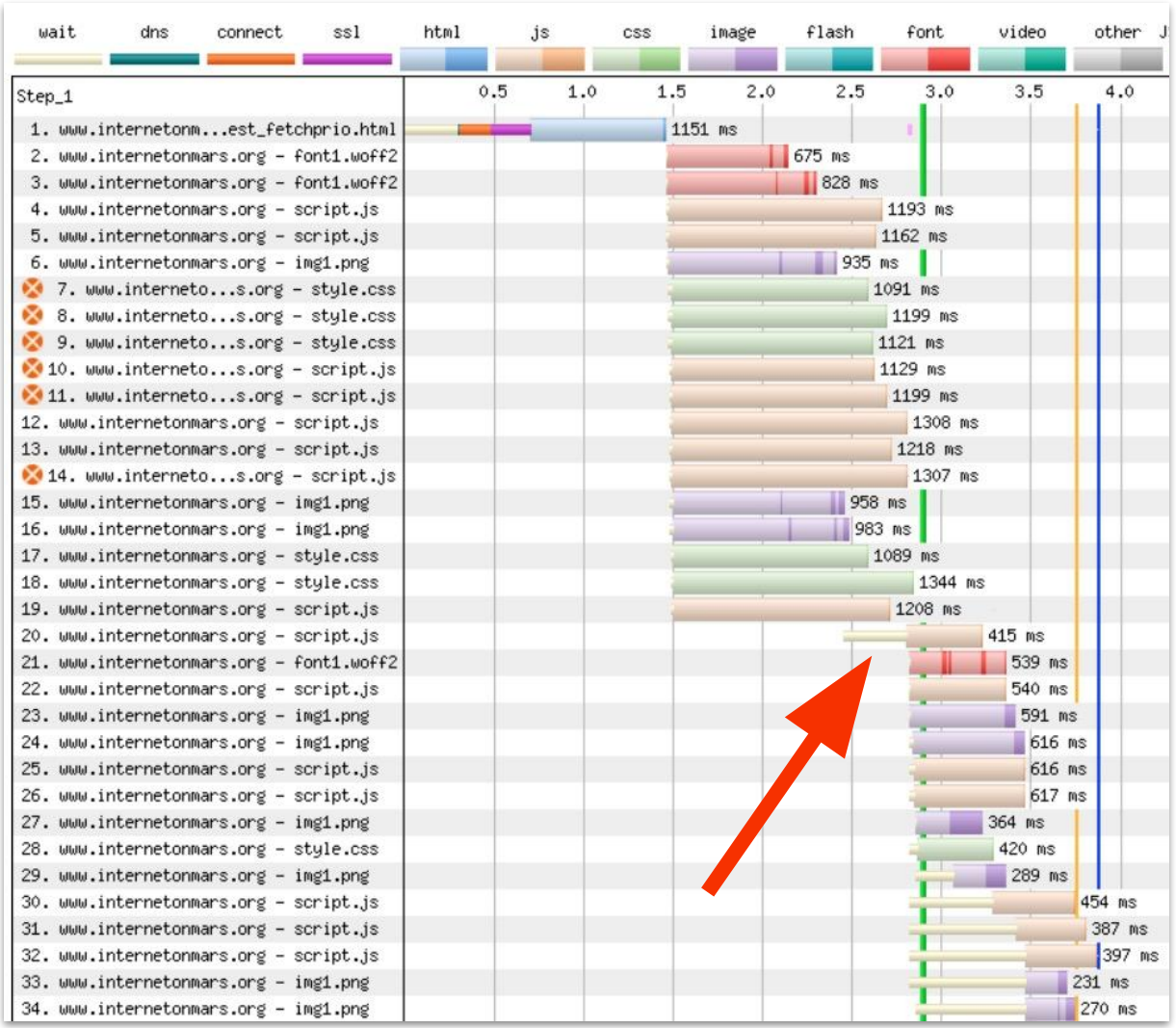
“Tight Mode”: How Browsers REALLY Load Web Pages

Robin Marx
@programmingart


HAIR TRANSPLANT
LOCALLY SOURCE FROM BROWSER DEVS
CLOWNS

Two-step waterfall

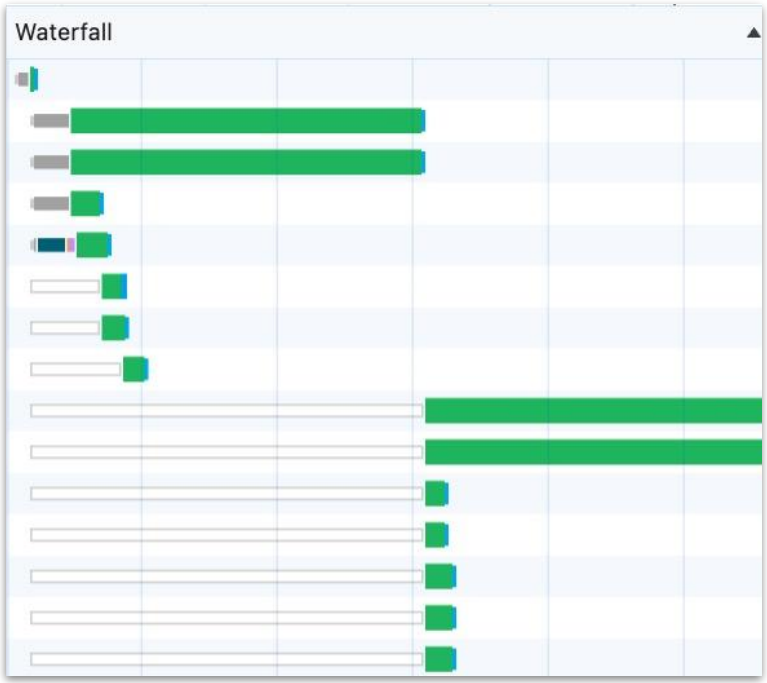


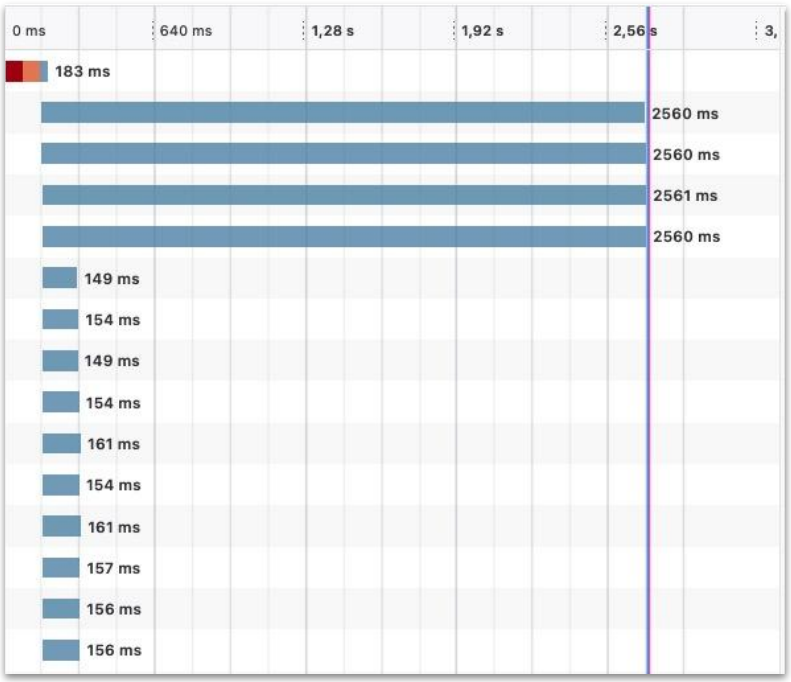
Name	Protocol	Type	Size	Time	Prio…	Waterfall
test_fetchprio.html	h3	docu…	6.8 kB	41 ms	Hig…	
font1.woff2?preload	h3	font	29.4 kB	143 ms	High	
font1.woff2?preload-prio-high	h3	font	29.3 kB	160 ms	High	
script.js?preload	h3	script	133 B	160 ms	High	
script.js?preload-prio-high	h3	script	58 B	160 ms	High	
img1.png?preload-prio-high	h3	png	42.9 kB	191 ms	High	
style.css?head	h3	styles…	187 B	191 ms	Hig…	
style.css?head-prio-high	h3	styles…	117 B	191 ms	Hig…	
style.css?head-prio-low	h3	styles…	117 B	196 ms	High	
script.js?head	h3	script	58 B	196 ms	High	
script.js?head-prio-high	h3	script	58 B	196 ms	High	
script.js?head-async-prio-high	h3	script	58 B	196 ms	High	
script.js?head-defer-prio-high	h3	script	58 B	196 ms	High	
script.js?head-prio-low	h3	script	58 B	196 ms	High	
img1.png?visible-eager	h3	png	42.8 kB	230 ms	Me…	
img1.png?visible-eager-prio-…	h3	png	42.8 kB	293 ms	High	
style.css?bottom	h3	styles…	117 B	293 ms	Me…	
style.css?bottom-prio-high	h3	styles…	117 B	293 ms	High	
script.js?bottom-prio-high	h3	script	58 B	293 ms	High	
script.js?bottom	h3	script	58 B	86 ms	Me…	
font1.woff2?preload-prio-low	h3	font	29.3 kB	109 ms	Low	
script.js?preload-prio-low	h3	script	58 B	109 ms	Low	
img1.png?preload	h3	png	42.8 kB	146 ms	Low	
img1.png?preload-prio-low	h3	png	42.8 kB	182 ms	Low	
script.js?head-async	h3	script	58 B	183 ms	Low	
script.js?head-defer	h3	script	58 B	183 ms	Low	
script.js?head-async-prio-low	h3	script	58 B	183 ms	Low	
script.js?head-defer-prio-low	h3	script	58 B	183 ms	Low	
img1.png?visible-eager-prio-…	h3	png	42.8 kB	221 ms	Low	
style.css?bottom-prio-low	h3	styles…	117 B	222 ms	Low	
script.js?bottom-prio-low	h3	script	58 B	222 ms	Low	
qlog-processor.js	h3	script	6.0 kB	232 ms	Low	
img1.png?visible-lazy	h3	png	42.8 kB	155 ms	High	

Akamai

Exact same HTML,

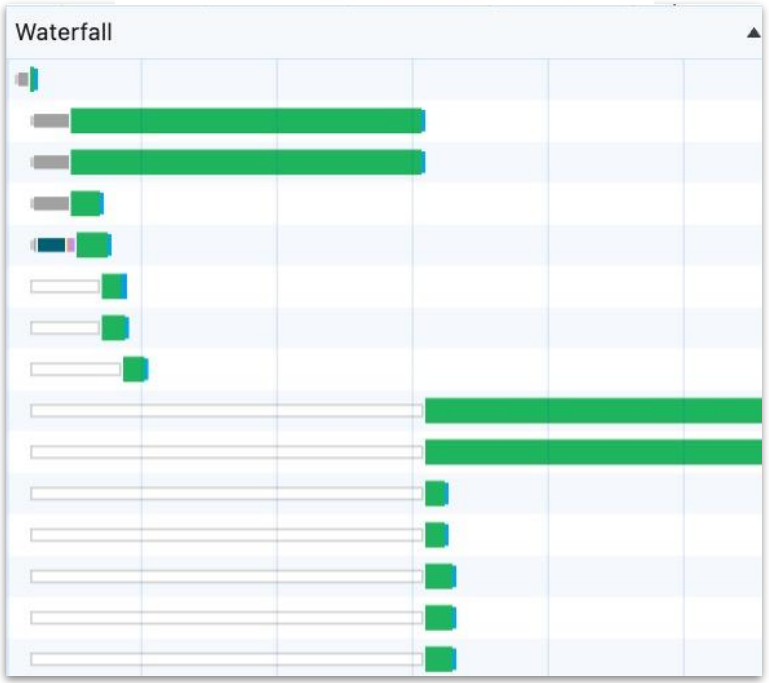
radically different behaviour

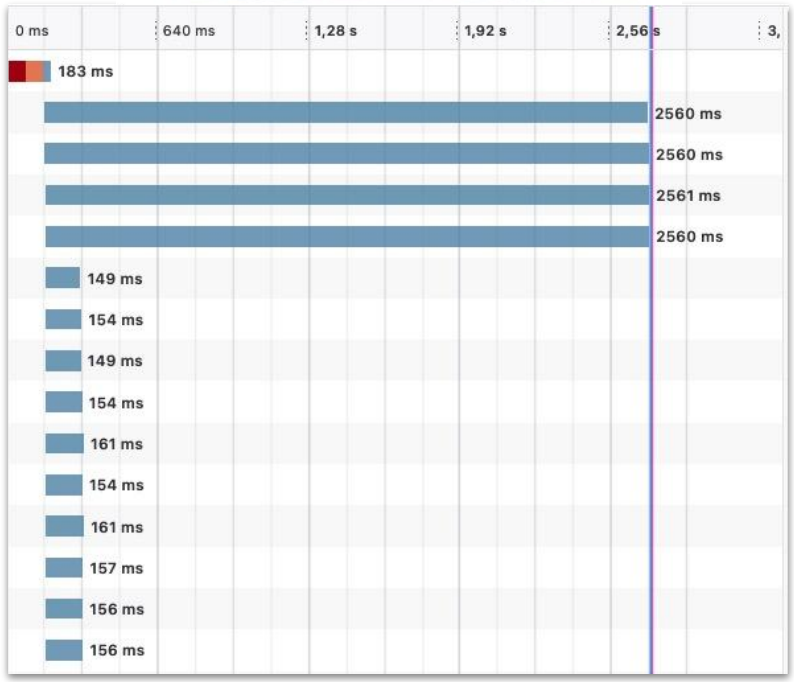




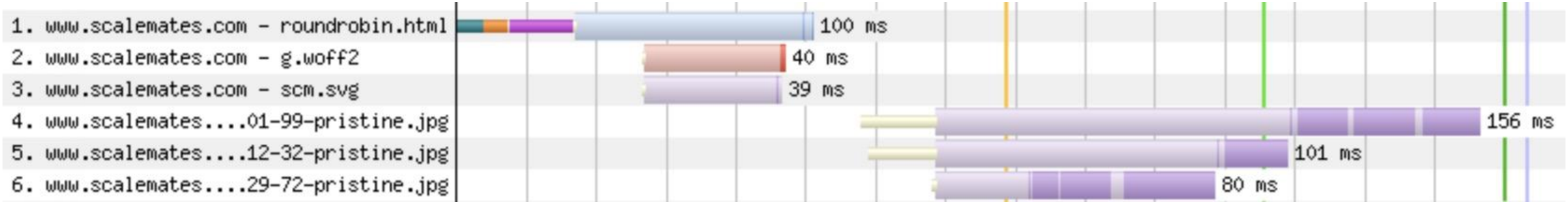
Exact same HTML,

radically different behaviour



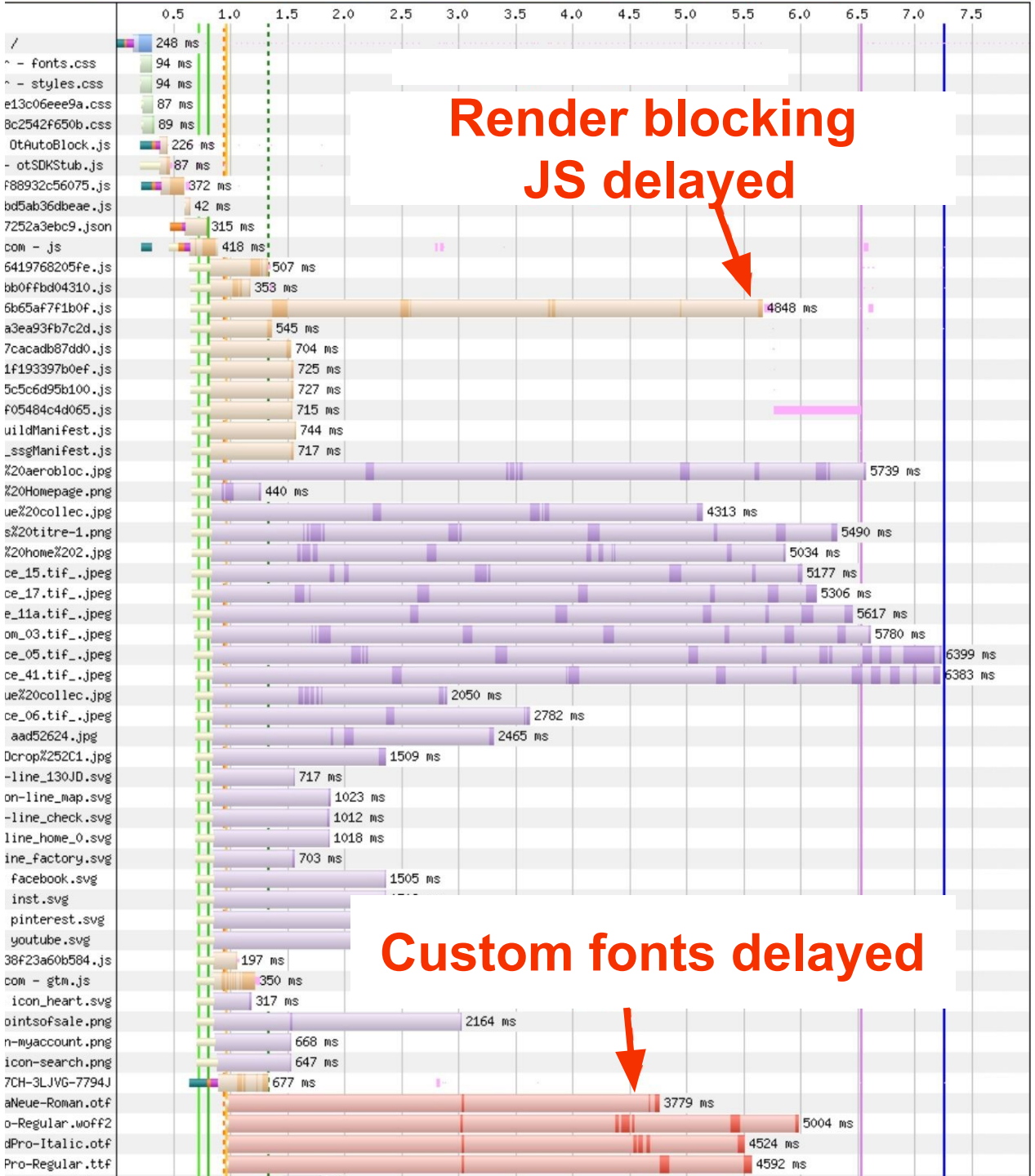


It’s all the servers’ fault!



response order =
INVERTED request order

Akamai


0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
/ 248 ms
- fonts.css 94 ms
- styles.css 94 ms
e13c06eee9a.css 87 ms
8c2542f650b.css 89 ms
OtAutoBlock.js 226 ms
- otSDKStub.js 87 ms
f88932c56075.js 372 ms
bd5ab36dbeae.js 42 ms
7252a3ebc9.json 315 ms
com - js 418 ms
6419768205fe.js 507 ms
bb0ffbd04310.js 353 ms
6b65af7f1b0f.js 4848 ms
a3ea93fb7c2d.js 545 ms
7cacadb87dd0.js 704 ms
1f193397b0ef.js 725 ms
5c5c6d95b100.js 727 ms
f05484c4d065.js 715 ms
uildManifest.js 744 ms
_ssgManifest.js 717 ms
%20aerobloc.jpg 5739 ms
%20Homepage.png 440 ms
ue%20collec.jpg 4313 ms
s%20titre-1.png 5490 ms
%20home%202.jpg 5034 ms
ce_15.tif_.jpeg 5177 ms
ce_17.tif_.jpeg 5306 ms
e_11a.tif_.jpeg 5617 ms
om_03.tif_.jpeg 5780 ms
ce_05.tif_.jpeg 6399 ms
ce_41.tif_.jpeg 6383 ms
ue%20collec.jpg 2050 ms
ce_06.tif_.jpeg 2782 ms
aad52624.jpg 2465 ms
Dcrop%252C1.jpg 1509 ms
-line_130JD.svg 717 ms
on-line_map.svg 1023 ms
-line_check.svg 1012 ms
line_home_0.svg 1018 ms
ine_factory.svg 703 ms
facebook.svg 1505 ms
inst.svg
pinterest.svg
youtube.svg
38f23a60b584.js 197 ms
com - gtm.js 350 ms
icon_heart.svg 317 ms
ointsofsale.png 2164 ms
n-myaccount.png 668 ms
icon-search.png 647 ms
7CH-3LJVG-7794J 677 ms
aNeue-Roman.otf 3779 ms
o-Regular.woff2 5004 ms
dPro-Italic.otf 4524 ms
Pro-Regular.ttf 4592 ms
Render blocking
JS delayed
Custom fonts delayed

PRIORITY:

```
1   <head>
2       <link rel=preload href=font1.woff2>        MEDIUM
3       <link rel=preload href=font2.woff2>        MEDIUM
4       <link rel=preload href=lcp.png>            MEDIUM
5
6       <link rel=stylesheet href=style1.css />    HIGHEST
7       <link rel=stylesheet href=style2.css />    HIGHEST
8       <link rel=stylesheet href=style3.css />    HIGHEST
9
10      <script src=script1.js defer></script>     LOW
11      <script src=script2.js defer></script>     LOW
12      <script src=script3.js defer></script>     LOW
13      <script src=script4.js defer></script>     LOW
14
15      <script src=script_critical.js></script>   HIGH
16  </head>
```

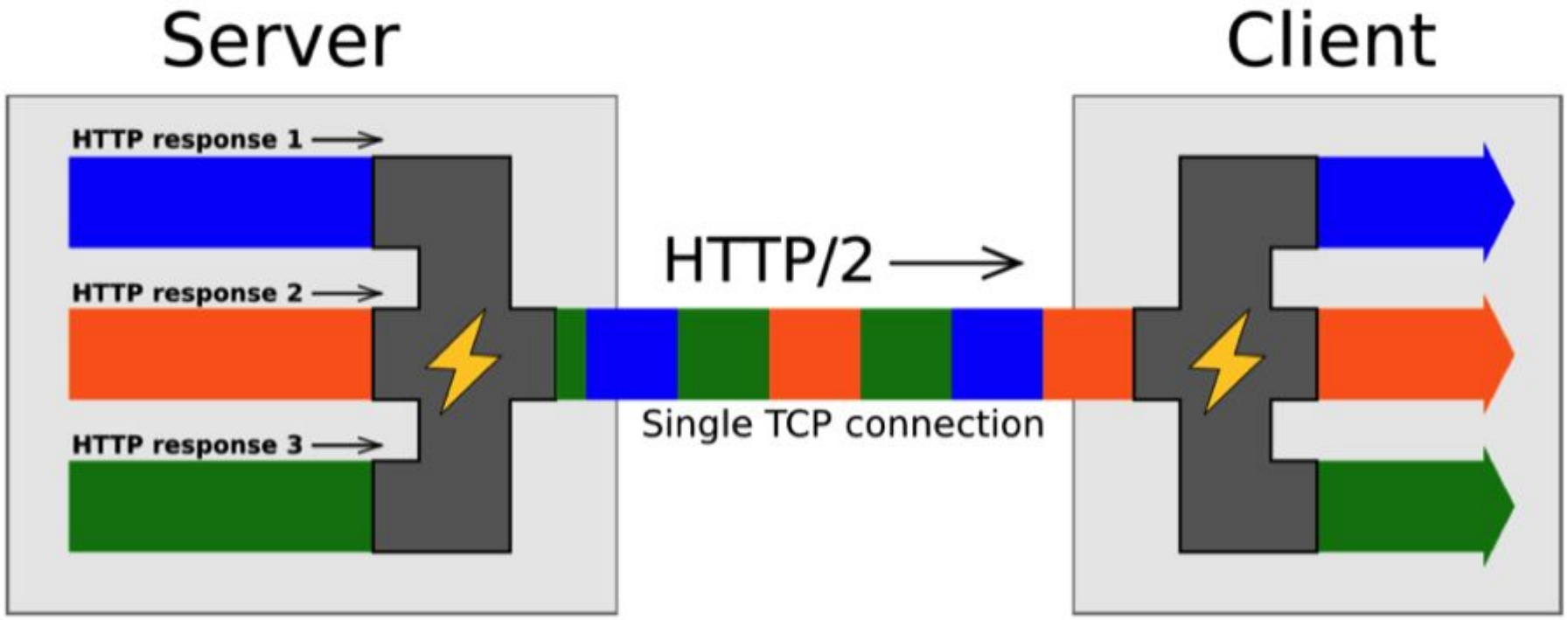
Akamai

PRIORITY:

```
1   <head>
2       <link rel=preload href=font1.woff2>          MEDIUM
3       <lin                                         MEDIUM
4       <lin                                         UM
5
6       <lin                                         EST
7       <lin                                         EST
8       <lin                                         EST
9
10      <scr
11      <scr
12      <scr
13      <scr
14
15      <script src=script_critical.js></script>     HIGH
16  </head>
```

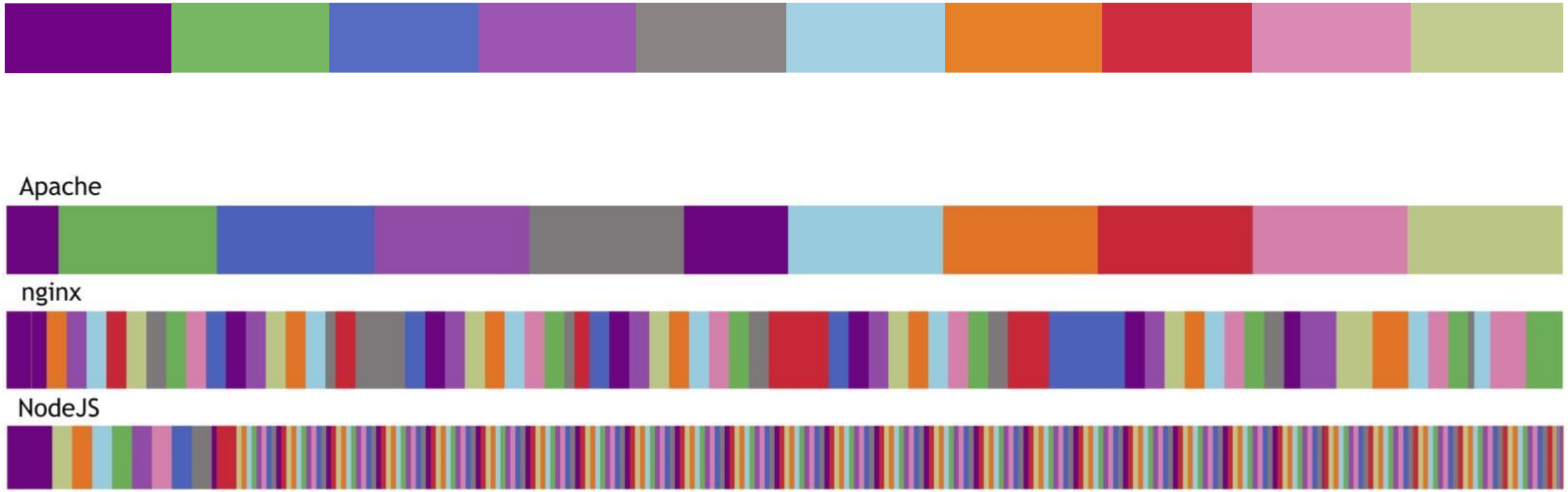
Akamai

HTTP/2 and /3 Multiplexing



(HTTP/2) Servers often don’t listen to browsers…

Browser instructions:



https://github.com/andydavies/http2-prioritization-issues
https://www.researchgate.net/publication/347519865_Debugging_Modern_Web_Protocols

Akamai

(HTTP/2) Servers often don't listen to browsers…

Browser instructions:

Apache

nginx

NodeJS

Terrible for Web performance

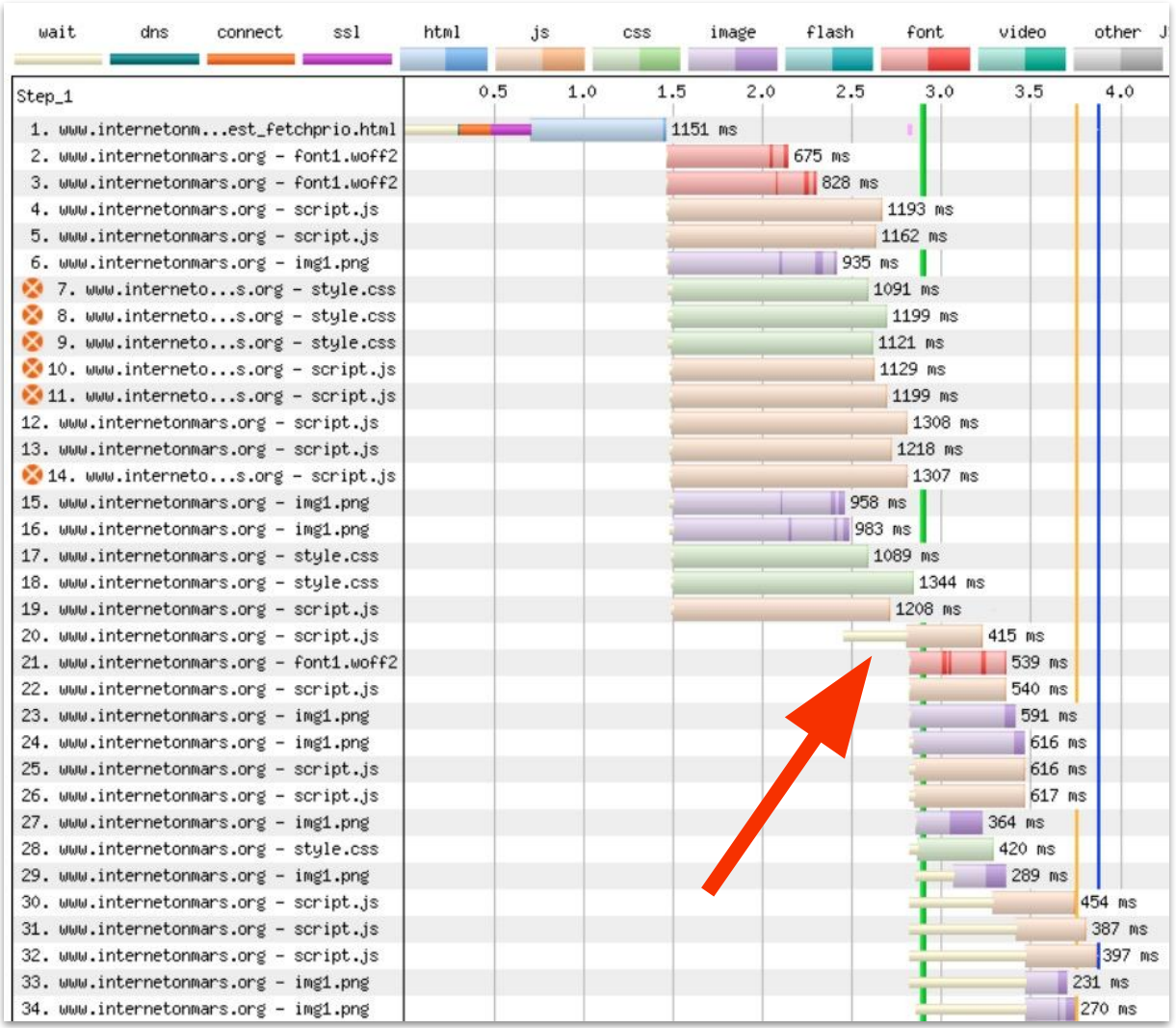
https://github.com/andydavies/http2-prioritization-issues
https://www.researchgate.net/publication/347519865_Debugging_Modern_Web_Protocols

Akamai

Only 2 of these companies do it (100%) correctly…

https://jherbots.info/public_media/research/anrw2024_h3-eps-in-the-wild_authorversion.pdf

Two-step waterfall

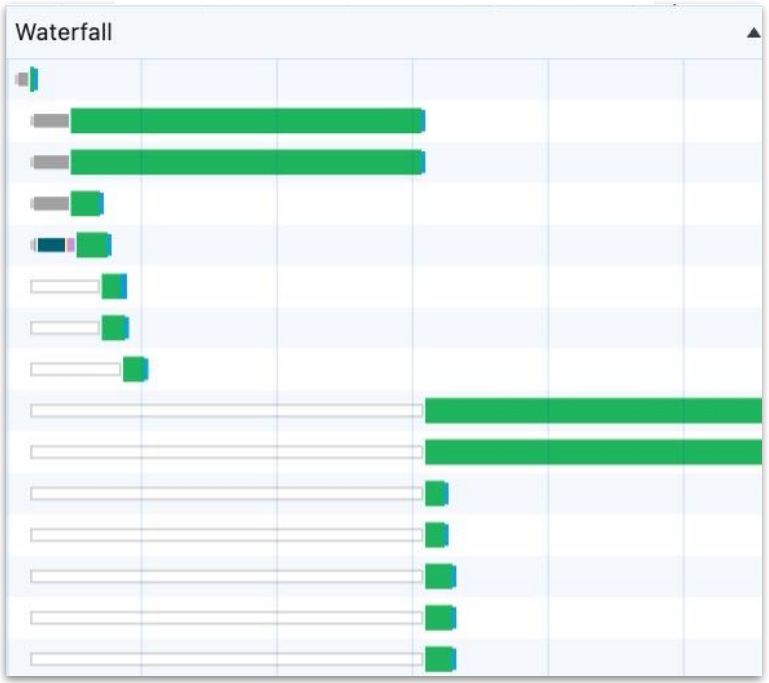


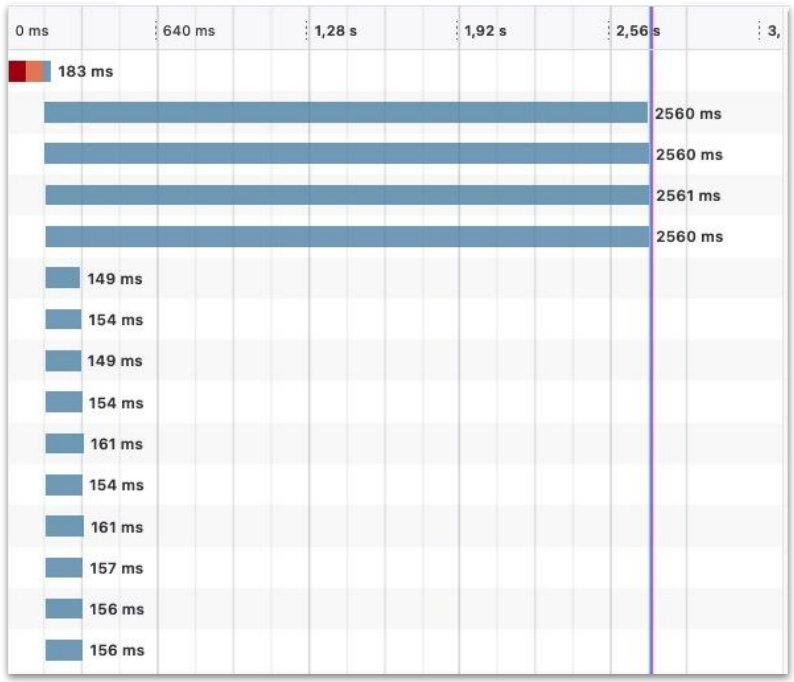
Name	Protocol	Type	Size	Time	Prio…	Waterfall
test_fetchprio.html	h3	docu…	6.8 kB	41 ms	Hig…	
font1.woff2?preload	h3	font	29.4 kB	143 ms	High	
font1.woff2?preload-prio-high	h3	font	29.3 kB	160 ms	High	
script.js?preload	h3	script	133 B	160 ms	High	
script.js?preload-prio-high	h3	script	58 B	160 ms	High	
img1.png?preload-prio-high	h3	png	42.9 kB	191 ms	High	
style.css?head	h3	styles…	187 B	191 ms	Hig…	
style.css?head-prio-high	h3	styles…	117 B	191 ms	Hig…	
style.css?head-prio-low	h3	styles…	117 B	196 ms	High	
script.js?head	h3	script	58 B	196 ms	High	
script.js?head-prio-high	h3	script	58 B	196 ms	High	
script.js?head-async-prio-high	h3	script	58 B	196 ms	High	
script.js?head-defer-prio-high	h3	script	58 B	196 ms	High	
script.js?head-prio-low	h3	script	58 B	196 ms	High	
img1.png?visible-eager	h3	png	42.8 kB	230 ms	Me…	
img1.png?visible-eager-prio-…	h3	png	42.8 kB	293 ms	High	
style.css?bottom	h3	styles…	117 B	293 ms	Me…	
style.css?bottom-prio-high	h3	styles…	117 B	293 ms	High	
script.js?bottom-prio-high	h3	script	58 B	293 ms	High	
script.js?bottom	h3	script	58 B	86 ms	Me…	
font1.woff2?preload-prio-low	h3	font	29.3 kB	109 ms	Low	
script.js?preload-prio-low	h3	script	58 B	109 ms	Low	
img1.png?preload	h3	png	42.8 kB	146 ms	Low	
img1.png?preload-prio-low	h3	png	42.8 kB	182 ms	Low	
script.js?head-async	h3	script	58 B	183 ms	Low	
script.js?head-defer	h3	script	58 B	183 ms	Low	
script.js?head-async-prio-low	h3	script	58 B	183 ms	Low	
script.js?head-defer-prio-low	h3	script	58 B	183 ms	Low	
img1.png?visible-eager-prio-…	h3	png	42.8 kB	221 ms	Low	
style.css?bottom-prio-low	h3	styles…	117 B	222 ms	Low	
script.js?bottom-prio-low	h3	script	58 B	222 ms	Low	
qlog-processor.js	h3	script	6.0 kB	232 ms	Low	
img1.png?visible-lazy	h3	png	42.8 kB	155 ms	High	

Akamai

Exact same HTML,

radically different behaviour





Resource Fetch Prioritization and Scheduling in Chrome

Author: Patrick Meenan

August 5, 2015 (Updated June 27, 2022)

Current State

As of April 2022, the table below represents how all resources in Chrome are handled:

	Load in "tight mode"		Conditionally load in "tight mode"		
Blink Priority	**VeryHigh**	**High**	**Medium**	**Low**	**VeryLow**
DevTools Priority	**Highest**	**High**	**Medium**	**Low**	**Lowest**
Main Resource	◉				
CSS*** (early**)	⬆◉	⬇			
CSS*** (late**)		⬆	◉	⬇	
Script (early** or not from preload scanner)		⬆◉		⬇	
Script (late**)		⬆	◉	⬇	
Script (async/defer)		⬆		◉⬇	

https://web.dev/articles/fetch-priority
https://imkev.dev/fetchpriority-opportunity
https://firefox-source-docs.mozilla.org/networking/http/prioritization.html
https://docs.google.com/document/d/1bCDuq9H1ih9iNjgzyAL0gpwNFiEP4TZS-YLRp_RuMlc


stablediffusionweb.com

Name	Protocol	Type	Size	Time	Prio…	Waterfall
test_fetchprio.html	h3	docu…	6.8 kB	41 ms	Hig…	
font1.woff2?preload	h3	font	29.4 kB	143 ms	High	
font1.woff2?preload-prio-high	h3	font	29.3 kB	160 ms	High	
script.js?preload	h3	script	133 B	160 ms	High	
script.js?preload-prio-high	h3	script	58 B	160 ms	High	
img1.png?preload-prio-high	h3	png	42.9 kB	191 ms	High	
style.css?head	h3	styles…	187 B	191 ms	Hig…	
style.css?head-prio-high	h3	styles…	117 B	191 ms	Hig…	
style.css?head-prio-low	h3	styles…	117 B	196 ms	High	
script.js?head	h3	script	58 B	196 ms	High	
script.js?head-prio-high	h3	script	58 B	196 ms	High	
script.js?head-async-prio-high	h3	script	58 B	196 ms	High	
script.js?head-defer-prio-high	h3	script	58 B	196 ms	High	
script.js?head-prio-low	h3	script	58 B	196 ms	High	
img1.png?visible-eager	h3	png	42.8 kB	230 ms	Me…	
img1.png?visible-eager-prio-…	h3	png	42.8 kB	293 ms	High	
style.css?bottom	h3	styles…	117 B	293 ms	Me…	
style.css?bottom-prio-high	h3	styles…	117 B	293 ms	High	
script.js?bottom-prio-high	h3	script	58 B	293 ms	High	
script.js?bottom	h3	script	58 B	86 ms	Me…	
font1.woff2?preload-prio-low	h3	font	29.3 kB	109 ms	Low	
script.js?preload-prio-low	h3	script	58 B	109 ms	Low	
img1.png?preload	h3	png	42.8 kB	146 ms	Low	
img1.png?preload-prio-low	h3	png	42.8 kB	182 ms	Low	
script.js?head-async	h3	script	58 B	183 ms	Low	
script.js?head-defer	h3	script	58 B	183 ms	Low	

Tight Mode

Chrome loads resources in 2 phases. “Tight mode” is the initial phase and constraints loading lower-priority resources until the body is attached to the document (essentially, after all blocking scripts in the head have been executed). In tight mode, low priority resources are only loaded if there are less than 2 in-flight requests at the time that they are discovered.

https://docs.google.com/document/d/1bCDuq9H1ih9iNjgzyAL0gpwNFiEP4TZS-YLRp_RuMlc

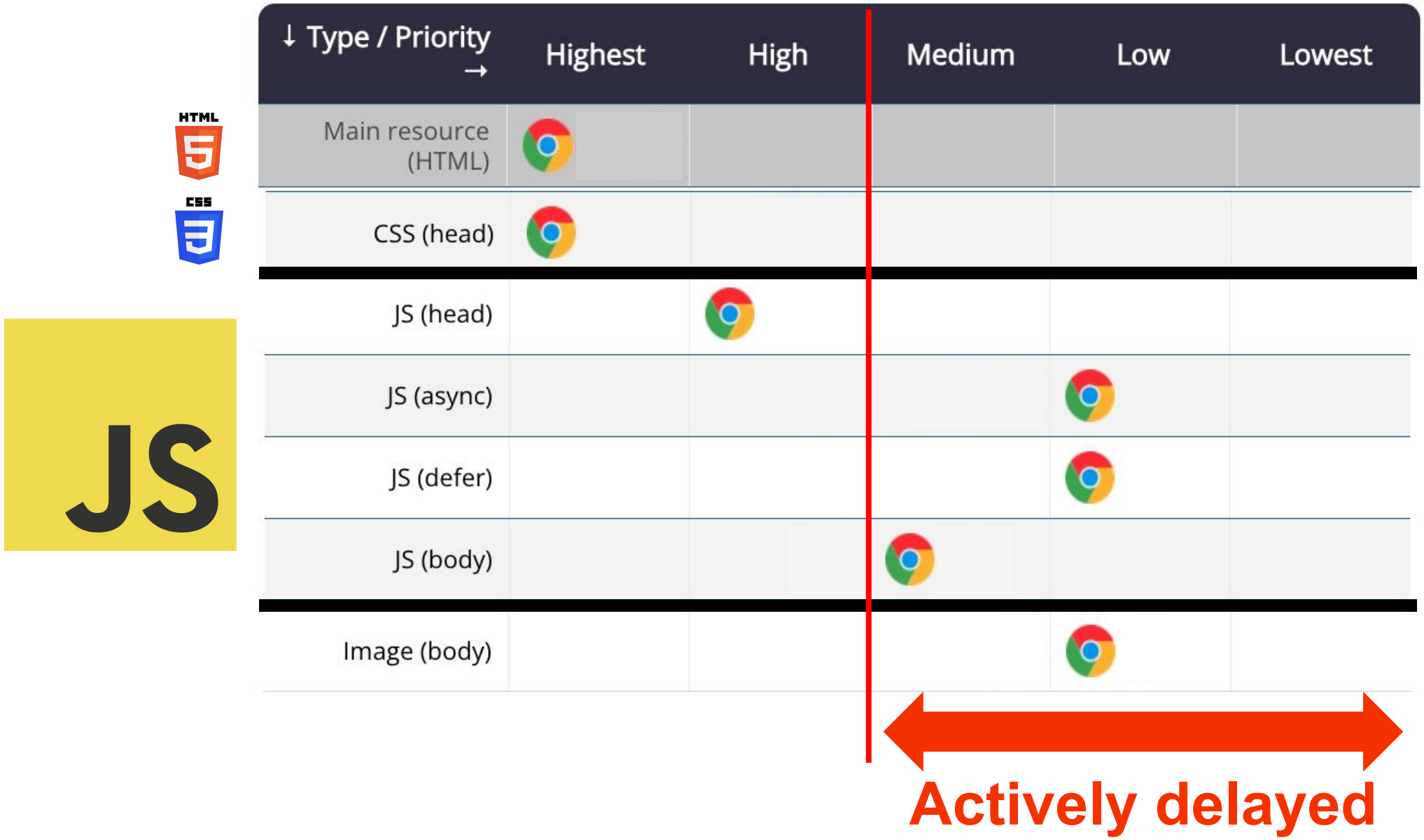
Akamai

Priority: *where* stuff is in HTML and *how* it's loaded

↓ Type / Priority →	Highest	High	Medium	Low	Lowest
Main resource (HTML)	●				
CSS (head)	●				
JS (head)		●			
JS (async)				●	
JS (defer)				●	
JS (body)			●		
Image (body)				●	

Akamai

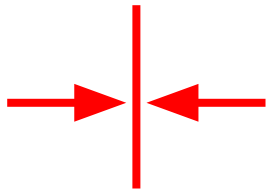
“Lower-priority”: medium + low + lowest



Name	Type	Time	Priority	Waterfall
tightmode_images_low.html	document	439 ms	Highest	
file?v=1&type=js&delay=2500	script	2.74 s	High	
file?v=2&type=js&delay=2500	script	2.75 s	High	
file?type=png&delay=100&v=1	png	140 ms	Low	
file?type=png&delay=100&v=2	png	139 ms	Low	
file?type=png&delay=100&v=3	png	208 ms	Low	
file?type=png&delay=100&v=4	png	210 ms	Low	
file?type=png&delay=100&v=5	png	208 ms	Low	
file?type=png&delay=100&v=6	png	200 ms	Low	
file?type=png&delay=100&v=7	png	200 ms	Low	
file?type=png&delay=100&v=8	png	202 ms	Low	
file?type=png&delay=100&v=9	png	200 ms	Low	
file?type=png&delay=100&v=10	png	218 ms	Low	

2 HIGH js

10 LOW img

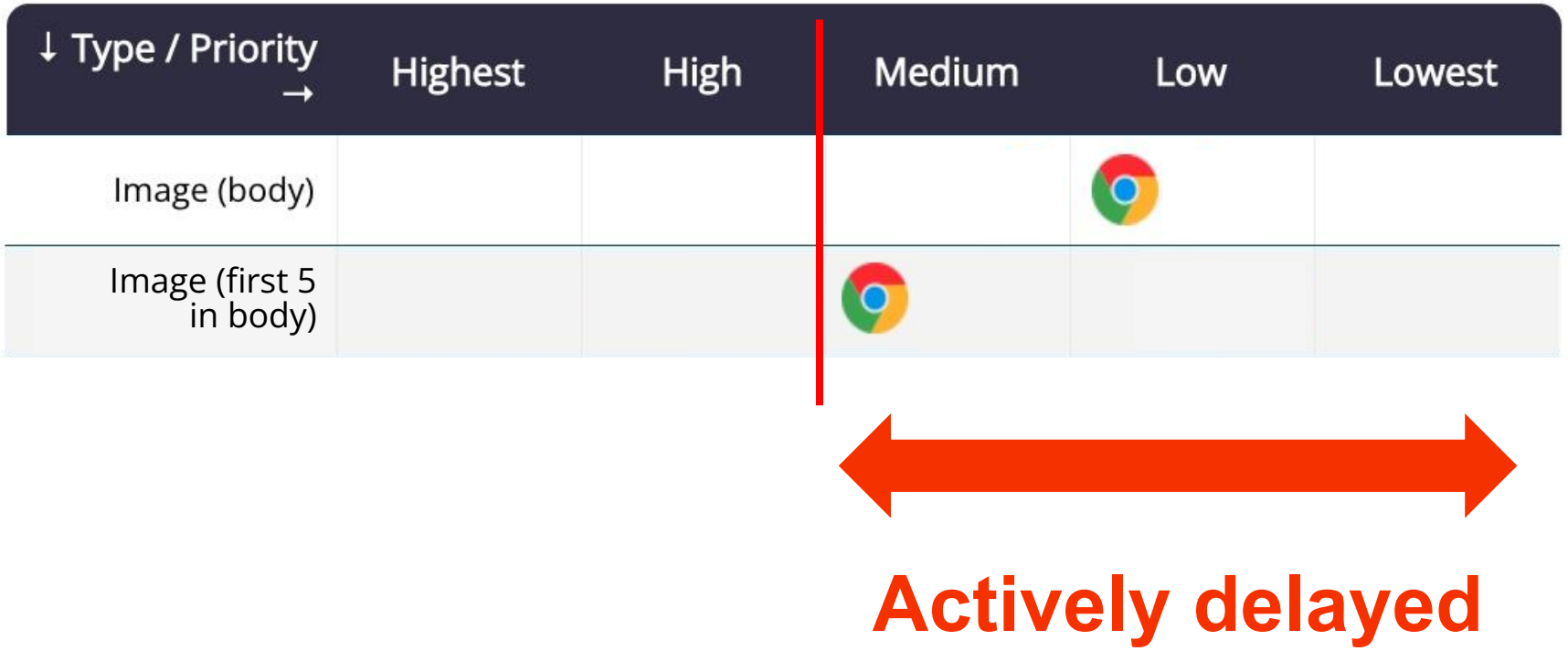


End of tight mode

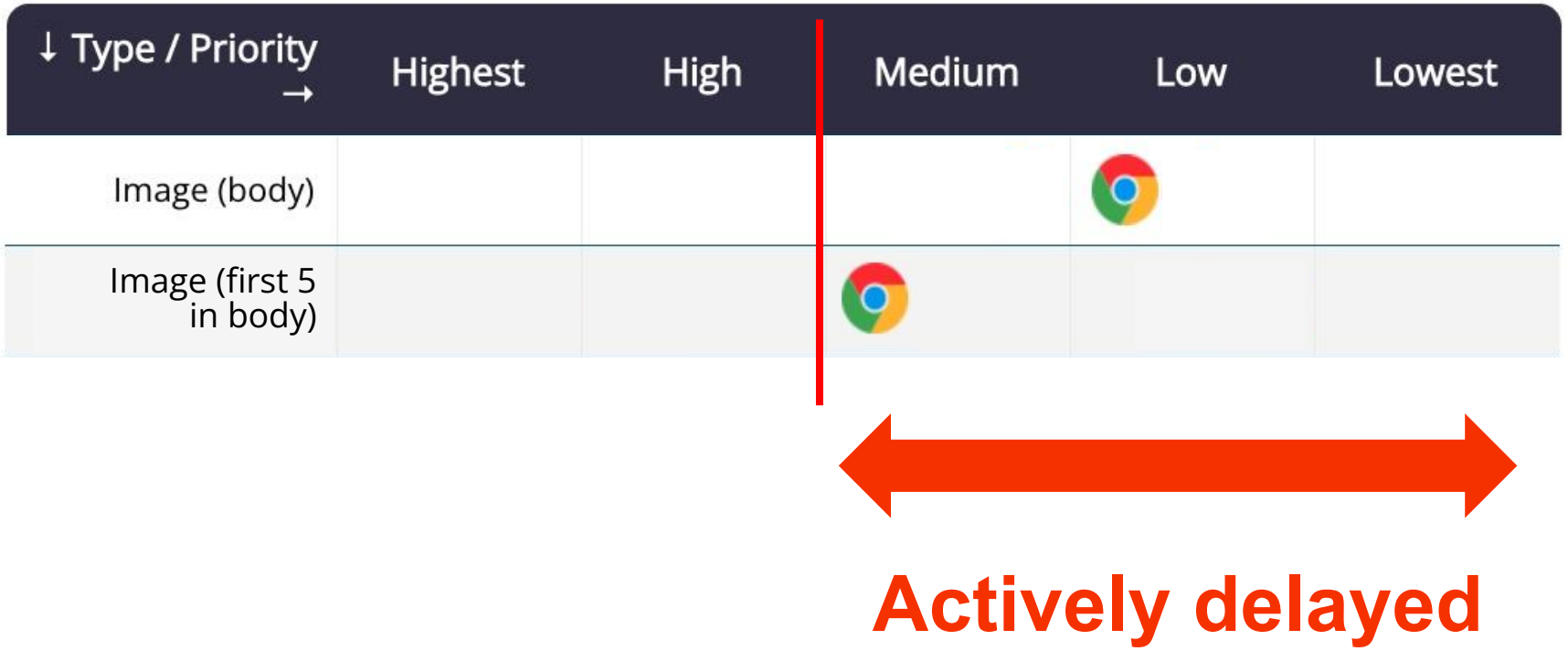
Akamai


Name
Type
Time
Priority
Waterfall
tightmode_simple.html document 59 ms Highest
file?v=1&type=js&delay=2500 script 2.60 s High
file?v=2&type=js&delay=2500 script 2.60 s High
file?type=png&delay=100&v=1 png 198 ms Medium
file?type=png&delay=100&v=2 png 198 ms Medium
file?type=png&delay=100&v=3 png 194 ms Medium
file?type=png&delay=100&v=4 png 194 ms Medium
file?type=png&delay=100&v=5 png 199 ms Medium
file?type=png&delay=100&v=6 png 162 ms Low
file?type=png&delay=100&v=7 png 158 ms Low
file?type=png&delay=100&v=8 png 163 ms Low
file?type=png&delay=100&v=9 png 162 ms Low
file?type=png&delay=100&v=10 png 162 ms Low
2 HIGH js
5 MEDIUM img
5 LOW img
End of
tight mode

Trying to improve LCP on the entire Web

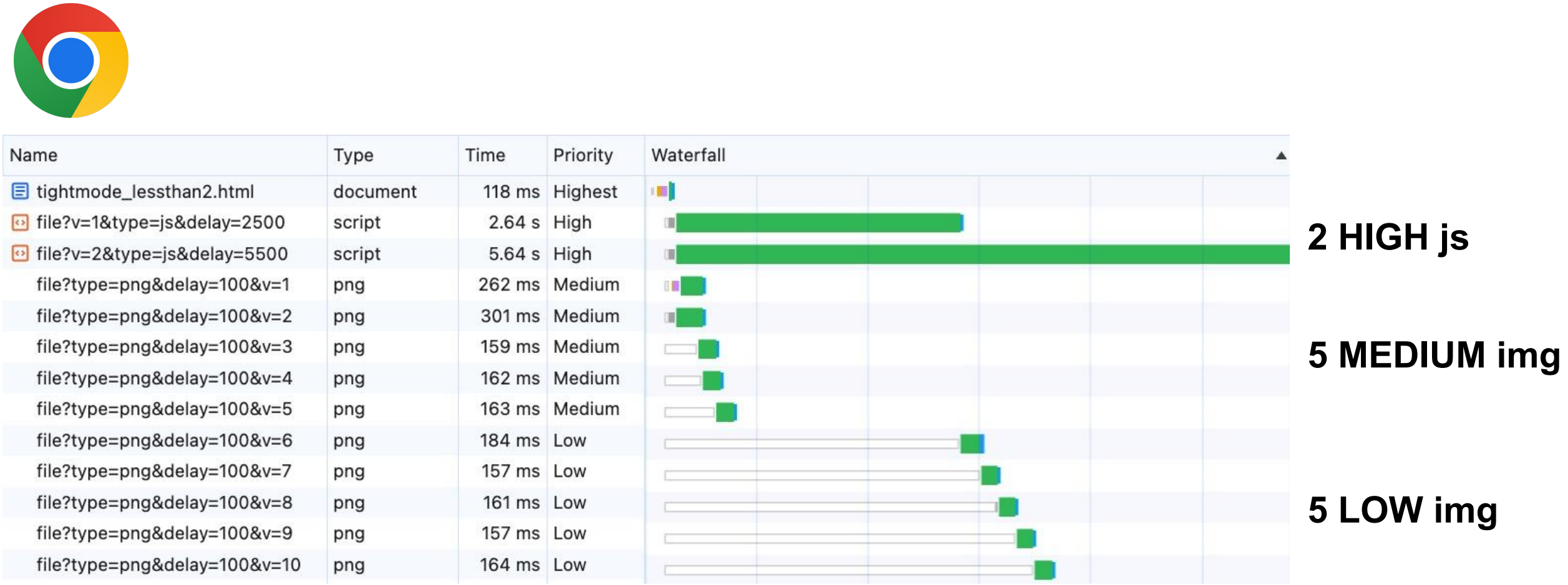


Trying to improve LCP on the entire Web



As of Chrome 117, Chrome will also load 2 Medium-priority requests at a time with no restrictions about other requests being in-flight.


Name
Type
Time
Priority
Waterfall
tightmode_simple.html document 59 ms Highest
file?v=1&type=js&delay=2500 script 2.60 s High
file?v=2&type=js&delay=2500 script 2.60 s High
file?type=png&delay=100&v=1 png 198 ms Medium
file?type=png&delay=100&v=2 png 198 ms Medium
file?type=png&delay=100&v=3 png 194 ms Medium
file?type=png&delay=100&v=4 png 194 ms Medium
file?type=png&delay=100&v=5 png 199 ms Medium
file?type=png&delay=100&v=6 png 162 ms Low
file?type=png&delay=100&v=7 png 158 ms Low
file?type=png&delay=100&v=8 png 163 ms Low
file?type=png&delay=100&v=9 png 162 ms Low
file?type=png&delay=100&v=10 png 162 ms Low
2 HIGH js
5 MEDIUM img
5 LOW img
End of
tight mode



"Low priority resources are only loaded if there are ***less than 2*** in-flight requests"

3 HIGHEST CSS

5 MEDIUM img

5 LOW img

End of tight mode

"After all blocking ***scripts*** have been executed"

```
1   <head>
2     <script src=script1.js></script>
3     <script src=script2.js></script>
4
5     <script src=script3.js defer></script>
6     <script src=script4.js defer></script>
7   </head>
8   <body>
9       <img src=img1.jpg />
10      <img src=img2.jpg />
11      ...
12      <img src=img9.jpg />
13      <img src=img10.jpg />
14  </body>
```

2 HIGH JS

2 LOW JS

5 MEDIUM IMG

5 LOW IMG

What will the waterfall look like for this HTML?

Akamai

```
1  <head>
2    <script src=script1.js></script>
3    <script src=script2.js></script>
4
5    <script src=script3.js defer></script>
6    <script src=script4.js defer></script>
7  </head>
8  <body>
9      <img src=img1.jpg />
10     <img src=img2.jpg />
11     ...
12     <img src=img9.jpg />
13     <img src=img10.jpg />
14 </body>
```

2 HIGH JS

2 LOW JS

5 MEDIUM IMG

5 LOW IMG

images in the <body>
delay
defer JS in the <head>

Name	Type	Time	Priority	Waterfall
tightmode_deferafterimg.html	document	131 ms	Highest	
file?v=1&type=js&delay=2500	script	2.71 s	High	
file?v=2&type=js&delay=2500	script	2.70 s	High	
file?type=png&delay=100&v=1	png	302 ms	Medium	
file?type=png&delay=100&v=2	png	302 ms	Medium	
file?type=png&delay=100&v=3	png	155 ms	Medium	
file?type=png&delay=100&v=4	png	194 ms	Medium	
file?type=png&delay=100&v=5	png	160 ms	Medium	
file?v=3&type=js&delay=2500&...	script	2.78 s	Low	
file?v=4&type=js&delay=2500&...	script	2.78 s	Low	
file?type=png&delay=100&v=6	png	315 ms	Low	
file?type=png&delay=100&v=7	png	314 ms	Low	
file?type=png&delay=100&v=8	png	314 ms	Low	
file?type=png&delay=100&v=9	png	315 ms	Low	
file?type=png&delay=100&v=10	png	313 ms	Low	

2 HIGH JS

5 MEDIUM IMG

2 LOW JS

5 LOW IMG

Akamai

Name	Type	Time	Priority	Waterfall
tightmode_simple.html	document	115 ms	Highest	
file?v=1&type=js&delay=2500	script	2.69 s	High	
file?v=2&type=js&delay=2500	script	2.69 s	High	
file?type=png&delay=100&v=1	png	253 ms	Medium	
file?type=png&delay=100&v=2	png	263 ms	Medium	
file?type=png&delay=100&v=3	png	151 ms	Medium	
file?type=png&delay=100&v=4	png	161 ms	Medium	
file?type=png&delay=100&v=5	png	149 ms	Medium	
file?type=png&delay=100&v=6	png	272 ms	Low	
file?type=png&delay=100&v=7	png	270 ms	Low	
file?type=png&delay=100&v=8	png	275 ms	Low	
file?type=png&delay=100&v=9	png	284 ms	Low	
file?type=png&delay=100&v=10	png	274 ms	Low	

Name	Type	Priority	Time	500.0ms	1000.0ms	1.50s	2.00s	2.50s
tightmode_simple.html	document	High	31.8ms					
file	js	High	2.57s					
file	js	High	2.56s					
file	png	Medium	2.74s					
file	png	Medium	2.75s					
file	png	Medium	2.75s					
file	png	Medium	2.76s					
file	png	Medium	2.75s					
file	png	Medium	2.75s					
file	png	Medium	2.76s					
file	png	Medium	2.75s					
file	png	Medium	2.74s					
file	png	Medium	2.76s					

No special casing of first 5 images

Akamai

Name	Type	Time	Priority	Waterfall
tightmode_lessthan2.html	document	205 ms	Highest	
file?v=1&type=js&delay=2500	script	2.66 s	High	
file?v=2&type=js&delay=5500	script	5.66 s	High	
file?type=png&delay=100&v=1	png	263 ms	Medium	
file?type=png&delay=100&v=2	png	275 ms	Medium	
file?type=png&delay=100&v=3	png	151 ms	Medium	
file?type=png&delay=100&v=4	png	157 ms	Medium	
file?type=png&delay=100&v=5	png	148 ms	Medium	
file?type=png&delay=100&v=6	png	153 ms	Low	
file?type=png&delay=100&v=7	png	161 ms	Low	
file?type=png&delay=100&v=8	png	143 ms	Low	
file?type=png&delay=100&v=9	png	148 ms	Low	
file?type=png&delay=100&v=10	png	148 ms	Low	

Name	Type	Priority	Time	1000.0ms	2.00s	3.00s	4.00s	5.00s
tightmode_lessthan2.html	document	High	73.6ms					
file	js	High	2.64s					
file	js	High	5.65s					
file	png	Low	2.80s					
file	png	Low	2.95s					
file	png	Low	3.10s					
file	png	Low	3.26s					
file	png	Low	3.41s					
file	png	Low	3.57s					
file	png	Low	3.72s					
file	png	Low	3.87s					
file	png	Low	4.03s					
file	png	Low	4.19s					

Max 2 things in flight

Akamai

Name	Type	Time	Priority	Waterfall
tightmode_cssonly	document	257 ms	Highest	
style-1.css?delay=2500	stylesheet	2.75 s	Highest	
style-2.css?delay=2500	stylesheet	2.85 s	Highest	
style-3.css?delay=5500	stylesheet	5.93 s	Highest	
image-1.jpg?type=png&delay=100&v=1	jpeg	503 ms	Medium	
image-1.jpg?type=png&delay=100&v=2	jpeg	372 ms	Medium	
image-1.jpg?type=png&delay=100&v=3	jpeg	807 ms	Medium	
image-1.jpg?type=png&delay=100&v=4	jpeg	807 ms	Medium	
image-1.jpg?type=png&delay=100&v=5	jpeg	805 ms	Medium	
image-1.jpg?type=png&delay=100&v=6	jpeg	805 ms	Low	
image-1.jpg?type=png&delay=100&v=7	jpeg	804 ms	Low	
image-1.jpg?type=png&delay=100&v=8	jpeg	849 ms	Low	
image-1.jpg?type=png&delay=100&v=9	jpeg	758 ms	Low	
image-1.jpg?type=png&delay=100&v=10	jpeg	849 ms	Low	

Name	Type	Priority	Time	2.00s	4.00s	6.00s
tightmode_cssonly	document	High	246ms			
style-1.css	css	High	2.71s			
style-2.css	css	High	2.79s			
style-3.css	css	High	5.78s			
image-1.jpg	jpg	Low	2.98s			
image-1.jpg	jpg	Low	3.12s			
image-1.jpg	jpg	Low	3.32s			
image-1.jpg	jpg	Low	3.46s			
image-1.jpg	jpg	Low	3.71s			
image-1.jpg	jpg	Low	3.86s			
image-1.jpg	jpg	Low	4.02s			
image-1.jpg	jpg	Low	4.17s			
image-1.jpg	jpg	Low	4.36s			
image-1.jpg	jpg	Low	4.52s			

CSS also triggers tight mode!

Akamai

Name	Type	Time	Priority	Waterfall
tightmode_bodyjs.html	document	129 ms	Highest	
file?v=1&type=js&delay=2500	script	2.68 s	High	
file?v=2&type=js&delay=2500	script	2.69 s	High	
file?type=png&delay=100&v=1	png	279 ms	Medium	
file?type=png&delay=100&v=2	png	292 ms	Medium	
file?type=png&delay=100&v=3	png	284 ms	Medium	
file?type=png&delay=100&v=4	png	279 ms	Medium	
file?type=png&delay=100&v=5	png	292 ms	Medium	
file?type=png&delay=100&v=6	png	292 ms	Low	
file?type=png&delay=100&v=7	png	279 ms	Low	
file?type=png&delay=100&v=8	png	277 ms	Low	
file?type=png&delay=100&v=9	png	282 ms	Low	
file?type=png&delay=100&v=10	png	283 ms	Low	

JS top of <body>

Name	Type	Priority	Time	1000.0ms	2.00s
tightmode_bodyjs.html	document	High	133ms		
file	js	High	2.57s		
file	js	High	2.57s		
file	png	Medium	2.73s		
file	png	Medium	2.74s		
file	png	Medium	2.75s		
file	png	Medium	2.76s		
file	png	Medium	2.76s		
file	png	Medium	2.74s		
file	png	Medium	2.76s		
file	png	Medium	2.76s		
file	png	Medium	2.75s		
file	png	Medium	2.75s		

Blocking JS or CSS delay whatever's behind them

Akamai

JS bottom of <body>

Name	Type	Time	Priority	Waterfall
tightmode_bottomjs.html	document	95 ms	Highest	
file?type=png&delay=100&v=1	png	163 ms	High	
file?type=png&delay=100&v=2	png	169 ms	High	
file?type=png&delay=100&v=3	png	171 ms	High	
file?type=png&delay=100&v=4	png	169 ms	High	
file?type=png&delay=100&v=5	png	170 ms	High	
file?type=png&delay=100&v=6	png	169 ms	High	
file?type=png&delay=100&v=7	png	167 ms	High	
file?type=png&delay=100&v=8	png	164 ms	High	
file?type=png&delay=100&v=9	png	163 ms	High	
file?type=png&delay=100&v=10	png	161 ms	High	
file?v=1&type=js&delay=2500	script	2.57 s	Medium	
file?v=2&type=js&delay=2500	script	2.57 s	Medium	

Name	Type	Priority	Time	1000.0ms	2.00s
tightmode_bottomjs.html	document	High	101ms		
file	png	Medium	312ms		
file	png	Medium	316ms		
file	png	Medium	364ms		
file	png	Medium	364ms		
file	png	Medium	367ms		
file	png	Medium	326ms		
file	png	Medium	370ms		
file	png	Medium	361ms		
file	png	Medium	357ms		
file	png	Medium	367ms		
file	js	High	2.74s		
file	js	High	2.73s		

Blocking JS or CSS delay whatever's behind them

Akamai

JS middle of <body>

Name	Type	Time	Priority	Waterfall
tightmode_jsinbetween.html	document	103 ms	Highest	
file?type=png&delay=100&v=1	png	303 ms	High	
file?type=png&delay=100&v=2	png	288 ms	High	
file?type=png&delay=100&v=3	png	288 ms	High	
file?type=png&delay=100&v=4	png	309 ms	High	
file?type=png&delay=100&v=5	png	284 ms	High	
file?v=1&type=js&delay=2500	script	2.71 s	Medium	
file?v=2&type=js&delay=2500	script	2.69 s	Medium	
file?type=png&delay=100&v=6	png	285 ms	Low	
file?type=png&delay=100&v=7	png	289 ms	Low	
file?type=png&delay=100&v=8	png	282 ms	Low	
file?type=png&delay=100&v=9	png	279 ms	Low	
file?type=png&delay=100&v=10	png	284 ms	Low	

Name	Type	Priority	Time	1000.0ms	2.00s
tightmode_jsinbetween.html	document	High	91.5ms		
file	png	Low	174ms		
file	png	Low	177ms		
file	png	Medium	2.78s		
file	png	Medium	2.79s		
file	png	Medium	2.80s		
file	js	High	2.62s		
file	js	High	2.63s		
file	png	Medium	2.80s		
file	png	Medium	2.80s		
file	png	Medium	2.79s		
file	png	Medium	2.79s		
file	png	Medium	2.80s		

Some weird heuristics at work here…

Akamai

```
<head>
  <script src=script1.js></script>
  <script src=script2.js></script>

  <script src=script3.js defer></script>
  <script src=script4.js defer></script>
</head>
<body>
    <img src=img1.jpg />
    <img src=img2.jpg />
    ...
    <img src=img9.jpg />
    <img src=img10.jpg />
</body>
```

What will the waterfall look like for this HTML?

Akamai

```html
<head>
  <script src=script1.js></script>
  <script src=script2.js></script>

  <script src=script3.js defer></script>
  <script src=script4.js defer></script>
</head>
<body>
    <img src=img1.jpg />
    <img src=img2.jpg />
    ...
    <img src=img9.jpg />
    <img src=img10.jpg />
</body>
```

Name	Type	Priority	Time
tightmode_deferafterimg.html	document	High	49.1ms
file	js	High	2.74s
file	js	High	2.75s
file	js	High	2.74s
file	js	High	2.73s
file	png	Medium	2.91s
file	png	Medium	2.92s
file	png	Medium	2.98s
file	png	Medium	2.97s
file	png	Medium	2.98s
file	png	Medium	2.91s
file	png	Medium	2.98s
file	png	Medium	2.99s
file	png	Medium	2.92s
file	png	Medium	2.98s

Async/Defer JS don't trigger tight mode by themselves, but are downloaded in it

Name	Type	Time	Priority	Waterfall
tightmode_deferafterimg.html	document	112 ms	Highest	
file?v=1&type=js&delay=2500	script	2.88 s	High	
file?v=2&type=js&delay=2500	script	2.88 s	High	
file?type=png&delay=100&v=1	png	478 ms	Medium	
file?type=png&delay=100&v=2	png	486 ms	Medium	
file?type=png&delay=100&v=3	png	156 ms	Medium	
file?type=png&delay=100&v=4	png	170 ms	Medium	
file?type=png&delay=100&v=5	png	163 ms	Medium	
file?v=3&type=js&delay=2500&defer=tr...	script	2.61 s	Low	
file?v=4&type=js&delay=2500&defer=tr...	script	2.61 s	Low	
file?type=png&delay=100&v=6	png	150 ms	High	
file?type=png&delay=100&v=7	png	152 ms	High	
file?type=png&delay=100&v=8	png	212 ms	High	
file?type=png&delay=100&v=9	png	205 ms	High	
file?type=png&delay=100&v=10	png	212 ms	High	

Name	Type	Priority	Time	1000.0ms 2.00s 3.00s
tightmode_deferafterimg.html	document	High	49.1ms	
file	js	High	2.74s	
file	js	High	2.75s	
file	js	High	2.74s	
file	js	High	2.73s	
file	png	Medium	2.91s	
file	png	Medium	2.92s	
file	png	Medium	2.98s	
file	png	Medium	2.97s	
file	png	Medium	2.98s	
file	png	Medium	2.91s	
file	png	Medium	2.98s	
file	png	Medium	2.99s	
file	png	Medium	2.92s	
file	png	Medium	2.98s	

Exact same HTML,

***radically* different behaviour**

Akamai

Tight mode

While blocking JS in the <head> is busy

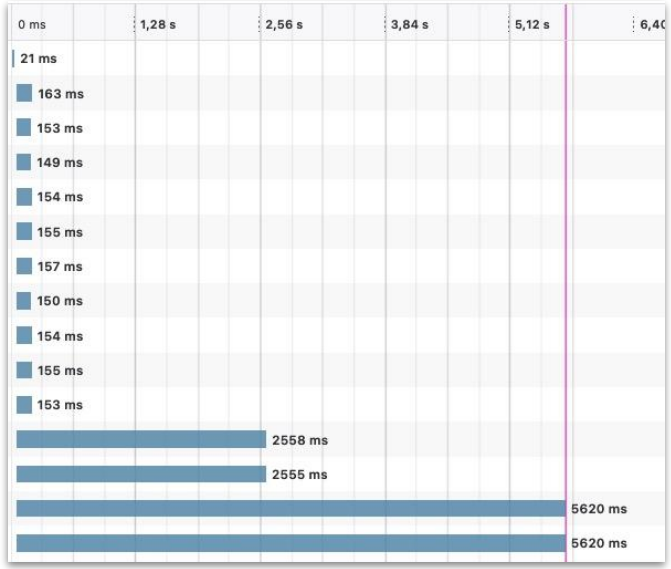
- Only LOW/LOWEST if fewer than 2 things in flight
- 2 MEDIUM at a time

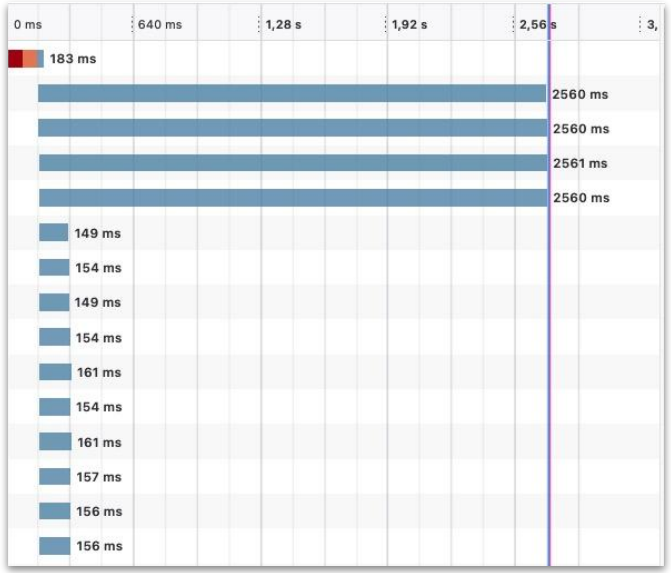
While blocking JS **or CSS** ~*anywhere* is busy

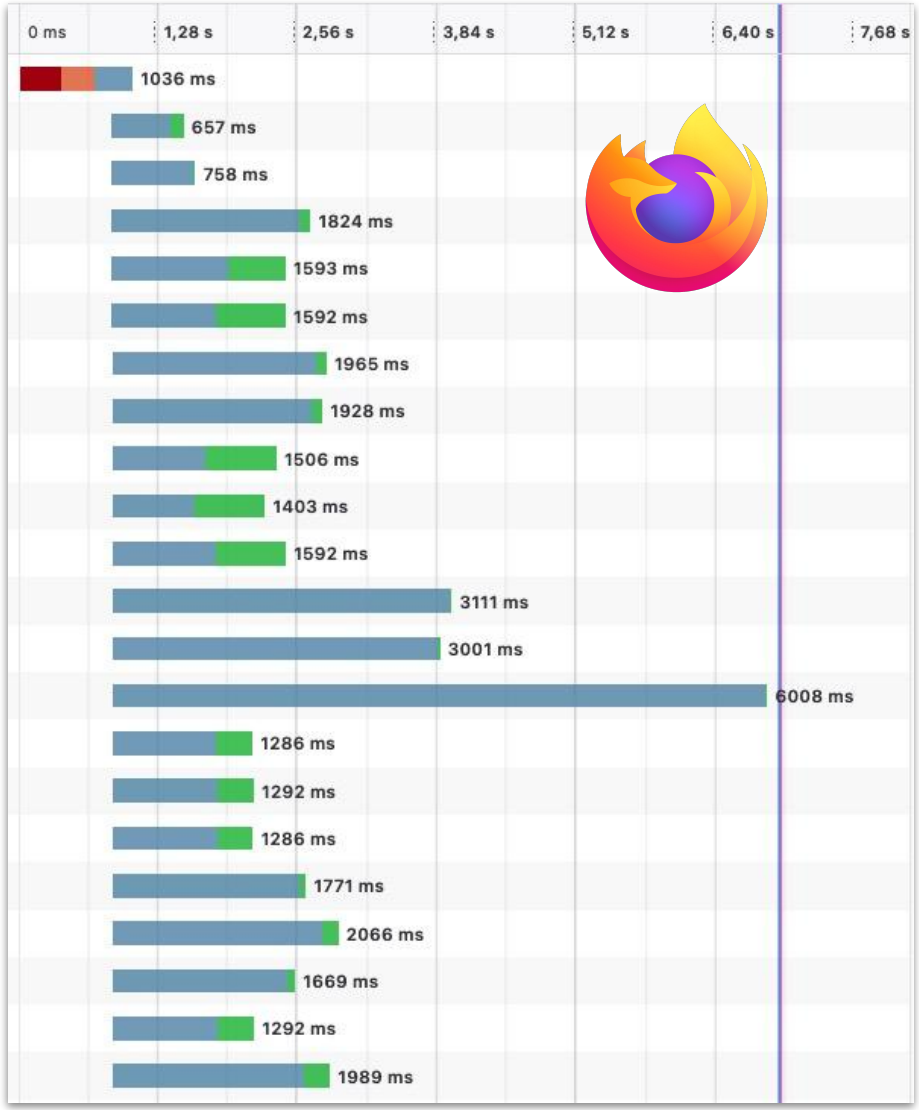
- Only MEDIUM/LOW/LOWEST if fewer than 2 things in flight
  - With the exception of async/defer JS, those always get requested asap

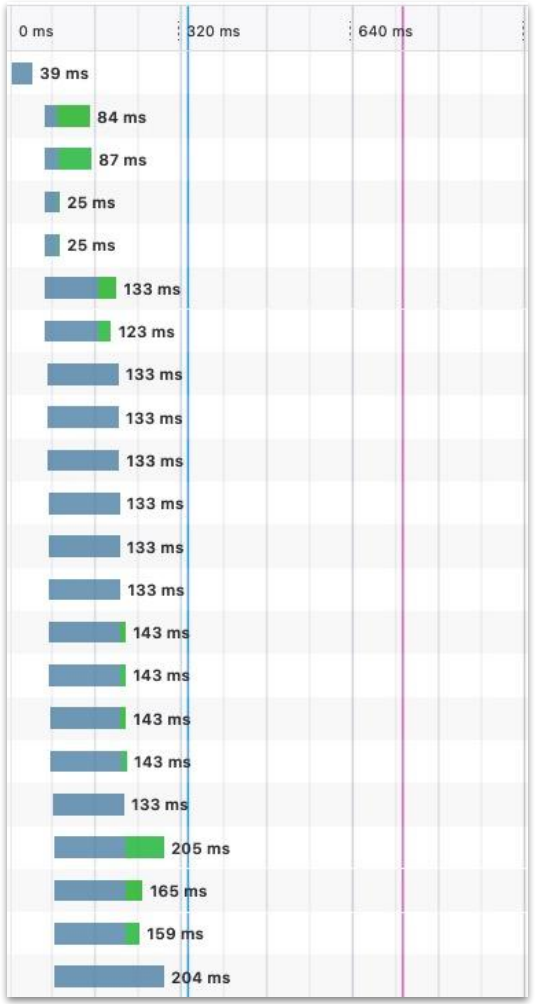
Akamai

Firefox doesn't do Tight Mode in HTTP/2 and /3





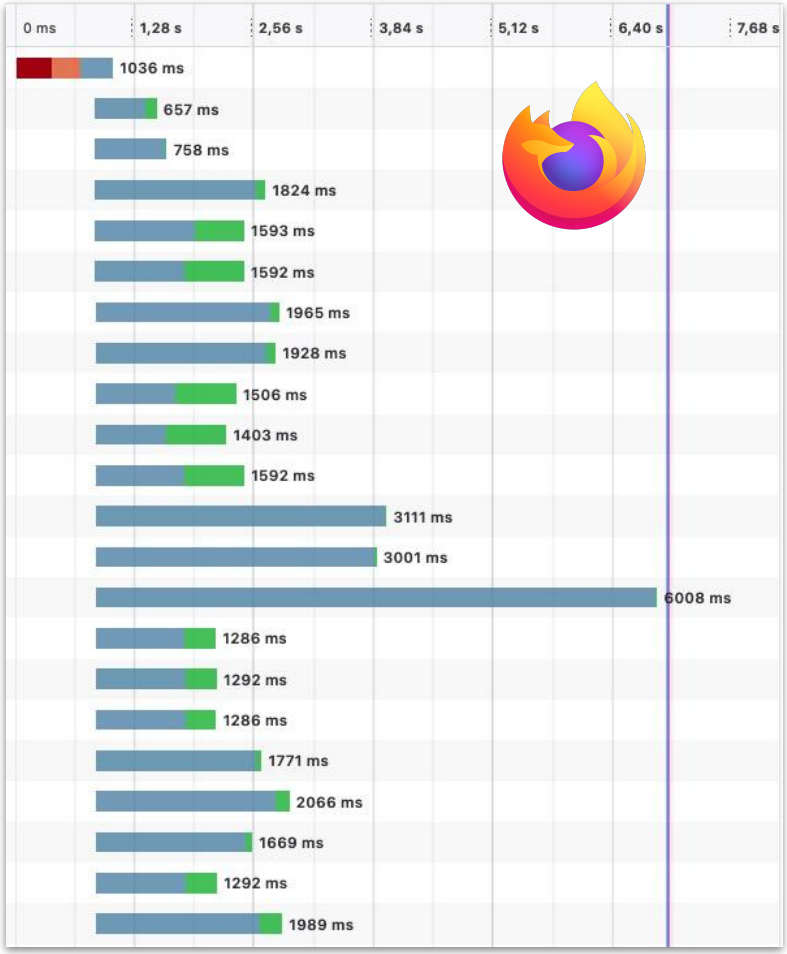




Akamai

That doesn’t (necessarily) make it slower ;)

(pre-empting the FireFox Fan Force Five on YouTube)



@saiv46 1 month ago (edited)

Firefox is actually better without fetchpriority. It's a solution to the problem created by Chrome/Safari themselves. I hate how much implementation details gets pushed on developers.

@HildeTheOkayish 1 month ago

Honestly prefer Firefox. For one, as webdeveloper you know what you get. No browser trying to outsmart you and ruining any optimisation you build in. But also, clearly the solution here should not be the browser trying to awkwardly fix bad server implementations but rather servers fixing it. And if they don't change it then as developer I would want to know if I can easily work around their …

@gg-gn3re 2 weeks ago

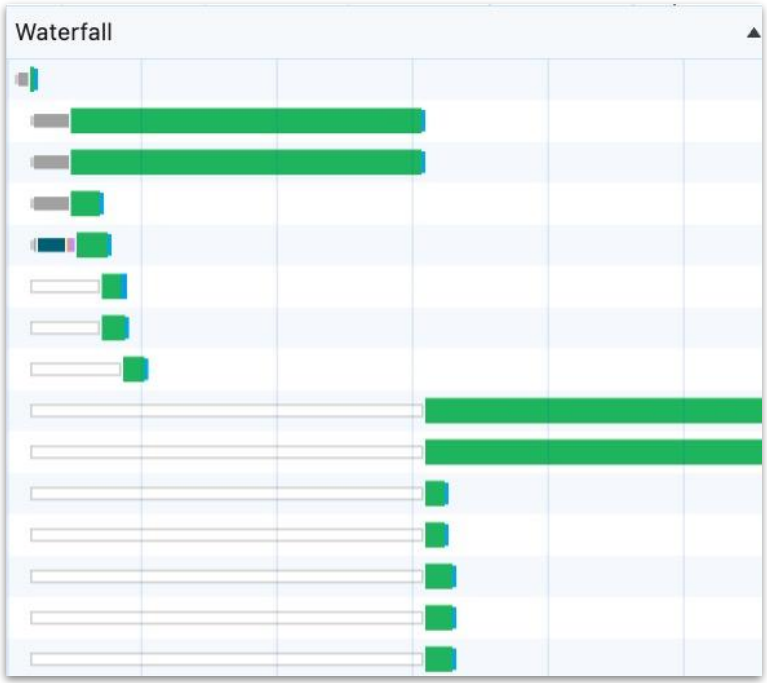
yep, always hilarious. FF has always loaded overall pages faster. Every single year for the past 15 years it has never once been slower. Chrome optimized their browser to load the "looks like the page is done" junk first which slows down overall load but appears to be done faster, this is what tricks people. …

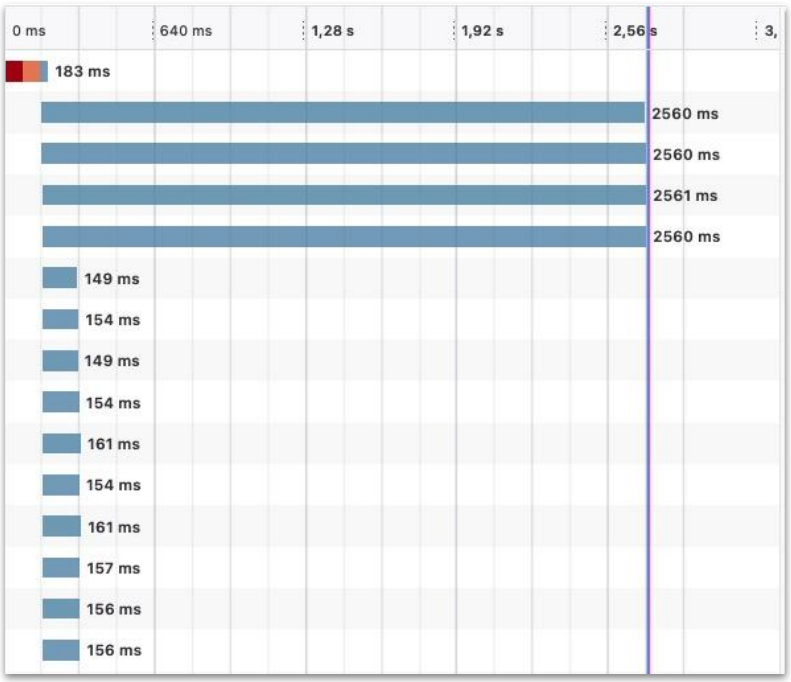
https://www.youtube.com/watch?v=p0lFyPuH8Zs

Akamai

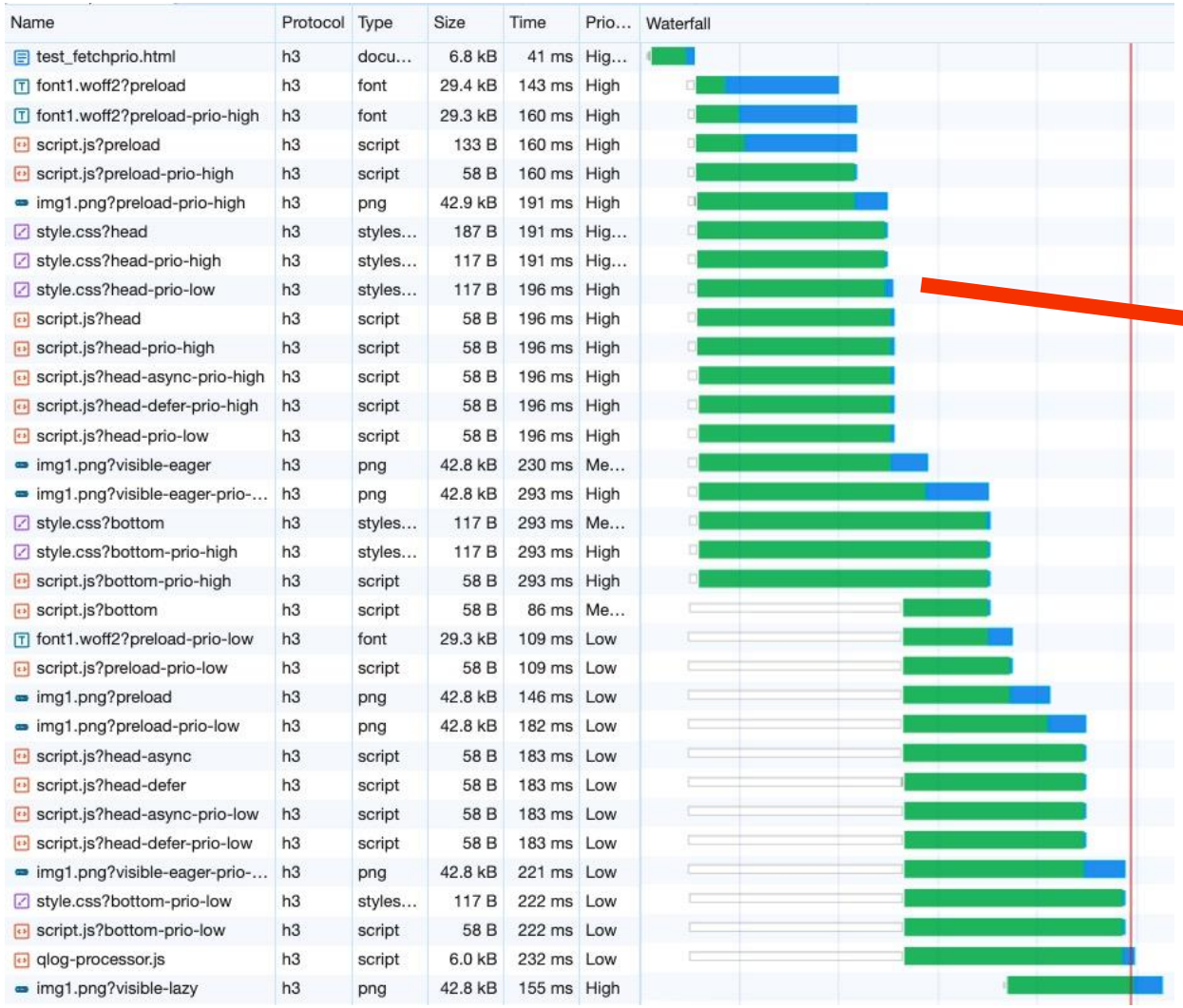
Exact same HTML,

radically different behaviour





How to fix wrong browser behaviour?



Name	Protocol	Type	Size	Time	Prio…	Waterfall
test_fetchprio.html	h3	docu…	6.8 kB	41 ms	Hig…	
font1.woff2?preload	h3	font	29.4 kB	143 ms	High	
font1.woff2?preload-prio-high	h3	font	29.3 kB	160 ms	High	
script.js?preload	h3	script	133 B	160 ms	High	
script.js?preload-prio-high	h3	script	58 B	160 ms	High	
img1.png?preload-prio-high	h3	png	42.9 kB	191 ms	High	
style.css?head	h3	styles…	187 B	191 ms	Hig…	
style.css?head-prio-high	h3	styles…	117 B	191 ms	Hig…	
style.css?head-prio-low	h3	styles…	117 B	196 ms	High	
script.js?head	h3	script	58 B	196 ms	High	
script.js?head-prio-high	h3	script	58 B	196 ms	High	
script.js?head-async-prio-high	h3	script	58 B	196 ms	High	
script.js?head-defer-prio-high	h3	script	58 B	196 ms	High	
script.js?head-prio-low	h3	script	58 B	196 ms	High	
img1.png?visible-eager	h3	png	42.8 kB	230 ms	Me…	
img1.png?visible-eager-prio-…	h3	png	42.8 kB	293 ms	High	
style.css?bottom	h3	styles…	117 B	293 ms	Me…	
style.css?bottom-prio-high	h3	styles…	117 B	293 ms	High	
script.js?bottom-prio-high	h3	script	58 B	293 ms	High	
script.js?bottom	h3	script	58 B	86 ms	Me…	
font1.woff2?preload-prio-low	h3	font	29.3 kB	109 ms	Low	
script.js?preload-prio-low	h3	script	58 B	109 ms	Low	
img1.png?preload	h3	png	42.8 kB	146 ms	Low	
img1.png?preload-prio-low	h3	png	42.8 kB	182 ms	Low	
script.js?head-async	h3	script	58 B	183 ms	Low	
script.js?head-defer	h3	script	58 B	183 ms	Low	
script.js?head-async-prio-low	h3	script	58 B	183 ms	Low	
script.js?head-defer-prio-low	h3	script	58 B	183 ms	Low	
img1.png?visible-eager-prio-…	h3	png	42.8 kB	221 ms	Low	
style.css?bottom-prio-low	h3	styles…	117 B	222 ms	Low	
script.js?bottom-prio-low	h3	script	58 B	222 ms	Low	
qlog-processor.js	h3	script	6.0 kB	232 ms	Low	
img1.png?visible-lazy	h3	png	42.8 kB	155 ms	High	

This should actually be down there
(or vice versa)

Akamai

FetchPriority to the rescue!?

```
<img src="lcp-image.jpg" fetchpriority="high">
```

```
<link rel="preload" href="/defer.js" as="script" fetchpriority="low">
```

https://web.dev/fetch-priority

Akamai

FetchPriority doesn't control priority *directly*

```
<img src="lcp-image.jpg" fetchpriority="high">
<img src="lcp-image.jpg" fetchpriority="low">
```

```
<img src="lcp-image.jpg" fetchpriority="highest">
<img src="lcp-image.jpg" fetchpriority="lowest">
```

https://web.dev/fetch-priority

How to get stuff INTO tight mode?

fetchpriority=high

- Images
- Defer/Async JS
- JS on the bottom of the <body>

- Images

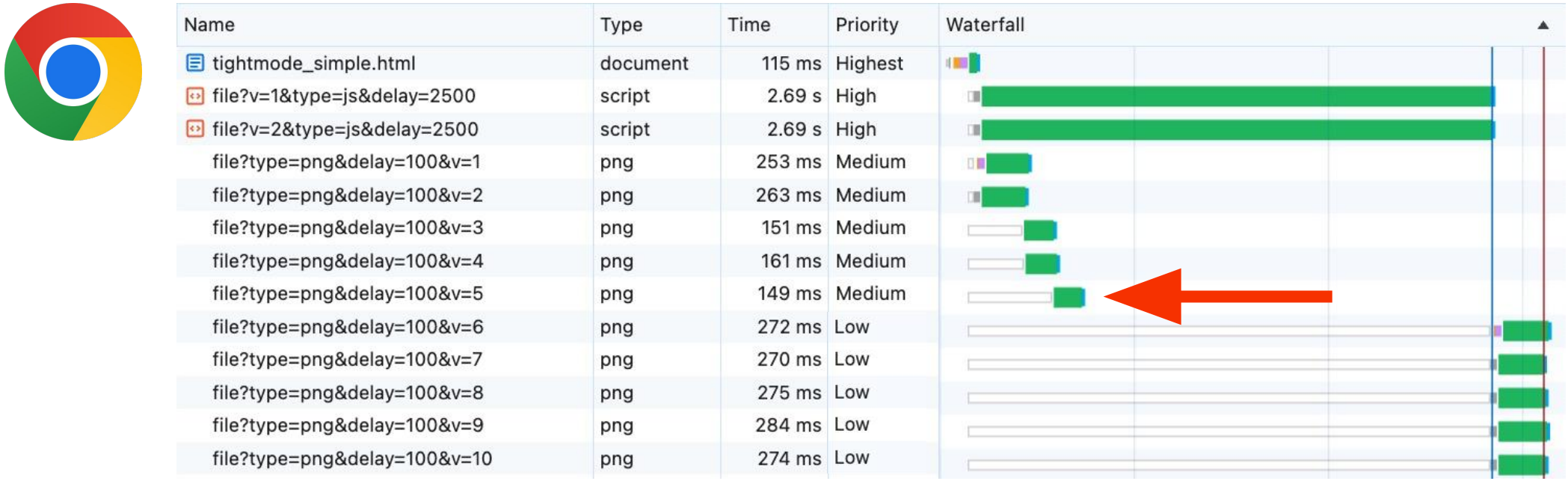
Name	Type	Priority	Time
tightmode_simple.html	document	High	31.8ms
file	js	High	2.57s
file	js	High	2.56s
file	png	Medium	2.74s
file	png	Medium	2.75s
file	png	Medium	2.75s
file	png	Medium	2.76s
file	png	Medium	2.75s
file	png	Medium	2.75s
file	png	Medium	2.76s
file	png	Medium	2.75s
file	png	Medium	2.74s
file	png	Medium	2.76s

Name	Type	Priority	Time
prio_fifthimage.html	document	High	411ms
file	js	High	2.87s
file	js	High	2.86s
file	png	Medium	3.02s
file	png	Medium	3.03s
file	png	Medium	3.11s
file	png	Medium	3.11s
file	png	High	467ms
file	png	Medium	3.12s
file	png	Medium	3.11s
file	png	Medium	3.12s
file	png	Medium	3.05s
file	png	Medium	3.11s

fetchpriority= high

causes image 5 to load in tight mode

Akamai



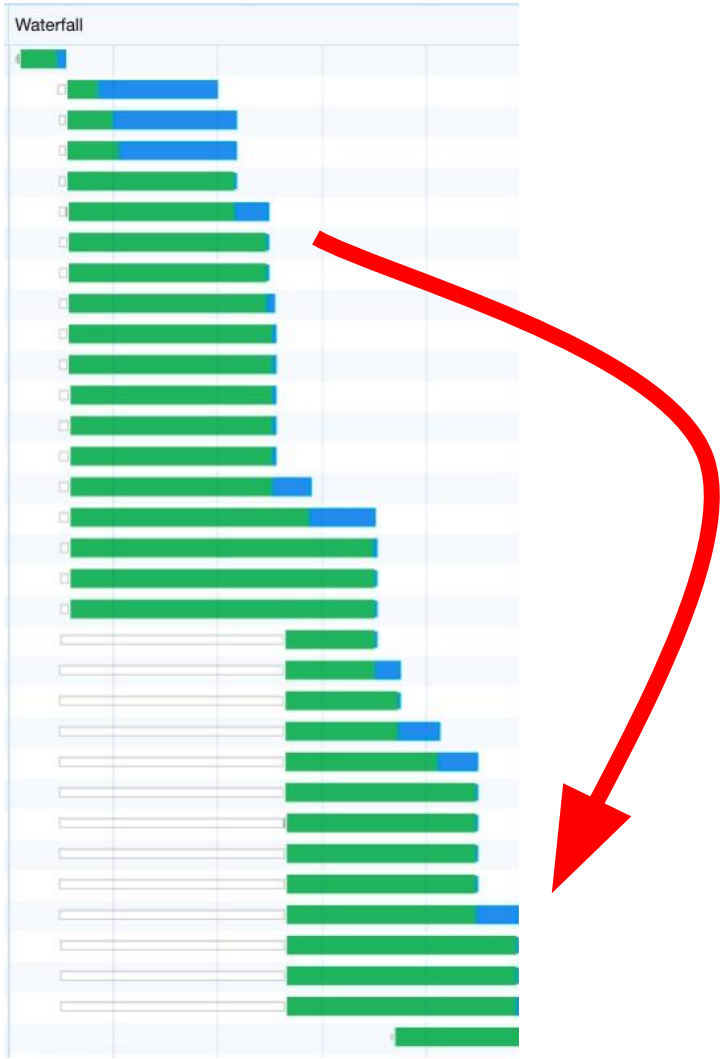
Name	Type	Time	Priority	Waterfall
tightmode_simple.html	document	115 ms	Highest	
file?v=1&type=js&delay=2500	script	2.69 s	High	
file?v=2&type=js&delay=2500	script	2.69 s	High	
file?type=png&delay=100&v=1	png	253 ms	Medium	
file?type=png&delay=100&v=2	png	263 ms	Medium	
file?type=png&delay=100&v=3	png	151 ms	Medium	
file?type=png&delay=100&v=4	png	161 ms	Medium	
file?type=png&delay=100&v=5	png	149 ms	Medium	
file?type=png&delay=100&v=6	png	272 ms	Low	
file?type=png&delay=100&v=7	png	270 ms	Low	
file?type=png&delay=100&v=8	png	275 ms	Low	
file?type=png&delay=100&v=9	png	284 ms	Low	
file?type=png&delay=100&v=10	png	274 ms	Low	

Name	Type	Time	Priority	Waterfall
prio_fifthimage.html	document	93 ms	Highest	
file?v=1&type=js&delay=2500	script	2.59 s	High	
file?v=2&type=js&delay=2500	script	2.59 s	High	
file?type=png&delay=100&v=1	png	185 ms	Medium	
file?type=png&delay=100&v=2	png	185 ms	Medium	
file?type=png&delay=100&v=5	png	185 ms	High	
file?type=png&delay=100&v=3	png	159 ms	Medium	
file?type=png&delay=100&v=4	png	159 ms	Medium	
file?type=png&delay=100&v=6	png	143 ms	High	
file?type=png&delay=100&v=7	png	150 ms	High	
file?type=png&delay=100&v=8	png	156 ms	High	
file?type=png&delay=100&v=9	png	155 ms	High	
file?type=png&delay=100&v=10	png	150 ms	High	

image 5 is requested before 3 and 4

Akamai

How to get stuff OUT OF tight mode?



fetchpriority=low

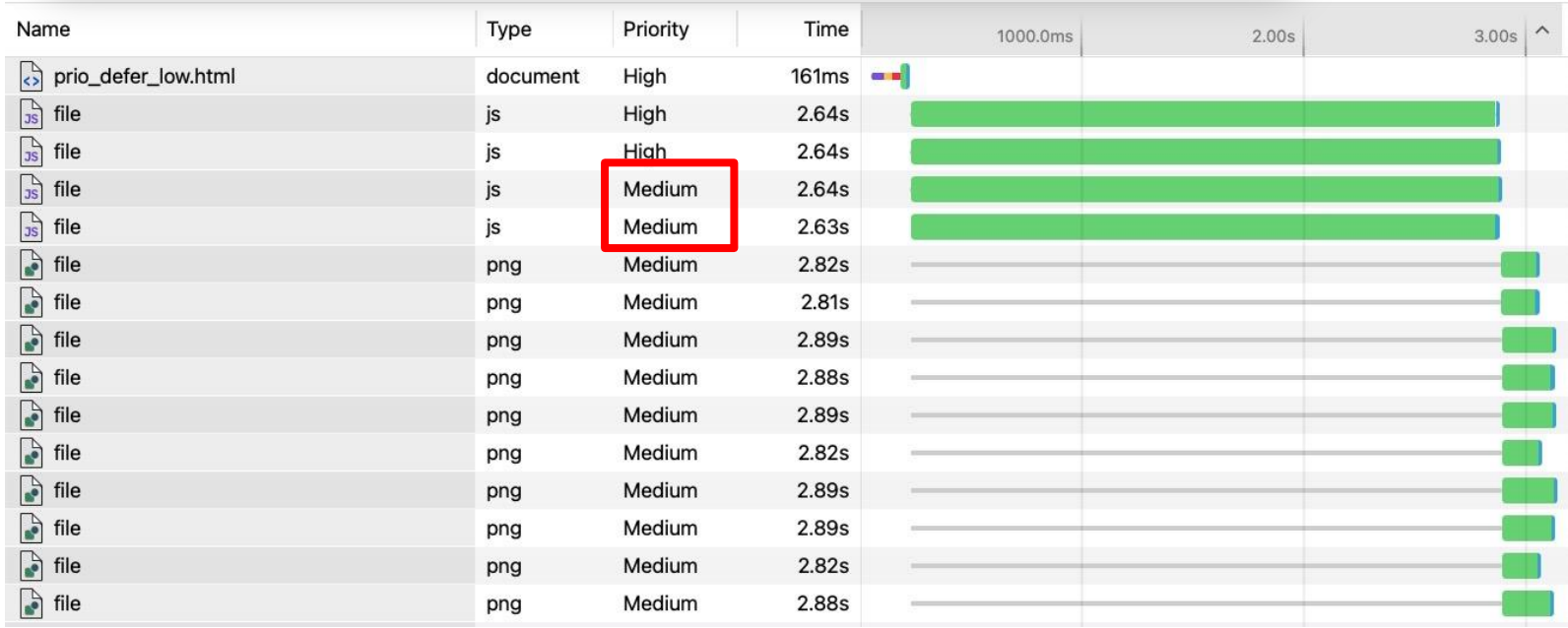
- First 5 images
- JS early and CSS late in <body>
- Preloaded fonts
- Preloaded async/defer JS

- **NOTHING AT ALL?!?**

Akamai

```html
<head>
  <script src=script1.js></script>
  <script src=script2.js></script>

  <script src=script3.js defer fetchpriority=low></script>
  <script src=script4.js defer fetchpriority=low></script>
</head>
<body>
    <img src=img1.jpg />
    <img src=img2.jpg />
    ...
    <img src=img9.jpg />
    <img src=img10.jpg />
</body>
```



Akamai

FetchPriority since October 2024!

https://caniuse.com/?search=fetchpriority
https://firefox-source-docs.mozilla.org/networking/http/prioritization.html

Akamai

Very extensive rework of prioritization system!

Resource Type	Class of Service	supportsPriority	Urgency	Incremental
HTML, Root Document	`UrgentStart (64)`	`PRIORITY_HIGHEST, -20`	`0`	`true`

CSS (`<head>` , Render-Blocking)	`Leader (1)`	`PRIORITY_NORMAL, 0`	`2`
			`fetchpriority=high: 0`
			`fetchpriority=low: 2`
				`false`

CSS (rel=preload)	`Leader (1)`	`PRIORITY_HIGHEST, -20`	`0`
			`fetchpriority=high: 0`
			`fetchpriority=low: 1`
				`false`

CSS (Body)	`Leader (1)`	`PRIORITY_NORMAL, 0`	`3`
			`fetchpriority=high: 2`
			`fetchpriority=low: 4`
				`false`

JavaScript (Blocking)	`Leader (1)`	`PRIORITY_NORMAL, 0`	`2`
			`fetchpriority=high: 1`
			`fetchpriority=low: 3`
				`false`

JavaScript (rel=preload)	`Unblocked (16)`	`PRIORITY_HIGHEST, -20`	`1`
			`fetchpriority=high: 1`
			`fetchpriority=low: 4`
				`false`

JavaScript (Async)	`TailAllowed (512), Unblocked (16)`	`PRIORITY_NORMAL, 0`	`3`
			`fetchpriority=high: 2`
			`fetchpriority=low: 4`
				`false`

JavaScript (Defer)	`Unblocked (16)`	`PRIORITY_NORMAL, 0`	`3`
			`fetchpriority=high: 2`
			`fetchpriority=low: 4`
				`false`

Font @font-face	`TailForbidden (1024)`	`PRIORITY_HIGH, -10`	`3`	`false`

Font (rel=preload)	`TailForbidden (1024), Unblocked (16)`	`PRIORITY_HIGH, -10`
		`fetchpriority=high: PRIORITY_HIGHEST, -20`
		`fetchpriority=low: PRIORITY_LOW, 10`
			`2`
			`fetchpriority=high: 1`
			`fetchpriority=low: 4`
				`false`

Image	`(0)`	`PRIORITY_LOW, 10`
		`fetchpriority=high: PRIORITY_HIGH, -10`
		`fetchpriority=low: PRIORITY_LOWEST, 20`
			`5`
			`fetchpriority=high: 3`
			`fetchpriority=low: 6`
				`true`

Image (rel=preload)	`(0)`	`PRIORITY_LOW, 10`
		`fetchpriority=high: PRIORITY_HIGH, -10`
		`fetchpriority=low: PRIORITY_LOWEST, 20`
			`4`
			`fetchpriority=high: 3`
			`fetchpriority=low: 5`
				`true`

Image (About to Be Rendered)	`(0)`	`PRIORITY_HIGH, -10`	`3`	`true`

Fetch	`(0)`	`PRIORITY_NORMAL, 0`
		`fetchpriority=high: PRIORITY_HIGH, -10`
		`fetchpriority=low: PRIORITY_LOW, 10`
			`4`
			`fetchpriority=high: 3`
			`fetchpriority=low: 5`
				`false`

https://firefox-source-docs.mozilla.org/networking/http/prioritization.html

Akamai

Browsers don’t agree on Priorities…

↓ Type / Priority →	Highest	High	Medium	Low	Lowest
Font (@font-face)					
Font preload			 		

```
<preload src="font.woff2" as="font" type="font/woff2" crossorigin>
```

https://calendar.perfplanet.com/2022/http-3-prioritization-demystified/

Akamai

…nor on the impact of FetchPriority


```
<preload src="font.woff2" as="font" type="font/woff2" crossorigin fetchpriority="high">
<preload src="font.woff2" as="font" type="font/woff2" crossorigin fetchpriority="low">
```

https://calendar.perfplanet.com/2022/http-3-prioritization-demystified/

Akamai

```html
<head>
  <script src=script1.js defer></script>
</head>
<body>
  <img src=lcp.png fetchpriority=high />
</body>

```

Name	Type	Time	Priority	Waterfall
ff_lcpbeforedefer.html	document	358 ms	Highest	
script_large.js?defer=true	script	1.32 s	Low	
img1.png?v=1	avif	60 ms	High	

Name	Type	Priority	Time	500.0ms	1000.0ms	1.50s	2.00s
ff_lcpbeforedefer.html	document	High	88.6ms				
script_large.js	js	High	1.53s				
img1.png	avif	High	391ms				

File	Type	Priority	Duration	0 ms	320 ms	640 ms
ff_lcpbeforedefer.html	html	Highest	85 ms	85 ms		
script_large.js?defer=true	js	Low	612 ms	612 ms		
img1.png?v=1	avif	Low	612 ms	612 ms		

from ultralow to low

That doesn’t (necessarily) make it worse ;)

(pre-empting the FireFox Fan Force Five on YouTube)

@saiv46 1 month ago (edited)

Firefox is actually better without fetchpriority. It's a solution to the problem created by Chrome/Safari themselves. I hate how much implementation details gets pushed on developers.

@HildeTheOkayish 1 month ago

Honestly prefer Firefox. For one, as webdeveloper you know what you get. No browser trying to outsmart you and ruining any optimisation you build in. But also, clearly the solution here should not be the browser trying to awkwardly fix bad server implementations but rather servers fixing it. And if they don't change it then as developer I would want to know if I can easily work around their …

@gg-gn3re 2 weeks ago

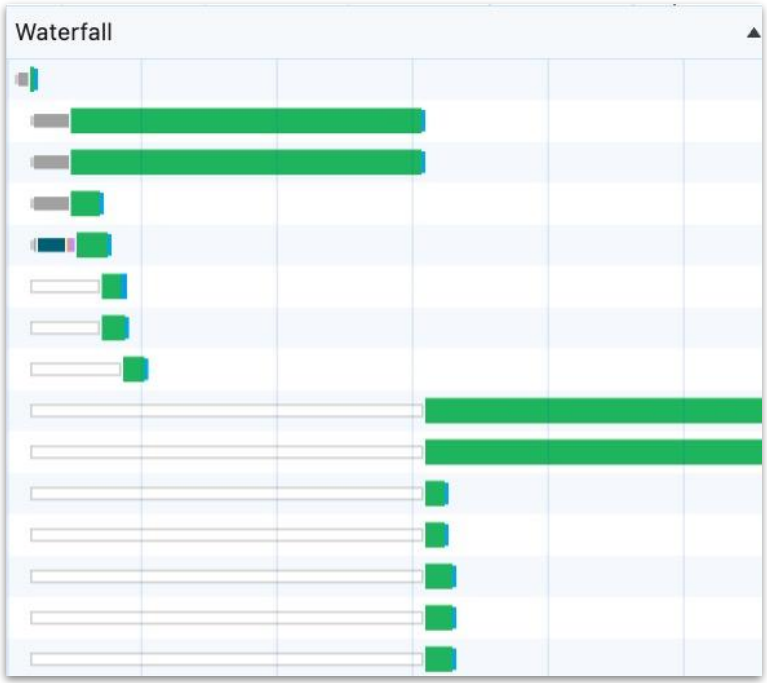
yep, always hilarious. FF has always loaded overall pages faster. Every single year for the past 15 years it has never once been slower. Chrome optimized their browser to load the "looks like the page is done" junk first which slows down overall load but appears to be done faster, this is what tricks people. …

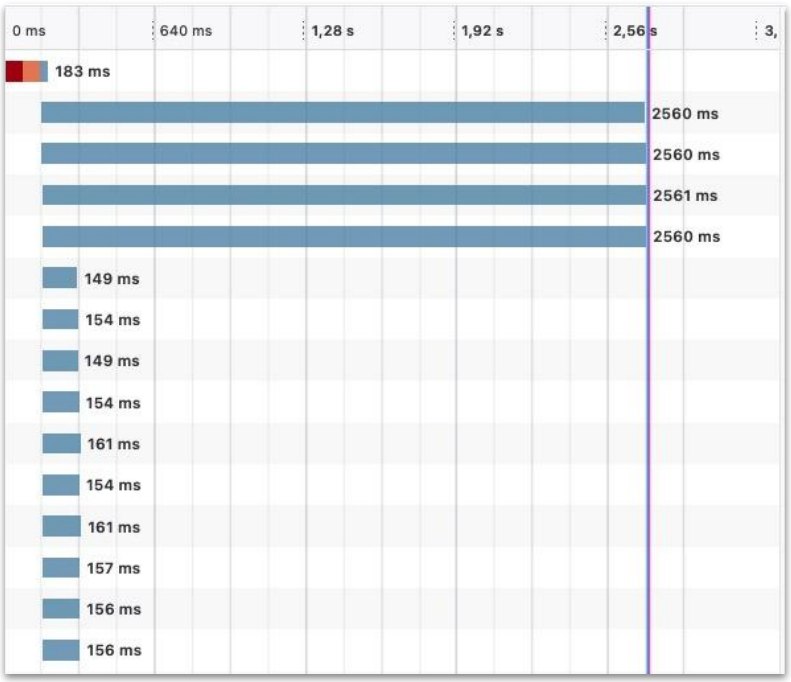
https://www.youtube.com/watch?v=p0lFyPuH8Zs

Akamai

Exact same HTML,

radically different behaviour





Other topics I researched

- Preload and its interactions with Tight Mode / fetchpriority
- 103 Early Hints
  - Tight mode impact?
  - Preloading responsive images?
- Why do font preloads need a crossorigin attribute?!?
  - Except on Safari?!!?!!!
  - Credentialed requests and CORS
  - Connection coalescing
- Tight mode across connections: chrome vs safari
- Tight mode impact for Speculation Rules API (prefetch/render)
- *How much I hate browser devtools sometimes :)*

Ask me about these sometime ;)

Akamai

As long as I employ
all the recent
Web standards

and use a well-known
hosting provider/CDN

my site will be fast on
all the browsers!

“Chrome” Web Vitals

(Loading)

LCP

Largest Contentful Paint

(Interactivity)

INP

Interaction to Next Paint

(Visual Stability)

CLS

Cumulative Layout Shift

Akamai

web.dev About Blog More Search / English Sign in

Guidance to build modern web experiences that work on any browser.

Thanks for tuning in to Google I/O! Watch content on-demand.

Building a better web, together

We want to help you build beautiful, accessible, fast, and secure websites that work cross-browser, and for all of your users. This site is our home for content to help you on that journey, written by members of the Chrome team, and external experts.

About web.dev

HTTP/3 connection in action

THANK YOU