BUILDING PERFORMANT WEBSITES
WHY ARE WE EVEN HERE?
OUR EXPERIENCE WITH THE WEB IS MADE UP OF A SERIES OF INTERACTIONS.
Walmart

100ms

1% REVENUE
2.2s 15.4% CONVERSIONS
Etsy

160kb

12%

BOUNCE RATE
Google

500MS  25%

SEARCHES
NETFLIX

GZIP

↓43%

BANDWIDTH BILL
ONE SECOND
SLOWER
CUSTOMER SATISFACTION
7% CONVERSIONS
$2,500,000
WPO stats

Case studies and experiments demonstrating the impact of web performance optimization (WPO) on user experience and business metrics.

Instagram increased impressions and user profile scroll interactions by decreasing the response size of the JSON needed for displaying comments (by 33% for the median and 50% for the 95th percentile for the main endpoint).

TRAC Research found, in a survey of 300 companies, that the average revenue loss for an hour of downtime was $21,000. For the same set of companies, average revenue loss due to an hour of slow performance (defined as load times exceeding 4.4 seconds) was $4,100. Website slowdowns occurred ten times more often than outages.
Estimate your revenue impact

Mobile site speed can affect conversion rates and as a result - revenue. Fill in your information to see how improving your site speed could impact revenue.

Impact Calculator

<table>
<thead>
<tr>
<th>Domain</th>
<th>Current Speed (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>amazon.com</td>
<td>1.8</td>
</tr>
</tbody>
</table>

Average Monthly Visitors: 1,500,000, Average Order Value ($): $50, Conversion Rate (%): 3%

Minimum Speed (0.6s)  Current Speed (1.8s)

$927,553 USD

Potential annual revenue impact from improving speed by 0.6 seconds
Wow! doing some research for my Delta V talk and found a website that costs $4.41!
I could buy a pint or a cup of coffee instead of downloading this website.
What Does My Site Cost?

Find out how much it costs for someone to use your site on mobile networks around the world.

WebPageTest Test ID

Site URL

OR

GRAB THE RESULTS

OR

TEST MY SITE

According to the June 15, 2017 run of HTTP Archive, the average site now weighs 2987kb. Here's the approximate cost around the world:
### Cost in USD, PPP

(POSTPAID DATA)

This is the cost of the site based on data from the [ITU](https://www.itu.int) and [World Bank](https://www.worldbank.org). The cost of data is standardized based on the PPP factor. Prices were collected from the operator with the largest marketshare in the country, using the least expensive plan with a (minimum) data allowance of 500 MB over (a minimum of) 30 days. Prices include taxes. Because these numbers are based on the least expensive plan, they are **best case scenarios**.

<table>
<thead>
<tr>
<th>Country</th>
<th>Cost (in USD, PPP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canada</td>
<td>$3.03</td>
</tr>
<tr>
<td>Switzerland</td>
<td>$2.01</td>
</tr>
<tr>
<td>Japan</td>
<td>$1.94</td>
</tr>
<tr>
<td>Germany</td>
<td>$1.64</td>
</tr>
<tr>
<td>United States</td>
<td>$1.31</td>
</tr>
</tbody>
</table>

This exceeds the $1.25 per day poverty level. Approximately:  
- 281,280 people in Canada live below this level.  
- 509,200 people in Japan live below this level.  
- 241,860 people in Germany live below this level.  
- 5,420,994 people in United States live below this level.
THE WEB WORLDWIDE

INDIA

THE PEOPLE

https://webworldwide.io

1.3 billion
CONNECTIVITY

- **23.4%**
  - Percentage of people with access to 4Mbps+ broadband
  - Source: Akamai State of the Internet

- **3465 kbps**
  - Average connection speed
  - Source: Akamai State of the Internet

- **26.0**
  - Internet users (per 100 people)
  - Source: World DataBank

- **17.1 million**
  - Fixed broadband subscriptions
  - Source: World DataBank

- **1.2**
  - Source: World DataBank

- **6.8**
  - Source: World DataBank
**AFFORDABILITY**

- **$3.28**
  - Mobile-broadband price, postpaid, 500MB (USD)
  - Source: ITU Measuring the Information Society Report

- **$11.41**
  - Mobile-broadband price, postpaid, 500MB (PPP $)
  - Source: ITU Measuring the Information Society Report

- **2.51**
  - Mobile-broadband price, postpaid, 500MB (% of GNI per capita)
  - Source: ITU Measuring the Information Society Report

- **$3.24**
  - Mobile-broadband price, prepaid, 500MB (USD)
  - Source: ITU Measuring the Information Society Report
WHERE DO WEBSITES COME FROM?
HTML → DOM → Javascript → CSS → CSSOM
1 BANDWIDTH
2 LATENCY
It's the Latency, Stupid

Stuart Cheshire, May 1996.

(Revised periodically)

Years ago David Cheriton at Stanford taught me something that seemed very obvious at the time -- that if you have a network link with low bandwidth then it's an easy matter of putting several in parallel to make a combined link with higher bandwidth, but if you have a network link with bad latency then no amount of money can turn any number of them into a link with good latency.

It's now many years later, and this obvious fact seems lost on the most companies making networking hardware and software for the home. I think it's time it was explained again in writing.

Fact One: Making more bandwidth is easy.

Imagine you live in a world where the only network connection you can get to your house is a 33kbit/sec modem running over a telephone line. Imagine that this is not enough for your needs. You have a problem.

The solution is easy. You can get two telephone lines, and use them together in parallel, giving you a total of 66kbit/sec. If you need even more you can get ten telephone lines, giving you 330kbit/sec. Sure, it's expensive, and having ten modems in a pile is inconvenient, and you may have to write your own networking software to share the data evenly between the ten lines, but if it was important enough to you, you could get it done.

It may not be cheap, but at least it's possible.

People with ISDN lines can already do this. It's called "bonding" and it uses two 56 (or 64) kbit/sec ISDN channels in parallel to give you a combined throughput of 112 (or 128) kbit/sec.

Fact Two: Once you have bad latency you're stuck with it.

If you want to transfer a large file over your modem it might take several seconds, or even minutes. The less data you send, the less time it takes, but there's a limit. No matter how small the amount of data, for any particular network device there's always a minimum time that you can never beat. That's called the latency of the device. For a typical Ethernet connection the latency is usually about 0.3ms (milliseconds -- thousandths of a second). For a typical modem link the latency is usually about 100ms, about 300 times worse than Ethernet.

If you wanted to send ten characters over your 33kbit/sec modem link you might think the total transmission time would be:

\[
\frac{80 \text{ bits}}{33000 \text{ bits per second}} = 2.4 \text{ms.}
\]
AVERAGE CONNECTION SPEED

Source: http://bit.ly/1WRjumM
4,148 KM
4,148 KM
299,792 KM/S
4,148 km / 299,792 km/s
14MS
ping 66.111.3.74
traceroute 66.111.3.74
tracert 66.111.3.74
BANDWIDTH IMPACT

Source: http://bit.ly/1rNCOiC
LATENCY IMPACT

Source: http://bit.ly/1rNCOlC
DNS
Lookup
DNS Lookup

www.amazon.com
DNS Lookup

176.32.103.205
DNS Lookup

BROWSER CACHE
OS CACHE
ROUTER CACHE
ISP DNS CACHE
RECURSIVE SEARCH
Histogram: AsyncDNS.ResolveSuccess recorded 9358 samples, average = 94.7 (flags = 0x1)

<table>
<thead>
<tr>
<th>Value</th>
<th>Count</th>
<th>Percent</th>
<th>Percentile</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0.0%</td>
<td>0.1%</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>116</td>
<td>1.2%</td>
<td>0.1%</td>
</tr>
<tr>
<td>58</td>
<td>383</td>
<td>4.1%</td>
<td>1.3%</td>
</tr>
<tr>
<td>58</td>
<td>389</td>
<td>4.2%</td>
<td>5.4%</td>
</tr>
<tr>
<td>58</td>
<td>178</td>
<td>1.9%</td>
<td>9.6%</td>
</tr>
<tr>
<td>58</td>
<td>157</td>
<td>1.7%</td>
<td>11.5%</td>
</tr>
<tr>
<td>58</td>
<td>1358</td>
<td>14.5%</td>
<td>13.1%</td>
</tr>
<tr>
<td>58</td>
<td>2060</td>
<td>22.0%</td>
<td>27.6%</td>
</tr>
<tr>
<td>67</td>
<td>1065</td>
<td>11.4%</td>
<td>49.7%</td>
</tr>
<tr>
<td>77</td>
<td>978</td>
<td>10.5%</td>
<td>61.0%</td>
</tr>
<tr>
<td>89</td>
<td>873</td>
<td>9.3%</td>
<td>71.5%</td>
</tr>
<tr>
<td>89</td>
<td>541</td>
<td>5.8%</td>
<td>80.8%</td>
</tr>
<tr>
<td>103</td>
<td>312</td>
<td>3.3%</td>
<td>86.6%</td>
</tr>
<tr>
<td>119</td>
<td>291</td>
<td>3.1%</td>
<td>89.9%</td>
</tr>
<tr>
<td>137</td>
<td>232</td>
<td>2.5%</td>
<td>93.0%</td>
</tr>
<tr>
<td>156</td>
<td>127</td>
<td>1.4%</td>
<td>95.5%</td>
</tr>
<tr>
<td>182</td>
<td>59</td>
<td>0.6%</td>
<td>96.9%</td>
</tr>
<tr>
<td>210</td>
<td>54</td>
<td>0.6%</td>
<td>97.5%</td>
</tr>
<tr>
<td>242</td>
<td>35</td>
<td>0.4%</td>
<td>98.1%</td>
</tr>
<tr>
<td>279</td>
<td>28</td>
<td>0.3%</td>
<td>98.5%</td>
</tr>
<tr>
<td>322</td>
<td>28</td>
<td>0.3%</td>
<td>98.8%</td>
</tr>
<tr>
<td>372</td>
<td>19</td>
<td>0.2%</td>
<td>98.9%</td>
</tr>
</tbody>
</table>

CHROME://HISTOGRAMS/
<table>
<thead>
<tr>
<th>Value</th>
<th>Count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>0.0%</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.0%</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.0%</td>
</tr>
<tr>
<td>24</td>
<td>116</td>
<td>1.2%</td>
</tr>
<tr>
<td>28</td>
<td>383</td>
<td>4.1%</td>
</tr>
<tr>
<td>32</td>
<td>389</td>
<td>4.2%</td>
</tr>
<tr>
<td>37</td>
<td>178</td>
<td>1.9%</td>
</tr>
<tr>
<td>43</td>
<td>157</td>
<td>1.7%</td>
</tr>
<tr>
<td>50</td>
<td>1358</td>
<td>14.5%</td>
</tr>
<tr>
<td>58</td>
<td>2060</td>
<td>22.0%</td>
</tr>
<tr>
<td>67</td>
<td>1065</td>
<td>11.4%</td>
</tr>
<tr>
<td>77</td>
<td>978</td>
<td>10.5%</td>
</tr>
<tr>
<td>89</td>
<td>873</td>
<td>9.3%</td>
</tr>
<tr>
<td>103</td>
<td>541</td>
<td>5.8%</td>
</tr>
<tr>
<td>119</td>
<td>312</td>
<td>3.3%</td>
</tr>
<tr>
<td>137</td>
<td>291</td>
<td>3.1%</td>
</tr>
<tr>
<td>156</td>
<td>232</td>
<td>2.5%</td>
</tr>
<tr>
<td>182</td>
<td>127</td>
<td>1.4%</td>
</tr>
<tr>
<td>210</td>
<td>59</td>
<td>0.6%</td>
</tr>
<tr>
<td>242</td>
<td>54</td>
<td>0.6%</td>
</tr>
<tr>
<td>279</td>
<td>35</td>
<td>0.4%</td>
</tr>
<tr>
<td>322</td>
<td>28</td>
<td>0.3%</td>
</tr>
<tr>
<td>372</td>
<td>28</td>
<td>0.3%</td>
</tr>
<tr>
<td>429</td>
<td>19</td>
<td>0.2%</td>
</tr>
<tr>
<td>495</td>
<td>6</td>
<td>0.1%</td>
</tr>
<tr>
<td>571</td>
<td>14</td>
<td>0.1%</td>
</tr>
<tr>
<td>659</td>
<td>11</td>
<td>0.1%</td>
</tr>
<tr>
<td>761</td>
<td>5</td>
<td>0.1%</td>
</tr>
<tr>
<td>878</td>
<td>5</td>
<td>0.1%</td>
</tr>
</tbody>
</table>

Histogram: AsyncDNS.ResolveSuccess recorded 9358 samples, average = 94.7 (flags = 0x1)
Histogram: AsyncDNS.ResolveSuccess recorded 9358 samples, average = 94.7 (flags = 0x1)

80%+

20-100MS
Future startups will prefetch DNS records for 10 hostnames

<table>
<thead>
<tr>
<th>Host name</th>
<th>How long ago (HH:MM:SS)</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://c.go-mpulse.net/">http://c.go-mpulse.net/</a></td>
<td>21:22</td>
<td>n/a</td>
</tr>
<tr>
<td><a href="http://smashingconf.com/">http://smashingconf.com/</a></td>
<td>23:02</td>
<td>n/a</td>
</tr>
<tr>
<td><a href="http://static.getclicky.com/">http://static.getclicky.com/</a></td>
<td>23:01</td>
<td>n/a</td>
</tr>
<tr>
<td><a href="https://apis.google.com/">https://apis.google.com/</a></td>
<td>23:34</td>
<td>n/a</td>
</tr>
<tr>
<td><a href="https://csi.gstatic.com/">https://csi.gstatic.com/</a></td>
<td>02:32</td>
<td>n/a</td>
</tr>
<tr>
<td><a href="https://lh5.googleusercontent.com">https://lh5.googleusercontent.com</a></td>
<td>23:34</td>
<td>n/a</td>
</tr>
<tr>
<td><a href="https://plus.google.com/">https://plus.google.com/</a></td>
<td>23:34</td>
<td>n/a</td>
</tr>
<tr>
<td><a href="https://ssl.gstatic.com/">https://ssl.gstatic.com/</a></td>
<td>23:35</td>
<td>n/a</td>
</tr>
<tr>
<td><a href="https://www.google.com/">https://www.google.com/</a></td>
<td>23:35</td>
<td>n/a</td>
</tr>
<tr>
<td><a href="https://www.gstatic.com/">https://www.gstatic.com/</a></td>
<td>23:34</td>
<td>n/a</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Host for Page</th>
<th>Page Load Count</th>
<th>Subresource Navigations</th>
<th>Subresource PreConnects</th>
<th>Subresource PreResolves</th>
<th>Expected Connects</th>
<th>Subresource Spec</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://smashingconf.com/">http://smashingconf.com/</a></td>
<td>3</td>
<td>165</td>
<td>2</td>
<td>0</td>
<td>40.059</td>
<td><a href="http://smashingconf.com/">http://smashingconf.com/</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>1.584</td>
<td><a href="http://static.getclicky.com/">http://static.getclicky.com/</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>1.584</td>
<td><a href="https://apis.google.com/">https://apis.google.com/</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>1.584</td>
<td><a href="https://plus.google.com/">https://plus.google.com/</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>5</td>
<td>0</td>
<td>1.168</td>
<td><a href="https://lh5.googleusercontent.com/">https://lh5.googleusercontent.com/</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>0.943</td>
<td><a href="https://apis.google.com/">https://apis.google.com/</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>11</td>
<td>5</td>
<td>0</td>
<td>1.531</td>
<td><a href="https://plus.google.com/">https://plus.google.com/</a></td>
</tr>
</tbody>
</table>
TCP Connection

SYN

SYN ACK

ACK
SYN, GET /file (30kb)
TCP SLOW-START

MAKE SURE WE’VE GOT THE BANDWIDTH BEFORE TRYING TO USE IT
Request

10x TCP Segments
(14600 bytes)
Request

10x ACK
Request

20x TCP Segments
(29200 bytes)
Request

20x ACK
Request

GET /file-2
Request

20x TCP Segments (23360 bytes)
response.set('Server-Timing', '${metric}${duration}${description}');
Server-Timing: db: 125
Server-Timing: serverName; edge.machinename.net
let entries = [], done = false
new PerformanceObserver(function(list, observer) {
  entries = entries.concat(list.getEntries())
  if (done) {
    observer.disconnect()
    beaconEntries(entries)
  }
}).observe({entryTypes: ['server']})
SSL Negotiation

ClientHello
SSL Negotiation

ServerHello
Certificate
ServerHelloDone
SSL Negotiation

ClientKeyExchange
ChangeCipherSpec
Finished
SSL Negotiation

ChangeCipherSpec

Finished
DNS Lookup
TCP Connection
SSL Negotiation
Request
1 USE A CDN
2 USE KEEP-ALIVE
TCP Connection Request
SSL Negotiation Request
Request Request
Request

20x TCP Segments (23360 bytes)
KeepAlive On

<ifModule mod_headers.c>
  Header set Connection keep-alive
</ifModule>
KeepAlive On

```
<ifModule mod_headers.c>
Header set Connection keep-alive
</ifModule>
```
KeepAlive On

<ifModule mod_headers.c>
Header set Connection keep-alive
</ifModule>
3 RESOURCE HINTS
header('Link: <path/to/critical.css>; rel=preload; as=stylesheet');
<link rel="preload" href="/path/to/styles.css"/>
function preload(url) {
    var hint = document.createElement("link");
hint.rel = "preload";
hint.href = url;
document.head.appendChild(hint);
}
Resource Hints: dns-prefetch

Gives a hint to the browser to perform a DNS lookup in the background to improve performance. This is indicated using `<link rel="dns-prefetch" href="http://example-domain.com/">`

<table>
<thead>
<tr>
<th></th>
<th>IE</th>
<th>Edge</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>Opera</th>
<th>iOS Safari</th>
<th>Opera Mini</th>
<th>Android Browser</th>
<th>Chrome for Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>11</td>
<td>13</td>
<td>47</td>
<td>51</td>
<td>9.1</td>
<td>38</td>
<td>9.3</td>
<td>8</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>14</td>
<td>48</td>
<td>52</td>
<td>49</td>
<td>53</td>
<td>TP</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes**: IE9 supported dns-prefetch as prefetch as the former wasn't defined yet.
<link rel="dns-prefetch" href="//host_name_to_prefetch.com"/>
### Resource Hints: preconnect

- **WD**

Gives a hint to the browser to begin the connection handshake (DNS, TCP, TLS) in the background to improve performance. This is indicated using `<link rel="preconnect" href="https://example-domain.com/">`

<table>
<thead>
<tr>
<th></th>
<th>IE</th>
<th>Edge</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>Opera</th>
<th>iOS Safari</th>
<th>Opera Mini</th>
<th>Android Browser</th>
<th>Chrome for Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current aligned</td>
<td>8</td>
<td>13</td>
<td>47</td>
<td>51</td>
<td>9.1</td>
<td>38</td>
<td>9.3</td>
<td>8</td>
<td>4.3</td>
<td>50</td>
</tr>
<tr>
<td>Usage relative</td>
<td>11</td>
<td>14</td>
<td>48</td>
<td>52</td>
<td>10</td>
<td>39</td>
<td>40</td>
<td>50</td>
<td>4.4</td>
<td>50</td>
</tr>
</tbody>
</table>

**Notes**: MS Edge status: Under Consideration

Firefox 39 did not support 'crossorigin' attribute and preconnects were not processed by the preload parser. Both of these features were enabled in Firefox 41.
<link rel="preconnect" href="//host_name_to_prefetch.com">
DNS Lookup 
TCP Connection 
SSL Negotiation 
Request
<link rel="dns-prefetch" href="//host_name_to_prefetch.com">

<link rel="preconnect" href="//host_name_to_prefetch.com">
# Resource Hints: preload

Using `<link rel="preload">`, browsers can be informed to prefetch resources without having to execute them, allowing fine-grained control over when and how resources are loaded.

<table>
<thead>
<tr>
<th></th>
<th>IE</th>
<th>Edge</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>Opera</th>
<th>iOS Safari</th>
<th>Opera Mini</th>
<th>Android Browser</th>
<th>Chrome for Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current aligned</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usage relative</td>
<td>8</td>
<td>13</td>
<td>47</td>
<td>51</td>
<td>9.1</td>
<td>38</td>
<td>9.3</td>
<td>8</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Known issues (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resources (4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feedback</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MS Edge status: Under Consideration
<link rel="preload" href="/style/other.css" as="style"/>
DNS Lookup → TCP Connection → SSL Negotiation → Request
# Resource Hints: prefetch

Informs the browsers that a given resource should be prefetched so it can be loaded more quickly. This is indicated using `<link rel="prefetch" href="(url)">`

<table>
<thead>
<tr>
<th></th>
<th>Current aligned</th>
<th>Usage relative</th>
<th>Date relative</th>
<th>Show all</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IE</td>
<td>Edge</td>
<td>Firefox</td>
<td>Chrome</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>14</td>
<td>51</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>56</td>
<td>55</td>
<td>54</td>
</tr>
</tbody>
</table>

Notes: No notes

Known issues: 0

Resources: 3

Feedback:
<link rel="prefetch" href="/style/other.css" as="style"/>
# Resource Hints: prerender

Gives a hint to the browser to render the specified page in the background, speeding up page load if the user navigates to it. This is indicated using `<link rel="prerender" href="(url)">`

<table>
<thead>
<tr>
<th></th>
<th>Current aligned</th>
<th>Usage relative</th>
<th>Show all</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE</td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Edge</td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Firefox</td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Chrome</td>
<td>29</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>Safari</td>
<td>48</td>
<td>49</td>
<td></td>
</tr>
<tr>
<td>Opera</td>
<td>9.1</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>iOS Safari</td>
<td>8.4</td>
<td>9.2</td>
<td></td>
</tr>
<tr>
<td>Opera Mini</td>
<td>*</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Android Browser</td>
<td>*</td>
<td>4.3</td>
<td>4.4</td>
</tr>
<tr>
<td>Chrome for Android</td>
<td>*</td>
<td>4.4</td>
<td>4.4, 50</td>
</tr>
</tbody>
</table>

Notes: No notes

Known issues: (1)

Resources: (2)

Feedback:
<link rel="prerender" href="//example.com/next-page.html"/>
DNS Lookup

TCP Connection

SSL Negotiation

Request
Intent to Deprecate and Remove: Prerender

Matthew Cary

Other recipients: pa...@chromium.org, dro...@chromium.org, kenji...@chromium.org

Contact emails
pa...@chromium.org, dro...@chromium.org, matt...@chromium.org

Summary
Disable prerendering in chrome. This will not break any sites, but may cause a performance regression.

Motivation
Prerender is difficult to maintain and wastes resources compared to its performance benefit. More efficient ways to achieve the same performance goal are in development, but as they will behave quite differently from prerender we need to resolve any unknown external dependencies by deprecating prerender cleanly.

Compatibility and Interoperability Risk
As prerender is a best-effort performance feature only, there is no compatibility risk. The most performance sensitive use of prerender we know of (invoking prerender through the CCT mayLaunchURL API) will not be immediately deprecated.

Usage Information
Detailed usage statistic are available only through Google Internal UMA counters. In summary, of the times link-rel prerender is triggered, under 15% are used. Approximately 0.2% of all navigations trigger prerender (these counts are from UMA users, a possibly-biased subset of all chrome users). Link-rel prerenders are the non-CCT externally facing prerender API.

Launch Tracking Bug
**NoState Prefetch**

get most of the benefits from prerendering without prerendering

**Status:** Public Draft

Authors: davidben@, pasko@

Last Updated: 2016-11-16

This document: [http://goo.gl/EJTCm](http://goo.gl/EJTCm)
dns-prefetch

Resolve DNS ahead of time

Ex: Third party content where you know the domain, but not the full URL
dns-prefetch
preconnect

TCP Handshake/SSL negotiation ahead of time

Ex: Third party content where you know the domain, but not the full URL
dns-prefetch
preconnect
preload

Fetch a resource ahead of time

Ex: Image or script you know you will need for current page
dns-prefetch
preconnect
preload

**prefetch**

Fetch a resource that will be needed for next navigation

*Ex: Image or script that will likely be needed on the next page*
dns-prefetch
preconnect
preload

prefetch
prerender

Prerender a page in the background for future navigation

Ex: Marketing funnel where next navigation is predictable
<html>
<head>
  <meta charset="UTF-8">
  <title>Hey there</title>
</head>
<body>
</body>
<html>
<head>
  <meta charset="UTF-8">
  <title>Hey there</title>
</head>
<body>
  <h1>Super awesome site</h1>
</body>

<html>
<head>
    <meta charset="UTF-8">
    <title>Hey there</title>
</head>
<body>
    <h1>Super awesome site</h1>
    <div id="main">
        Super...
    </div>
</body>
<html>
<head>
  <meta charset="UTF-8">
  <title>Hey there</title>
</head>
<body>
  <h1>Super awesome site</h1>
  <div id="main">
    <p>Super interesting stuff, I'm sure.</p>
  </div>
</body>
</html>
Network

HTML

DOM

Javascript

CSS

CSSOM

Render Tree

Layout

Paint
<html>
<head>
  <meta charset="UTF-8">
  <title>Hey there</title>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <h1>Super awesome site</h1>
  <div id="main">
    <p>Super interesting stuff, I'm sure.</p>
  </div>
</body>
</html>
#main{
  color: green;
}

h1 {
  font-size: 2em;
}

p{
  color: orange;
}
#main{
  color: green;
}

h1 {
  font-size: 2em;
}

p{
  color: orange;
}
Have a question?

Enter your question here

Amazon's Choice

Amazon's Choice recommends highly rated, well-priced products for "gag gifts for teens"

1.

2.
Archie McPhee - Accoutrements
Accoutrements Handihorse

Amazon's Choice for "gag gifts for teens"

Size:
Pack of 1

$6.23
$18.95 Save $12.72 (67%)

Prime & FREE Shipping on orders above $25.

Only 18 left in stock - order soon.
CSS IS RENDER BLOCKING
<link rel="stylesheet" href="style.css" />

<link rel="stylesheet" href="print.css" media="print" />

<link rel="stylesheet" href="large.css" media="(min-width: 30em)" />
document.write("Take my breath awaaaaaaayyyyyy");

document.getElementById("#main").style.height = "100px";
<script>
<script>
JS IS PARSER
BLOCKING
CSS BLOCKS JS EXECUTION

JS EXECUTION BLOCKS DOM
### async attribute for external scripts - L5

The boolean async attribute on script elements allows the external JavaScript file to run when it's available, without delaying page load first.

#### Current aligned

<table>
<thead>
<tr>
<th>Browser</th>
<th>IE</th>
<th>Chrome</th>
<th>Safari</th>
<th>Opera</th>
<th>iOS Safari</th>
<th>Opera Mini</th>
<th>Android Browser</th>
<th>Chrome for Android</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8</td>
<td>29</td>
<td>8.4</td>
<td>9.2</td>
<td>4.3</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>45</td>
<td>49</td>
<td>9.1</td>
<td>4.4</td>
<td>8</td>
<td>8</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>48</td>
<td>48</td>
<td>38</td>
<td>4.4</td>
<td>8</td>
<td>8</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>50</td>
<td>49</td>
<td>9.3</td>
<td>4.4</td>
<td>8</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

#### Known issues (0) Resources (4) Feedback

Using `script.async = false;` to maintain execution order for dynamically-added scripts isn't supported in Safari 5.0.
<script type="text/javascript" async src="blah.js.php"></script>
<script async>
<script async>
The boolean `defer` attribute on script elements allows the external JavaScript file to run when the DOM is loaded, without delaying page load first.

<table>
<thead>
<tr>
<th>Browser</th>
<th>IE</th>
<th>Edge</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>Opera</th>
<th>iOS Safari</th>
<th>Opera Mini</th>
<th>Android Browser</th>
<th>Chrome for Android</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8</td>
<td>13</td>
<td>47</td>
<td>51</td>
<td>9.1</td>
<td>38</td>
<td>9.3</td>
<td>8</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>45</td>
<td>49</td>
<td>50</td>
<td>4.3</td>
<td>4.4</td>
<td>4.4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes: Partial support in older IE refers to a buggy implementation (see issue).
<script type="text/javascript" defer src="blah.js.php"></script>
<script defer>
<script defer>
#main{
   color: green;
}
h1 {
   font-size: 2em;
}
p{
   color: orange;
}
#main{
    color: green;
}

h1 {
    font-size: 2em;
}

p{
    display: none;
}
DO YOU WANNA BUILD A RENDER TREE?
<html>
  <head>
    <meta charset="utf-8" />
    <title>Best Workshop Ever</title>
    <link rel="stylesheet" type="text/css" href="blah.css" />
  </head>
  <body>
    <h1>Building Performant Websites</h1>
    <div id="main">
      <p>This is the best workshop I have <strong>ever</strong> been in!</p>
      <p class="hide">Seriously.</p>
    </div>
    <script>
      document.getElementById('main').style.backgroundColor = "pink"
    </script>
  </body>
</html>

h1{
  color: green;
}
#main{
  display: inline-block;
}
#main p{
  color: blue;
}
p {
  color: orange;
}
p strong{
  font-weight: bold;
}
.hide{
  display: none;
}
Super..
CRITICAL PATH
Archie McPhee - Accoutrements
Accoutrements Handihorse

Amazon's Choice for "gag gifts for teens"

Size:
Pack of 1

$6.23
$18.95 Save $12.72 (67%)

Fast delivery & FREE Shipping on orders above $25.

Only 18 left in stock - order soon.
ANYTHING IN THE WAY OF INITIAL RENDER IS CRITICAL
<html>
<head>
  <meta charset="UTF-8">
  <title>Hey there</title>
  <link rel="stylesheet" href="style.css" />
  <script type="text/javascript" src="jquereactugular.js"></script>
</head>
<body>
  <h1>Super awesome site</h1>
  <div id="main">
    <p>Super interesting stuff, I'm sure.</p>
  </div>
</body>
</html>
Super awesome site

Super interesting stuff, I'm sure.
Hey there

Super awesome site

Super interesting stuff, I'm sure.
Super awesome site

Super interesting stuff, I'm sure.
Hey there

Super awesome site

Super interesting stuff, I'm sure.
<style type="text/css">
  //critical css inlined and minimized
</style>

<script type="text/javascript">
  loadCSS("/path/to/full/css");
  //set a cookie
</script>
<html>
<head>
  <meta charset="UTF-8">
  <title>Hey there</title>
  <style type="text/css">
    //critical css inlined and minimized
  </style>
  <script type="text/javascript" async src="jqueryreactangular.js"></script>
</head>
<body>
  <h1>Super awesome site</h1>
  <div id="main">
    <p>Super interesting stuff, I'm sure.</p>
  </div>
</body>
</html>
EVERYTHING ELSE
51%

IMAGES
49%
FORMATS
RASTER
PNG, JPG, GIF

VECTOR
SVG
<svg xmlns="http://www.w3.org/2000/svg"
     xmlns:xlink="http://www.w3.org/1999/xlink">
  <rect x="10" y="10" height="100" width="100"
       style="stroke:#ff0000; fill: #0000ff"/>
</svg>
WHEN TO USE VECTOR?

Images made of simple geometric shapes.

ex: logos, icons, etc.
JPEG

Photos and other detailed imagery

Lossy (information/details get lost)
PNG-8/PNG-24

256 PNG-8, no color palette limit PNG-24

Lossless

Transparency
GIF

256-color limit

Lossless

Perfect marriage of high file-size and low quality
Evolution of <img>: Gif without the GIF

by Colin Bendell

tl;dr

- GIFs are awesome but terrible for quality and performance
- Replacing GIFs with `<video>` is better but has perf. drawbacks: not preloaded, uses range requests
- Now you can `<img src=".mp4">` s in Safari Technology Preview
- Early results show mp4s in `<img>` tags display 20x faster and decode 7x faster than the GIF equivalent – in addition to being 1/14th the file size!
- Background CSS video & Responsive Video can now be a “thing”.
- Finally cinemagraphs without the downsides of GIFs!
- Now we wait for the other browsers to catch-up: This post is 46MB on Chrome but 2MB in Safari TP

Intro

I both love and hate animated GIFs. 🎨 🎨
WEBP

Blink-based browsers only
A new image format for the Web

WebP is a modern image format that provides superior lossless and lossy compression for images on the web. Using WebP, webmasters and web developers can create smaller, richer images that make the web faster.

WebP lossless images are 26% smaller in size compared to PNGs. WebP lossy images are 25-34% smaller than comparable JPEG images at equivalent SSIM quality index.

Lossless WebP supports transparency (also known as alpha channel) at a cost of just 22% additional bytes. For cases when lossy RGB compression is acceptable, lossy WebP also supports transparency, typically providing 3× smaller file sizes compared to PNG.

- More Info for Webmasters

How WebP Works

Lossy WebP compression uses predictive coding to encode an image, the same method used by the VP8 video codec to compress keyframes in videos. Predictive coding uses the values in neighboring blocks of pixels to predict the values in a block, and then encodes only the difference.

Lossless WebP compression uses already seen image fragments in order to exactly reconstruct the original image from the compressed data.
<table>
<thead>
<tr>
<th>EXIF DATA</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera make</td>
<td>Canon</td>
</tr>
<tr>
<td>Camera model</td>
<td>Canon EOS-1D X</td>
</tr>
<tr>
<td>Date/Time</td>
<td>2018/03/25 16:39:12</td>
</tr>
<tr>
<td>Resolution</td>
<td>460 x 259</td>
</tr>
<tr>
<td>Flash used</td>
<td>No</td>
</tr>
<tr>
<td>Focal length</td>
<td>165.0mm (35mm equivalent 276mm)</td>
</tr>
<tr>
<td>CCD width</td>
<td>21.49mm</td>
</tr>
<tr>
<td>Exposure time</td>
<td>0.0006 s (1/1600)</td>
</tr>
<tr>
<td>Aperture</td>
<td>f/3.2</td>
</tr>
<tr>
<td>Focus dist.</td>
<td>23.80m</td>
</tr>
<tr>
<td>ISO equiv.</td>
<td>5000</td>
</tr>
<tr>
<td>Exposure bias</td>
<td>-0.67</td>
</tr>
<tr>
<td>Whitebalance</td>
<td>Manual</td>
</tr>
<tr>
<td>Mode</td>
<td>pattern</td>
</tr>
<tr>
<td>Exposure</td>
<td>shutter priority (semi-auto)</td>
</tr>
</tbody>
</table>

There is no GPS info
ImageOptim makes images load faster

Removes bloated metadata. Saves disk space & bandwidth by compressing images without losing quality.

Download for Free

Requires macOS 10.9+. It's Free and Open Source. Donate.

Faster web pages and apps

- Reduces image file sizes — so they load faster and download sooner by applying advanced compression that preserves quality.

Image files scrubbed clean

- Removes invisible junk: private EXIF metadata from digital cameras, embedded thumbnails, comments, and unnecessary color profiles.

Best tools with drag’n'drop

Seamlessly combines all the best image optimization tools: MozJPEG, ppnguant, Pngcrush, 7zip, SVGO and Google Zopfli.
ImageAlpha greatly reduces file sizes of 24-bit PNG files (including alpha transparency) by applying lossy compression and conversion to a more efficient PNG8-alpha format. Such images are compatible with iOS, all browsers, and even degrade well in IE6.

ImageAlpha can achieve better quality than similar function in Macromedia Fireworks by employing latest pngquant and pngquant-s9 and alpha-channel-aware posterizer.

How to use it

Drag true-color PNG image into main part of the window. Small images should be converted instantly. If image is large, there might be progress spinner visible for few seconds.

If the quality isn’t good enough, choose between Median Cut and Posterizer compression type and try Dithered mode.

If quality is great, try reducing number of colors, which will also further reduce file size.
Your Photos on a Diet
Reduce image size by up to 80%, without compromising quality

Photo: Zach Dishner, Photo Resolution: 5184x3456 | Download

Original vs. JPEGmini

https://bit.ly/1o2LqAU
Nodejs-based tool for optimizing SVG vector graphics files

<table>
<thead>
<tr>
<th>Folder</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>bin</td>
<td></td>
<td>5 years ago</td>
</tr>
<tr>
<td>docs/how-it-works</td>
<td>Update ko.md</td>
<td>17 days ago</td>
</tr>
<tr>
<td>examples</td>
<td>Added config example to node example file</td>
<td>2 months ago</td>
</tr>
<tr>
<td>lib</td>
<td>Update tools.js</td>
<td>4 days ago</td>
</tr>
<tr>
<td>plugins</td>
<td>fix</td>
<td>2 days ago</td>
</tr>
<tr>
<td>test</td>
<td>Fixes #915 for collapseGroups (#916)</td>
<td>23 days ago</td>
</tr>
<tr>
<td>.editorconfig</td>
<td>Istanbul + coveralls.io</td>
<td>5 years ago</td>
</tr>
<tr>
<td>.jshintignore</td>
<td>Prefix ids plugin don't add prefix if prefix is false (#907)</td>
<td>9 days ago</td>
</tr>
</tbody>
</table>
1. REQUEST IMAGE
2. DOWNLOAD IMAGE
3. DECODE IMAGE
4. STORE IN MEMORY
5. COPY TO GPU
6. PAINT TO DISPLAY
RGB to YUV
RGB to YUV → Chroma Subsampling
RGB to YUV → Chroma Subsampling → DCT / Quantization
RGB to YUV → Chroma Subsampling → DCT / Quantization → Huffman Encoding
YUV to RGB

Chroma Upsampling

iDCT / Dequantization

Huffman Decoding
Testing the impact of image scaling: Large Sized

The images below are large images (1200px wide). The browser scales them down to 200px.

Image count: 10
2GB RAM
Quad-core 1.5 GHz Krait CPU
Adreno 320 GPU
Testing the impact of image scaling: Large Sized

The images below are large images (1300px wide). The browser scales them down to 330px.

Image count: 10

Resized: 30.38ms
Resized: 30.38ms
Double: 102.77ms
Large (6x): 1,534.99ms
Testing the impact of image scaling: Large Sized

The images below are large images (1200x600). The browser scales them down to 300x.

Image count: 10

Resized: 30.38ms

Double: +238.3%

Large (6x): +4952.6%
RESIZE IMAGES
Request → Decode → Copy to GPU → Display

Store in memory
W \times H \times 4
734,400
4.38 MB
“...25% of new Android phones have only 512MB of RAM.”

Jen Fitzpatrick
VP of product management for Google Maps
RESIZE IMAGES
230,000 x 550px

70,000 x 150px
YUV to RGB  →  Chroma Upsampling  →  iDCT / Dequantization  →  Huffman Decoding
Luma
(light)

Chroma
(color)
Chroma Subsampling
8 total
4:4:4 + Chroma = Image

4:2:2 + Chroma = Image

4:2:0 + Chroma = Image
\[(W \times H \times 3) - (W \times H \times \text{SUBSAMPLE\_LEVEL} \times 2)\]
<table>
<thead>
<tr>
<th>Color Format</th>
<th>Resolution (px x px)</th>
<th>Total Pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGBA</td>
<td>1300 x 1024</td>
<td>5,324,800</td>
</tr>
<tr>
<td>4:4:4</td>
<td>1300 x 1024</td>
<td>3,993,600</td>
</tr>
<tr>
<td>4:2:2</td>
<td>1300 x 1024</td>
<td>2,662,400</td>
</tr>
<tr>
<td>4:2:0</td>
<td>1300 x 1024</td>
<td>1,996,800</td>
</tr>
</tbody>
</table>
4:2:0
memory savings
62.5%
Source: http://time.com/3445111/#1
<table>
<thead>
<tr>
<th>Type</th>
<th>Image Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Time</td>
<td>3.266 ms</td>
</tr>
<tr>
<td>Self Time</td>
<td>3.266 ms</td>
</tr>
<tr>
<td>Image URL</td>
<td>mobile.gannett-cdn.com/-mm-/e05e8a455d93baa2d689973eab3fad3b2c3/c=194-24/USATODAY/USATODAY/535689026075773485-AP-Pope-Climate-Change-Science.jpg</td>
</tr>
<tr>
<td>Type</td>
<td>Image Decode</td>
</tr>
<tr>
<td>-------</td>
<td>--------------</td>
</tr>
<tr>
<td>Total Time</td>
<td>3.266 ms</td>
</tr>
<tr>
<td>Self Time</td>
<td>3.266 ms</td>
</tr>
<tr>
<td>Image URL</td>
<td>mobile.gannett-cdn.com/-mm-/e05e8a455d93ba8a2d6b9753aeb3fad383c62c3/c=194-24/USATODAY/USATODAY/63568926075773485-AP-Pope-Climate-Change-Science.jpg</td>
</tr>
</tbody>
</table>
Psst....need some memory?
Yeah, I’ll take some of that.
YUV to RGB

Chroma Upsampling

iDCT / Dequantization

Huffman Decoding
Testing the impact of image scaling: Large Sized

The images below are large images (7200px x 4666). The browser scales them down to 250px.

Image count: 10
<meta name="viewport" content="width=device-width, minimum-scale=1.0"/>
Testing the impact of image scaling: Large Sized

The images below are large images (1200px wide). The browser scales them down to 200px.

Add ten more images

Image count: 10
Testing the impact of image scaling: Large Sized

The images below are large images (1200px wide). The browser scales them down to 200px.

Image count: 10
Testing the impact of image scaling: Large Sized

The images below are large images (1200px wide). The browser scales them down to 200px.

Add ten more images

Image count: 10
Testing the impact of image scaling: "Large Sized"

The images below are large images (1200px wide). The browser scales them down to 200px.

Image count: 10
Testing the impact of image scaling: Large Sized

The images below are large images (1200px wide). The browser scales them down to 200px.

Add ten more images

Image count: 10
Testing the impact of image scaling: Large Sized

The images below are large images (1200px wide). The browser scales them down to 200px.

Image count: 10

Resized: 5.62ms
Double: 27.74ms
Large (6x): 1534.99ms
Testing the impact of image scaling: Large Sized

The images below are large images (1200px wide). The browser scales them down to 200px.

Add ten more images

Image count: 10

**Resized:** 5.62ms

**Double:** 27.74ms

**Large (6x):** 462.11ms
Testing the impact of image scaling: "Large Sized"

The images below are large images (1200px wide). The browser scales them down to 200px.

Resized: 5.62ms

Double: +393.6%

Large (6x): +8122.6%
<meta name="viewport" content="initial-scale = 1.0, maximum-scale = 1.0"/>

<meta name="viewport" content="initial-scale = 1.0, maximum-scale = 1.0, width=device-width"/>

<meta name="viewport" content="initial-scale = 1.0, maximum-scale = 1.0, width=device-width"/>
1 RESIZE IMAGES
1. RESIZE IMAGES
2. SMALL BREAKPOINTS FOR LARGE IMAGES
1. Resize images
2. Small breakpoints for large images
3. Meta viewport is your friend
1. Resize images
2. Small breakpoints for large images
3. Meta viewport is your friend
4. Use 4:2:0 subsampling
RESPONSIVE
<img src="cat.jpg" alt="cat" srcset="cat-160.jpg 160w, cat-320.jpg 320w, cat-640.jpg 640w, cat-1280.jpg 1280w"/>
<img src="cat.jpg" alt="cat"
srcset="cat-160.jpg 160w,
cat-320.jpg 320w,
cat-640.jpg 640w,
cat-1280.jpg 1280w"
sizes="(max-width: 480px) 100vw,
(max-width: 900px) 33vw,
254px"/>
<picture>
  <source type="image/svg+xml" srcset="logo.xml">
  <source type="image/webp" srcset="logo.webp">
  <img src="logo.png" alt="ACME Corp">
</picture>
Two weeks on a slow 2G connection have been eye-opening. Worst offender so far: Web fonts.
@font-face {
  font-family: "My font";
  font-style: normal;
  font-weight: 400;
  src: <list of sources>;
}
@font-face {
  font-family: "My font";
  font-style: normal;
  font-weight: 400;
  src: <list of sources>;
}

@font-face {
  font-family: "My font";
  font-style: italic;
  font-weight: 400;
  src: <list of sources>;
}
Use not, waste not.
@font-face {
  font-family: "some-font";
}

@media (min-width: 400px) {
  body {
    font-family: "some-font", Georgia, serif;
  }
}
FOUT

FLASH OF UNSTYLED TEXT
FOIT
FLASH OF INVISIBLE TEXT
FOIT vs. FOUT

This demo was created to show the functional differences between FOIT and FOUT. It’s using Open Sans loaded from Google Web Fonts. *(Not sure what FOIT and FOUT are?)* Decide for yourself—which feels faster? Which feels more stable? Take note of the difference in page stability when you group your repaints.

Created by @zachleat. Read more at [FOIT vs. FOUT (the blog post)](https://zachleat.com/foit-vs-fout) or the [Comprehensive Guide to Font Loading Strategies](https://zachleat.com). You can always return to zachleat.com.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Roman</td>
<td>4000</td>
<td>ms</td>
</tr>
<tr>
<td>Bold</td>
<td>4000</td>
<td>ms</td>
</tr>
<tr>
<td>Italic</td>
<td>4000</td>
<td>ms</td>
</tr>
<tr>
<td>Bold Italic</td>
<td>4000</td>
<td>ms</td>
</tr>
</tbody>
</table>

Randomize Load Times
Force a Grouped Repaint
Start Font Load

**FOIT** with 3s Timeout

**FOUT**
FOIT vs. FOUT

This demo was created to show the functional differences between FOIT and FOUT. It's using Open Sans loaded from Google Web Fonts. (Not sure what FOIT and FOUT are?) Decide for yourself—which feels faster? Which feels more stable? Take note of the difference in page stability when you group your repaints.

Created by @zachleat. Read more at FOIT vs. FOUT (the blog post) or the Comprehensive Guide to Font Loading Strategies. You can always return to zachleat.com.

<table>
<thead>
<tr>
<th>Font Style</th>
<th>Time (ms)</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roman</td>
<td>4000</td>
<td></td>
</tr>
<tr>
<td>Bold</td>
<td>4000</td>
<td>Randomize Load Times</td>
</tr>
<tr>
<td>Italic</td>
<td>4000</td>
<td>Force a Grouped Repaint</td>
</tr>
<tr>
<td>Bold Italic</td>
<td>4000</td>
<td>Start Font Load</td>
</tr>
</tbody>
</table>

**FOIT** with 3s Timeout

**FOUT**
<table>
<thead>
<tr>
<th>Browser</th>
<th>Timeout</th>
<th>Fallback</th>
<th>Swap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chrome 35+</td>
<td>3 seconds</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Opera</td>
<td>3 seconds</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Firefox</td>
<td>3 seconds</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Internet Explorer</td>
<td>0 seconds</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Safari</td>
<td>No timeout</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Browser</td>
<td>Timeout</td>
<td>Fallback</td>
<td>Swap</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------</td>
<td>----------</td>
<td>------</td>
</tr>
<tr>
<td>Chrome 35+</td>
<td>3 seconds</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Opera</td>
<td>3 seconds</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Firefox</td>
<td>3 seconds</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Internet Explorer</td>
<td>0 seconds</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Safari</td>
<td>No timeout</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
@font-face {
  font-family: 'League Gothic';
  font-display: auto;
  src: ...
...
}

1. FONT BLOCK
   How long to block

2. FONT SWAP
   How long to swap

3. FONT FAILURE
   When to give up
font-display: auto;

Whatever the browser was gonna do.
font-display: auto;

font-display: fallback;

Tiny block (100ms) then show fallback, small swap.
Block 3s or so, then display fallback. Infinite swap.
Don’t block. Infinite swap.
font-display: auto;
font-display: fallback;
font-display: block;
font-display: swap;
font-display: optional;

Like fallback, but browser can decide not to load.
MINIMIZE VARIANTS
Open Sans

Regular (300)
Italic (300)
Light (400)
Light Italic (400)
Semibold (600)
Semibold Italic (600)
Bold (700)
Bold Italic (700)
Extrabold (800)
Extrabold Italic (800)
<table>
<thead>
<tr>
<th>Style</th>
<th>Actual font</th>
<th>Synthesized</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
</tr>
<tr>
<td>300, italic</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
</tr>
<tr>
<td>400</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
</tr>
<tr>
<td>400, italic</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
</tr>
<tr>
<td>600</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
</tr>
<tr>
<td>600, italic</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
</tr>
<tr>
<td>700</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
</tr>
<tr>
<td>700, italic</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
</tr>
<tr>
<td>800</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
</tr>
<tr>
<td>800, italic</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
<td><em>Lorem ipsum dolor fit amet</em>...</td>
</tr>
</tbody>
</table>
VARIABLE FONTS
ONE FILE

weight (wght)
width (wdth)
optical sizing (opsz)
italics (ital)
slant (slnt)

and more....
h1 {
  font-variation-settings: "wght" 850, "wdth" 100, "ital" 1;
}
@font-face {
    font-family: "My Font";
    src: local("My Font"),
        url("/fonts/my-font.woff2") format("woff-variations"),
        url("/fonts/my-font.woff2") format("woff2"),
        url("/fonts/my-font.woff") format("woff");
}
SUBSETTING
Language Support

- All Characters: 260k
- Default: 177k

Which should I choose?

- Catalan
- Italian
- Czech
- Polish
- Dutch
- Portuguese
- English
- Slovenian
- French
- Spanish
- German
- Swedish
- Hungarian
- Turkish

OpenType Features

Which should I choose?
<link href="http://fonts.googleapis.com/css?family=Open+Sans&text=Hello" rel="stylesheet">
<link href="http://fonts.googleapis.com/css?family=Open+Sans&text=Hello" rel="stylesheet">
@font-face {
  font-family: "My font";
  font-style: normal;
  font-weight: 400;
  src: <list of sources>;
}
@font-face {
    font-family: "My font";
    src: <list of sources>;
}
FORMATS

TTF
EOT
OTF
WOFF
WOFF2
FORMATS

TTF

EOT

OTF

WOFF

WOFF2
FORMATS

TTF
EOT
OTF
WOFF
WOFF2
WOFF - Web Open Font Format

Compressed TrueType/OpenType font that contains information about the font's source.

<table>
<thead>
<tr>
<th>Browser</th>
<th>IE</th>
<th>Edge</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>iOS Safari</th>
<th>Opera Mini</th>
<th>Chrome for Android</th>
<th>UC Browser for Android</th>
<th>Samsung Internet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>16</td>
<td>59</td>
<td>65</td>
<td>11</td>
<td>11.2</td>
<td>all</td>
<td>64</td>
<td>11.8</td>
<td>6.2</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>60</td>
<td>66</td>
<td>11.1</td>
<td>11.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Usage % of all users:** 95.14%
TrueType/OpenType font that provides better compression than WOFF 1.0.

<table>
<thead>
<tr>
<th></th>
<th>IE</th>
<th>Edge</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>iOS Safari</th>
<th>Opera Mini</th>
<th>Chrome for Android</th>
<th>UC Browser for Android</th>
<th>Samsung Internet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11</td>
<td>16</td>
<td>59</td>
<td>65</td>
<td>11</td>
<td>11</td>
<td>all</td>
<td>64</td>
<td>11.8</td>
<td>6.2</td>
</tr>
<tr>
<td>Notes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
local("My Font")
@font-face {
  font-family: "My Font";
  ...
  src: local("My Font"),
       url("fonts/my-font.woff2") format("woff2"),
       url("fonts/my-font.woff") format("woff");
}
Is it local? Then we don’t need to download.

```css
@font-face {
    font-family: "My Font";
    ...
    src: local("My Font"),
         url("fonts/my-font.woff2") format("woff2"),
         url("fonts/my-font.woff") format("woff");
}
```
@font-face {
  font-family: "My Font";
  ...
  src: local("My Font"),
       url("fonts/my-font.woff2") format("woff2"),
       url("fonts/my-font.woff") format("woff");
}
@font-face {
  font-family: "My Font";
  ...
  src: local("My Font"),
       url("/fonts/my-font-l.woff2") format("woff2"),
       url("/fonts/my-font-l.woff") format("woff");
  unicode-range: U+000-5FF; /* Latin glyphs */
}
@font-face {
  font-family: "My Font";
  font-style: normal;
  font-weight: 400;
  src: local("My Font"),
       url("/fonts/my-font-j.woff2") format("woff2"),
       url("/fonts/my-font-j.woff") format("woff");
  unicode-range: U+3000-9FFF, U+ff??;
  /* Japanese glyphs */
}
Combining fonts

I love the font Just Another Hand, I use it a lot in diagrams during my talks:

Here it is! Yay!

The thing is, I don’t like the positioning of the hyphen & equals glyphs...

Cache-Control: max-age=3600

They look awkwardly positioned – they sit too high.

Thankfully CSS lets you merge fonts together, so I can create a single font family that’s like Just Another Hand, except it takes the hyphen & equals glyphs from a different font, Architects

Daughter:

Cache-Control: max-age=3600
Cache-Control: max-age=3600
Cache-Control: max-age=3600
@font-face {
    font-family: 'Just Another Hand Fixed';
    font-style: normal;
    font-weight: 400;
    src: local('Just Another Hand'),
        local('JustAnotherHand-Regular'),
        url('jah.woff2') format('woff2'),
        url('jah.woff') format('woff');
}
@font-face {
  font-family: 'Just Another Hand Fixed';
  src: local('Architects Daughter'),
       local('ArchitectsDaughter'),
       url('ad.woff2') format('woff2'),
       url('ad.woff') format('woff');
  unicode-range: U+2d, U+3d;
}
<link rel="preload" href="lato.woff2" as="font" type="font/woff2" crossorigin>
```javascript
var font = new FontFace("Lato", "url(/fonts/lato.woff2)", {
  style: 'normal', unicodeRange: 'U+000-5FF', weight: '400' 
});

font.load(); // don't wait for render tree, initiate immediate fetch!

font.ready().then(function() {
  // apply the font (which may re-render text and cause a page reflow)
  // once the font has finished downloading
  document.fonts.add(font);
  document.body.style.fontFamily = "Lato, serif";

  // OR... by default content is hidden, and rendered once font is available
  var content = document.getElementById("content");
  content.style.visibility = "visible";

  // OR... apply own render strategy here...

});
```
```javascript
var font = new FontFace("Lato", "url(/fonts/lato.woff2)", {
  style: 'normal', unicodeRange: 'U+000-5FF', weight: '400'
});

font.load(); // don't wait for render tree, initiate immediate fetch!

font.ready().then(function() {
  // apply the font (which may re-render text and cause a page reflow)
  // once the font has finished downloading
  document.fonts.add(font);
  document.body.style.fontFamily = "Lato, serif";

  // OR... by default content is hidden, and rendered once font is available
  var content = document.getElementById("content");
  content.style.visibility = "visible";

  // OR... apply own render strategy here...
});
```
var font = new FontFace("Lato", "url(/fonts/lato.woff2)", {
    style: 'normal', unicodeRange: 'U+000-5FF', weight: '400'
});

font.load(); // don't wait for render tree, initiate immediate fetch!

font.ready().then(function() {
    // apply the font (which may re-render text and cause a page reflow)
    // once the font has finished downloading
    document.fonts.add(font);
    document.body.style.fontFamily = "Lato, serif";

    // OR... by default content is hidden, and rendered once font is available
    var content = document.getElementById("content");
    content.style.visibility = "visible";

    // OR... apply own render strategy here...
});
```javascript
var font = new FontFace("Lato", "url(/fonts/lato.woff2)", {
    style: 'normal',
    unicodeRange: 'U+000-5FF',
    weight: '400'
});

font.load(); // don't wait for render tree, initiate immediate fetch!

font.ready().then(function() {
    // apply the font (which may re-render text and cause a page reflow)
    // once the font has finished downloading
    document.fonts.add(font);
    document.body.style.fontFamily = "Lato, serif";

    // OR... by default content is hidden, and rendered once font is available
    var content = document.getElementById("content");
    content.style.visibility = "visible";

    // OR... apply own render strategy here...
});
```
This CSS module defines a scripting interface to font faces in CSS, allowing font faces to be easily created and loaded from script. It also provides methods to track the loading status of an individual font, or of all the fonts on an entire page.
TURN IT UP TO 11
CRITICAL PATH
& preload
& font loading API
#perfMatters
Web Performance Conference in Redwood City, CA on March 27-28, 2018

What is #perfMatters?

#perfMatters Conference is the web performance conference in Redwood City, California with talks by internationally renowned performance developers. Our focus is on front-end web performance.

Register Now  Workshop
<style>
@font-face {
    font-family: "My Font";
    ...
    src: url("data:application/x-font-woff; charset=utf-8; base64,d09GRgABA");
}
</style>
var font = new FontFace("Awesome Font", "url(/fonts/awesome.woff2)", {
    style: 'normal', unicodeRange: 'U+000-5FF', weight: '400'
});

font.load(); // don't wait for render tree, initiate immediate fetch!

font.ready().then(function() {
    // apply the font (which may re-render text and cause a page reflow) // once the font has finished downloading
document.fonts.add(font);
document.body.style.fontFamily = "Awesome Font, serif";

    // OR... by default content is hidden, and rendered once font is available
    var content = document.getElementById("content");
    content.style.visibility = "visible";

    // OR... apply own render strategy here...
});
### Before

| 591 ms | 802 ms | 1.01 s | 1.30 s | 1.80 s | 1.87 s | 2.01 s | 2.10 s | 2.17 s | 2.39 s |

### After

| 551 ms | 726 ms | 970 ms | 1.06 s | 1.56 s | 2.44 s | 2.46 s | 2.56 s | 2.64 s | 2.80 s |
MEASURING PERFORMANCE
SYNTHETIC
Active

RUM
Passive
SYNTHETIC

You control the variables

Connection, device, browser

Sterile Environment

Means more consistent results
Limited visibility

*Only monitors pages you chose*

Limited accuracy

*CDN cheats, real vs emulated, etc*
RUM

Full coverage

*All pages visited by users*

Real-world accuracy

*Real devices, real networks, etc*
RUM

No competitive reference

Noisier data

Real world means real problems
Write code → Test in the lab

Validate via RUM ← Release to users
onLoad
SpeedIndex
Visually Complete
DOMContentLoaded
DOM Ready
Start Render
Lighthouse
Requests
Weight
Critical Requests
First Paint
PageSpeed
TTFB
YSlow
Visually Complete
FOUR TYPES OF METRICS

1. QUANTITY BASED
2. RULE BASED
3. MILESTONE TIMINGS
4. SPEED INDEX
QUANTITY BASED

METRICS RELATED TO THE QUANTITY OF REQUESTS, BYTES, ETC OF A PAGE
### Performance Results (Median Run)

<table>
<thead>
<tr>
<th></th>
<th>Load Time</th>
<th>First Byte</th>
<th>Start Render</th>
<th>Speed Index</th>
<th>First Interactive (beta)</th>
<th>Document Complete</th>
<th>Fully Loaded</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Time</td>
<td></td>
</tr>
<tr>
<td>First View (Run 3)</td>
<td>8.806s</td>
<td>1.226s</td>
<td>2.796s</td>
<td>2900</td>
<td>10.386s</td>
<td>8.806s</td>
<td>19.339s</td>
</tr>
<tr>
<td>Repeat View (Run 3)</td>
<td>3.381s</td>
<td>1.187s</td>
<td>1.585s</td>
<td>1733</td>
<td>3.760s</td>
<td>3.381s</td>
<td>14.987s</td>
</tr>
</tbody>
</table>

#### Test Results

**Run 1:**

<table>
<thead>
<tr>
<th>Waterfall</th>
<th>Screen Shot</th>
<th>Video</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
QUANTITY BASED

EASY TO UNDERSTAND

CLEAR TIE TO DESIGN/DEVELOPMENT DECISIONS
More Weight Doesn't Mean More Wait

Posted by Scott on 04/17/2015

When I talk about web performance, I like to use the phrase, “weight does not need to increase wait.” To be clear, that’s not because the weight of a webpage doesn’t matter—it most definitely does—but rather because we can often (usually, even) deliver a usable representation of a web page’s content very quickly, even if that page is quite large and asset-heavy as a whole. At the root of this distinction is a performance metric that the web community has only recently begun to discuss and prioritize, known as perceived performance.

Formerly, much of the focus in web performance was concerned with optimizing assets like images and fonts, which does make for a shorter overall page load time. But today there are techniques we can use in addition to file optimization that have an arguably larger impact on how soon our users can see and use the...
QUANTITY BASED

EASY TO UNDERSTAND

ALMOST NO

CLEAR TIE TO DESIGN/ CONNECTION TO

DEVELOPMENT USER EXPERIENCE

DECISIONS
Quantity Based
RULE BASED

METRICS BASED ON ADHERENCE TO A SET OF RULES OR GUIDELINES
PageSpeed Insights

http://technical.highedweb.org/

Should Fix:
Eliminate render-blocking JavaScript and CSS in above-the-fold content

Consider Fixing:
Prioritize visible content
Optimize images
Reduce server response time
Minify CSS
PageSpeed Insights

http://technical.highedweb.org/

66 / 100 Speed

Should Fix:
Eliminate render-blocking JavaScript and CSS in above-the-fold content
  ▶ Show how to fix

Consider Fixing:
Priority visible content
  ▶ Show how to fix

Optimize images
  ▶ Show how to fix

Reduce server response time
  ▶ Show how to fix

Minify CSS
  ▶ Show how to fix

5 Passed Rules
  ▶ Show details
Grade F on Make fewer HTTP requests

This page has 12 external Javascript scripts. Try combining them into one. This page has 6 external stylesheets. Try combining them into one.

Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

Read More
RULE BASED

GREAT CHECKLIST OF BASIC OPTIMIZATIONS

EASY TO UNDERSTAND
RULE BASED

GREAT CHECKLIST OF BASIC OPTIMIZATIONS  ALMOST NO CONNECTION TO USER EXPERIENCE

EASY TO UNDERSTAND
WEIGHT:  +44%
ONLOAD:  +46%
REQUESTS:  +21%
SPEED INDEX:  +34%
MILESTONE TIMINGS

TIMING OF A VERY SPECIFIC MILESTONE IN THE COURSE OF LOADING A PAGE
TIME TO FIRST BYTE

WHEN DOES THE BROWSER RECEIVE THE FIRST BYTE OF THE FIRST RESPONSE
### Performance Results (Median Run)

<table>
<thead>
<tr>
<th></th>
<th>Load Time (s)</th>
<th>First Byte (ms)</th>
<th>Start Render (ms)</th>
<th>Speed Index</th>
<th>First Interactive (beta)</th>
<th>Document Complete</th>
<th>Fully Loaded</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>First View (Run 3)</td>
<td>8.806</td>
<td>1.228</td>
<td>2.796</td>
<td>2900</td>
<td>10.386</td>
<td>3.606</td>
<td>19.339</td>
</tr>
<tr>
<td>Repeat View (Run 3)</td>
<td>3.381</td>
<td>1.187</td>
<td>1.585</td>
<td>1733</td>
<td>3.760</td>
<td>3.381</td>
<td>14.987</td>
</tr>
</tbody>
</table>

- **Cost**: $5555-

### Test Results

**Run 1:**

- **Waterfall**
- **Screen Shot**
- **Video**
START RENDER

WHEN DOES STUFF START SHOWING UP
<table>
<thead>
<tr>
<th>Step 1</th>
<th>dns</th>
<th>connect</th>
<th>ssl</th>
<th>html</th>
<th>js</th>
<th>css</th>
<th>image</th>
<th>flash</th>
<th>font</th>
<th>video</th>
<th>other</th>
<th>JS Execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td></td>
<td></td>
<td></td>
<td><a href="http://www.amazon.com">www.amazon.com</a> /</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
<td></td>
<td>images-na.ssl----</td>
<td>411</td>
<td>dmP0D7L.css</td>
<td>433</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td></td>
<td></td>
<td></td>
<td>images-na.ssl----</td>
<td>72+uC9L.css</td>
<td>909</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td></td>
<td></td>
<td></td>
<td>images-na.ssl----</td>
<td>21SX%2BxN0sHl.css</td>
<td>450</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td></td>
<td></td>
<td></td>
<td>images-na.ssl----</td>
<td>jbs0jo-l.css</td>
<td>484</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td></td>
<td></td>
<td></td>
<td>images-na.ssl----</td>
<td>CB487546343.png</td>
<td>246</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td></td>
<td></td>
<td></td>
<td>m.media-amazon..2d280cie_Y2.png</td>
<td>632</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td></td>
<td></td>
<td></td>
<td>fls.amazon..pti%30mobile:1000</td>
<td>610</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td></td>
<td></td>
<td></td>
<td>images-na.ssl----</td>
<td>V497204129.js</td>
<td>614</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td></td>
<td></td>
<td></td>
<td>images-na.ssl----</td>
<td>12YXwM7uJL.js</td>
<td>565</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td></td>
<td></td>
<td></td>
<td>images-na.ssl----</td>
<td>5issqyvb26l.js</td>
<td>593</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td></td>
<td></td>
<td></td>
<td>images-na.ssl----</td>
<td>18Ww-h28-L.js</td>
<td>576</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13.</td>
<td></td>
<td></td>
<td></td>
<td>images-na.ssl----</td>
<td>a72lenyyl.js</td>
<td>607</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.</td>
<td></td>
<td></td>
<td></td>
<td>images-na.ssl----</td>
<td>2hyeuTd9l.js</td>
<td>590</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.</td>
<td></td>
<td></td>
<td></td>
<td>images-na.ssl----</td>
<td>110Xqg0681l.js</td>
<td>601</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16.</td>
<td></td>
<td></td>
<td></td>
<td>m.media-amazon..0b7efa_Y2_woff2</td>
<td>1181</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17.</td>
<td></td>
<td></td>
<td></td>
<td>images-na.ssl----</td>
<td>V497204134.html</td>
<td>233</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19.</td>
<td></td>
<td></td>
<td></td>
<td>images-na.ssl----</td>
<td>CB4972s1099.jpg</td>
<td>292</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20.</td>
<td></td>
<td></td>
<td></td>
<td>images-na.ssl----</td>
<td>om - ads.TTH.js</td>
<td>182</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21.</td>
<td></td>
<td></td>
<td></td>
<td>images-na.ssl----</td>
<td>CB497294260.jpg</td>
<td>739</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22.</td>
<td></td>
<td></td>
<td></td>
<td><a href="http://www.amazon.com">www.amazon.com</a> - 0-3180543-1857141</td>
<td>580</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.</td>
<td></td>
<td></td>
<td></td>
<td><a href="http://www.amazon.com">www.amazon.com</a> - display.html</td>
<td>1034</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24.</td>
<td></td>
<td></td>
<td></td>
<td>m.media-amazon..7e699c73_Y2.png</td>
<td>285</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25.</td>
<td></td>
<td></td>
<td></td>
<td>images-na.ssl----</td>
<td>CB50111281.jpg</td>
<td>457</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26.</td>
<td></td>
<td></td>
<td></td>
<td>aax-us-east.am..B2x22n22x22true77D</td>
<td>608</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27.</td>
<td></td>
<td></td>
<td></td>
<td><a href="http://www.amazon.com">www.amazon.com</a> - 0-3180543-1857141</td>
<td>972</td>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DOCUMENT COMPLETE

WHEN DOCUMENT FIRES ONLOAD EVENT
FULLY LOADED
NOTHING MORE TO SEE HERE
let perfData = window.performance.timing;

let renderTime = perfData.domComplete - perfData.domLoading;
window.performance.getEntriesByType('resource');
MILESTONE TIMINGS

EASY TO TRACK

EASY TO COMMUNICATE

USER TIMING ALLOWS FOR VERY SPECIFIC METRICS
MILESTONE TIMINGS

EASY TO TRACK

LIMITED VIEW OF THE
USER EXPERIENCE

EASY TO COMMUNICATE

USER TIMING ALLOWS FOR
VERY SPECIFIC METRICS
SPEED INDEX

A REPRESENTATION OF THE PERCEIVED LOAD OF A PAGE, FROM START TO FINISH
## Web Page Performance Test for https://www.amazon.com/

### Summary
- **Tester:** Nexus5_2-192.168.0.172
- **Test runs:** 3

### Performance Results (Median Run)

<table>
<thead>
<tr>
<th></th>
<th>Load Time</th>
<th>First Byte</th>
<th>Start Render</th>
<th>Speed Index</th>
<th>First Interactive (beta)</th>
<th>Document Complete</th>
<th>Fully Loaded</th>
</tr>
</thead>
<tbody>
<tr>
<td>First View (Run 3)</td>
<td>8.806s</td>
<td>1.226s</td>
<td>2.796s</td>
<td>2900</td>
<td>10.586s</td>
<td>8.806s 43</td>
<td>19.339s 134</td>
</tr>
<tr>
<td>Repeat View (Run 3)</td>
<td>3.381s</td>
<td>1.187s</td>
<td>1.585s</td>
<td>1733</td>
<td>3.760s</td>
<td>3.381s 8</td>
<td>14.987s 46</td>
</tr>
</tbody>
</table>

### Test Results

**Run 1:**

- **Waterfall**
- **Screen Shot**
- **Video**
SPEED INDEX

VERY CLOSELY TIED TO USER EXPERIENCE
SPEED INDEX

Very closely tied to user experience

difficult to communicate

limited tracking
Quantity Based
Rule Based
Milestone Timings
Speed Index
NO ONE METRIC
to rule them all
...and in the darkness bind them
IS IT happening?
performance.getEntriesByType('paint');

/*
[
  {name: "first-paint", startTime: 378.79},
  {name: "first-contentful-paint",
   startTime: 807.205}
]
*/
const observer = new PerformanceObserver((list) => {
    for (const entry of list.getEntries()) {
        console.log(entry.entryType);
        console.log(entry.startTime);
        console.log(entry.duration);
    }
});

// Start observing paint
observer.observe({entryTypes: ['paint']});
IS IT useful?
Introducing Last Painted Hero

WEDNESDAY 4TH OF APRIL 2018

We’re excited to announce that we’ve launched Last Painted Hero as an official metric. Last Painted Hero is a synthetic metric that shows you when the last piece of critical content is painted. Keep reading to learn how Last Painted Hero works, why (and how) we created it, and how it can help you understand how your users perceive the speed of your pages.

The case for smarter heuristics

When choosing the right performance metric, my soapbox for the last few
<table>
<thead>
<tr>
<th>Service</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Painted Hero (Syn), Airbnb</td>
<td>18.78s</td>
</tr>
<tr>
<td>Last Painted Hero (Syn), ESPN</td>
<td>9.6s</td>
</tr>
<tr>
<td>Last Painted Hero (Syn), Youtube</td>
<td>5.9s</td>
</tr>
<tr>
<td>Last Painted Hero (Syn), Zillow</td>
<td>7.38s</td>
</tr>
</tbody>
</table>
Accoutrements Yodelling Pickle

635 customer reviews | 42 answered questions

Amazon’s Choice for “yodelling pickle”

List Price: $14.99
Price: $11.03 prime

FREE Shipping on orders over $25—or get FREE Two-Day Shipping with Amazon Prime

You Save: $3.96 (26%)

In Stock.

Want it Tuesday, April 24? Order within 7 hrs 49 mins and choose Two-Day Shipping at checkout. Details

Ships from and sold by Amazon.com.
- Hours of mindless entertainment
- Batteries included
- Great gift for the person who has everything except a yodelling pickle

New (22) from $10.40 & FREE shipping.
performance.mark();

performance.measure();
<link rel="stylesheet" href="/sheet1.css"/>
<link rel="stylesheet" href="/sheet4.css"/>
<script>
performance.mark("stylesheets done blocking");
</script>
http://localhost:3000?react_perf
IS IT usable?
TIME TO INTERACTIVE

Layout has stabilized, key webfonts are visible, and the main thread is available enough to handle user input.
import ttiPolyfill from './path/to/tti-polyfill.js';

ttiPolyfill.getFirstConsistentlyInteractive().then((tti) => {
  // process
});
IS IT delightful?
IS IT delightful?
const observer = new PerformanceObserver((list) => {
  for (const entry of list.getEntries()) {
    ga('send', 'event', {
      eventCategory: 'Performance Metrics',
      eventAction: 'longtask',
      eventValue: Math.round(entry.startTime + entry.duration),
      eventLabel: JSON.stringify(entry.attribution),
    });
  }
});

observer.observe({entryTypes: ['longtask']});
```javascript
const subscribeBtn = document.querySelector('#subscribe');

subscribeBtn.addEventListener('click', (event) => {
  // Event listener logic goes here...

  const lag = performance.now() - event.timeStamp;
  if (lag > 100) {
    //send to tracking
  }
});
```
THANK YOU

Tim Kadlec | @tkadlec