



# Need for speed

## Тестирование и оптимизация скорости загрузки мобильных сайтов



Иван Бригидा,  
Mobile UX Manager,  
13 июня, Москва

HTTP<sub>s</sub> splitting  
CSS bundling  
inline browser back-end  
gzip push caching  
critical code server  
fonts page rendering  
redirects lazy loading  
images webpack  
minification

# Почему скорость важна?

**Почему скорость важна?**

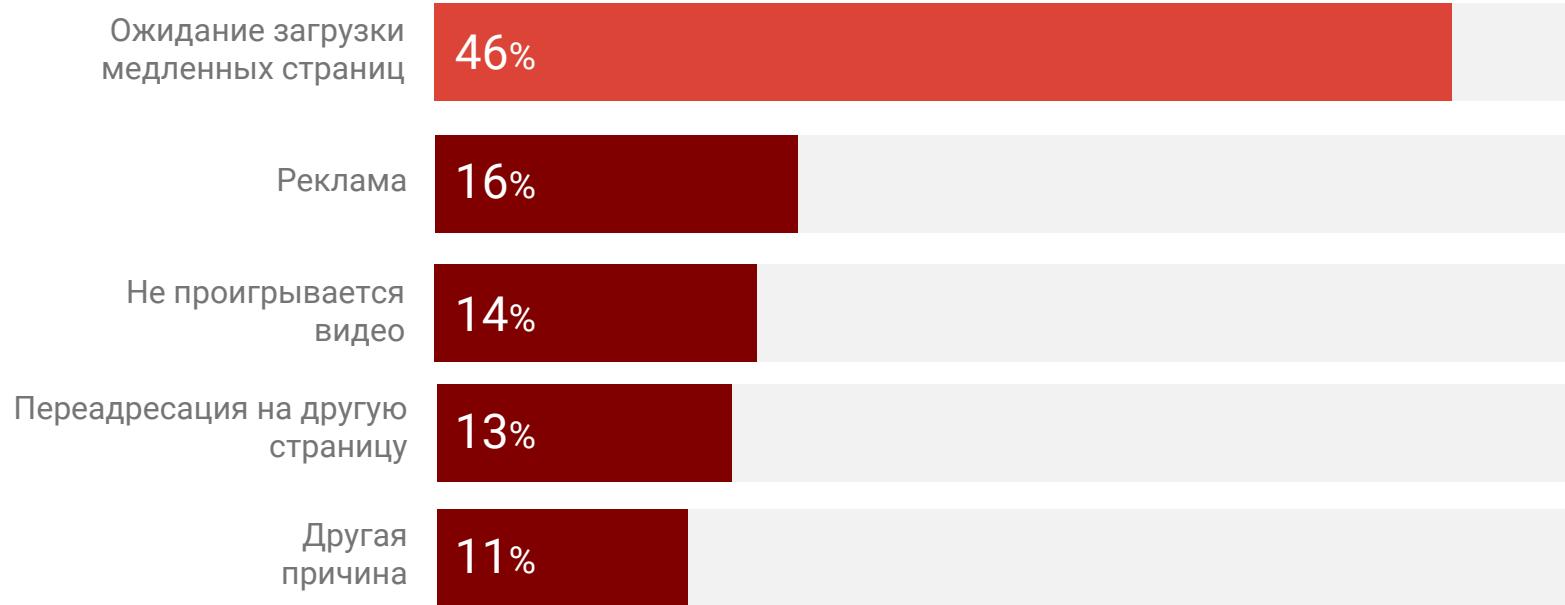
Как измерять скорость?

Оптимизация изображений

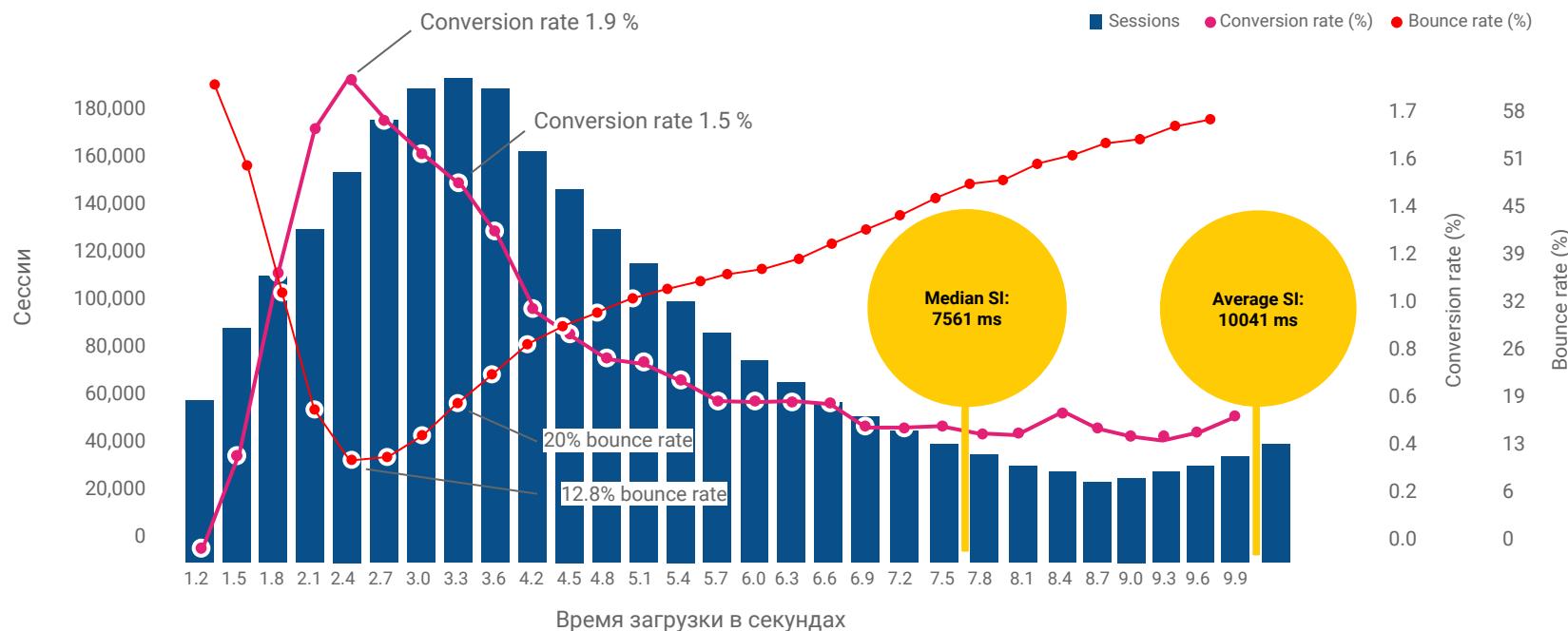
Critical Rendering Path

Выводы

# Опрос: “Что вы не любите больше всего при просмотре мобильных страниц?”



# Мобильные страницы, которые загружаются на 1 секунду быстрее имеют до 27% улучшение CvR



# CEO: скорость мобильной страницы является ранжирующим сигналом в Google (в разработке)

google page load speed

Web Shopping News Images Videos More ▾ Search tools

About 63,700,000 results (0.51 seconds)

[PageSpeed Tools | Google Developers](#)  
https://developers.google.com/speed/pagespeed/?hl=en ▾ Google ▾  
Except as otherwise noted, the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code samples are licensed under the ...

[PageSpeed Insights](#)  
Have comments or questions about PageSpeed Insights ...

[Get Started](#)  
Get Started with the PageSpeed Insights API. Contents  
[More results from google.com »](#)

[About PageSpeed Insights](#)  
The PageSpeed Insights tools help you diagnose speed and ...

[Measure Speed up](#)  
The extent of speed up PageSpeed Service provides ...

[Make the Web Faster | Google Developers](#)  
https://developers.google.com/speed/?hl=en ▾ Google ▾  
May 27, 2015 - Leverage Google's Public DNS to improve security and speed of your ...  
Except as otherwise noted, the content of this page is licensed under ...

# Как измерять скорость?

Почему скорость важна?

**Как измерять скорость?**

Оптимизация изображений

Critical Rendering Path

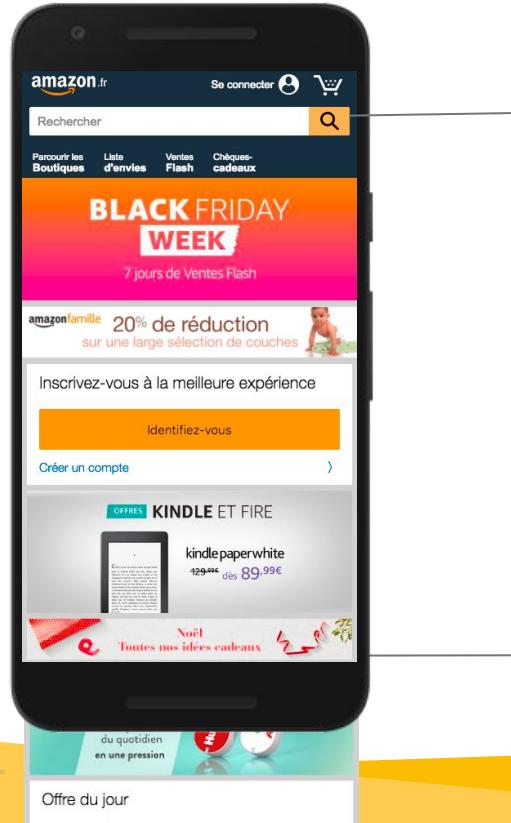
Выводы

# Какая метрика скорости самая лучшая?



-  Load Time: **4.033s**
-  First Byte: **0.805s**
-  Start Render: **2.282s**
-  Fully Loaded: **10.041s**
-  Speed Index: **2309**

# Speed Index - наиболее значимая метрика



Speed Index:  
2309

**Speed Index** - это  
среднее время  
отрисовки **видимой**  
части страницы

# Пример: восприятие скорости пользователем похожих страниц может сильно отличаться

0.9s



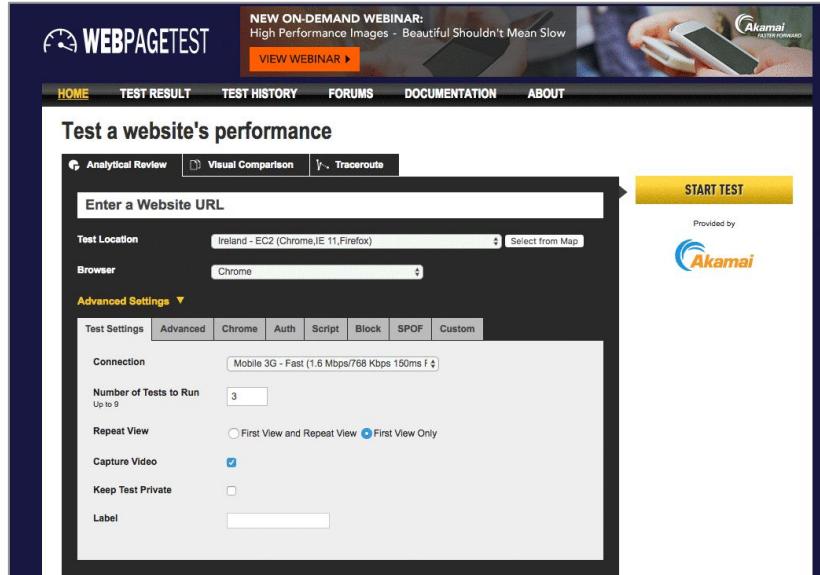
1.6s

Speed Index : 750

Speed Index : 1500

<https://github.com/Antoinebr/speed-lab>

# Измерение скорости с помощью Webpagetest.org



 Соединение: **3G**

 Число тестов: **3**

 Первичная загрузка  
(без кэша браузера)

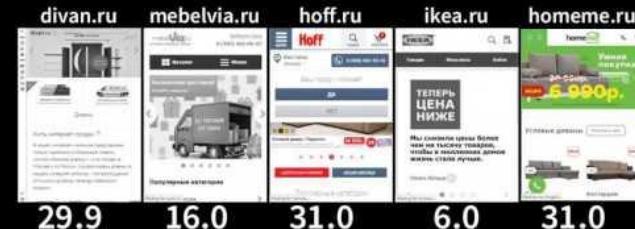
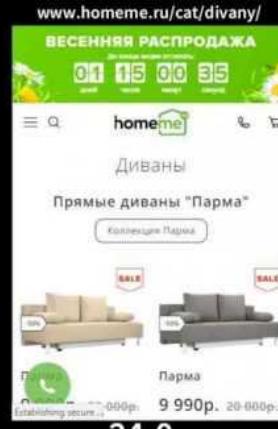
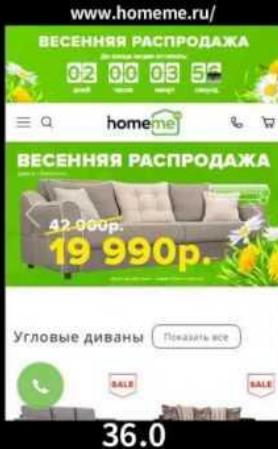
 Chrome (tab) :  
**Emulate Mobile Browser**

[webpagetest.org/mobiletest](http://webpagetest.org/mobiletest)

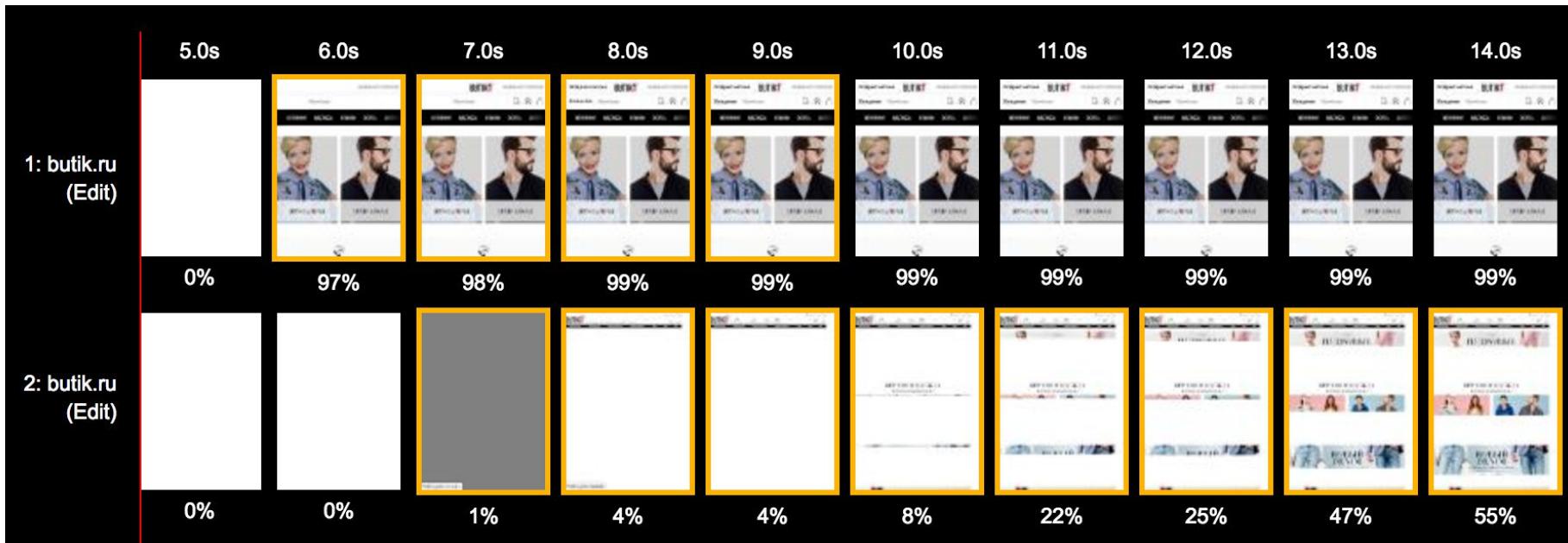
# Speed Index: наиболее значимая метрика для пользователя

Performance Results (Median Run)											
	Load Time	First Byte	Start Render	Speed Index	DOM Elements	Document Complete			Fully Loaded		
	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost				
First View (Run 2)	4.174s	0.805s	2.386s	2413	335	4.174s	30	394 KB	10.003s	96	817 KB
<a href="#">Plot Full Results</a>											

# С помощью WebPageTest можно делать видео загрузки страниц



... и сравнивать одну и ту же страницу в разные периоды времени



# Бенчмарки скорости для мобильного сайта

1

## Speed Index

На основе Webpagetest на 3G Fast\*



5000 мс  
или 5 секунд

2

Общее число запросов



80 запросов

3

Объем страницы

Общий вес в байтах



1.0 Мб

# Основные проблемы при достижении метрик

 5000 мс



 80 запросов



**Первичная загрузка  
(above the fold) не  
приоритезирована**

Слишком много запросов, особенно Javascript и CSS вставлены в верхнюю часть страницы (head)

 1.0 Мб



**Используется  
слишком много  
внешних ресурсов**

Каждый дополнительный ресурс (файл) требует времени на передачу по сети Интернет

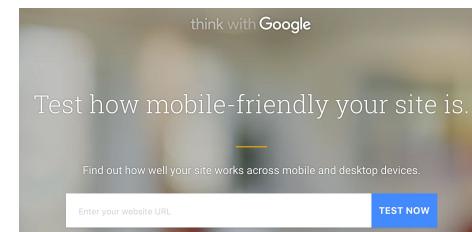
**Слишком тяжелая  
страница**

Картинки либо слишком большие по размеру (Мб и разрешение), либо их требуется слишком много

# Другие инструменты измерения скорости



## Mobile-Friendly Test



# Оптимизация изображений (на мобильных устройствах)

Почему скорость важна?

Как измерять скорость?

**Оптимизация изображений**

Critical Rendering Path

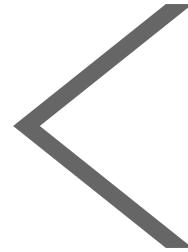
Выводы

# Две команды CSS, влияющие на скорость загрузки изображений

```
Img{  
display: none;  
}
```

```
Img{  
max-width: 100%;  
}
```

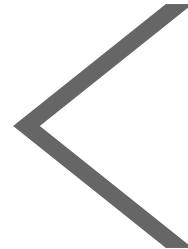
# Базовое правило: обрезайте изображение под размер мобильного экрана!



1800 x 1200px

```
Img{  
    max-width:100%;  
}
```

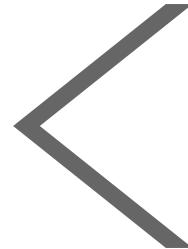
# Тег srcset: задание разных изображений под тип устройства



```

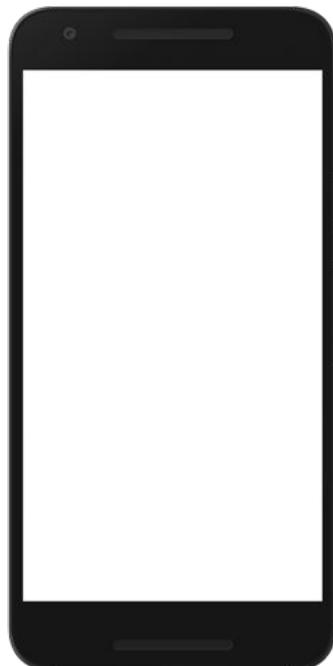
```

# Другой способ - background images



```
@media (min-width: 768px) {  
    .myImage { background-image: url(http://placehold.it/992x150); }  
  
}  
  
@media (min-width: 992px) {  
    .myImage { background-image: url(http://placehold.it/1200x150); }  
  
}  
  
@media (min-width: 1200px) {  
    .myImage { background-image: url(http://placehold.it/2000x150); }  
}
```

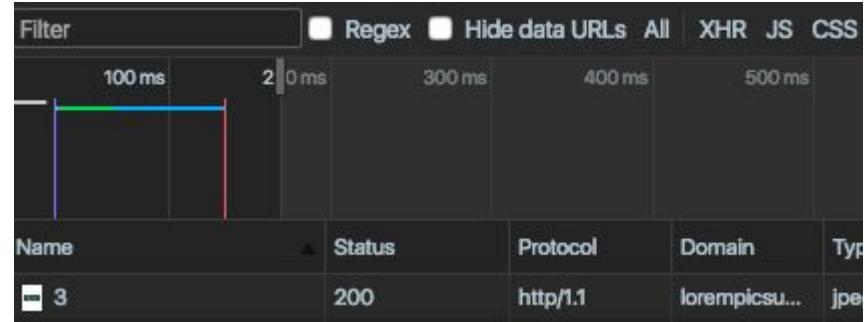
# Неправильный подход: скачать и скрыть



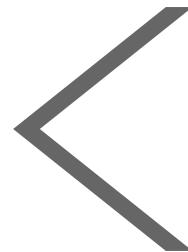
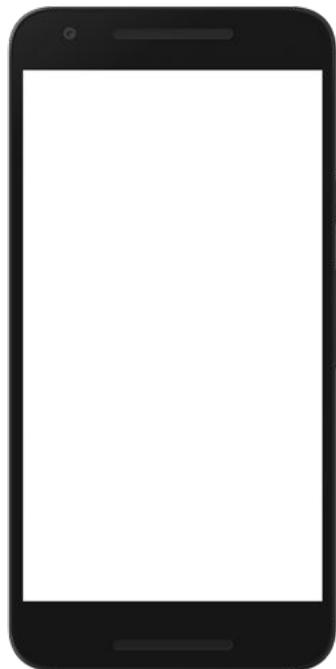
style.css

```
@media only screen and (max-width: 768px){  
    img{  
        display: none;  
    }  
}
```

network tab



# Решение: background images



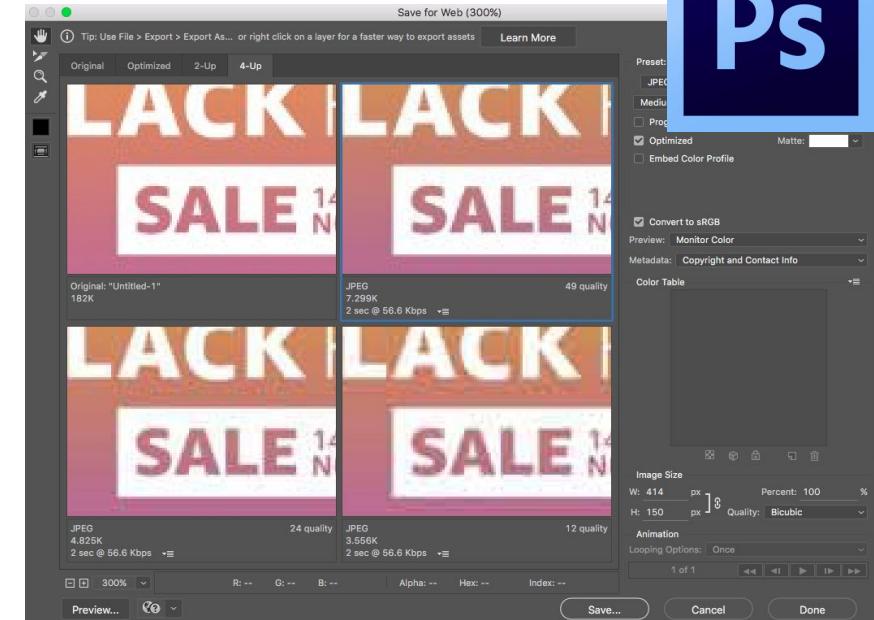
```
@media (max-width: 768px) {  
    .myImage { background-image: none; }  
  
}  
  
@media (min-width: 768px) {  
    .myImage { background-image: url(http://placehold.it/992x150); }  
}
```

Попробуйте **отказаться от**  
**высокого качества**  
изображения в пользу  
**производительности**

# Сжимайте изображения!



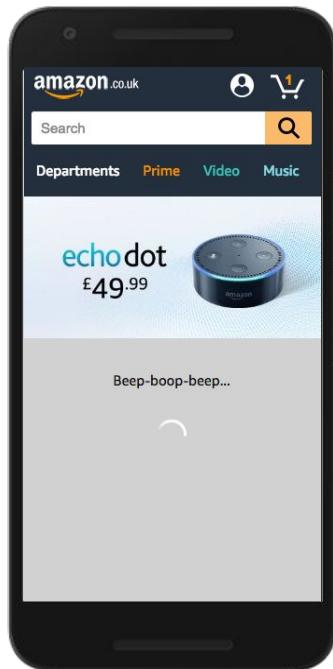
**8.4 KB**  
414 x 150



Оптимизируйте  
**видимые на первом**  
**экране** изображения  
(above the fold)



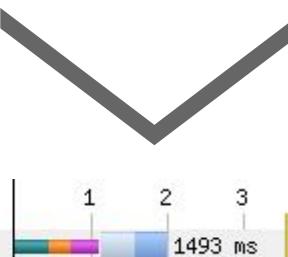
# Base64: способ закодировать изображение в html



```

```

<https://www.amazon.co.uk>  
1. www.amazon.co.uk - /

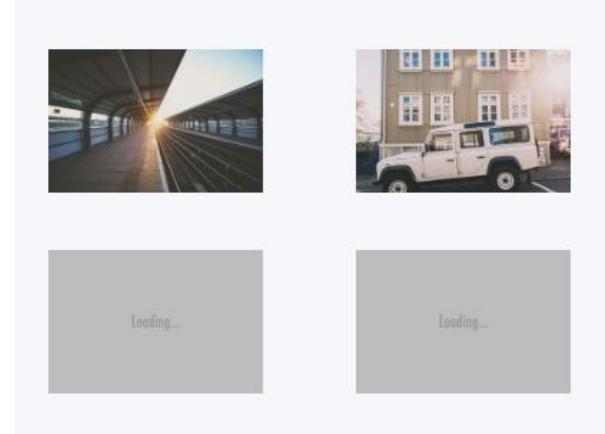


# Как закодировать изображение в Base64?

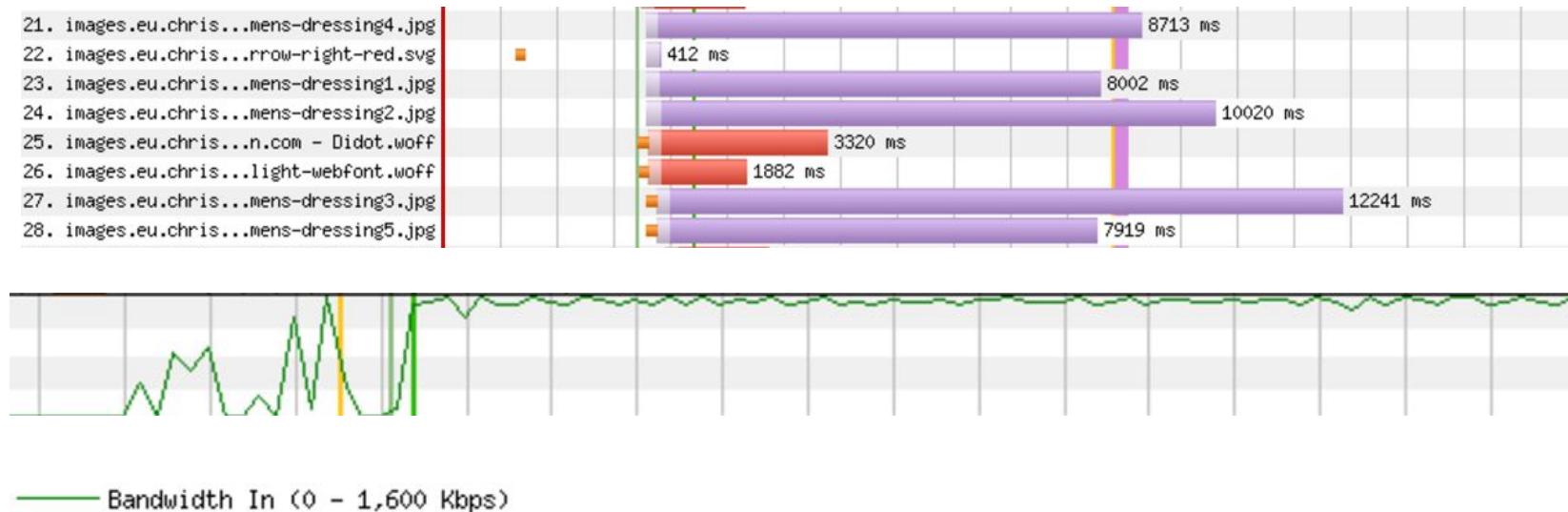
The screenshot shows the homepage of the [BASE64 IMAGE](https://www.base64-image.de/) website. At the top, there is a navigation bar with links for HOME, TUTORIAL, FAQ, and CONTACT. Below the navigation, a large banner features a cartoon character named "MR BASE" (a red, round character with a white shirt and tie) holding a briefcase. The banner has a call-to-action button that says "DRAG & DROP IMAGES ANYWHERE" and another button that says "OR CLICK HERE". The main text on the banner reads "CONVERT YOUR IMAGES TO BASE64".

<https://www.base64-image.de/>  
<https://www.base64encode.org/>

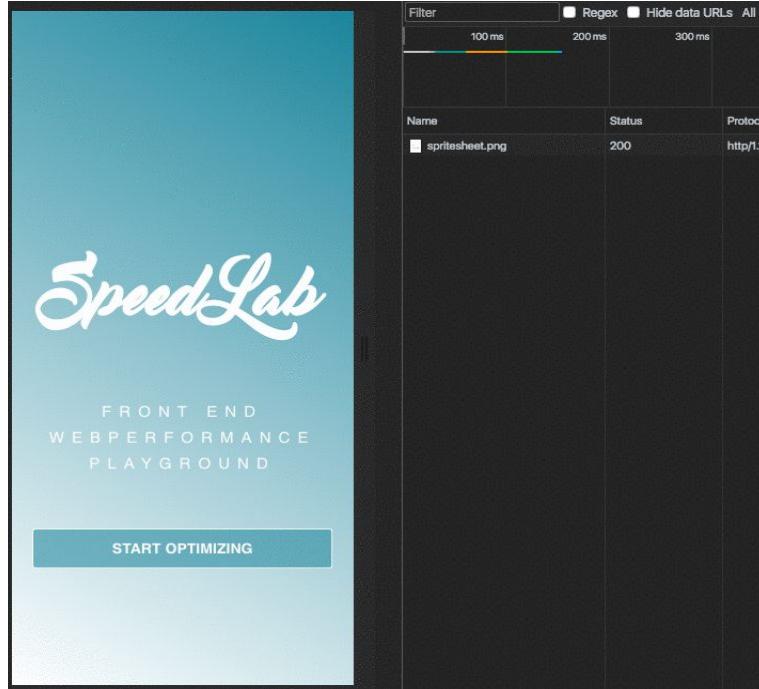
# Загрузка изображений по мере необходимости (Image lazy loading)



# Из-за картинок происходит насыщение полосы пропускания



# Подход Lazy loading



<https://github.com/aFarkas/lazysizesw>

**Lazy loading** позволяет загрузить изображение непосредственно **перед** просмотром пользователем

# Установка Lazy loading

The screenshot shows the GitHub README page for the `lazysizes` project. It includes the project title, a detailed description of what `lazysizes` is and how it works, and several code snippets demonstrating its usage. At the bottom, there are links for 'View all of README.md', 'Pulse' (activity), and 'Issues'.

**lazysizes**

**lazysizes** is a fast (jank-free), SEO-friendly and self-initializing lazyloader for images (including responsive images `picture` / `srcset`), iframes, scripts/widgets and much more. It also prioritizes resources by differentiating between crucial in view and near view elements to make perceived performance even faster.

It may become also your number one tool to integrate responsive images. It can automatically calculate the `sizes` attribute for your responsive images, it allows you to share media queries for your `media` attributes with your CSS, helping to separate layout (CSS) from content/structure (HTML) and it makes integrating responsive images into any environment really simple. It also includes a set of optional plugins to further extend its functionality.

[View all of README.md](#)

[Pulse](#) Past week

[Issues](#)

The screenshot displays three code snippets illustrating different ways to implement lazy loading using the `lazysizes` library:

```
<script src="lazysizes.min.js" async=""></script>
```

```

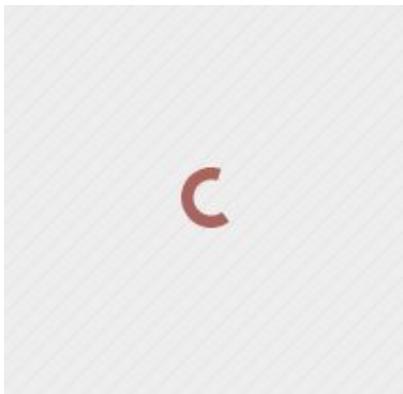
```

```

```

<https://github.com/aFarkas/lazysizesw>

# Lazy loading: что рисуется на месте картинок?



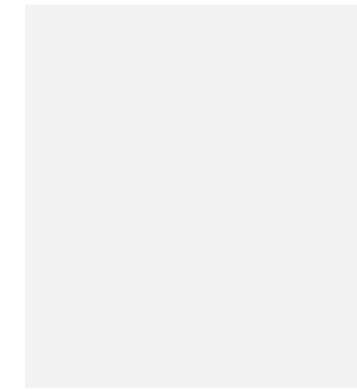
**Spinner**



**Placeholder**



**Solid Color**



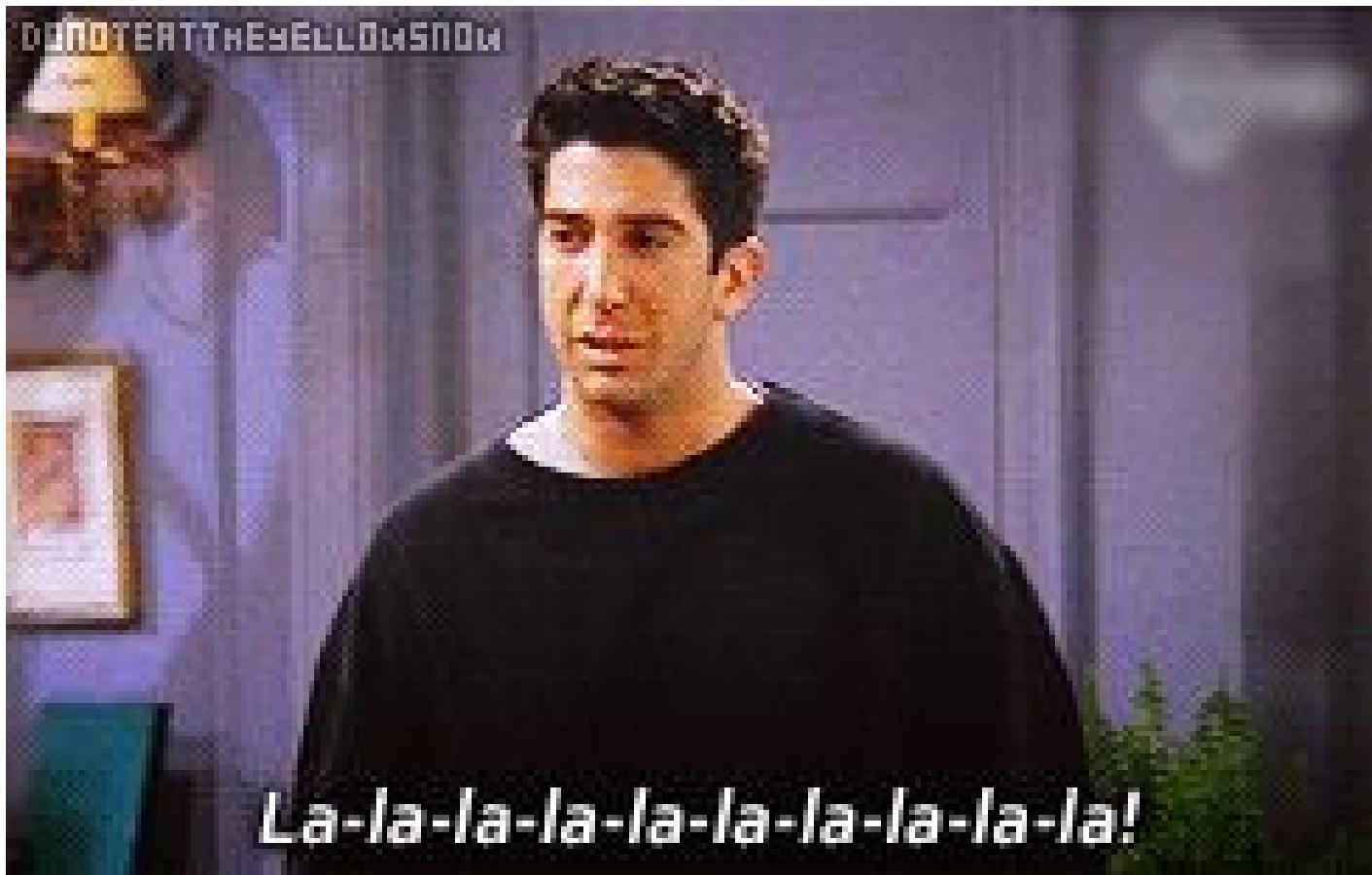
**Progressive / Blur**

Spinner demo :

[goo.gl/VNtZSI](http://goo.gl/VNtZSI)

Кто считает, что  
**карусель (slider)**  
**хорошо подходит к**  
мобильному сайту?





# Почему нужно **убирать** карусель с мобильного сайта?



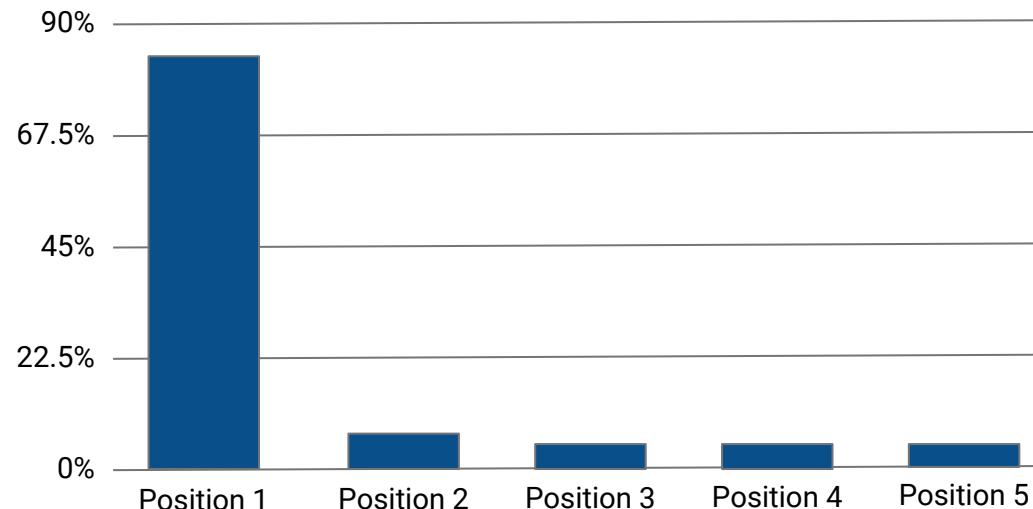
-  Плохая производительность
-  Первый слайд обычно заполняет все клики
-  Движение отвлекает
-  Информация изменяется в процессе чтения
-  Похоже на рекламу
-  Нужно загружать несколько картинок и скрипты

# До и после слайдера: улучшение загрузки в разы



# Первая позиция в слайдере - 80% кликов

**ND.edu Feature Click-Through Rate**



<https://www.orbitmedia.com/blog/rotating-sliders-hurt-website/>

# Кто уже отказался от слайдеров?

amazon.co.uk Hello, BROSS... 8

Search

Shop By Department Your Lists Deals Gift Card

**ALL-NEW fire TV stick** £39.99

Amazon Fashion **PERFECT BASICS** Meet the style staples that will take a starring role in your new season wardrobe > SHOP NOW

Related to items you've viewed

- LEDMO LED Strip light, SMD2835 Warm White, 3000K... £14.48
- TOPLUS 5050 led strip lights RGB led strips lighting full kit... £17.99
- NICEKER LED TV Backlight Bias lighting Kit - 100CM 5V LED... £8.99

FREE DELIVERY. On orders over €50 or every time with Nike+. Details.

**NEW ARRIVALS**

New styles in stock, and ready to rock.

SHOP NOW

SEPHORA

#TRENDINGatSEPHORA HIGH HIGHLIGHTING SHOP NOW >

Just Arrived

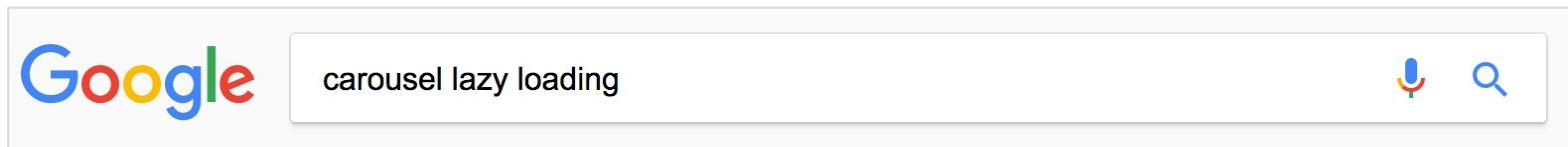
See More

**NEW** TOO FACED limited edition Natural Love Ultimate Neutral Eye Shadow

**NEW** BENEFIT COSMETICS Work Kit, Girl! Work-Day Essentials Makeup Kit

# Carousel lazy loading

- **Slick** → <https://github.com/kenwheeler/slick>
- **Lazy loader for Bootstrap** → <https://goo.gl/V2s62K>



# Critical Rendering Path (поведение браузера при отрисовке)

Почему скорость важна?

Как измерять скорость?

Оптимизация изображений

**Critical Rendering Path**

Выводы

# Web 101: три главных элемента любого сайта

## HTML



## HTML

Формирует структуру сайта, которая улучшается и модифицируется другими технологиями (CSS и Javascript)

<р>Это параграф </р>

## CSS



## CSS

Используется для стиля, форматирования и *layout* страницы. CSS - это список правил, применяемых к HTML-тегам на уровне одного элемента, многих, или всей страницы

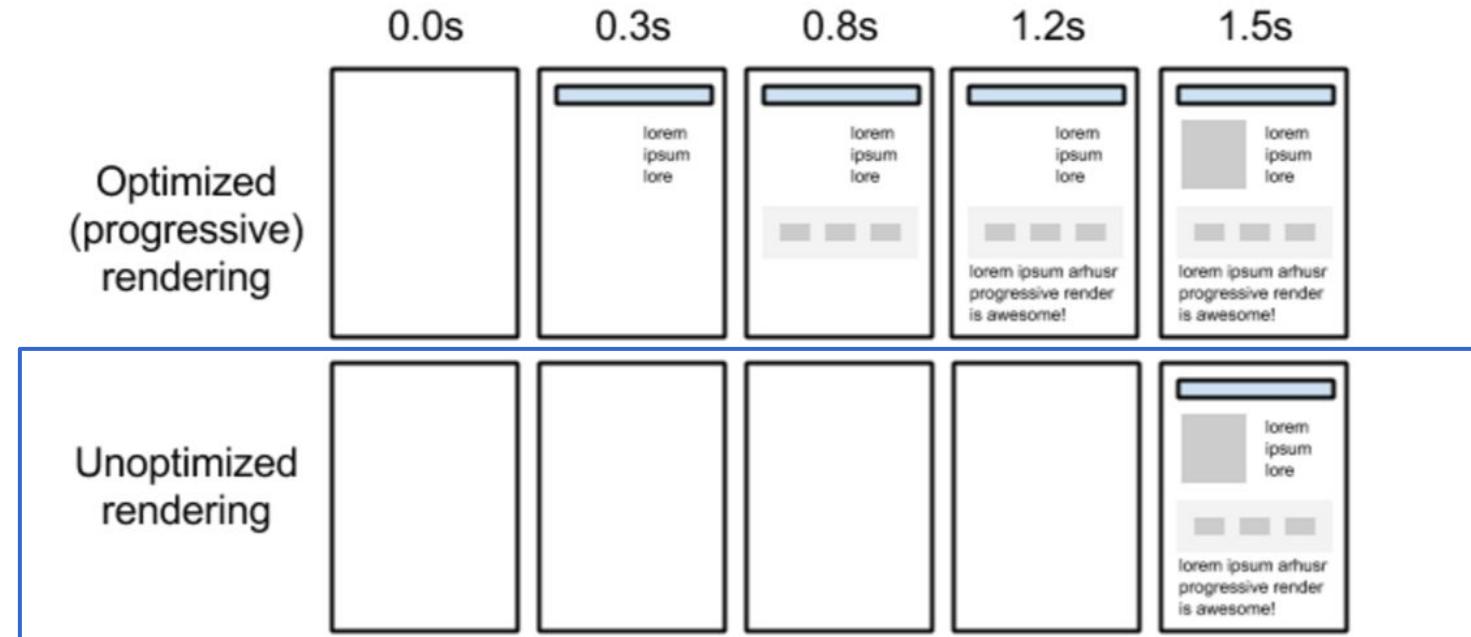
## JS



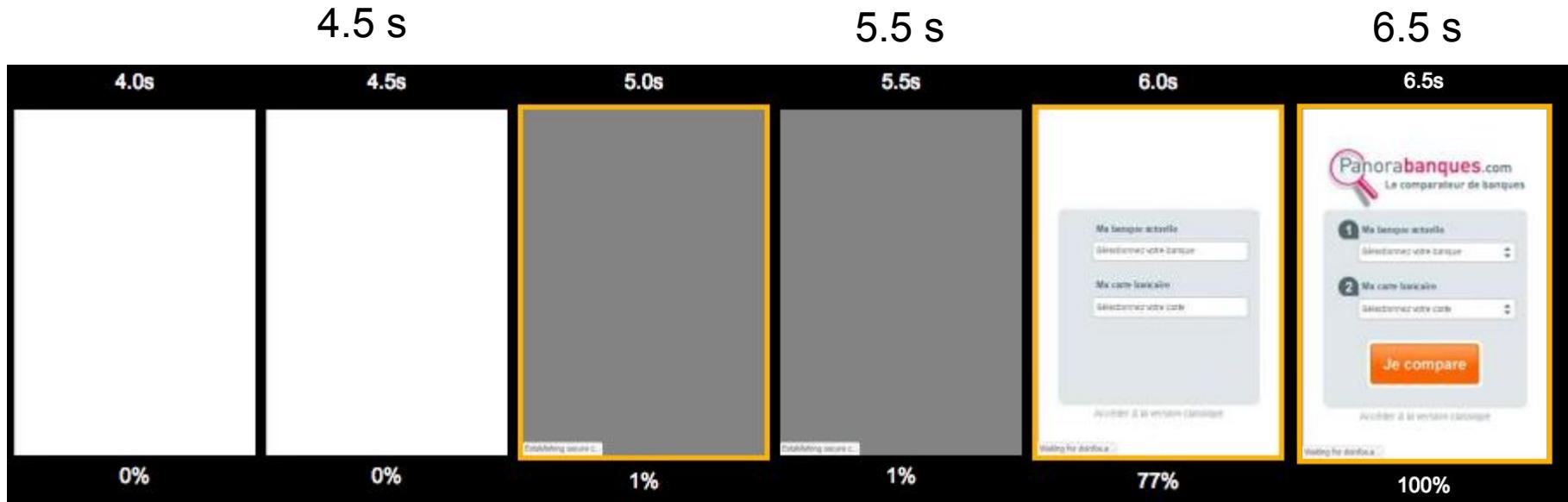
## Javascript

Используется для управления поведением разных элементов. Может быть использован для всплывающих меню, проверки форм, тегов рекламы, и т.д.

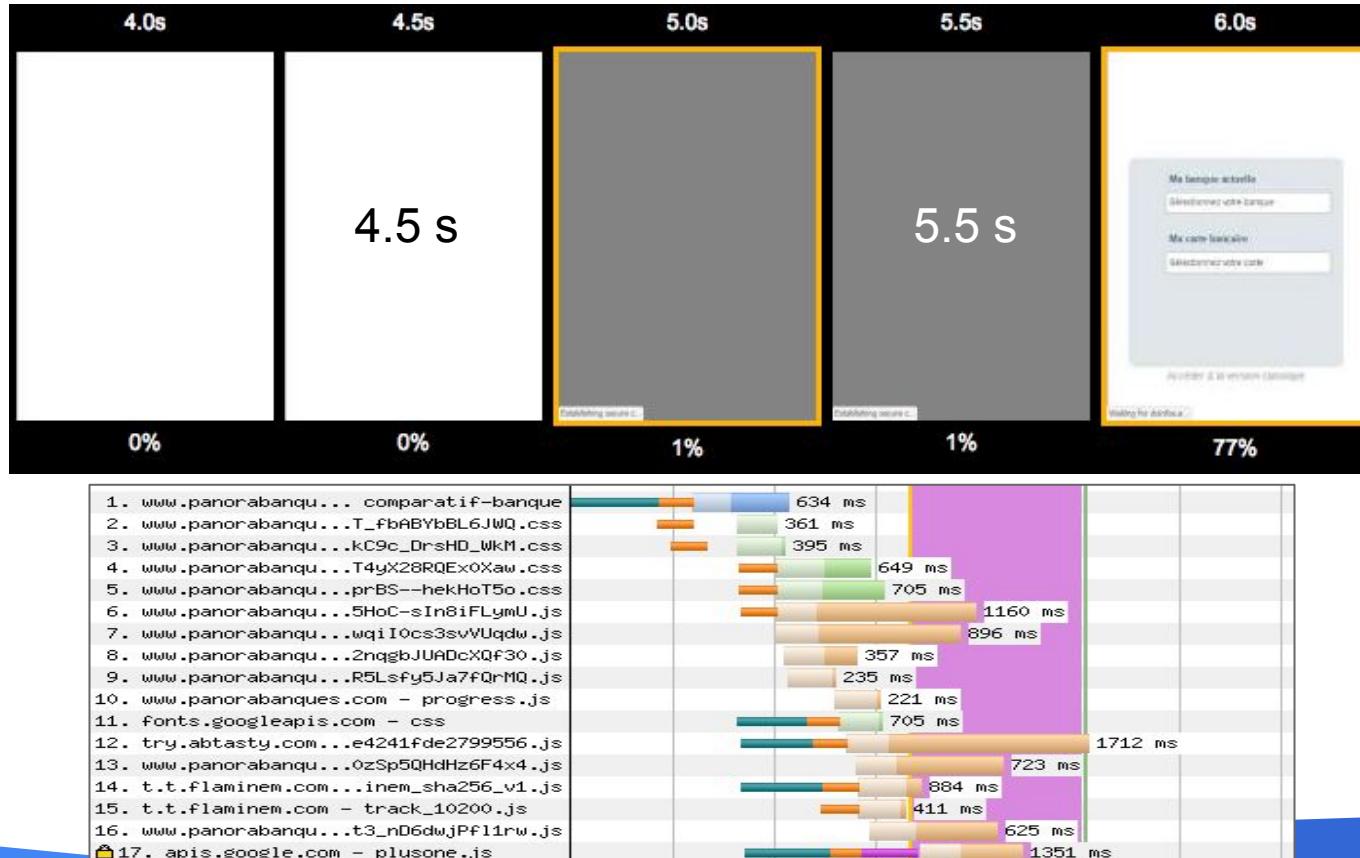
# Critical Rendering Path: приоритизация загрузки первой страницы



# Пример: что тут можно улучшить?



# Пример: что тут можно улучшить?



# Процесс отрисовки страницы браузером

index.html

```
<!DOCTYPE html>
<head>
  <link rel="stylesheet" href="app.css" />
  <script src="app.js"></script>
</head>

<body>
  ...
</body>
```

DOM building  
( HTML parsing)

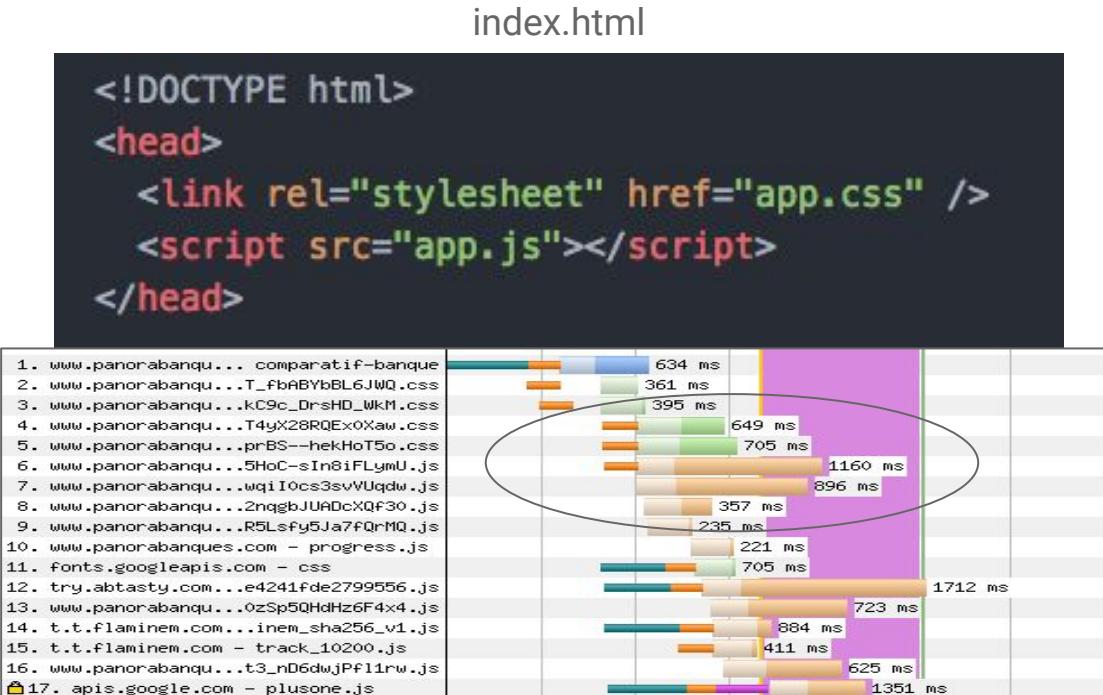
CSS

JavaScript

CSSOM + DOM  
= render Tree

Layout & Paint

# CSS и JavaScript блокируют отрисовку страницы



DOM building  
( HTML parsing)

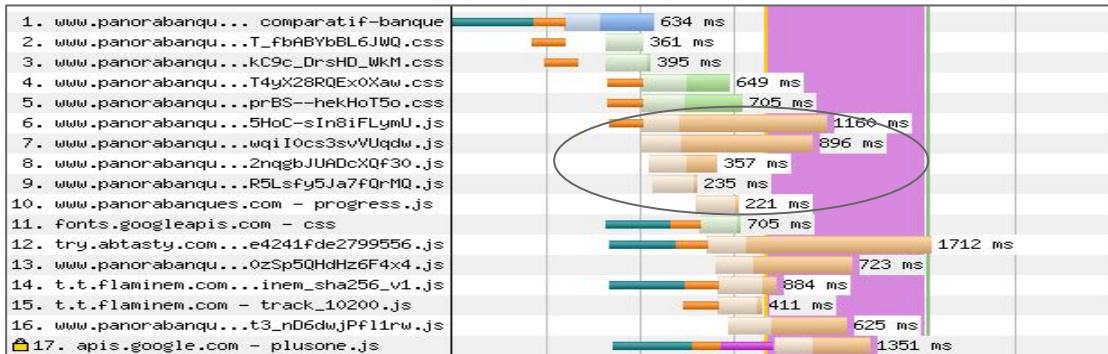
CSS

JavaScript

CSSOM + DOM  
= render Tree

Layout & Paint

# Выполнение JavaScript



<span style="color: green;">█</span> HTML parsing	<span style="color: blue;">█</span> Script download
<span style="color: grey;">█</span> HTML parsing paused	<span style="color: red;">█</span> Script execution



DOM building  
( HTML parsing)

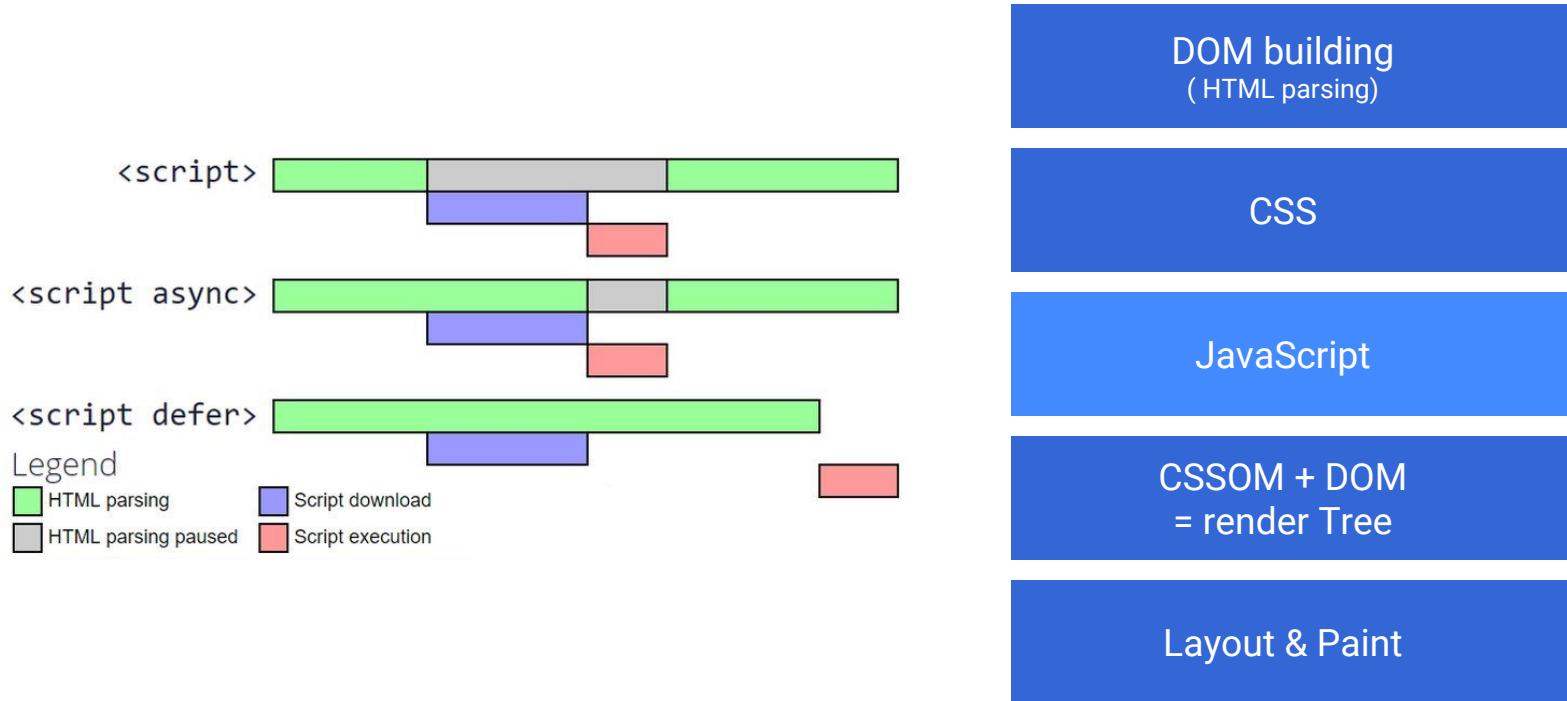
CSS

JavaScript

CSSOM + DOM  
= render Tree

Layout & Paint

# Варианты оптимизации Javascript: defer и async



# Когда что-то может пойти не так...

```
<h1>Hello</h1>

<script defer="defer" src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>

<script>
$(document).ready(() => {
    $('h1').append(' world !');
});
</script>
```

✖ Uncaught ReferenceError: \$ is not defined  
at index.html:13

DOM building  
( HTML parsing)

CSS

JavaScript

CSSOM + DOM  
= render Tree

Layout & Paint

● ► Uncaught ReferenceError: \$ is not defined  
at index.html:13

Proprietary + Confidential



```
<script>
  function whenAvailable(name, callback) {
    var interval = 10; // ms
    window.setTimeout(function() {
      if (window[name]) {
        callback(window[name]);
      } else {
        window.setTimeout(arguments.callee, interval);
      }
    }, interval);
  }
</script>

<h1>Hello</h1>

<script defer="defer" src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>

<script>
  whenAvailable('$', ()=>{
    $(document).ready(() => {
      $('h1').append(' world !');
    });
  })
</script>
```

demo :  
[goo.gl/Xy2jaf](http://goo.gl/Xy2jaf)

# Как оптимизировать CSS?



DOM building  
(HTML parsing)

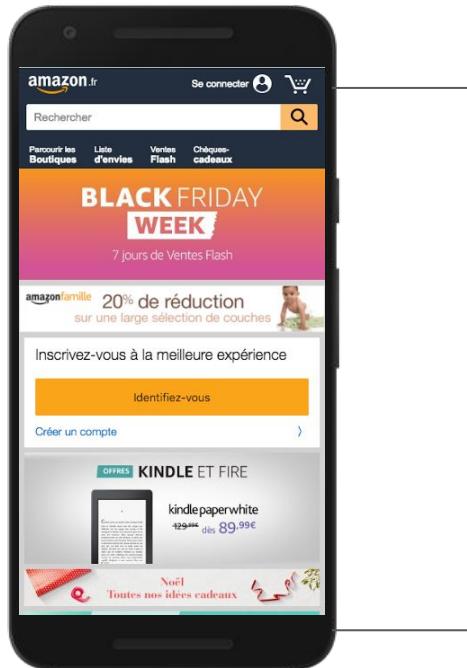
CSS

JavaScript

CSSOM + DOM  
= render Tree

Layout & Paint

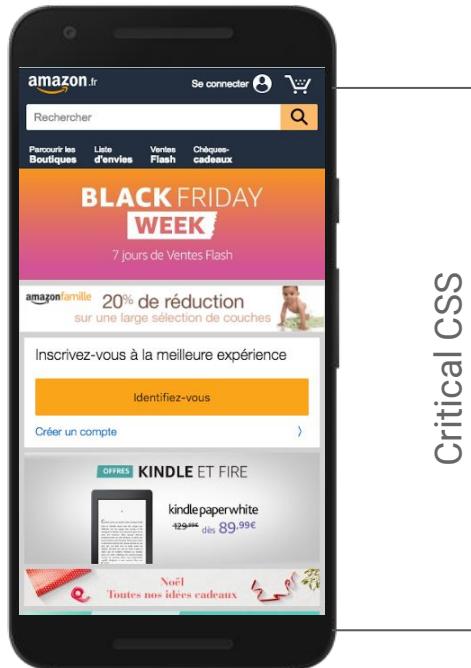
# Оптимизация CSS - загрузка самых критических правил в первую очередь



`<style> Inlined critical CSS </style>`

`<script> Load every other CSS file in async </script>`

# Выделение критических правил CSS

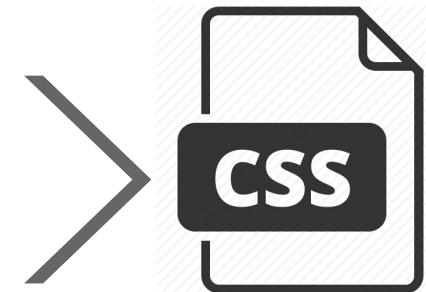


```
var critical = require('critical');

critical.generate({
  src: "http://amazon.fr",
  dest: 'critical.css',
  width: 412,
  height: 732
});
```

extraction (node.js)

<https://github.com/addyosmani/critical>



Node module:  
critical.css

# Загрузка остальных правил вторична



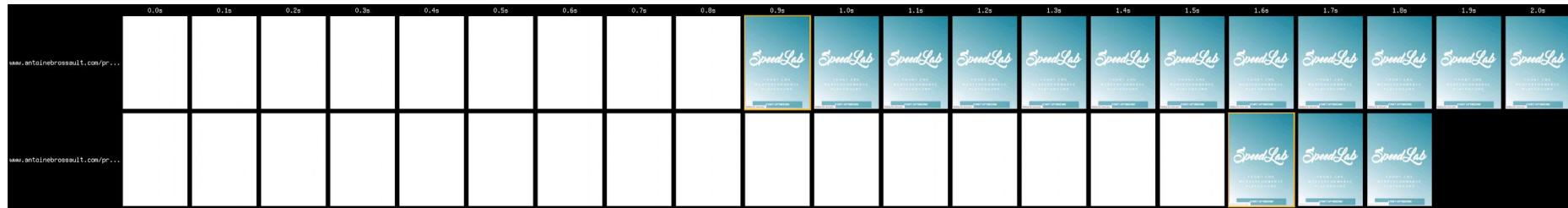
Non-critical CSS

```
<script>loadCSS( "css/app.css" );</script>

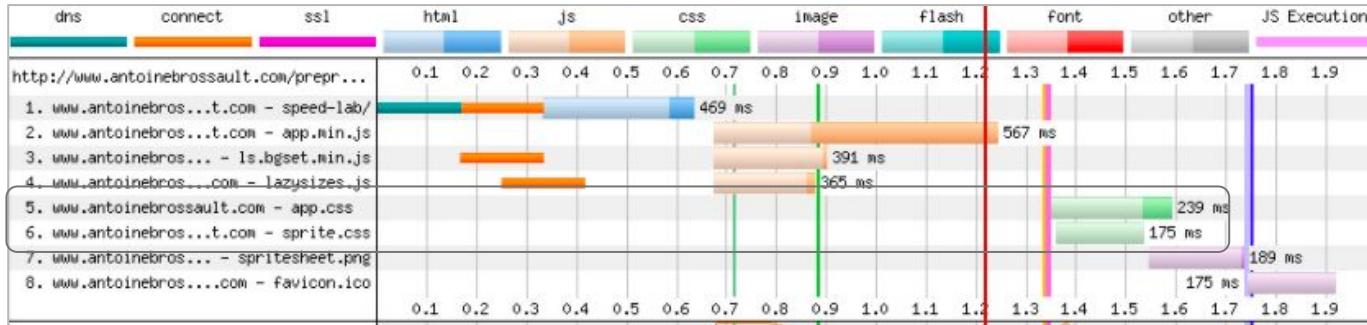
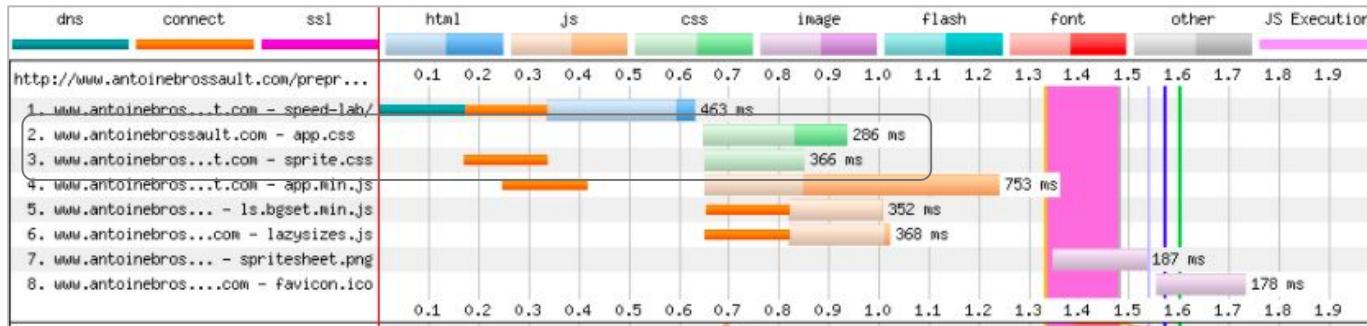
<script>loadCSS( "css/sprite.css" );</script>
```

<https://github.com/filamentgroup/loadCSS>

# Результат: ускорение визуальной загрузки с 1.6с до 0.9с



# Результат: сравнение тестов Webpagetest



# Выводы

Почему скорость важна?

Как измерять скорость?

Оптимизация изображений

Critical Rendering Path

**Выводы**

## Вывод 1: Мобильная скорость важна из-за

- ❑ ожиданий пользователя
- ❑ конверсии и мобильной выручки
- ❑ более низкого CPA и эффективной рекламы



## Вывод 2: Измерение скорости

- ❑ нужно проводить на *WebPageTest.org*
- ❑ стремиться к бенчмарку 5 секунд для SpeedIndex, 1Мб для изображений, 80 запросов в сумме



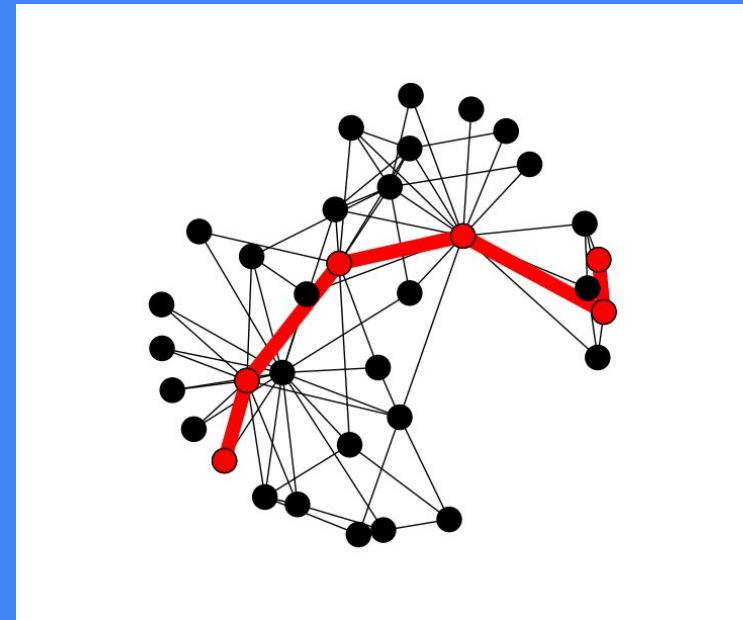
## Вывод 3: оптимизация сайта начинается с изображений через

- ❑ задание правильного размера (resize) и качества (compression)
- ❑ отказа от слайдера (карусели)



## Вывод 4: оптимизация продолжается через Critical Rendering Path,

- ❑ улучшая загрузку скриптов Javascript (defer, async)
- ❑ упрощая работу стилей (inline critical CSS)



Delete

End

P  
D