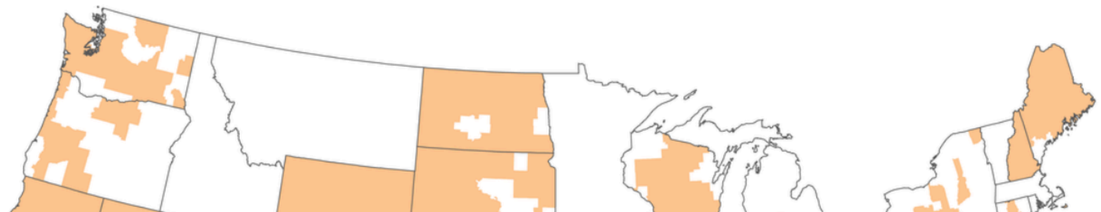
At this quarry, how to spend G.O.P. tax cut isn't set in stone.

The failure to repeal Obamacare provided Senate Republicans with a road map of what not to do when pursuing their tax agenda.

Tom Brenner/The New York Times

Republicans Try to Avert Federal Shutdown, Due Within Days

- With government funding set to expire Friday, Republicans are pushing for a stopgap spending measure that would extend



Opinion

Rob Kim/Getty Images

Billy Bush
Yes, Donald Trump, You Said That

The president is currently engaging in some revisionist history.

Dec. 3

Robert P. George and Sherif

The New York Times

Dec. 4, 2017

G.O.P. TAX PLAN

Tom Brenner/The New York Times

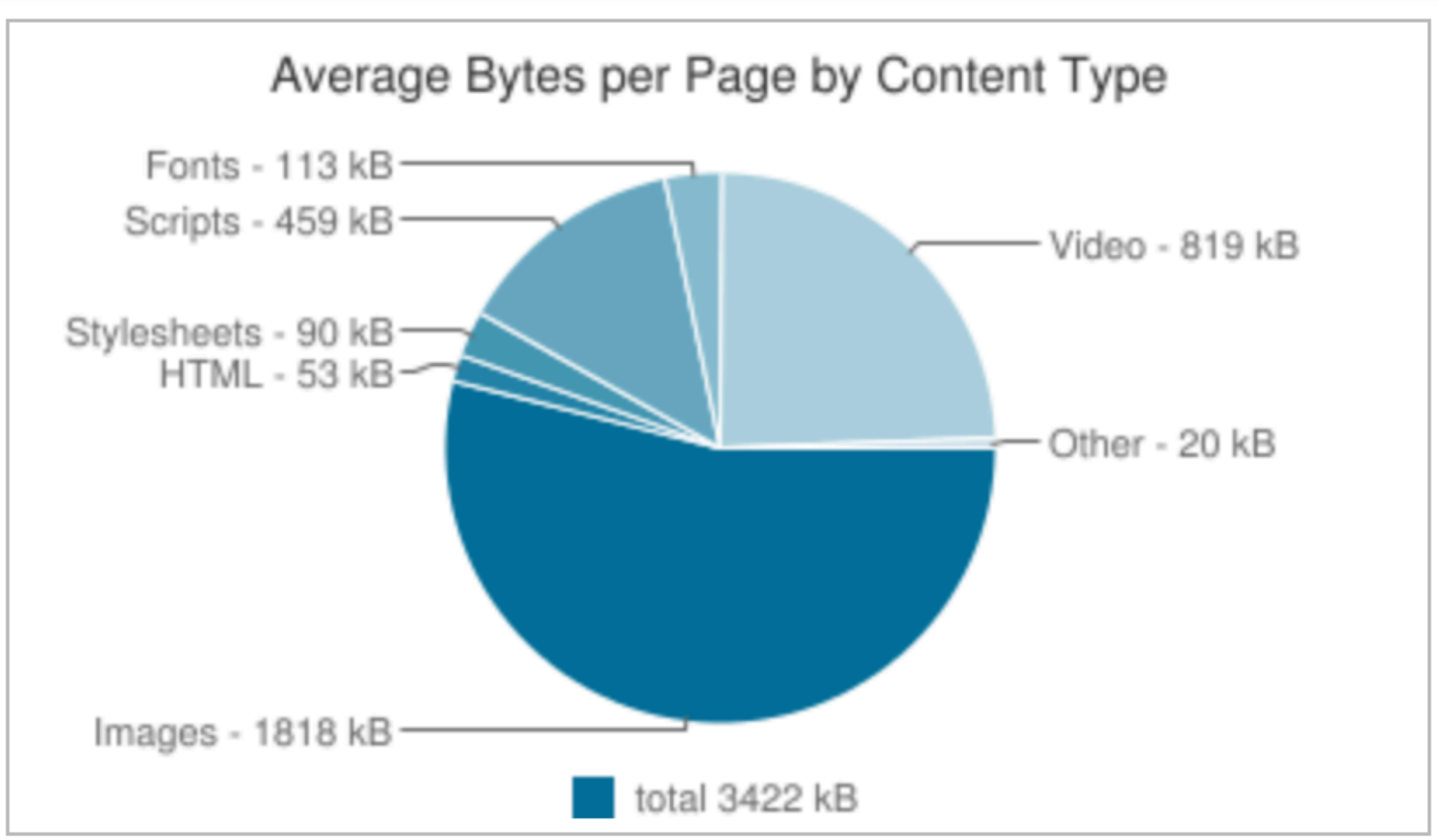
Adverse Report Threatened Tax Bill, So G.O.P. Went on Attack

- A Republican requirement that Congress consider the full cost of major legislation threatened to derail the party's $1.5 trillion tax rewrite last week.
- Lawmakers went on the offensive to discredit the agency performing the analysis.

Evaluate images case-by-case

- Does the image reiterate information found in the surrounding text?
- Is the image necessary to understand the surrounding content?
- Does the image contain text?
- Is the image a graph, chart, or table?
- Could the content of the image be presented in a different format that would not require an image?
- Is the image purely presentational?

53% of the average web page



Source: Internet Archive

And they don't always fit

Source: *The Outline*

If you can avoid using an image, do it

If you need an image, choose the best format

Quick format recap

- **GIF**
  - for images with large swaths of solid colors
  - Binary transparency
- **JPG**
  - For photographs and images with gradations of color
  - Can be compressed (introduces artifacts)

Quick format recap

- **PNG (8-Bit)**
  - Alternative to GIF
  - Can support alpha transparency (with the right creation software)
- **PNG (24-bit)**
  - Alternative to JPG
  - Usually larger than JPGs
  - Supports alpha tranparency

Quick format recap

- **WebP**
  - Newer format, not universally supported
  - Smaller than JPGs and 24-bit PNGs
  - Support alpha transparency
  - and so much more…

Sometimes images are “nice to have”

Who judged this thing?

Who judged this thing?

- Rachel Andrew
- Lara Hogan
- Mat Marquis
- Heydon Pickering
- Jen Simmons
- Una Kravets

How it works

```
<li class="gallery__item h-card"
    data-img="/i/j/r.jpg||y"
    >
  <a href="https://twitter.com/rachelandrew">
    <b>Rachel Andrew</b>
  </a>
</li>
```

image path

no alt necessary

prepend to this list item

Getting CSS & JS in sync

```
var $watcher = document.createElement('div'),
    re = /['"]/g;

$watcher.setAttribute( 'id', 'getActiveMQ-watcher' );
$watcher.style.display = 'none';
document.body.appendChild( $watcher );

window.getActiveMQ = function() {
  return window.getComputedStyle( $watcher, null )
               .getPropertyValue( 'font-family' )
               .replace( re, '' );
};
```

Getting CSS & JS in sync

```
#getActiveMQ-watcher { font-family: global; }
@media (min-width: 15em) {
  #getActiveMQ-watcher { font-family: tiny; }
}
@media (min-width: 20em) {
  #getActiveMQ-watcher { font-family: small; }
}
@media (min-width: 30em) {
  #getActiveMQ-watcher { font-family: medium; }
}
@media (min-width: 40em) {
  #getActiveMQ-watcher { font-family: large; }
}
@media (min-width: 48.75em) {
  #getActiveMQ-watcher { font-family: larger; }
}
@media (min-width: 60em) {
  #getActiveMQ-watcher { font-family: full; }
}
```

Getting CSS & JS in sync

```
var $watcher = document.createElement('div'),
    re = /['"]/g;

$watcher.setAttribute( 'id', 'getActiveMQ-watcher' );
$watcher.style.display = 'none';
document.body.appendChild( $watcher );

window.getActiveMQ = function() {
  return window.getComputedStyle( $watcher, null )
               .getPropertyValue( 'font-family' )
               .replace( re, '' );
};
```

Getting CSS & JS in sync

```
var MQ = getActiveMQ();

if ( MQ == 'larger' || MQ == 'full' ) {
  lazyLoadImages();
}
```

How it works

```
<li class="gallery__item h-card"
    data-img="/i/j/r.jpg||y"
    >
  <a href="https://twitter.com/rachelandrew">
    <b>Rachel Andrew</b>
  </a>
</li>
```

Result!

```
<li class="gallery__item h-card"
    data-img="/i/j/r.jpg||y"
    data-imaged="true"
    >
  <img src="/i/j/r.jpg" alt="">
  <a href="https://twitter.com/rachelandrew">
    <b>Rachel Andrew</b>
  </a>
</li>
```

Oh wait...
optimize everything

Rachel Andrew

Source: 38 kB JPG

B&W: 35 kB JPG (-7%)

Crop & Resize: 12 kB JPG (-68%)

Blur & optimize: 9 kB JPG (-76%)

Who judged this thing?

JPG: 9 kB (-76%)

WebP: 4 kB (-89%)

Not every browser supports WebP

Indicating alternate formats

```
<picture>
  <source type="image/webp" srcset="my.webp">
  <img src="my.jpg" alt="Alt text goes here">
</picture>
```

first choice if WebP is supported

fallback image if it isn't

Indicating alternate formats

```
<picture>
  <source type="image/svg+xml"
          srcset="my.svg">
  <source type="image/webp" srcset="my.webp">
  <img src="my.jpg" alt="Alt text goes here">
</picture>
```

Indicating alternate formats

```
<li class="gallery__item h-card"
    data-img="/i/j/r.jpg||y"
    data-imaged="true"
    >
  <picture>
    <source type="image/webp"
            srcset="/i/j/r.webp">
    <img src="/i/j/r.jpg" alt="">
  </picture>
  <a href="https://twitter.com/rachelandrew">
    <b>Rachel Andrew</b>
  </a>
</li>
```

Steps for better performance

1. Use native features whenever possible
2. Only include assets you actually need
3. Optimize everything
4. Think about when you load assets
5. Consider how you load assets
6. Only load assets when they add value

Every choice we make affects our users' experiences

Let’s spend our time to save it for our users

Speedy performance increases accessibility for everyone

Thank you!

@AaronGustafson

aaron-gustafson.com

slideshare.net/AaronGustafson