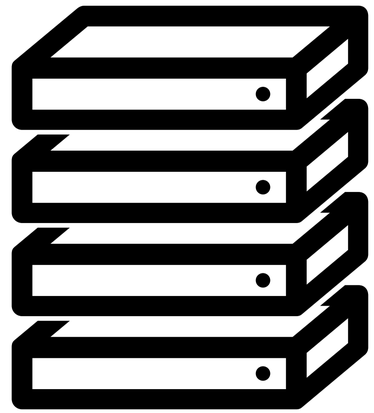
Start Render, FCP

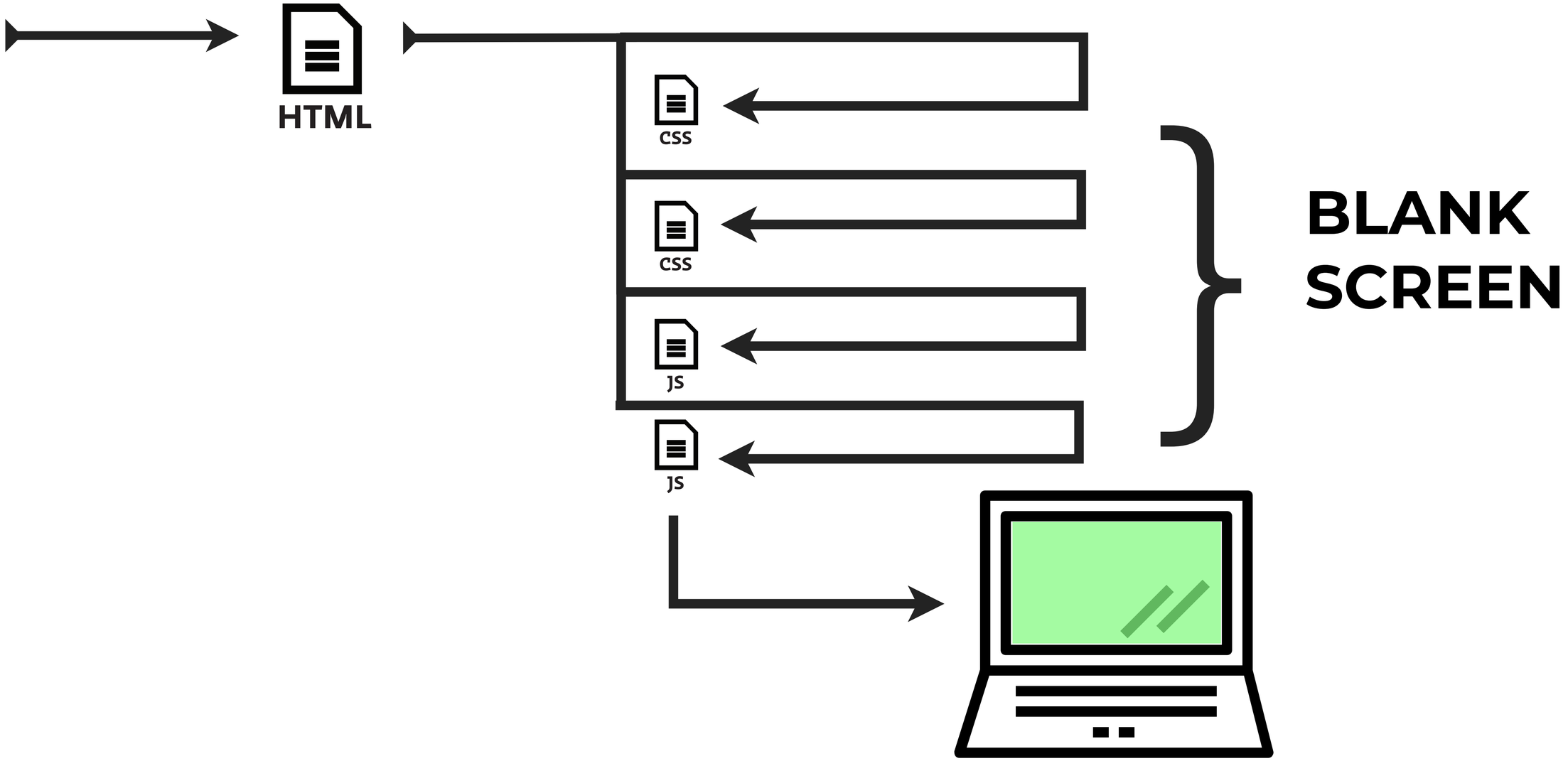
Initial Paints

The first time pixels and content start to become visible to the user.

Example.com please!



Sure! Here's the HTML first.


HTML
CSS
CSS
JS
JS
BLANK
SCREEN

Scott Jehl
@scottjehl

I was curious how long browsers these days will wait for a JavaScript file to load before displaying content that comes after it in the page. So I tested it!

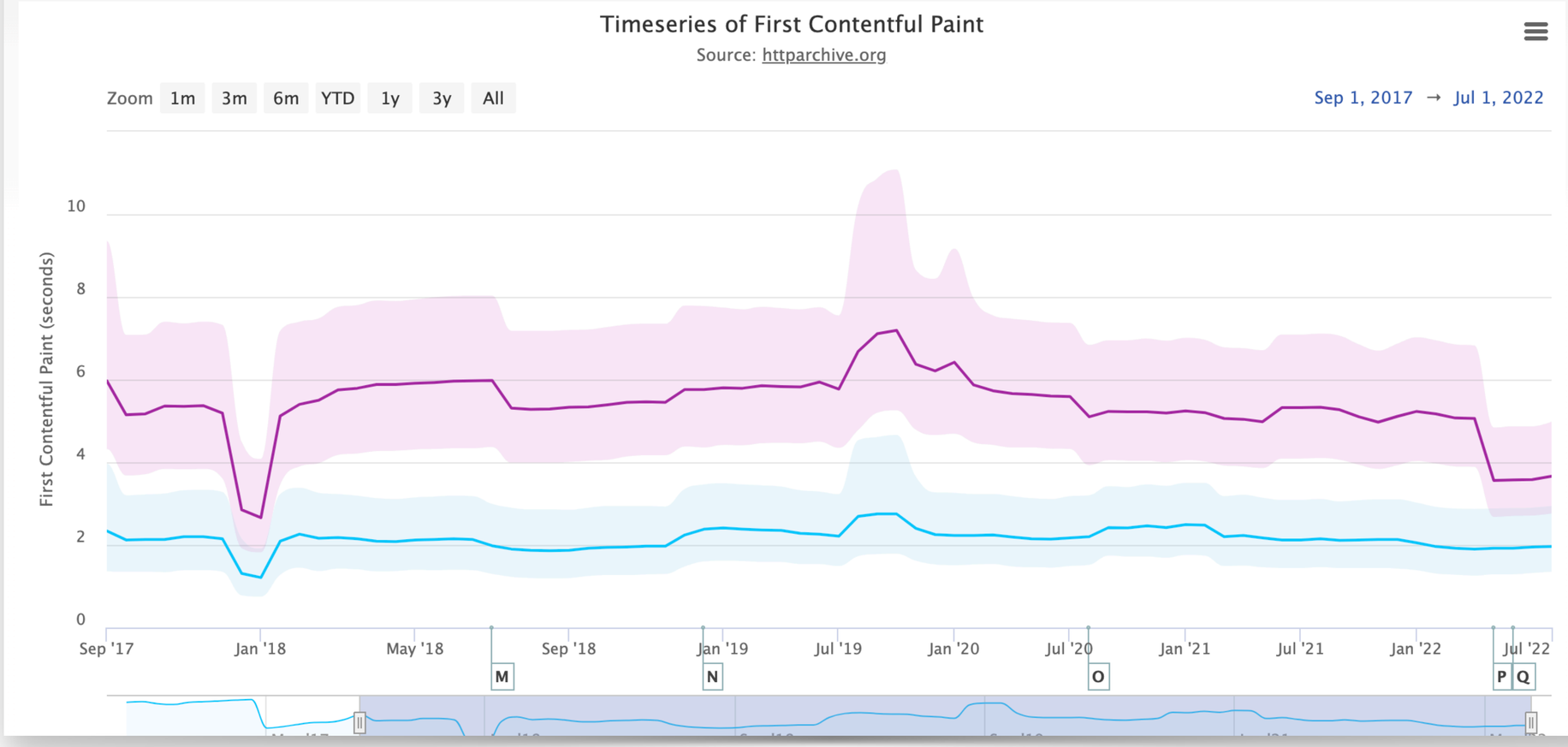
Chrome: 30s
Firefox: 30s
Edge: 20s
IE 11: 7s
Safari: 60s
Android Chrome: 63s
iOS: 75s

Test runs tinyurl.com/swngjl3

First Contentful Paint

The number of seconds from the time the navigation started until the page's primary content appears on the screen.

MEDIAN DESKTOP	MEDIAN MOBILE
2.0 seconds	3.7 seconds
▼13.0%	▼38.3%



https://httparchive.org/reports/loading-speed

The render blockers

```
<head>
    <link rel="stylesheet" href="site.css">

    <script src="site.js"></script>



</head>
```

Unblocking render: async JS

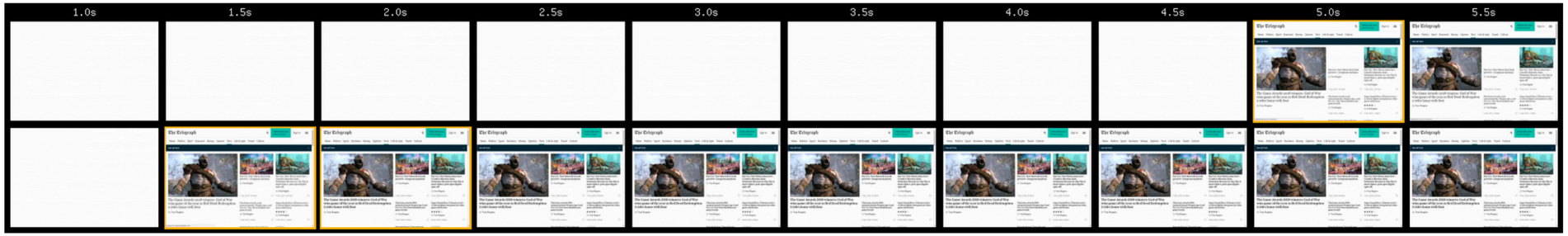
```
<head>
      <script src="site.js" defer></script>

      <script src="site.js" async></script>



</head>
```

Async or Defer?

- **Async:** load in parallel and execute the script whenever it arrives.
- **Defer:** load in parallel and execute the script later, just before domContentLoaded, in the order referenced

Deferring all JavaScript

The single biggest improvement (and easiest to implement technically) came from deferring all JavaScript, including our own, by adding the defer attribute to each script tag.



WebPageTest filmstip comparing before and after deferred scripts.

https://medium.com/the-telegraph-engineering/improving-third-party-web-performance-at-the-telegraph-a0a1000be5

Another way to defer…

```
<head>
        <script src="site.js" type="module"></script>




</head>
```

And another...

```
<head>
    <script>
    var script = document.createElement('script');
    script.src = "site.js";
    script.async = false;
    document.body.append(script);
    </script>
</head>
```

...which is nice for conditional loading

```
<head>
    <link rel="stylesheet" href="site.css">
    <script>
    if( …some condition… ){
      var script = document.createElement('script');
      script.src = "site.js";
      script.async = false;
      document.body.append(script);
    }
    </script>
```

Or reference it later.

```
…

<script src="site.js"></script>
</body>
```

Unblocking render: async CSS

```
<head>
    <link rel="stylesheet" href="site.css"
    media="print">




</head>
```

Unblocking render: defer

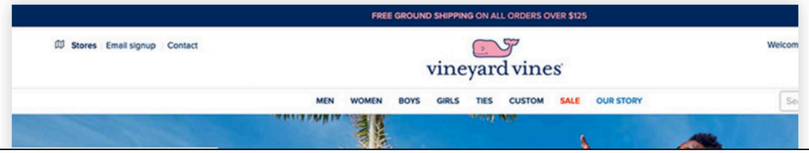
```
<head>
    <link rel="stylesheet" href="site.css"
    media="print" onload="this.media='all'">
</head>
```

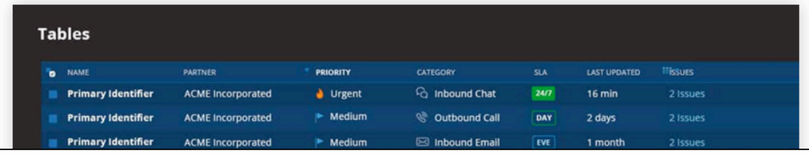
filament group

ABOUT WORK BLOG CODE HIRE US

we design
responsive, accessible,
performant, resilient, & beautiful **websites.**

FEATURED PROJECTS





Which JavaScript is **“critical?”**

- Ideally, none! But…
- Feature tests
- Polyfills
- File loaders
- Conditional logic to bootstrap the page

Enhancing Optimistically

Posted by **Scott** on 06/10/2016

Every so often, we come across ways to improve our more well-trodden core progressive enhancement patterns. Sometimes, we'll utilize a new web standard to address problems we'd previously approached in a less-optimized manner, while other times we'll make adjustments to address browser-or-network conditions that could be handled in more fault-tolerant ways. Recently, I came across an example of the latter, and this post will document a small but meaningful way I worked to accommodate it.

Serving Condiments

For **quite a while now**, we've been progressively enhancing sites using a pattern that these days many of us refer to as "Cutting the Mustard," per **Tom Maslen's great metaphor**. As the pattern goes, we run a series of feature tests relevant to

https://www.filamentgroup.com/lab/enhancing-optimistically.html

Avoid FOUC with an early class

```
<script>
window.document.documentElement.className += "
enhanced";
</script>

<style>
.foo {
  /* basic styles for .foo go here */
}
.enhanced .foo {
  /* enhanced styles for .foo go here */
}
</style>
```

```
if( "querySelector" in window.document && "addEventListener" in wind
  // This is a capable browser, let's improve the UI further!
  window.document.documentElement.className += " enhanced";

  // load the enhanced scripting
  loadJS( "/path/to/enhancements.js" );

  // set a timeout to degrade the ui after 8 seconds
  setTimeout( function(){
    // remove the enhanced class
    window.document.documentElement.className = window.document.docu
  }, 8000 );
}
```

https://www.filamentgroup.com/lab/enhancing-optimistically.html


Music Choice

Inlining files

“

If the external CSS resources are small, you can insert those directly into the HTML document, which is called inlining.

PageSpeed Insights

https://developers.google.com/speed/docs/insights/OptimizeCSSDelivery

Inlining CSS

```
<head>
    <style>
    .header { background: #09878}
    h1 { font-size: 1.2em; col… }
    h2 { margin: 0; }
    …
    </style>
</head>
```

“

In the case of a large CSS file,... Identify and “inline” the CSS necessary for rendering the **above-the-fold** content

PageSpeed Insights

https://developers.google.com/speed/docs/insights/OptimizeCSSDelivery

Thinking Critically

FREE GROUND SHIPPING ON ALL ORDERS OVER $125

Stores | Email signup | Contact

vineyard vines

Welcome, Sign In

Bag (5)

MEN WOMEN BOYS GIRLS TIES SALE GIFTS OUR STORY MORE

Search...

SUMMER
ON THE COAST

This season's styles are ready for your Good Life!

- Tops
- Bottoms
- Swim
- Accessories

SHOP BY CATEGORY

MEN | WOMEN | BOYS
GIRLS | CUSTOM | SALE

Shark Week™ is celebrating its 30th anniversary, we're celebrating our 20th, and we've made some killer gear for some epic shows.

SHOP COLLECTION

FATHER'S DAY IS JUNE 17th! Get Dad the coolest gear and beat the heat with some performance gear built for The Good Life!

SHOP PERFORMANCE

FREE GROUND SHIPPING ON ALL ORDERS OVER $125

Stores | Email signup | Contact

vineyard vines

Welcome, Sign In

Bag (5)

MEN WOMEN BOYS GIRLS TIES SALE GIFTS OUR STORY MORE

Search...

SUMMER
ON THE COAST

This season's styles are ready for your Good Life!

- Tops
- Bottoms
- Swim
- Accessories

SHOP BY CATEGORY

MEN | WOMEN | BOYS
GIRLS | CUSTOM | SALE

Shark Week™ is celebrating its 30th anniversary, we're celebrating our 20th, and we've made some killer gear for some epic shows.

SHOP COLLECTION

FATHER'S DAY IS JUNE 17th! Get Dad the coolest gear and beat the heat with some performance gear built for The Good Life!

SHOP PERFORMANCE

Full CSS

```
.header { background: #09878}
h1 { font-size: 1.2em; col… }
h2 { margin: 0; }
ol { color: red; }
li { color: blue; backgrou… }
li:hover { color: purple; … }
li:first-child { color: gr… }
li:last-child { color: pin… }
.footer { border-top: 1px … }
.copyright { font-size: 1.… }
.legal { font-size: 1.… }
.smallprint { font-size: 1.… }
.social { font-size: 1.… }
```

Critical Home

```
.header { background: #09878}
h1 { font-size: 1.2em; col… }
h2 { margin: 0; }
ol { color: red; }
li { color: blue; backgrou… }
li:first-child { color: gr… }
```

GitHub This repository Search Explore Features Enterprise Blog Sign up Sign in

filamentgroup / grunt-criticalcss ★ Star 230 Fork 9

Grunt wrapper for criticalcss

35 commits 1 branch 17 releases 5 contributors

branch: master grunt-criticalcss / +

Update README.md

jefflembeck authored 5 days ago latest commit 30a6e31c61

tasks	Fix buffer bug, update to 0.4.1	5 days ago
test	Update criticalcss	a month ago
.gitignore	gitignore	4 months ago
.jshintrc	Scaffold	4 months ago
Gruntfile.js	Move up to 0.4, add support for buffer control for memory issues	6 days ago
LICENSE-MIT	Scaffold	4 months ago
README.md	Update README.md	5 days ago

https://github.com/filamentgroup/grunt-criticalcss

README.md

Code
Issues 2
Pull Requests 0
Pulse
Graphs

HTTPS clone URL
https://github.com/f

You can clone with HTTPS or Subversion.

Clone in Desktop
Download ZIP

Extracting critical CSS

```
criticalcss: {
  home: {
    options:  {
      outputfile : 'css/critical/critical-home.css',
      filename : 'all.css',
      url : 'http://fgwebsite.local'
    }
  },
  services: {
    options:  {
      outputfile : 'css/critical/critical-services.css',
      filename : 'all.css',
      url : 'http://fgwebsite.local/services/'
    }
  },
  about: {
```

Inlining lightweight CSS

```
<head>
    <style>
      <% include “critical-home.css” %>
    </style>
</head>
```

Inlining critical CSS, async the rest

```
<head>
    <style>
      <% include “critical-home.css” %>
    </style>

    <link rel="stylesheet" href="site.css"
    media="print" onload="this.media='all'">
</head>
```

Break files by global / local

```
<head>
    <link rel="stylesheet" href="global.css">
    <link rel="stylesheet" href="homepage.css">

    <script src="global.js"></script>
    <script src="homepage.js"></script>

</head>
```

Break files by global / local

```
<head>
    <link rel="stylesheet" href="global.css">
    <link rel="stylesheet" href="product.css">

    <script src="global.js"></script>
    <script src="product.js"></script>

</head>
```

Splitting in other ways..

```
<head>
    <link rel="stylesheet" href="global.css">

    <link rel="stylesheet" href="global-small.css"
    media="(max-width: 400px)">

    <link rel="stylesheet" href="global-large.css"
    media="(min-width: 401px)">
```

To recap, free up first paint times:

- Identify which css & JS is critical to block render
- Serve those critical css and JS files either by:
  - Inlining
  - Referencing in contextual ways by template, media queries, etc.
- Load anything async/deferred that you can