

# Varje millisekund (kan) räknas

Prestandabudget, critical rendering path och debugging



Hur lång tid vill en  
användare vänta?

“2 seconds is the threshold for e-commerce website acceptability. At Google, we aim for under a half second.”

– Maile Ohye, Google (*Google’s Site Performance for Webmasters*)

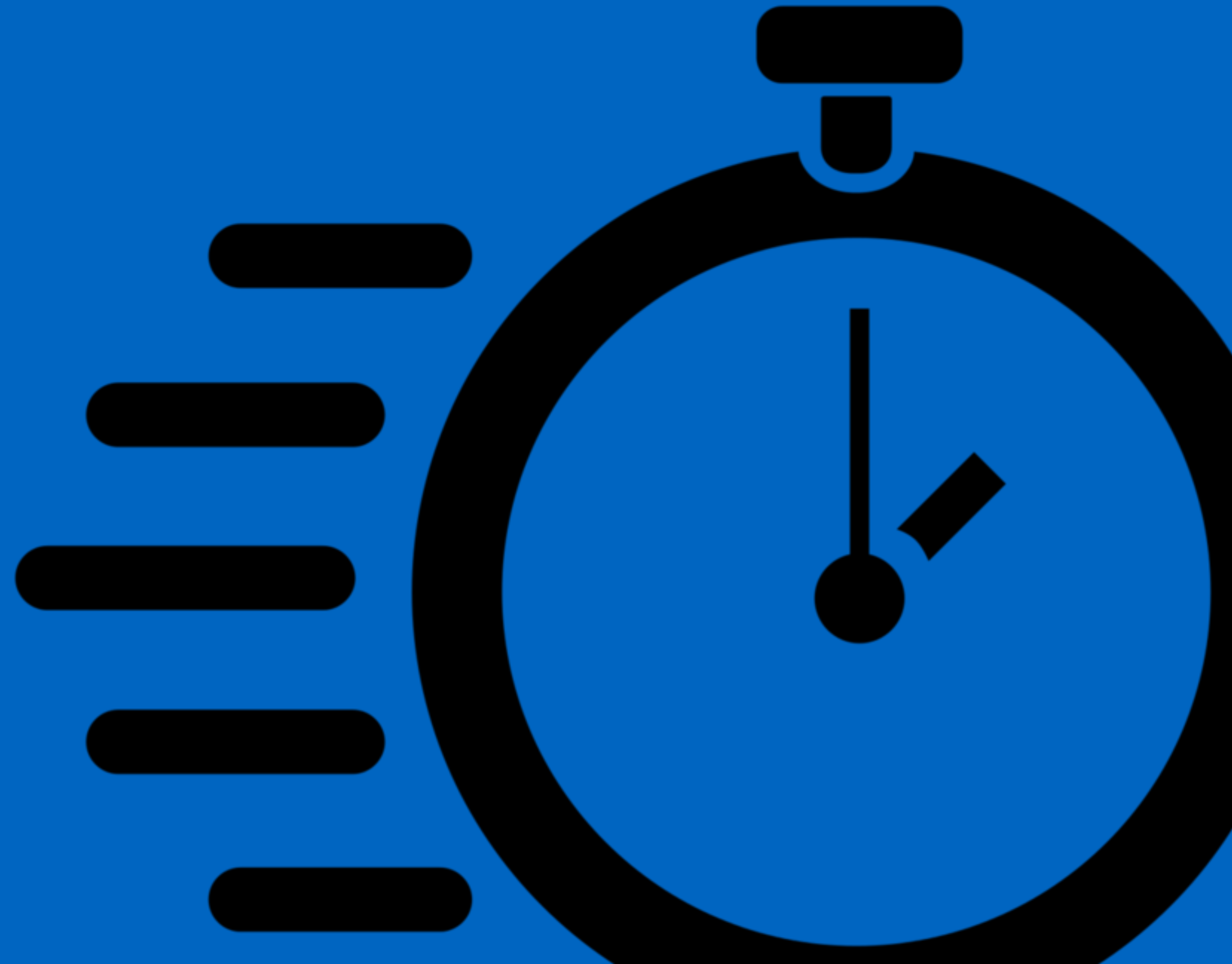
Vilket år?

2010.

# Kostnaden för väntetid

# Jakob Nielsens tumregler

- 0.1 sekund
- 1 sekund
- 10 sekunder (idag: 3-5 sek.)



Upplevd väntetid



# System state och feedback loop

Du behöver ha en faktisk skillnad  
på 20% för att något skall  
upplevas som snabbare

5000MB - 1.5MB per sida

Kostnaden för väntetid: Bandbredd

Demo:

<https://whatdoesmysitecost.com/>

+1% = 18M SEK

Kostnaden för väntetid: Kundförluster

1 sekund = -\$1.6B

Kostnaden för väntetid: Kundförluster

“Amazon's calculated that a page load slowdown of just one second could cost it \$1.6 billion in sales each year. Google has calculated that by slowing its search results by just four tenths of a second they could lose 8 million searches per day—meaning they'd serve up many millions fewer online adverts.”

– Kit Eaton, Fast Company

(<https://www.fastcompany.com/1825005/how-one-second-could-cost-amazon-16-billion-sales>)

# Prestanda i praktiken



# Servertechnologien

- Gzip
- Brotli
- Zopfli
- HTTPS
- HTTP/2

# Web fonts

- Font subsetting
- Ladda inte många samtidigt (2-3 max)

# Ny best-in-class font fallback

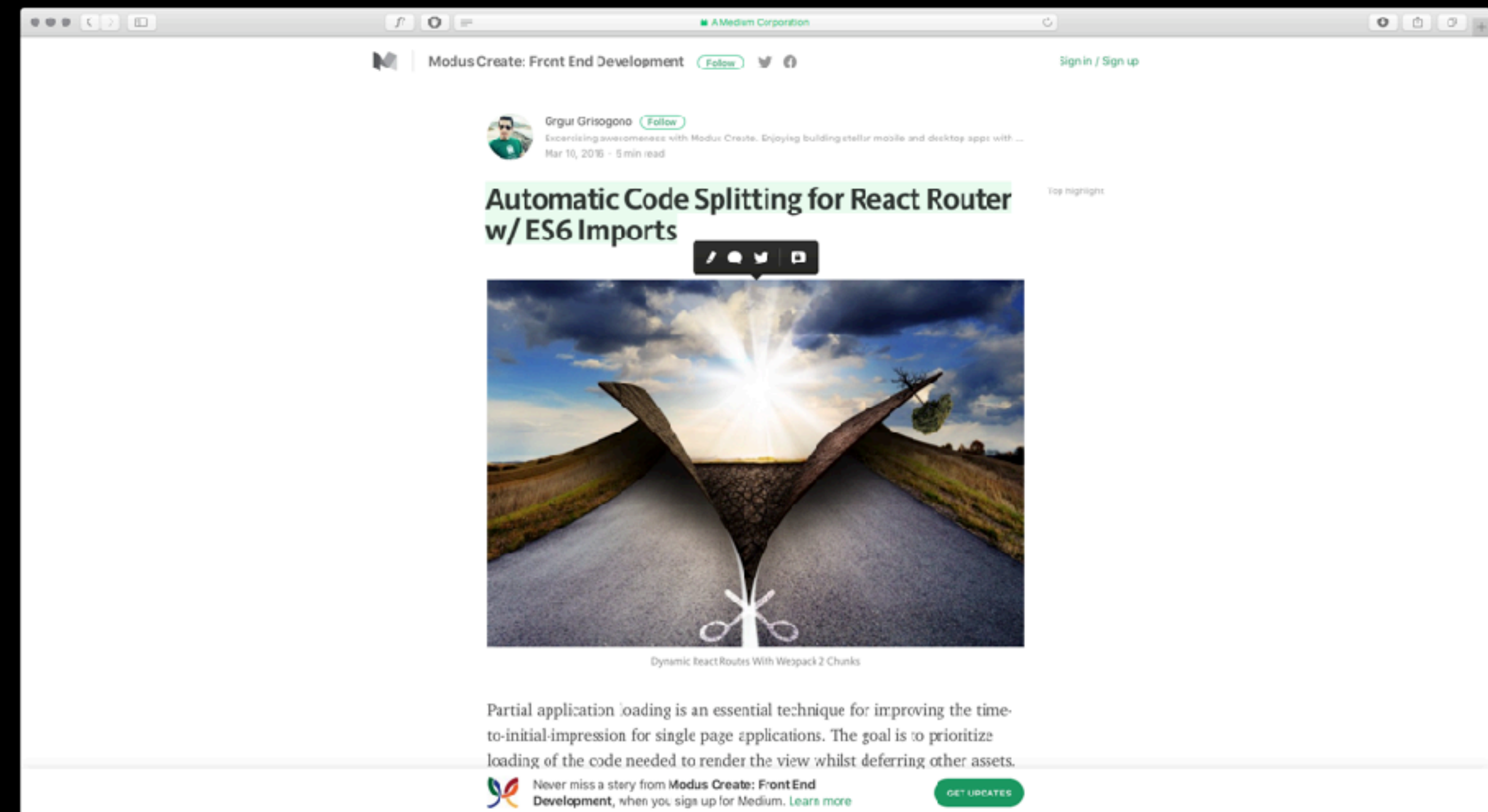
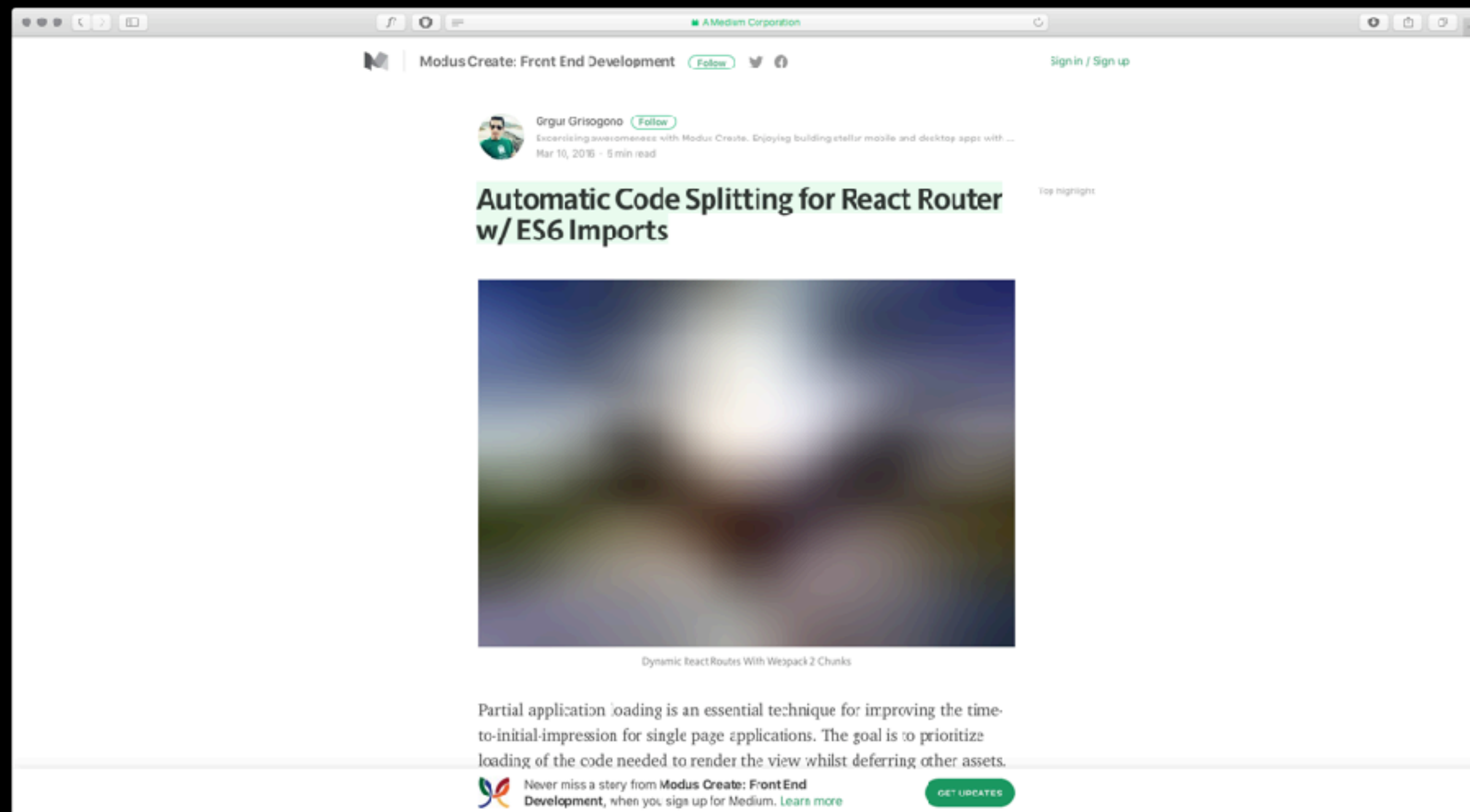
- Använder endast system-fonts
- Gäller alla system i vilka typsnitt kan specificeras

# SVG

- Helt kodbaserade
- Kan skrivas för hand
- Innehåller ofta stora mängder skräpdata

Demo: SVG + SVGOMG

# Low Quality Image Preview (LQIP)



# Lazy loading (modern monster)

```

```

# Request count

- Så få som möjligt: minification, concatenation...
- Helst över 14KB styck
- Använd GZIP eller annan kompression



RC med HTTP/2

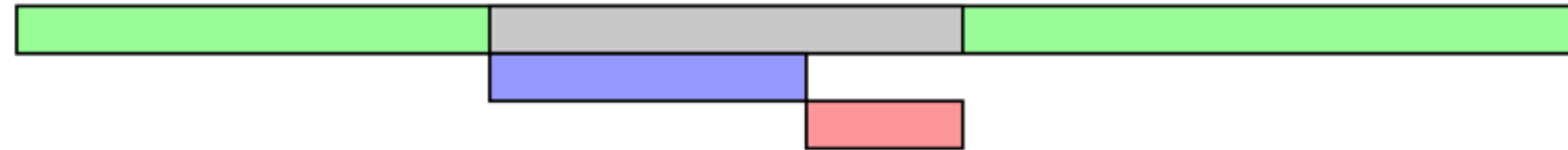
# Base64

- Behöver inte requestas
- Är ren binär data
- Tar CPU-kraft att dekodas
- Funkar dåligt att minifiera

Defer, async

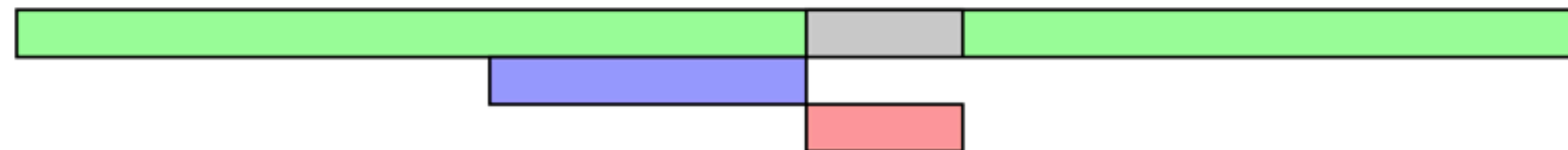
## <script>

Let's start by defining what <script> without any attributes does. The HTML file will be parsed until the script file is hit, at that point parsing will stop and a request will be made to fetch the file (if it's external). The script will then be executed before parsing is resumed.



## <script async>

async downloads the file during HTML parsing and will pause the HTML parser to execute it when it has finished downloading.



## <script defer>

defer downloads the file during HTML parsing and will only execute it after the parser has completed. defer scripts are also guaranteed to execute in the order that they appear in the document.



**Prefetch, preload, subresource**

# PWA och AMP

- Progressive Web Apps
- Accelerated Mobile Pages

# AMP

Instant delivery

Optimized discovery

No user scripts

Static content

# PWA

Advanced platform features

Highly dynamic

Slower first delivery

Not easily embedded

Parallel trådbehandling  
+ (service/web) workers



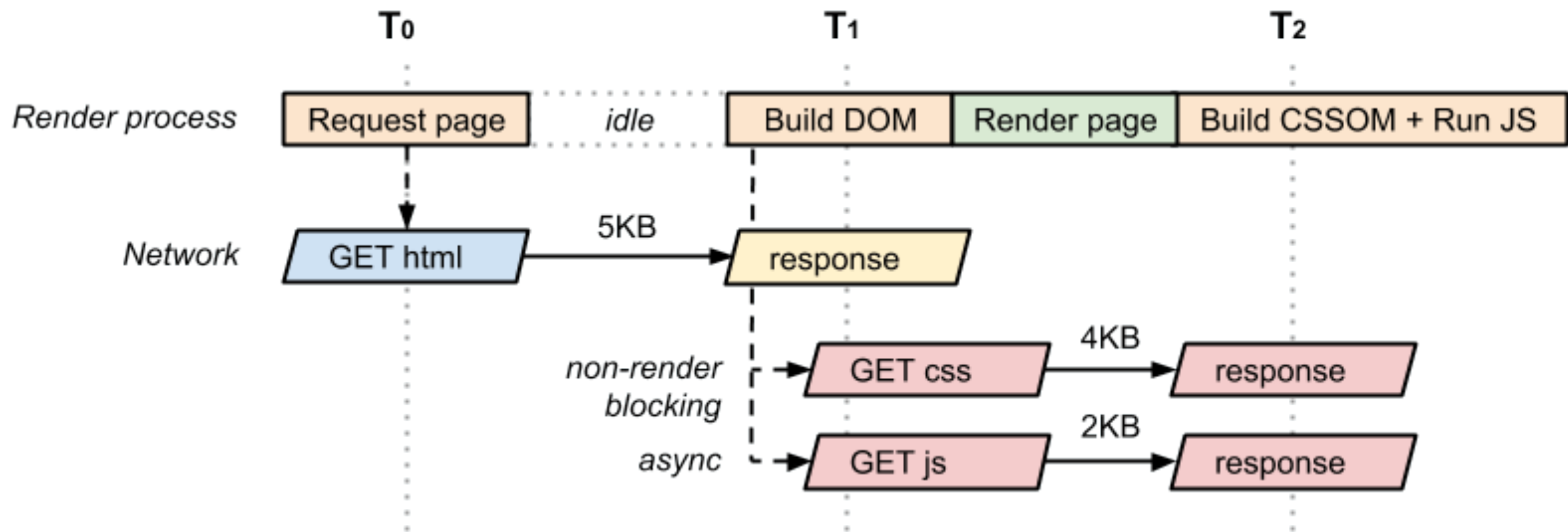
# Prestandamätning

# Mätvärden

- Filstorlekar
- HTTP request count
- Tid tills interaktiv
- Tid tills helt laddad
- Speed index (målsättning)
- Resursmängd/antal

# Critical rendering path

- CSSOM
- DOM
- Network Roundtrip
- Resurser
- (Non-)Render blocking



# Critical rendering path

- Radera
- Minifiera
- Prioritera
- Asynkronisera

Prestandabudget

Demo:

[www.performancebudget.io](http://www.performancebudget.io)

# Demo: Chromes nätverkspanel



# PageSpeed Insights

WebPageTest

# GT Metrix

Louis

Lighthouse

# Best practices

# Trafik/rendering

- En TCP-hämtning är max 14kb så sidan måste initialt laddas på den mängden data för att vara omedelbart synlig
- Max 8 simultana hämtningar
- 16ms frame render time för att uppnå 60fps
- Använd aldrig JS för att kolla windowscroll, ex. “scroll jacking”
- Använd CSS transform3d och andra GPU-accelererade animationer

# Laddning

- Använd Critical CSS
- Behovsstyrd och asynkron laddning (async/defer, end of body...)
- Bundla, mangla, minifiera, hash, chunka, splitta (om möjligt för routade sidor), använd tree-shaking
- Ladda aldrig saker som inte skall användas



# Utseende

- Använd Critical CSS
- Använd inte ikonfonter
- Använd font subsetting om möjligt
- Ladda inte mer än 2-3 egna web fonts
- Använd bara WOFF1+2
- Använd bara SVG, helst inline, för grafik (alternativt Base-64)
- Använd helst Webp för bilder med JPG som fallback

# Framtiden för prestanda

Teknologier i stil med PWA/  
AMP kommer utforskas mer.

**Prestanda kommer fortsätta vara  
en viktig del av kundupplevelsen.**

Nätverkshastigheten kommer  
stiga de kommande fem åren.

Grundläggande teknologier är/  
kommer inom kort vara stabila  
över både desktop och mobil.

# SCORES

Most used

All browsers

## current

Score	Browser
492	Chrome 52
461	Firefox 48
460	Edge 14
383	Safari 10.0

## upcoming

Score	Browser
473	Edge 15

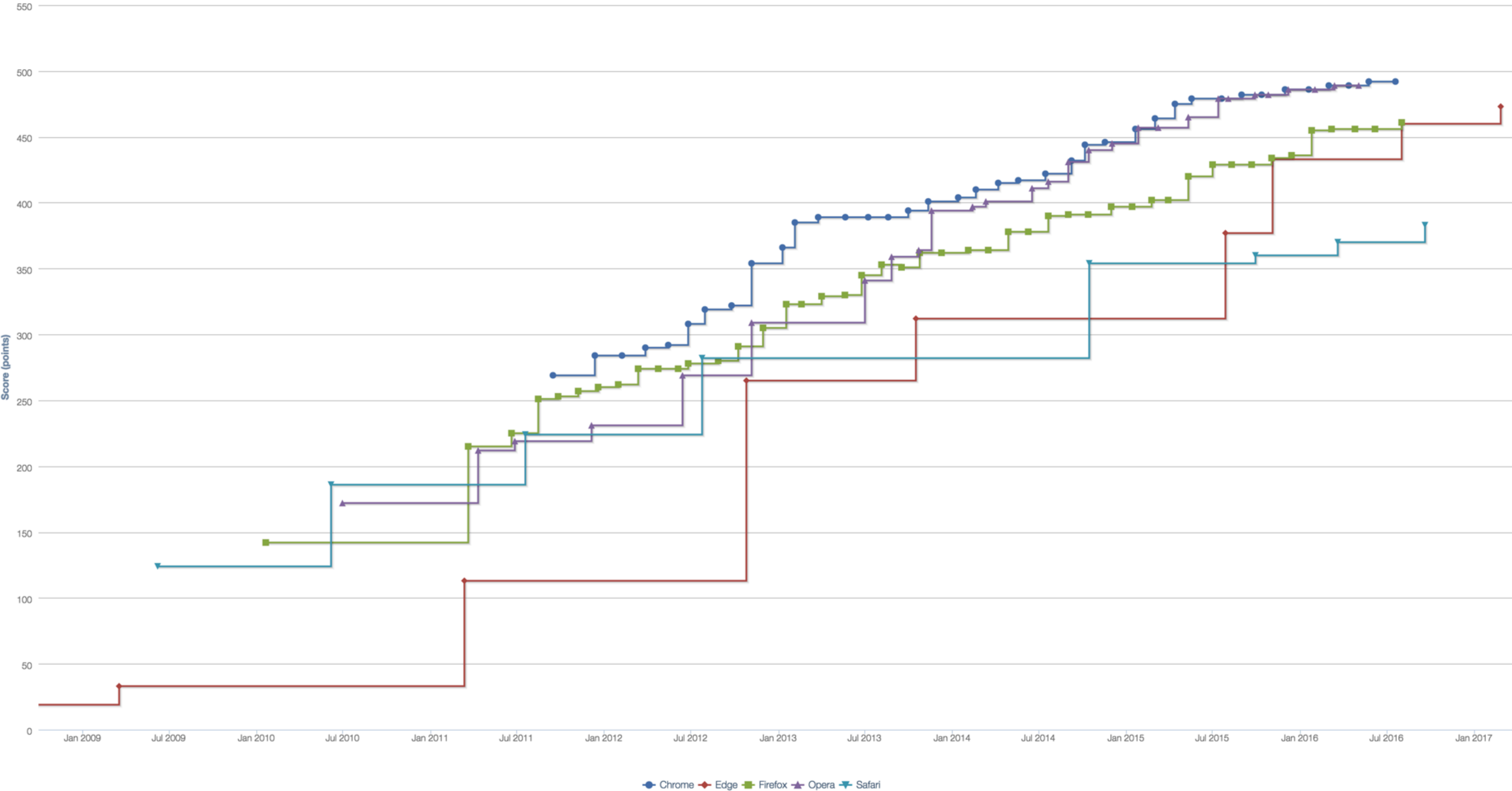
## experimental

Score	Browser
490	Edge
484	Firefox

## older

Score	Browser
492	Chrome 51
489	Chrome 50
489	Opera 37
482	Chrome 45
479	Opera 30
456	Chrome 40
456	Firefox 47
456	Firefox 46
456	Firefox 45
433	Edge 13
429	Firefox 40
397	Firefox 35
394	Chrome 30
378	Firefox 30
377	Edge 12
370	Safari 9.1
360	Safari 9.0
354	Safari 8.0
312	Internet Explorer 11
309	Opera 12.10
265	Internet Explorer 10
113	Internet Explorer 9

HTML5test.com score over the years





Grundläggande teknologier är/  
kommer inom kort vara stabila  
över både desktop och mobil.

“Unlike the other sections of the survey which have at least one or two options people haven’t heard of, practically every developer surveyed was at least aware of the existence of the four survey options, which all clocked in at 91% awareness or higher.”

- <http://stateofjs.com/2016/buildtools/>

# Workshop

Diskutera vad som är den “givna  
nivån” av prestanda idag i ett  
utvecklingsuppdrag.

Diskutera vad man kan göra för att kunder aktivt skall köpa prestandaförbättringar.

**Diskutera hur man bör arbeta med  
prestanda i teamet och inom/inför  
uppdrag.**

Gör en audit i Chrome.

Använd Google PSI.



Använd Louis.

Använd Font Squirrel font  
subsetting.

Exportera kritisk CSS.

Refaktorerera en av de nya Suez-  
sidorna till maximal prestanda.