

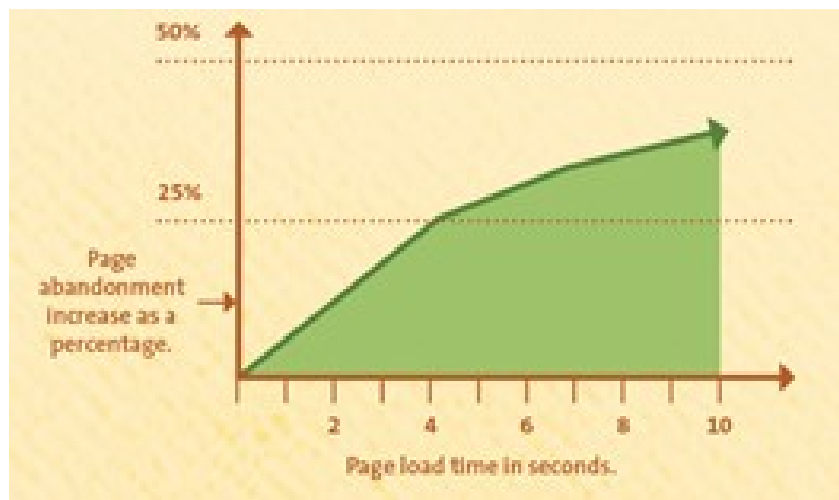
školení frontend

**Rychlost načítání**

# rychlost načítání

## Proč to řešit

- 47% uživatelů očekává, že se stránka načte nejpozději do 2s
- 40% uživatelů odejde, pokud načtení trvá více než 3s
- zpomalení o 1s při načtení snižuje míru konverze o 7% a snižuje návštěvnost o 11%
- rychlost načtení stránky je součástí hodnocení při vyhledávání na Google



Zdroj: <https://headspin.io/resources/marketing/reports/5136-RR-performance-web-application.pdf>  
<https://neilpatel.com/blog/loading-time/>

# rychlost načítání

## Proč to řešit

**0 - 100ms:** *Okamžité načtení*

**100 - 300ms:** *Mírná prodleva*

**300 - 1000ms:** *Uživatel ztrácí soustředění, znatelná prodleva*

**1s+:** *Mentální přepnutí*

**10s+:** *Uživatel odchází*

Zdroj: [https://medium.com/@luisvieira\\_gmr/understanding-the-critical-rendering-path-rendering-pages-in-1-second-735c6e45b47a](https://medium.com/@luisvieira_gmr/understanding-the-critical-rendering-path-rendering-pages-in-1-second-735c6e45b47a)

# rychlost načítání

## Jak se načítá stránka

### 1. Document Object Model (DOM)

- načtení a vybudování struktury stránky
- shromáždění informací o CSS, JS
- s vykreslením se čeká na CSSOM

### 2. CSS object model (CSSOM)

- buduje strukturu z hlediska CSS
- blokuje vykreslení

### 3. Render Tree

- propojení DOM a CSSOM dohromady

### 4. Layout

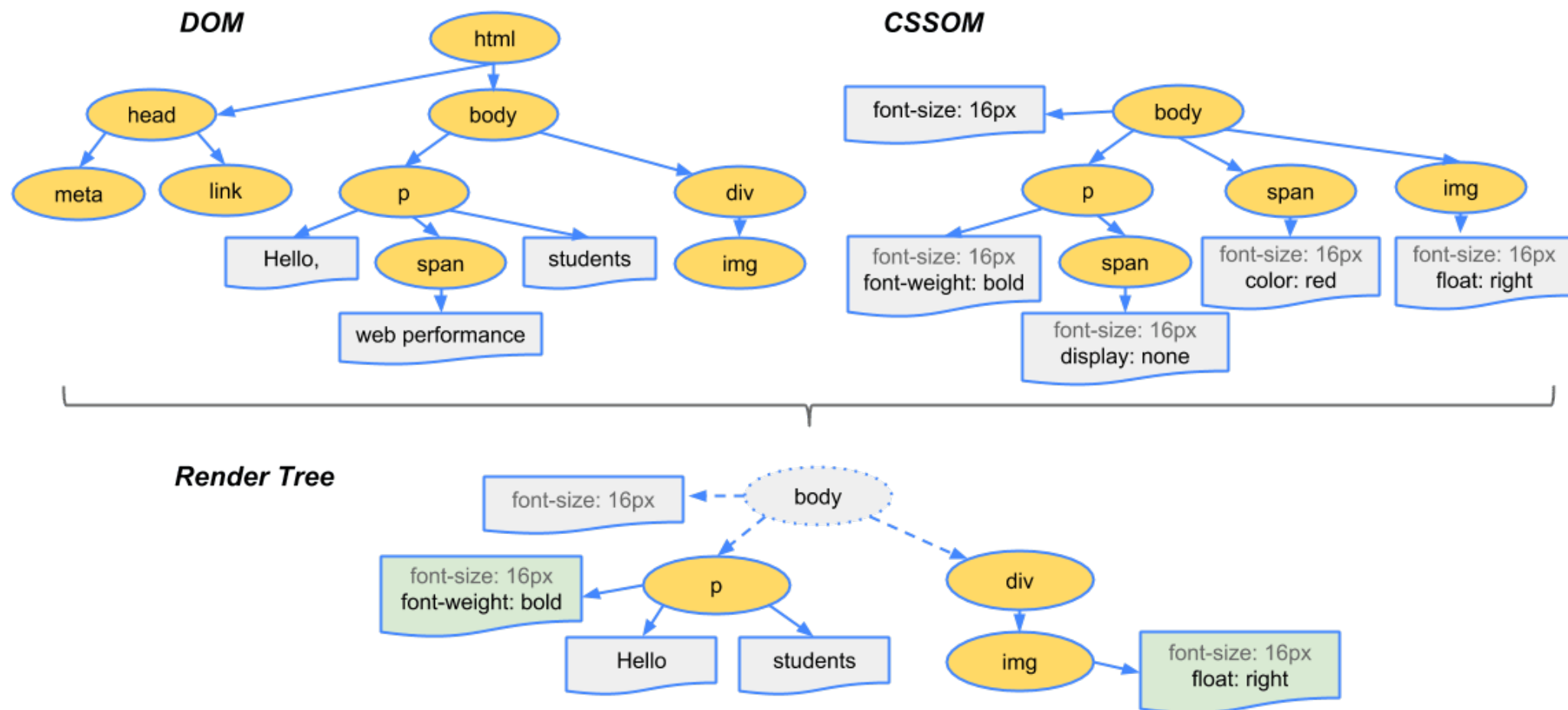
- výpočet rozměrů všech viditelných prvků stránky
- probíhá při každé změně prvků stránky nebo při změně viewportu

### 5. Paint

- vykreslení jednotlivých prvků postupně na obrazovku

# rychlost načítání

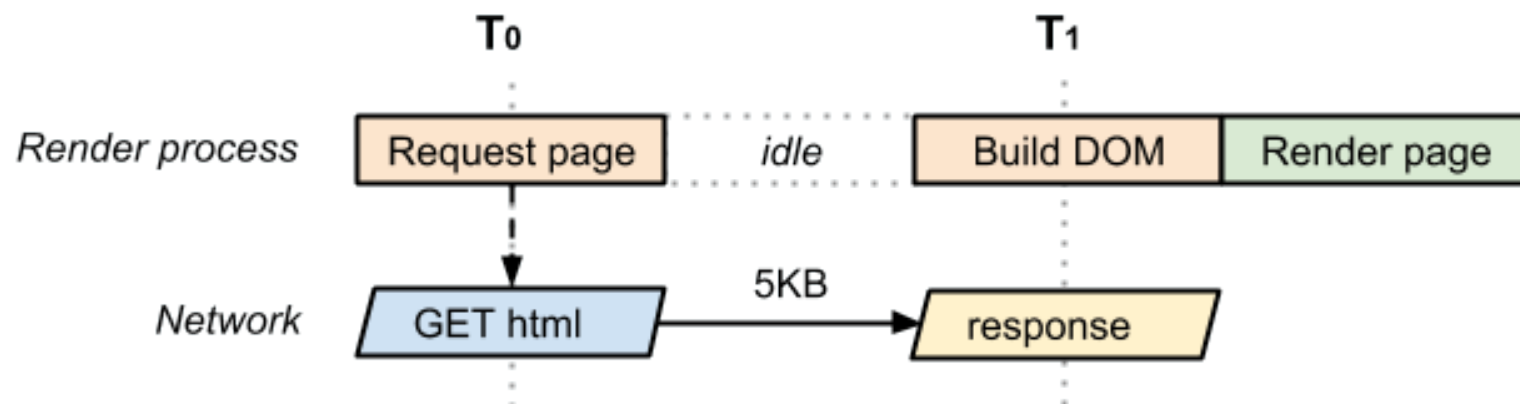
## Jak se načítá stránka



# rychlost načítání

## Jak se načítá stránka

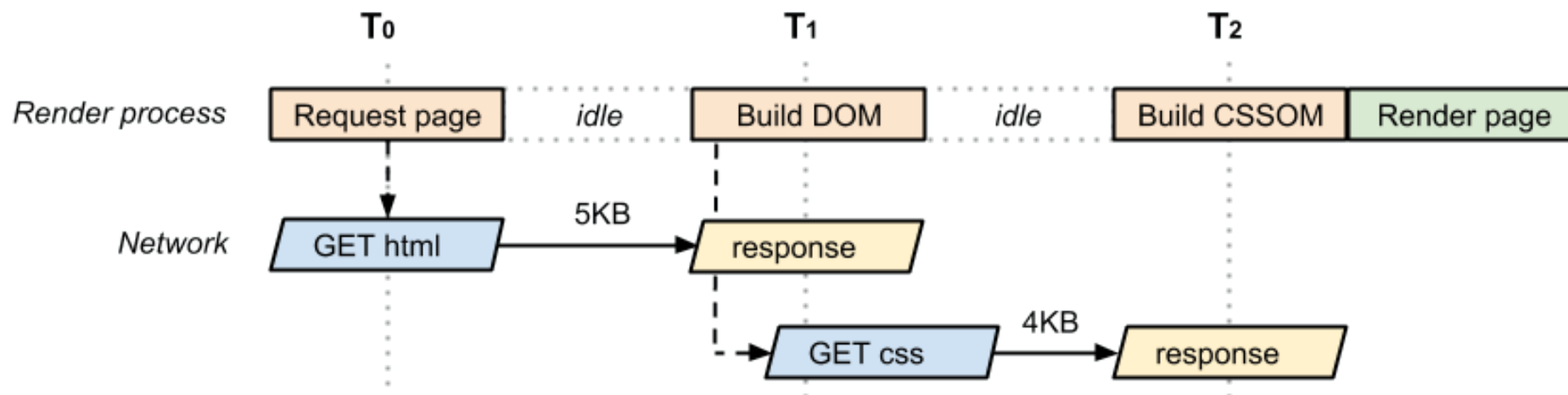
```
<html>
<head>
  <meta name="viewport" content="width=device-width,initial-scale=1">
  <title>Critical Path: No Style</title>
</head>
<body>
  <p>Hello <span>web performance</span> students!</p>
  <div></div>
</body>
</html>
```



# rychlost načítání

## Jak se načítá stránka

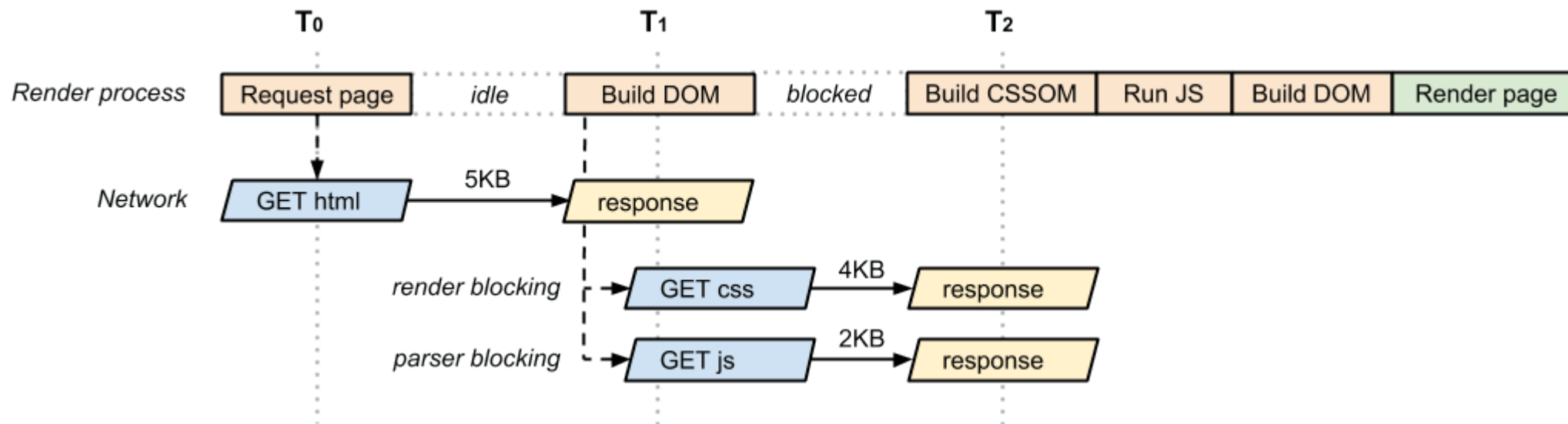
```
<html>
<head>
  <meta name="viewport" content="width=device-width,initial-scale=1">
  <link href="style.css" rel="stylesheet">
</head>
<body>
  <p>Hello <span>web performance</span> students!</p>
  <div></div>
</body>
</html>
```



# rychlost načítání

## Jak se načítá stránka

```
<html>
<head>
  <meta name="viewport" content="width=device-width,initial-scale=1">
  <link href="style.css" rel="stylesheet">
</head>
<body>
  <p>Hello <span>web performance</span> students!</p>
  <div></div>
  <script src="app.js"></script>
</body>
</html>
```

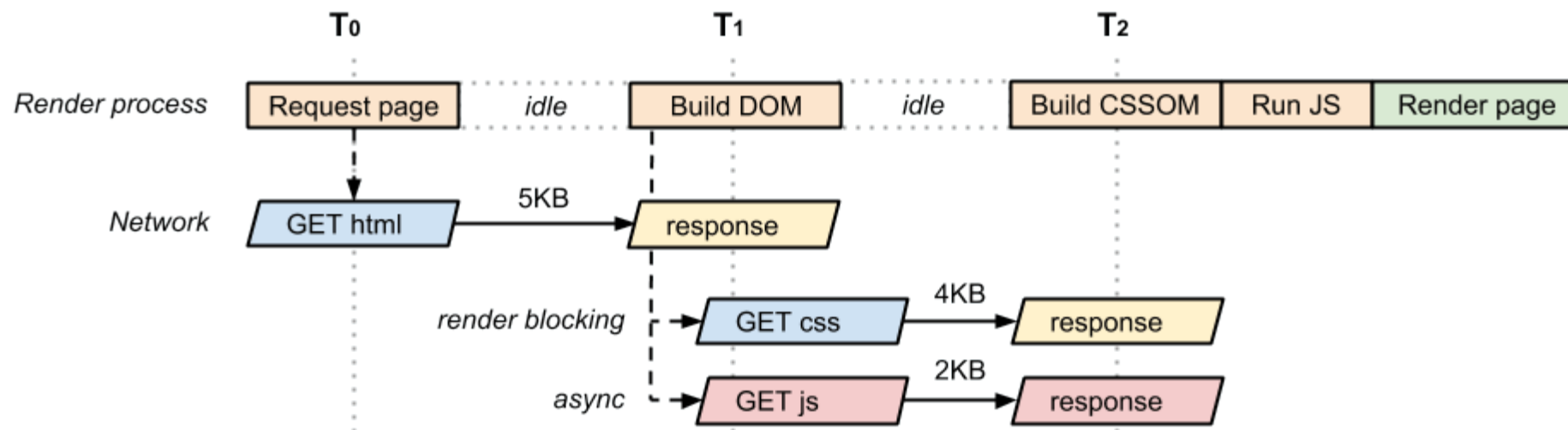




# rychlost načítání

## Jak se načítá stránka

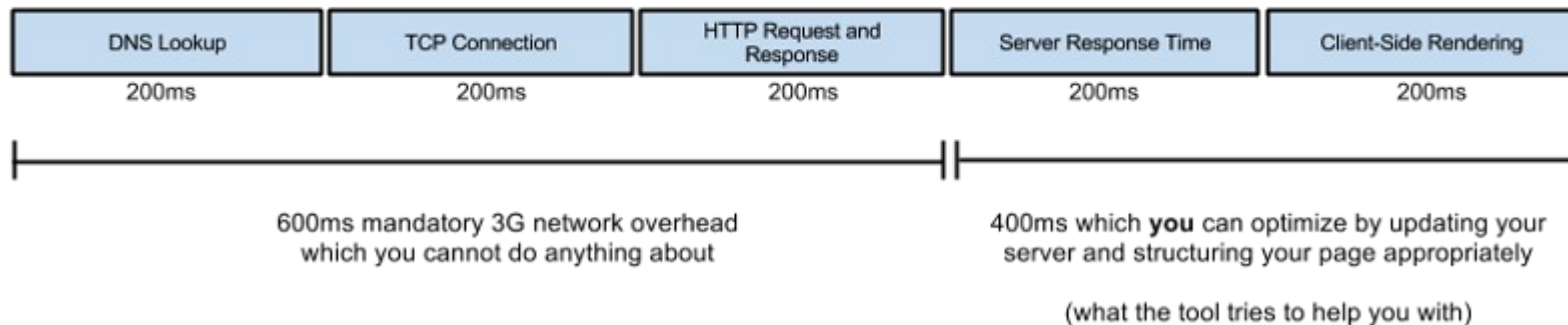
```
<html>
<head>
  <meta name="viewport" content="width=device-width,initial-scale=1">
  <link href="style.css" rel="stylesheet">
</head>
<body>
  <p>Hello <span>web performance</span> students!</p>
  <div></div>
  <script src="app.js" async></script>
</body>
</html>
```



# rychlost načítání

## Jak se načítá stránka – mobil

Rendering a mobile page in 1 second



# rychlost načítání

## Metriky

- **Time To First Byte (TTFB)**
  - Rychlost serveru a sítě – za jak dlouho dorazí první bajt
  - Ideálně pod 0.5s
- **DOM Content Loaded (DCL)**
  - Rozparsování HTML prohlížečem (nečeká se na CSS, JS)
- **First Paint (FP)**
  - První moment, kdy se něco změní na obrazovce
- **First Contentful Paint (FCP)**
  - První okamžik, kdy je vykreslena hlavní část stránky (např. titulek, ...)
- **Time to Interactive (TTI)**
  - První možnost na něco kliknout
- **Speed Index**
  - Celkové skóre stránky z hlediska rychlosti načtení
- **Load**
  - Stránka je kompletní

# rychlost načítání PageSpeed Insights

<https://developers.google.com/speed/pagespeed/insights/>

- jednoduchý webový nástroj
- kombinuje statistiky uživatelů Chrome, kteří web navštívili
- testujte více stránek webu, nejen homepage
- podrobné vysvětlení ke všem upozorněním
- cílem není dosáhnout 100
- pokud jste v červených číslech je něco špatně

PageSpeed Tools > Insights

DOMOVSKÁ STRÁNKA PŘÍRUČKY REFERENCE PODPORA

PageSpeed Insights

<http://www.aspectworks.cz/>

ANALYZOVAT



Mobilní zařízení



Počítač

Rychlost stránek

Unavailable

Optimalizace

Medium

67 / 100

PSI se v současné době zobrazuje jako přehled na jednu stránku. Přehled uživatelského dojmu Chromu nemá dostatek dat o reálné rychlosti této stránky, ale přehled PSI dokázal tuto stránku analyzovat a rozpoznat možnosti optimalizace, které by rychlost stránky zvýšily. Prozkoumejte níže uvedená doporučení. [Další informace](#)

Statistika stránky

Podle statistik vyžaduje střední stránka na internetu k načtení tento počet odezev, které blokují vykreslení: 4 a tento počet zdrojů: 75 (1). Tato stránka však pravděpodobně používá méně zdrojů. Podle odhadu PSI tato stránka k načtení vyžaduje tento počet odezev, které blokují vykreslení: 4 a tento počet zdrojů: 73 (1,6 MB). Stránky s menším počtem odezev a bajtů jsou rychlejší.

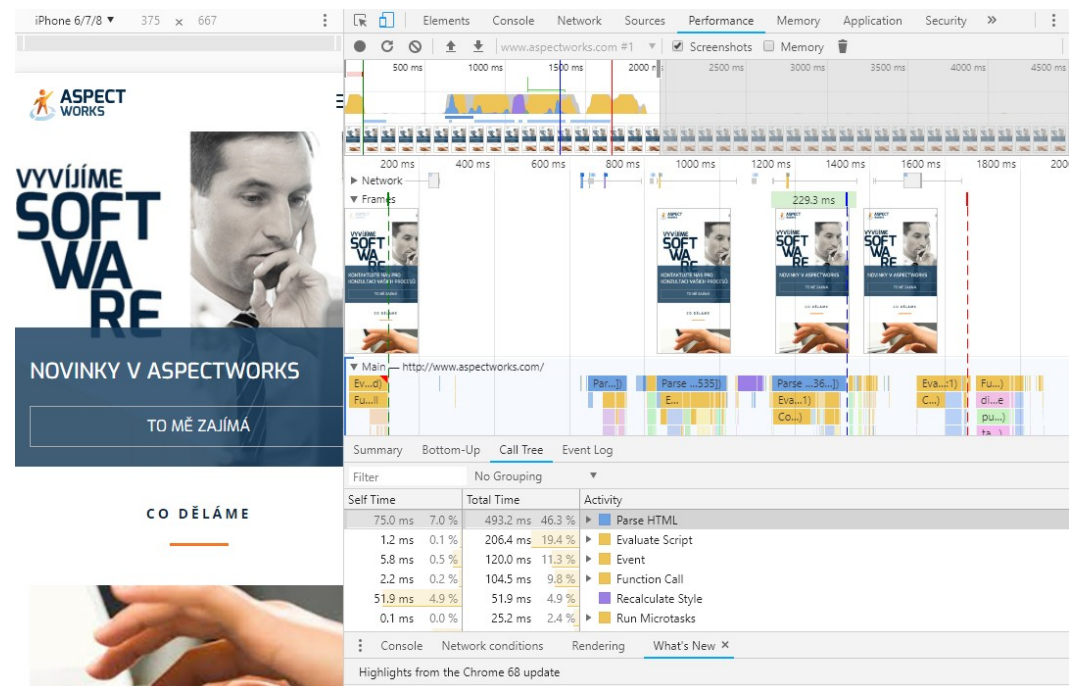
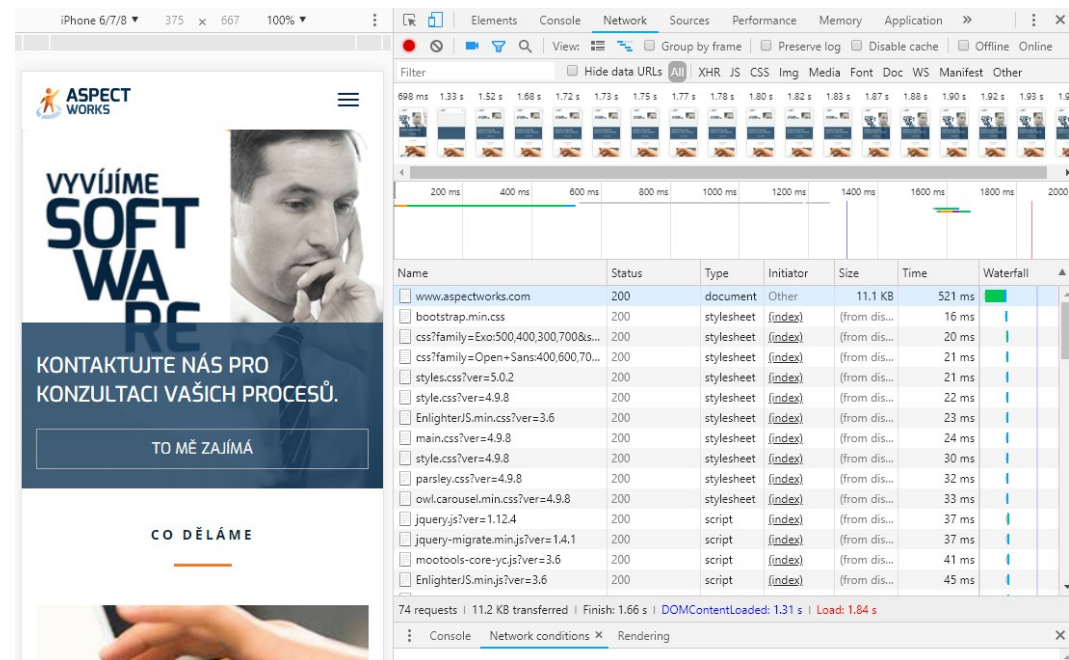


<https://www.vzhurudolu.cz/prirucka/pagespeed-insights>

# rychlost načítání

## Chrome DevTools

- přímo v prohlížeči
- pokročilé informace o rychlosti a celkovém procesu načtení stránky
- testy pomalého připojení
- filmový záznam načtení

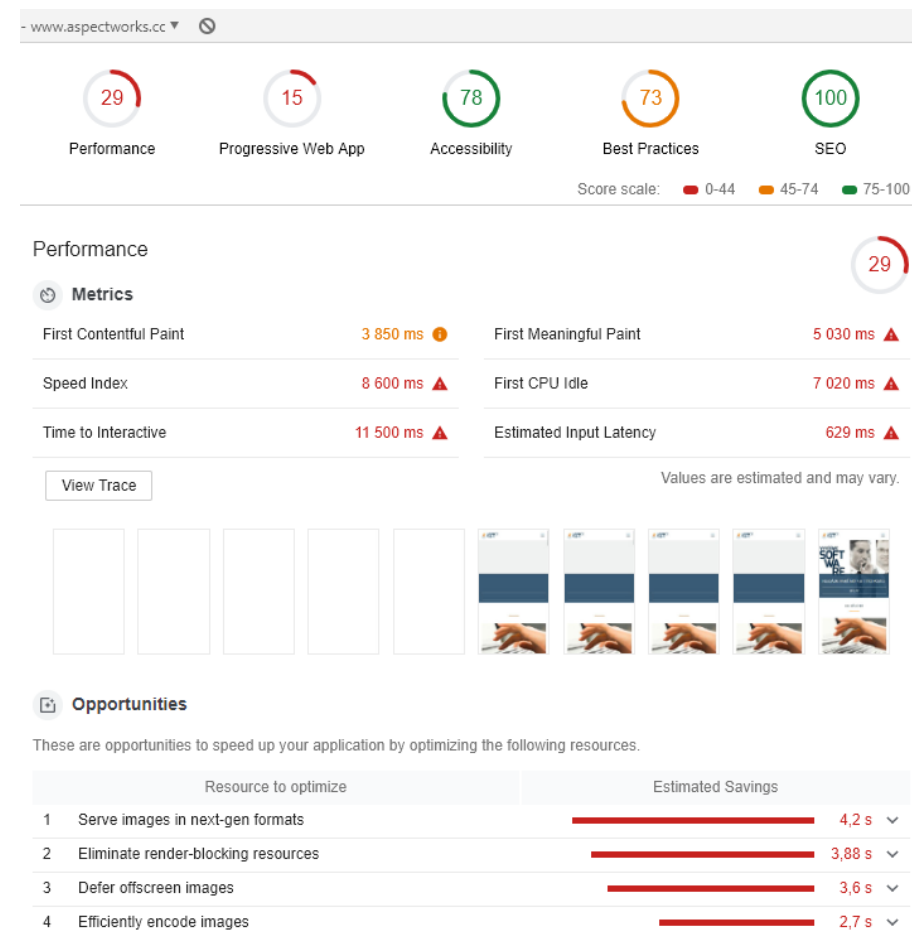


# rychlost načítání

# Google Lighthouse

<https://developers.google.com/web/tools/lighthouse>

- komplexní nástroj pro analýzu webu
- součást Chrome DevTools
- možnost spuštění také v příkazové řádce nebo jako node plugin (může být součástí buildu)
- SpeedIndex – celkové skóre stránky



<https://www.vzhurudolu.cz/prirucka/pagespeed-insights>

# rychlost načítání WebPagetest.org

<https://www.webpagetest.org/>

- webové prostředí
- pokročilé informace o rychlosti
- možnost testování z jiné lokality, testy pomalého připojení
- SpeedIndex – celkové skóre stránky

**WEBPAGETEST**

HOME **TEST RESULT** TEST HISTORY FORUMS DOCUMENTATION ABOUT

Web Page Performance Test for [www.aspectworks.com/](http://www.aspectworks.com/)

From: Prague, Czech Republic - Chrome - Cable  
24. 9. 2018 10:23:45

Need help improving?

**A** **A** **A** **F** **B** **X**

First Byte Time Keep-alive Enabled Compress Transfer Compress Images Cache static content Effective use of CDN

**Summary** Details Performance Review Content Breakdown Domains Processing Breakdown Screen Shot Image Analysis Request Map

Tester: WPTORGPRG-194.212.90.66  
First View only  
Test runs: 3  
[Re-run the test](#)

[Raw page data](#) - [Raw object data](#)  
[Export HTTP Archive \(.har\)](#)  
[View Test Log](#)

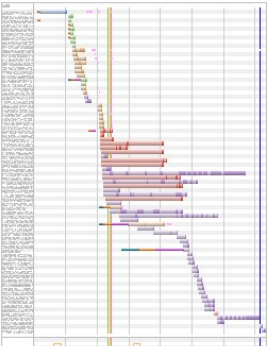

**Performance Results (Median Run)**

	Load Time	First Byte	Start Render	Speed Index	First Interactive (beta)	Document Complete			Fully Loaded			
						Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View (Run 3)	3.545s	0.492s	1.200s	2178	> 2.065s	3.545s	75	1,626 KB	3.623s	76	1,630 KB	\$\$\$\$\$

[Plot Full Results](#)

**Test Results**

Run 1:

	Waterfall	Screen Shot	Video
First View (3.577s) <a href="#">Timeline (view)</a> <a href="#">Processing Breakdown</a> <a href="#">Trace (view)</a>			<a href="#">Filmstrip View</a> <a href="#">Watch Video</a>

# rychlost načítání

## Jak zrychlit – JS

- v ideálním případě by načtení stránky nemělo být závislé na JS
- minifikace jako samozřejmost
- zvážit, zda se kvůli jednomu pluginu vyplatí linkovat jQuery nebo celý Bootstrap
- rozdělit JS na kritické pro fungování a zbytek
- využívat asynchronní nebo odložené načtení JS

**<script>** blokuje načtení HTML, čeká se na stažení i spuštění

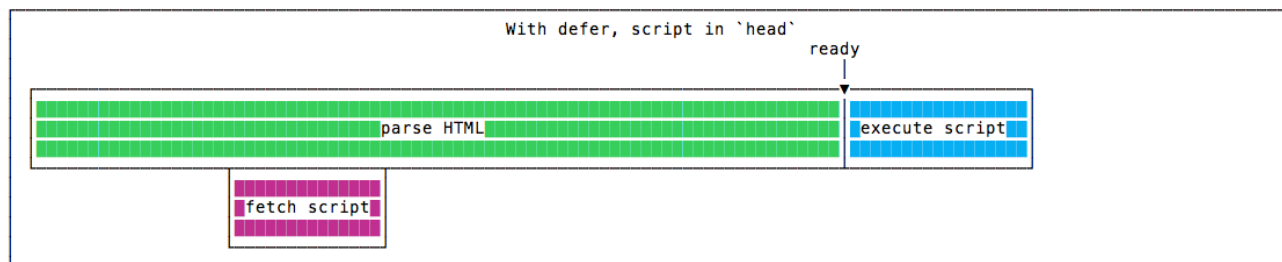
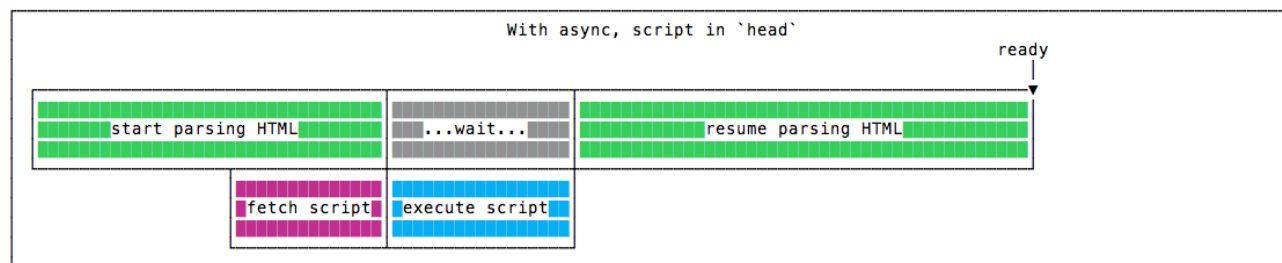
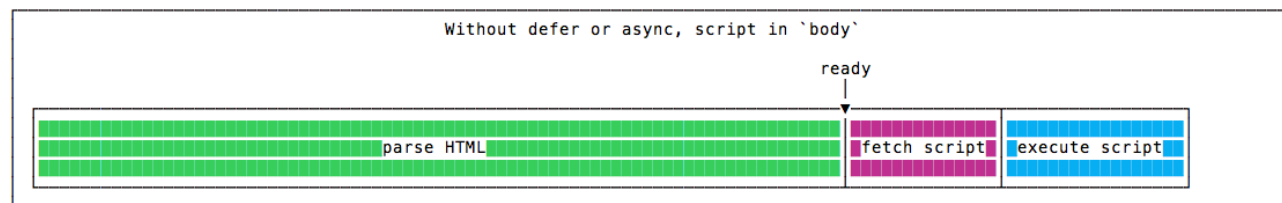
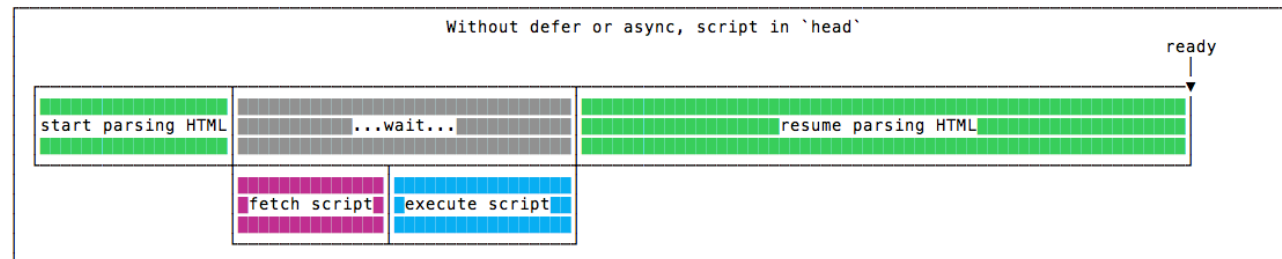
**<script async>** neblokuje načtení HTML, spustí se po stažení, nelze zajistit pořadí

**<script defer>** neblokuje načtení HTML, spustí se až po načtení HTML, jistota pořadí



# rychlost načítání

## Jak zrychlit - JS



# rychlost načítání

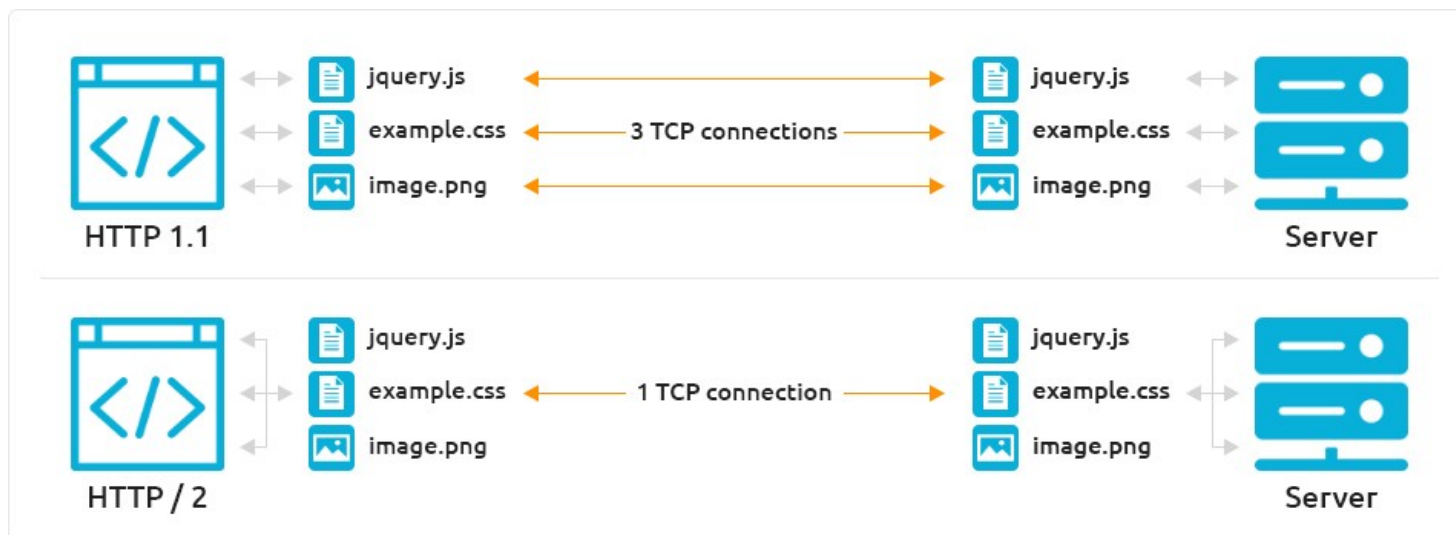
## Jak zrychlit CSS

- **Critical CSS**
  - načtení viditelné části přímo v hlavičce a zbytek na konci HTML
  - načtení celé stránky může trvat o něco více, ale subjektivní dojem načtení je lepší
  - <https://jonassebastianohlsson.com/criticalpathcssgenerator/>
  - <https://github.com/addyosmani/critical>
- **Asynchronní načtení CSS pomocí JS**
  - nutné použít v kombinaci s Critical CSS
  - uložit informaci do cookie, aby uživatel podruhé už použil celé CSS z cache

# rychlost načítání

## Jak zrychlit

- v jednu chvíli jde více požadavků i odpovědí
- rychleji se parsuje
- není problém s více malými soubory (není nutný image sprite)
- možnost rozdělit CSS do více malých souborů podle obsahu stránky
- nutné HTTPS



# rychlost načítání

## Lazyloading

- Načtení obsahu až ve chvíli, kdy je potřeba (resp. když by jej měl uživatel vidět)
- Nejčastěji se využívá pro obrázky a je navázáno na skrolování stránky
- Mnoho JS knihoven pro lazyloading:  
[Lazy.js](#), [Unveil](#), [lazySizes](#), [Recliner](#), [jQuery Lazy](#), ...

Jak to funguje?

```

```

Jakmile bude obrázek ve viewportu, tak vyměníme `data-src` za `src`.

# rychlost načítání

## Lazyloading

Knihovna	Velikost	jQuery/ Vanilla JS	Elementy
Unveil	0,7 kB	jQuery	obrázky
LazyLoad	3,5 kB	JS	obrázky, scrset
jQuery Lazy	4,9 kB	jQuery	obrázky, iframe
Recliner	1,2 kB	jQuery	obrázky, iframe, ajax
lazySizes	6,5 kB	JS	obrázky, iframe, ajax
lazyframe	5,1 kB	JS	iframe

# rychlost načítání Cvičení

Doplňte lazyloading pro galerii a fotky kadeřnic - použijte libovolný skript / plugin.

Vytvořte si **CriticalCSS**, vložte jej do záhlaví a CSS vložte na konec HTML stránky.

Můžete využít codepen:

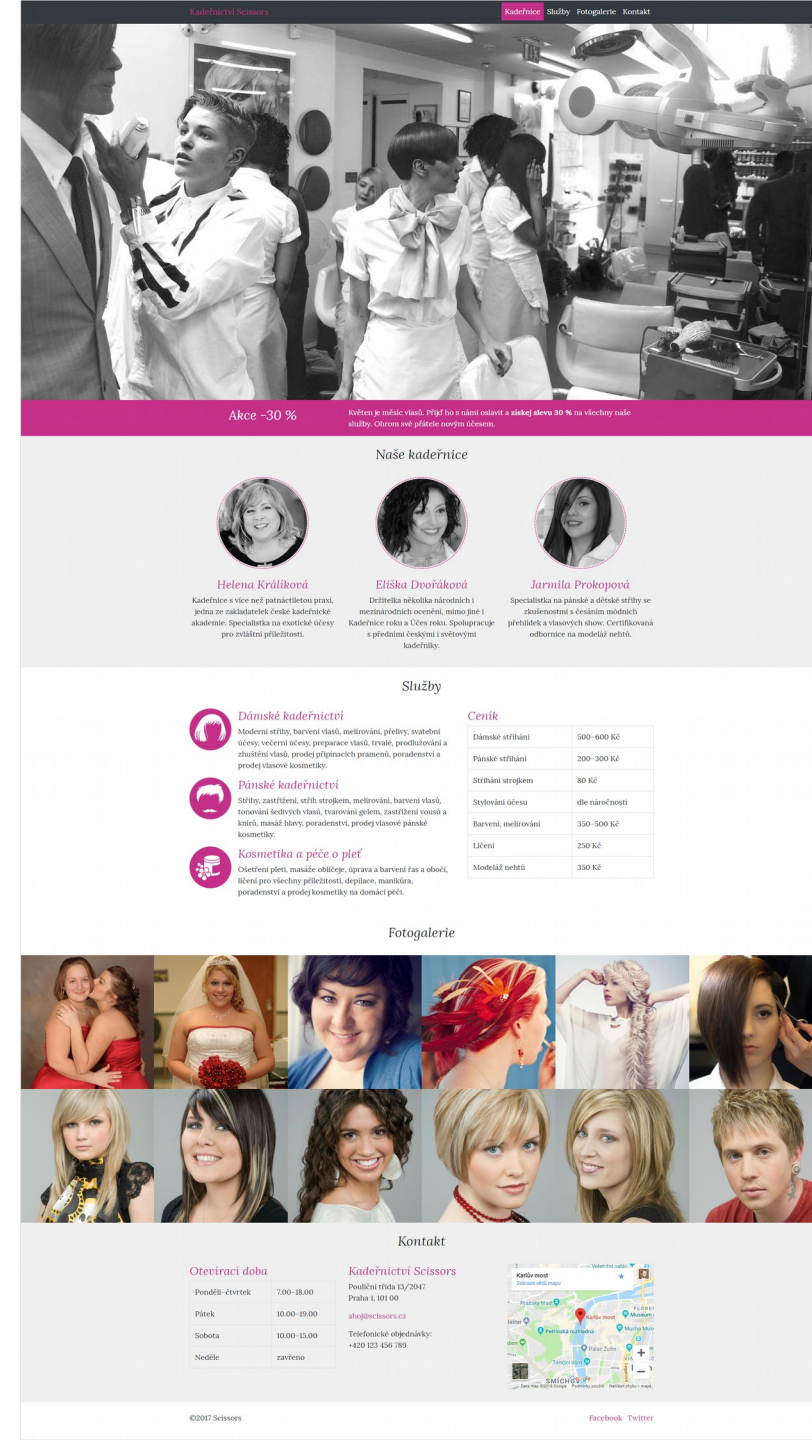
<https://codepen.io/svobodalukas/pen/BqwYLg>

Generování CriticalCSS:

<https://jonassebastianohlsson.com/criticalpathcssgenerator/>

pro generování CriticalCSS použijte URL:

<http://www.lukassvoboda.cz/coding/frontend-aspect-works/cviceni-bootstrap>



# rychlost načítání

## FOIT versus FOUT

- **Flash of Invisible Text** – dokud není font načten, není text zobrazen
- **Flash of Unstyled Text** – text je zobrazen systémovým fontem, poté webfontem
- <https://www.zachleat.com/foitfout/#4000,4000,4000,4000>



Více informací:

<https://www.filamentgroup.com/lab/font-events.html>

<https://www.zachleat.com/web/comprehensive-webfonts/>

# rychlost načítání

## FOIT versus FOUT

```
<style>
body {
  font-family: Helvetica, Arial, sans-serif;
}

.fonts-loaded body {
  font-family: "Roboto", Helvetica, Arial, sans-serif;
}
</style>

<script>
var roboto = new FontFaceObserver("Roboto", {
  weight: 400
});

roboto.check().then(function() {
  document.documentElement.className += "fonts-loaded";
});
</script>
```

<https://jonsuh.com/blog/font-loading-with-font-events/>