

Как НЕ удовлетворить Google PageSpeed

@Tchernyavsky









Скорость загрузки



Оценка скорости загрузки основана на данных, полученных методом имитации загрузки сайта с помощью инструмента [Lighthouse](#).

Затраченное время: 17.02.2019, 19:05:30

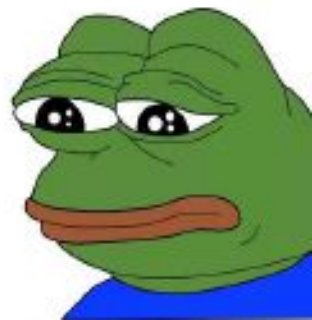
Scale: ■ 90-100 (высокая скорость) ■ 50-89 (средняя скорость) ■ 0-49 (низкая скорость)



Оценка скорости загрузки основана на данных, полученных методом имитации загрузки сайта с помощью инструмента [Lighthouse](#).

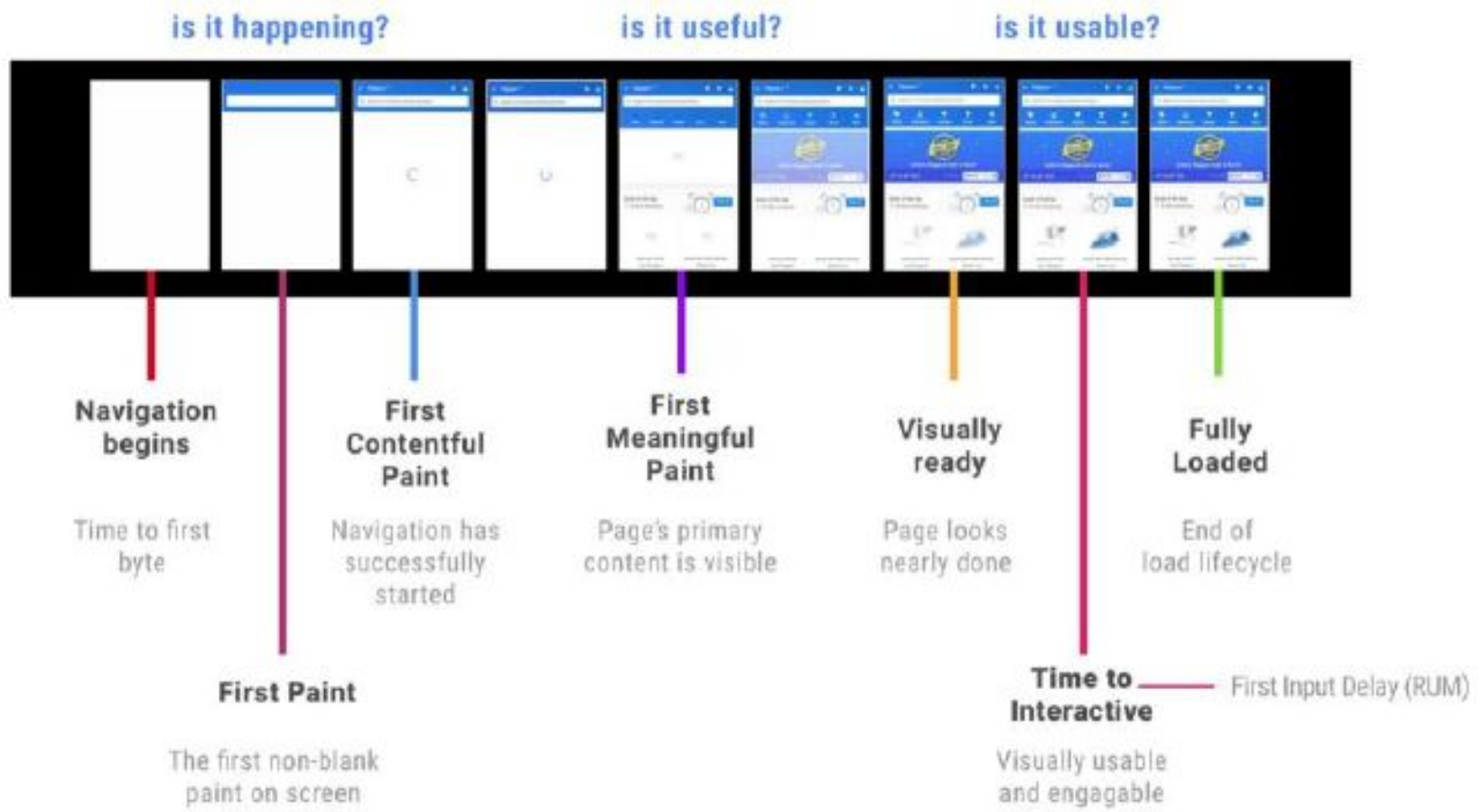
Затраченное время: 17.02.2019, 14:37:20

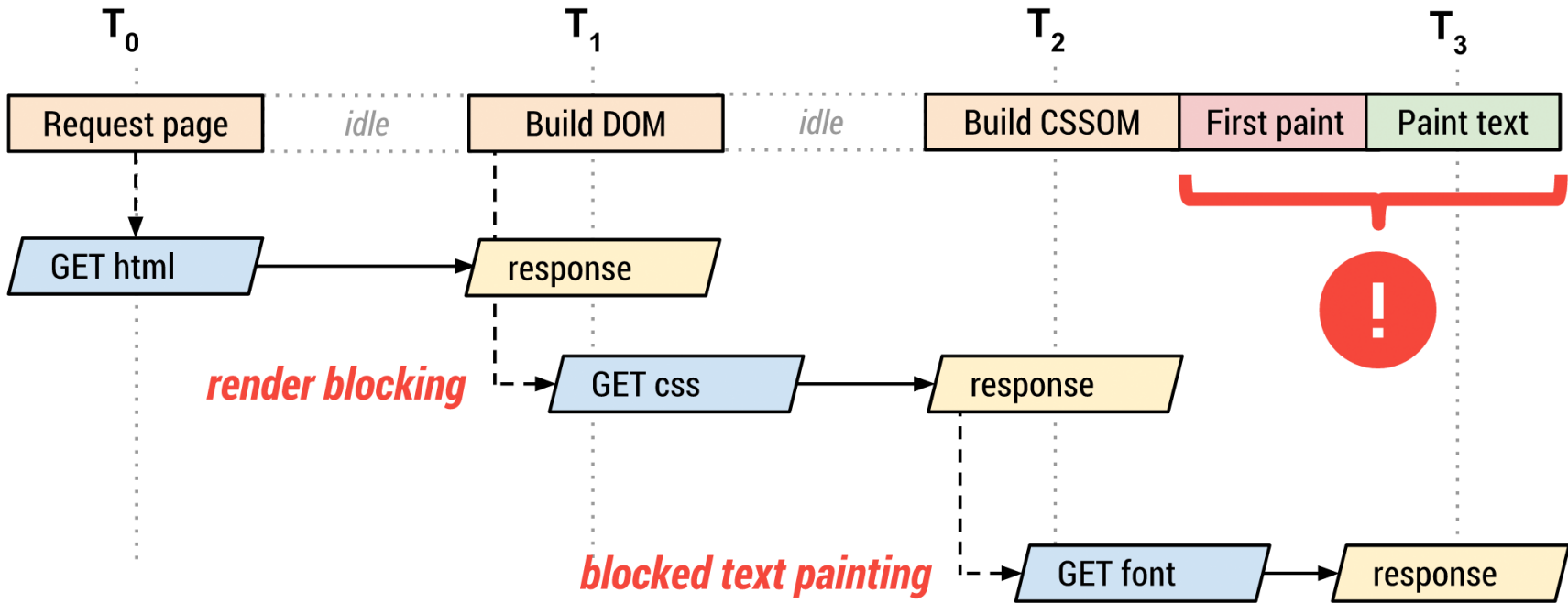
Scale: ■ 90-100 (высокая скорость) ■ 50-89 (средняя скорость) ■ 0-49 (низкая скорость)



Что мы сегодня будем оптимизировать?

- First Contentful Paint
- Preload/prefetch/preconnect
- Шрифты
- Lazy loading
- <picture>
- WebP
- YouTube iframe
- Time to Interactive







Critical CSS

Обычный

Critical CSS



Critical by addyosmani



Critters

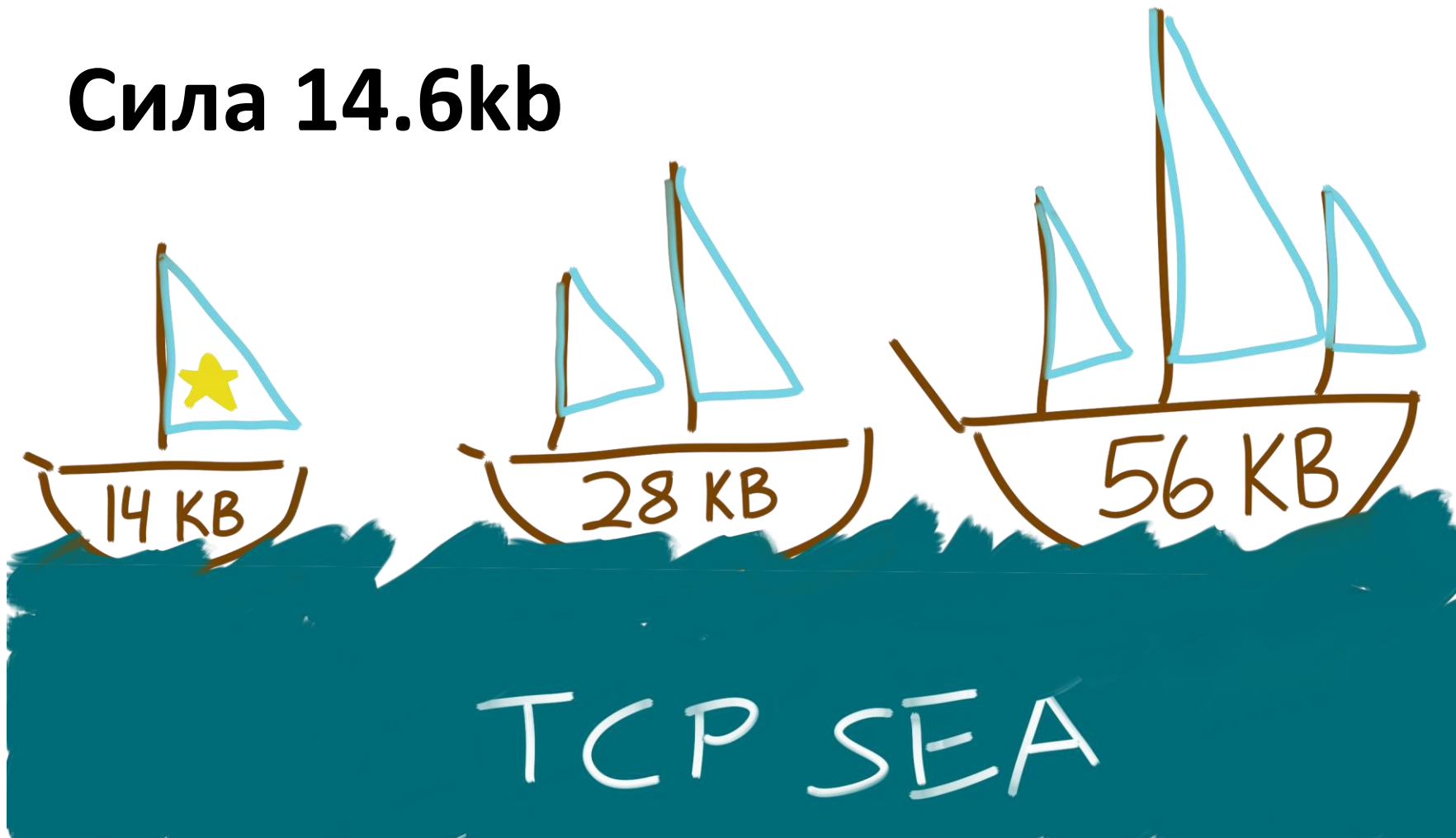


Critical

Non-Critical


```
<head>
  <style> /* inlined critical CSS */ </style>
  <script> loadCSS('non-critical.css'); </script>
</head>
```


Сила 14.6kb



H2

HTTP/1.1
3.09s

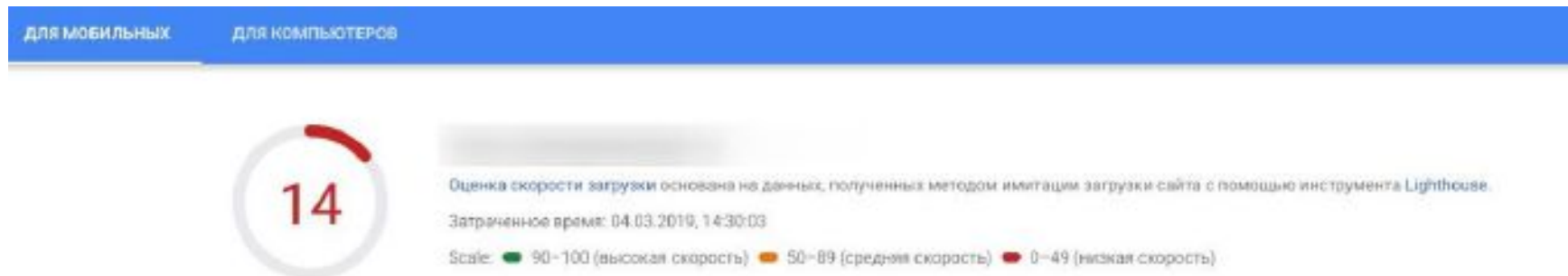

REFRESH



Run HTTP/2 test

Critical CSS

Результаты оптимизации:



Прогрессивный CSS


```
<body>
  <link rel="stylesheet" href="header.css">
  // header component

  <link rel="stylesheet" href="article.css">
  // article component

  <link rel="stylesheet" href="comments.css">
  // comments component






  <link rel="stylesheet" href="footer.css">
  // footer component
</body>
```

Медиавыражения по типам

Responsive 425 x 554

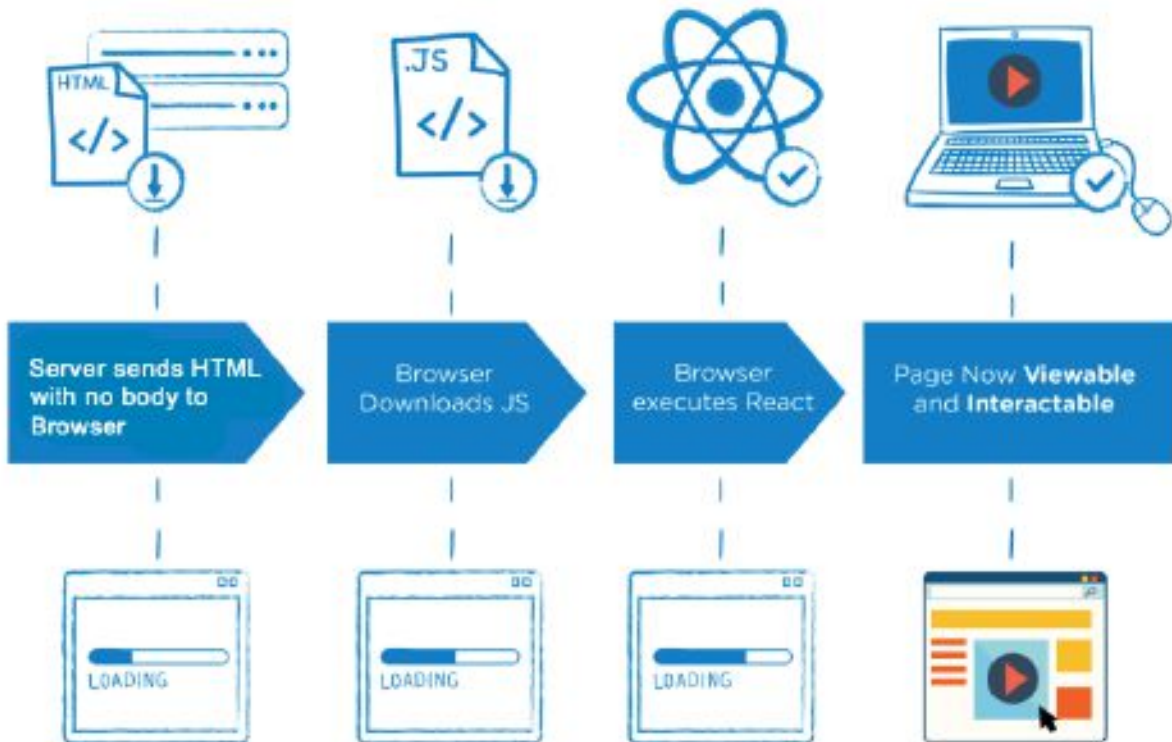
Elements Console Sources

Filter Hide data URLs

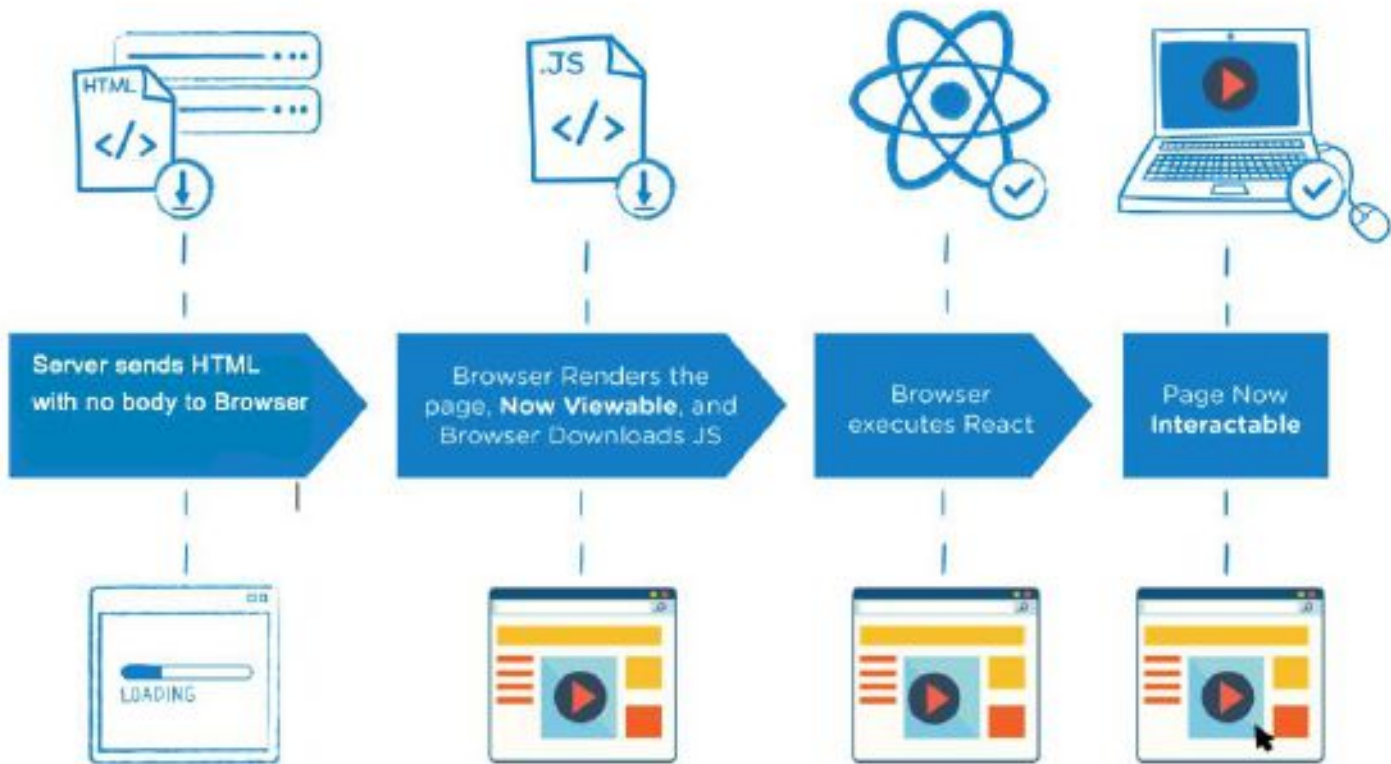
Name	Type	Time	Priority
 all.css /Users/da...	style...	4 ms 4 ms	Highest
 small.css /Users/da...	style...	5 ms 5 ms	Highest
 medium.css /Users/da...	style...	5 ms 5 ms	Lowest
 large.css /Users/da...	style...	5 ms 5 ms	Lowest
 another.css /Users/da...	style...	7 ms 7 ms	Lowest

```
<link rel="stylesheet" href="all.css"> media="all" />  
<link rel="stylesheet" href="small.css" media="(min-width: 20em)" />  
<link rel="stylesheet" href="medium.css" media="(min-width: 64em)" />  
<link rel="stylesheet" href="large.css" media="(min-width: 90em)" />
```

CSR



SSR



Директива Preload/Prefetch




Prefetch

Preload

На самом деле все ситуативно

Preload работает с текущей навигацией и ресурсами с высоким приоритетом



```
<link rel="preload" href="style.css" as="style">  
<link rel="preload" href="main.js" as="script">  
<link rel="preload" href="some-image.png" as="image" media="(max-width: 600px)">
```

Prefetch — ресурсы с низким приоритетом для следующей навигации

Without prefetching



With prefetching



Preload ~ 77%

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android
		2-55		3.1-10.1		3.2-10.3						
	12-16	56	4-49	11	10-36	11.2						
6-10	17	57-64	50-71	11.1	37-56	11.4		2.1-4.4.4	7	12-12.1		
11	18	65	72	12	57	12.1	all	67	10	46	71	64
		66-67	73-75	12.1-TP		12.2						

Prefetch ~ 79%

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android
								2.1-3				
								4				
			4-7		10-12.1			4.1-4.3				
6-10	12-17	2-64	8-71	3.1-11.1	15-56	3.2-11.4		4.4-4.4.4	7	12-12.1		
11	18	65	72	12	57	12.1	all	67	10	46	71	64
		66-67	73-75	12.1-TP		12.2						

Preconnect



```
<link rel="preconnect" href="https://www.youtube.com">  
<link rel="preconnect" href="https://i.ytimg.com">
```

Preconnect ~ 88%

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android
		2-38									
	12-14	¹ 39	4-45	3.1-11	10-32	3.2-11.2					
6-10	² 15-17	40-67	46-74	11.1-12	33-60	11.3-12.1		2.1-4.4.4	7	12-12.1	
11	² 18	68	75	12.1	62	12.3	all	67	10	46	75
	76	69-70	76-78	13-TP		13					

Webpack



Webpack

Takes routes you code-split

```
import('lazy-loaded-sean.js')  
import('lazy-loaded-tobias.js')
```



webpack

and preloads them up using `<link rel='preload'>`

```
<link rel="preload" href="chunk.31132ae6680e598f8879.js" as="script">  
<link rel="preload" href="chunk.d15e7fd9c91b34bb78c4.js" as="script">
```

or `<link rel='prefetch'>`

```
<link rel="prefetch" href="chunk.31132ae6680e598f8879.js">  
<link rel="prefetch" href="chunk.d15e7fd9c91b34bb78c4.js">
```



[html-webpack-plugin](#)



[preload-webpack-plugin](#)

Webpack

```
plugins: [  
  new HtmlWebpackPlugin({  
    prefetch: ['**/*.*.'],  
    preload: ['**/*.*.'],  
  }),  
  new ResourceHintWebpackPlugin()  
]
```



[resource-hints-webpack-plugin](#)

```
plugins: [  
  new HtmlWebpackPlugin(),  
  new ScriptExtHtmlWebpackPlugin({  
    inline: 'startup',  
    sync: [/!mp(1|2){1,3}/, 'initial'],  
    defer: ['slow', /big.*andslow/],  
    module: [/^((?!sync).)*$/, 'mod'],  
    prefetch: 'indirectly-referenced.js',  
    defaultAttribute: 'async'  
  })  
]
```



[script-ext-html-webpack-plugin](#)



Webpack magic comments



```
import(/* webpackPreload: true */ 'ChartingLibrary');  
// => <link rel="preload" as="script" href="charting-library-chunk.js" />
```



```
import(/* webpackPrefetch: true */ 'LoginModal');  
// => <link rel="prefetch" href="login-modal-chunk.js" />
```

preload/prefetch

Результаты оптимизации:

- 1 Настройте предварительную загрузку ключевых запросов

1,5 s ^

Чтобы загружать требуемые ресурсы в порядке приоритета, вам следует использовать `<link rel=предварительную загрузку>`. [Подробнее....](#)

для мобильных

для компьютеров



Оценка скорости загрузки основана на данных, полученных методом имитации загрузки сайта с помощью инструмента Lighthouse.

Затраченное время: 04.03.2019, 14:24:47

Scale: ● 90–100 (высокая скорость) ● 50–89 (средняя скорость) ● 0–49 (низкая скорость)

Шрифты

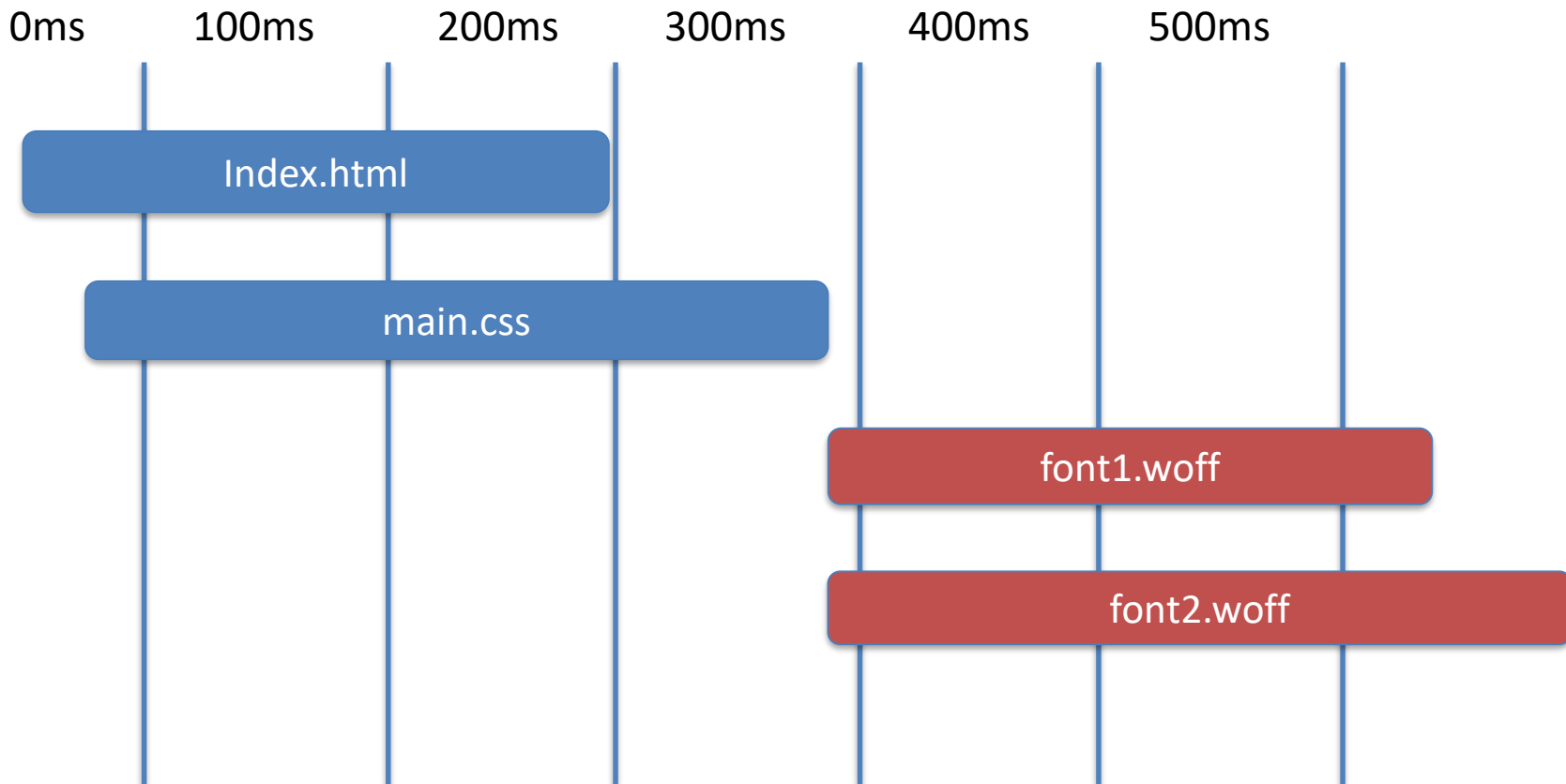
1 Настройте показ всего текста во время загрузки веб-шрифтов



Чтобы пользователь мог видеть текст, пока веб-шрифты не загрузились, используйте функцию отображения шрифтов CSS. [Подробнее....](#)

URL	Потенциальная экономия времени (мс)
...fonts/Lato-Bold.woff2	620 ms
...fonts/Lato-Regular.woff2	620 ms
...fonts/Lato-Semibold.woff2	620 ms
...fonts/Lato-Black.woff2	620 ms

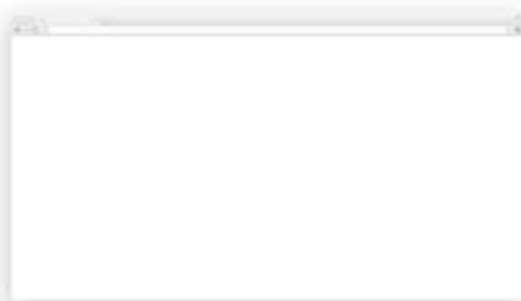
Загрузка веб-шрифтов ожидает загрузки файлов HTML и CSS.



Проблемы с Миттом Ромни из-за веб-шрифта



Flash Of Unstyled Text (FOUT)



запрос



загрузка контента



шрифт загружен

Flash Of Invisible Text (FOIT)



запрос



загрузка контента



шрифт загружен

FOUT vs FOIT

Chrome	Firefox	Safari	Opera	IE9-11	Edge
FOIT	FOIT	FOIT	FOIT	FOUT	FOUT
3 sec.	3 sec.	∞	3 sec.	n/a	n/a

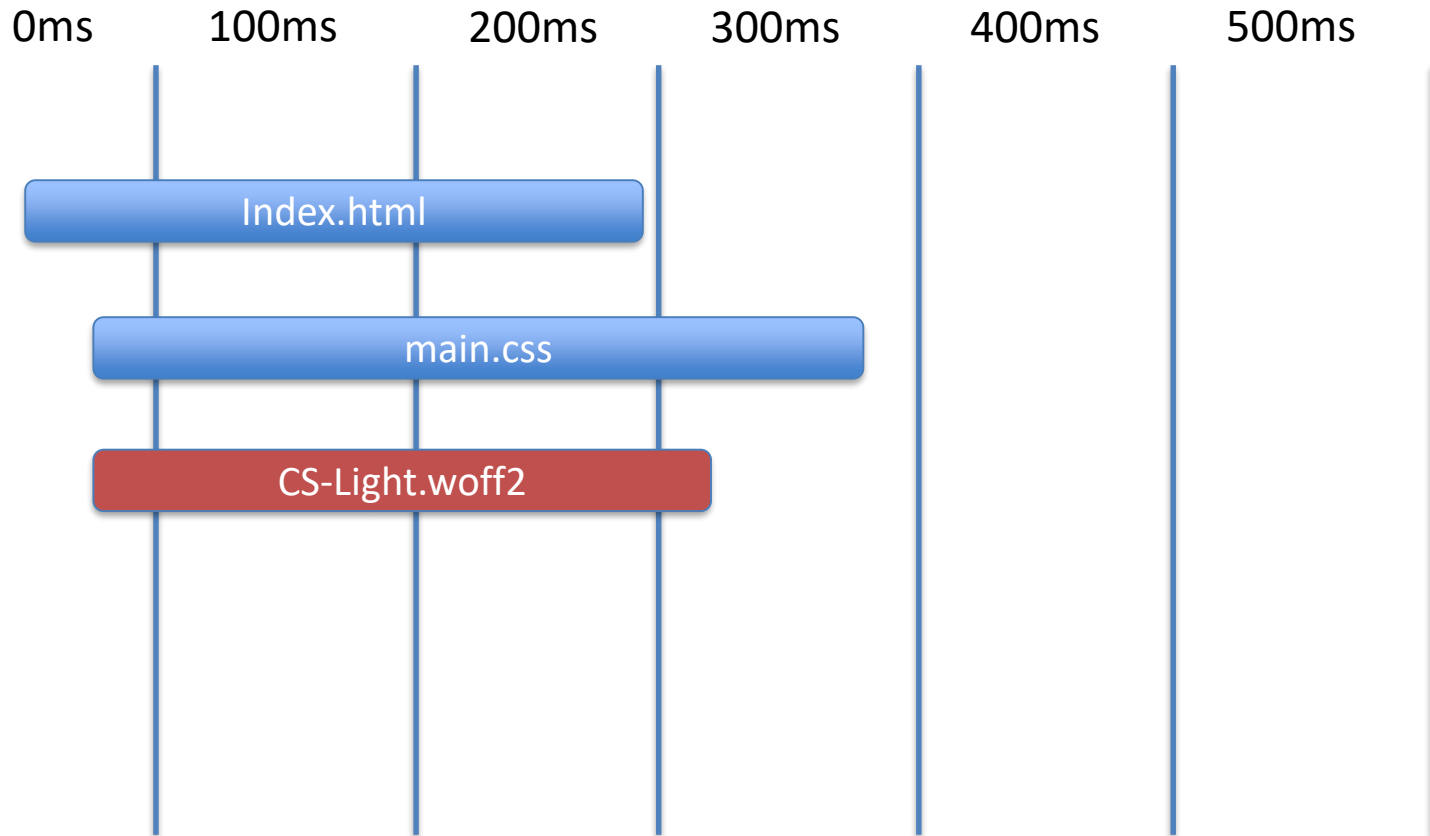
Предзагрузка с preload

```
<head>
  <link rel="preload" href="CS-Light.woff2" as="font" type="font/woff2" crossorigin="anonymous">
</head>
```

Name	Size	Time	Waterfall	5.00 s	10.00 s
<input type="checkbox"/> CS-Light.woff2	72.0 KB	2.14 s			
<input type="checkbox"/> tt?<v=6303277870179704	32.3 KB	1.88 s			
<input type="checkbox"/> CAC-Regular.woff2	93.4 KB	4.25 s			
<input type="checkbox"/> CS-Regular.woff2	73.0 KB	3.68 s			
<input type="checkbox"/> CS-Demi.woff2	74.0 KB	3.74 s			

Скорость отрисовки лучше, чем у **@font-face**
Легко реализовать, просто добавь **<link>**

preload



unicode-range

38 попугаев

```
@font-face{  
  font-family: 'WebFont';           /* определили название шрифта */  
  src: local('Arial Black');        /* указываем нужный шрифт */  
  unicode-range: U+0030-0039;      /* указываем диапазон символов, только цифры */  
}  
  
@font-face{  
  font-family: 'WebFont';  
  src: local('Georgia');  
  unicode-range: U+0410-044F;      /* диапазон кириллицы */  
}
```


Wakamai Fondue

Lato Bold

This is a **TrueType** font with **398 characters**. It has **5 layout features**.

Characters

398

Glyphs

399

Filename

Lato-Bold.woff2

Layout features

5

Languages

12

Size

49 KB

Format

WOFF2



[Wakamai fondue](#)

Fonttools



Fonttools



```
$ pyftsubset "Lato-Bold.ttf" --output-file="Lato-Bold-cy.woff2" --flavor=woff2 --layout-features="*"
--unicodes=some-unicodes-for-cyrillic
```



```
$ pyftsubset "Lato-Bold.ttf" --output-file="Lato-Bold-all.woff2" --flavor=woff2 --layout-features="*"
--unicodes=defaults-unicodes-for-you
```

Critical FOFT

Первая часть шрифта (кириллица, числа)



Целый шрифт или оставшаяся часть

Font-display

font-display: auto

font-display: block

font-display: swap

font-display: optional

font-display: fallback

Браузер	Время ожидания	Запасной шрифт	Замена шрифта
Chrome 35+	3 секунды	+	+
Opera	3 секунды	+	+
Firefox 3 секунды	+	+	
Internet Explorer	0 секунд	+	+
Safari	нет ожидания	-	-

'auto'

Использует поведение браузера по умолчанию. Обычно это поведение 'block'.

'block'

Устанавливает короткий период блокировки (3 секунды) и бесконечный период замены.

Если шрифт не загрузился, то браузер отрисовывает невидимый текст, но после загрузки заменяет его как можно раньше. Это поведение не подходит для иконочных шрифтов, ведь если шрифт не загрузится и пройдет 3 секунды, то пользователь увидит квадраты вместо иконок.

'swap'

Устанавливает период блокировки в 0 секунд и бесконечный период замены.

Браузер сразу отрисовывает текст безопасным шрифтом и пользователь не видит пустого экрана дожидаясь загрузки шрифта. Но когда шрифт загрузится, он сразу заменится.

'fallback'

Устанавливает очень короткий период блокировки (100 мс или меньше) и короткий период замены (3 секунды).

Работает так-же как и swap, но если шрифт не загрузился в течении 3-х секунд, то браузер отрисует текст запасным шрифтом. Подходит для текста предназначенного для удобного чтения. В таком случае шрифт играет второстепенную роль. Если он быстро не загрузился, то не стоит огорчать пользователя мерцанием от замены шрифта.

'optional'

Устанавливает очень короткий период блокировки (100 мс или меньше) и период замены в 0 секунды.

Похоже на 'fallback', но по истечению 3-х секунд, браузер может или прервать загрузку (при медленном интернете) или повысить приоритет загрузки. Когда браузер загрузит шрифт, он все равно продолжит отображать текст запасным шрифтом, но если вы обновите текущую или перейдете на другую страницу, для которой установлено значение 'optional' и используется тот-же шрифт, то браузер отрисует текст загруженным шрифтом.

Can I use font-display ~ 78%

@font-face descriptor (currently defined as font-display) that allows control over how a downloadable font renders before it is fully loaded.

Current aligned

Usage relative

Date relative

Apply filters

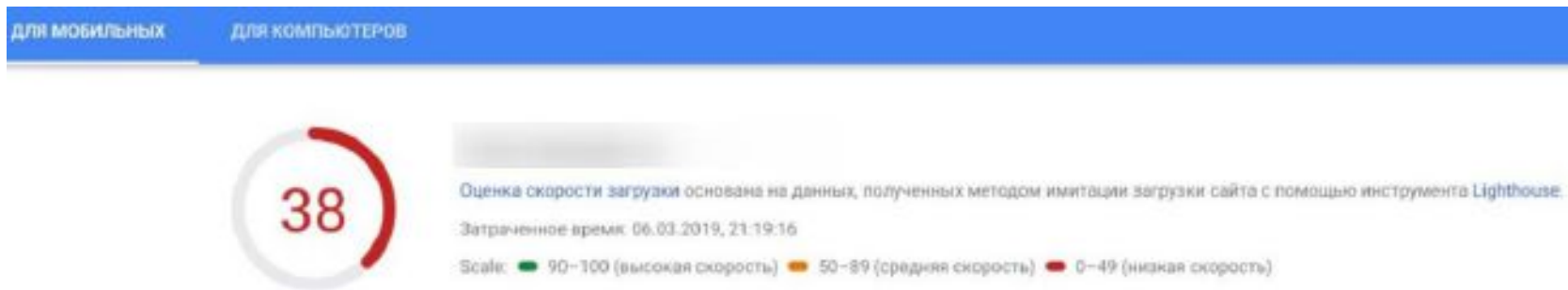
Show all

?



IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android
		2-45	4-48		10-35							
		46-57	49-59	3.1-11	36-46	3.2-11.2						
6-10	12-17	58-64	60-71	11.1	47-56	11.4		2.1-4.4.4	7	12-12.1		
11	18	65	72	12	57	12.1	all	67	10	46	71	64
		66-67	73-75	12.1-TP		12.2						

Критический FOFT preload + swap

Результаты оптимизации:



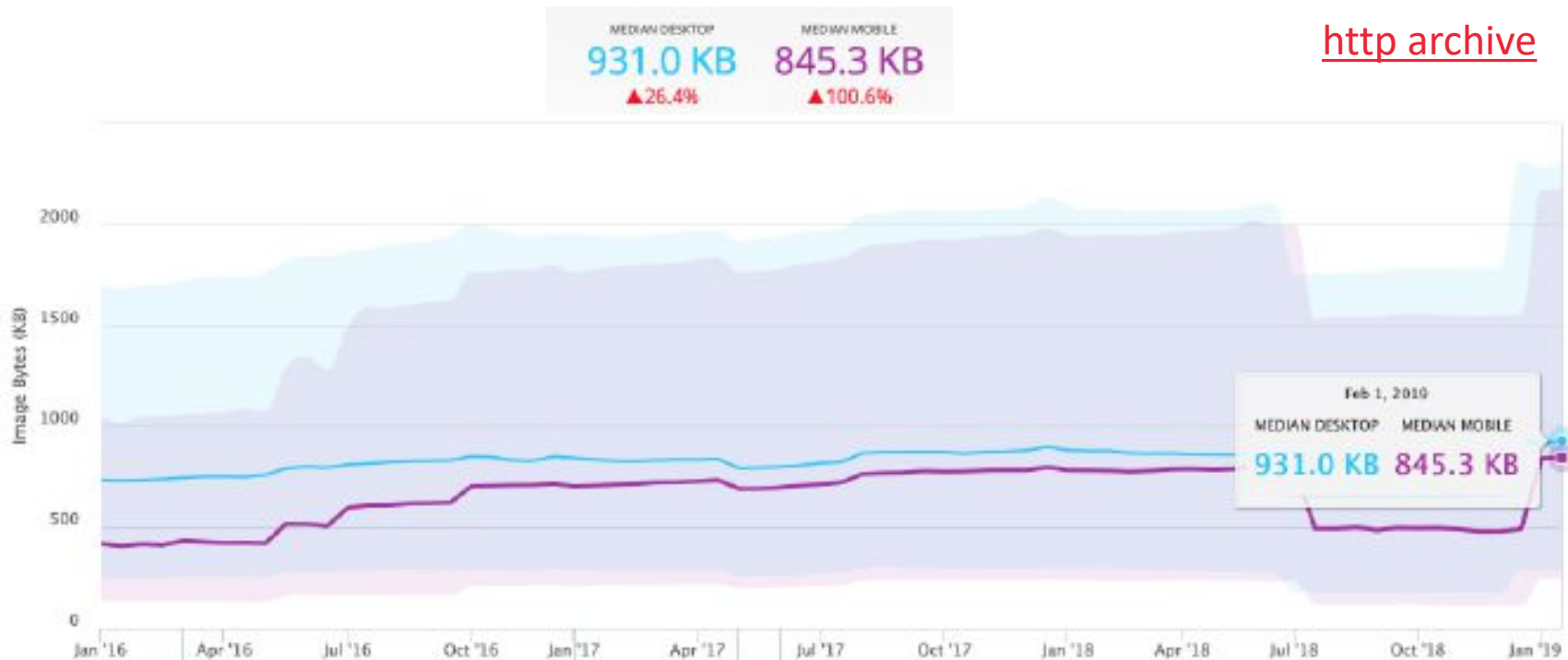
Lazy loading images

Возможности		Приблизительная экономия	
1	Отложите загрузку скрытых изображений	<div><div></div></div>	4,8 s ✓
3	Отложите загрузку скрытых изображений	<div><div></div></div>	0,3 s ^
Чтобы уменьшить время до начала взаимодействия, рекомендуем использовать принцип lazy loading для скрытых изображений после того, как все важные ресурсы будут загружены. Подробнее...			
URL		Размер (КБ)	Потенциальная экономия данных (КБ)
	...-bmU_EZFF2I/mqdefault.jpg (img.youtube.com)	20 КБ	20 КБ
	...VbburZ2qRXM/mqdefault.jpg (img.youtube.com)	20 КБ	20 КБ



[http archive](http://archive)

Средний объем изображений



Концепт lazy loading



Intersection Observer API

```
document.addEventListener("DOMContentLoaded", () => {
  let lazyloadImages;

  if ("IntersectionObserver" in window) {
    lazyloadImages = document.querySelectorAll("img.lazy");
    let imageObserver = new IntersectionObserver((entries, observer) => {
      entries.forEach(entry => {
        if (entry.isIntersecting) {
          var image = entry.target;
          image.src = image.dataset.src;
          image.classList.remove("lazy");
          imageObserver.unobserve(image);
        }
      });
    });

    lazyloadImages.forEach(function(image) {
      imageObserver.observe(image);
    });
  } else {
    // default method (1 img)
  }
})
```

Intersection Observer API

```
document.addEventListener("DOMContentLoaded", () => {
  const lazyloadImages = document.querySelectorAll("img.lazy");
  let lazyloadThrottleTimeout;

  function lazyload () {
    if (lazyloadThrottleTimeout) {
      clearTimeout(lazyloadThrottleTimeout);
    }

    lazyloadThrottleTimeout = setTimeout(() => {
      const scrollTop = window.pageYOffset;
      lazyloadImages.forEach(img => {
        if (img.offsetTop < (window.innerHeight + scrollTop)) {
          img.src = img.dataset.src;
          img.classList.remove('lazy');
        }
      });
      if (lazyloadImages.length == 0) {
        document.removeEventListener("scroll", lazyload);
        window.removeEventListener("resize", lazyload);
        window.removeEventListener("orientationchange", lazyload);
      }
    }, 200);
  }

  document.addEventListener("scroll", lazyload);
  window.addEventListener("resize", lazyload);
  window.addEventListener("orientationchange", lazyload);
});
```

Загрузка изображений CSS

```
document.addEventListener("DOMContentLoaded", () => {
  let lazyloadImages;

  if ("IntersectionObserver" in window) {
    lazyloadImages = document.querySelectorAll(".lazy");
    let imageObserver = new IntersectionObserver((entries, observer) => {
      entries.forEach(entry => {
        if (entry.isIntersecting) {
          let image = entry.target;
          image.classList.remove("lazy");
          imageObserver.unobserve(image);
        }
      });
    });

    lazyloadImages.forEach(image => {
      imageObserver.observe(image);
    });
  } else {
    // lazyload method (1 lag)
  }

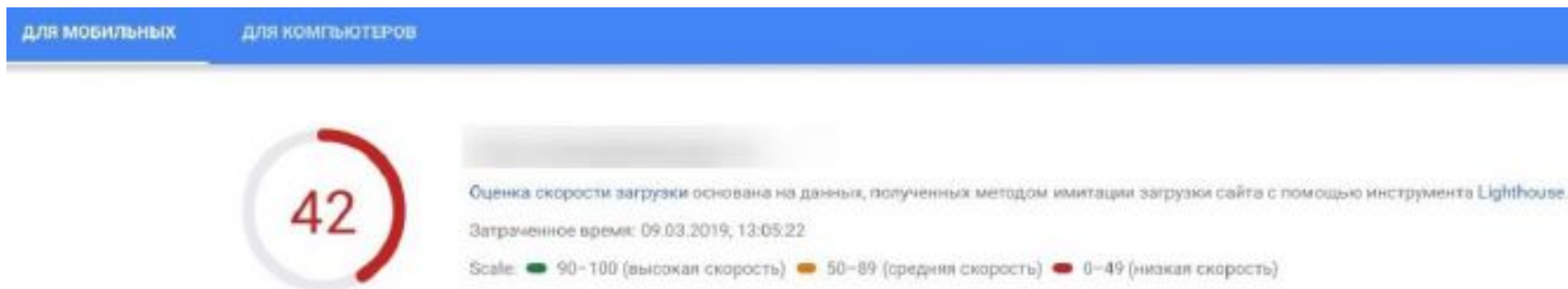
  // if not support
  // document.addEventListener("scroll", lazyload);
  // window.addEventListener("resize", lazyload);
  // window.addEventListener("orientationChange", lazyload);
})
```

Can I use Intersection Observer API ~ 73%

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android
			4-50									
	12-14	2-51	51-57		10-37							
	15	52-54	58-64		38-44							
6-10	16-17	55-64	65-71	3.1-11.1	45-56	3.2-11.4		2.1-4.4.4	7	12-12.1		
11	18	65	72	12	57	12.1	all	67	10	46	71	64
		66-67	73-75	12.1-TP		12.2						

Lazy loading изображений

Результаты оптимизации:

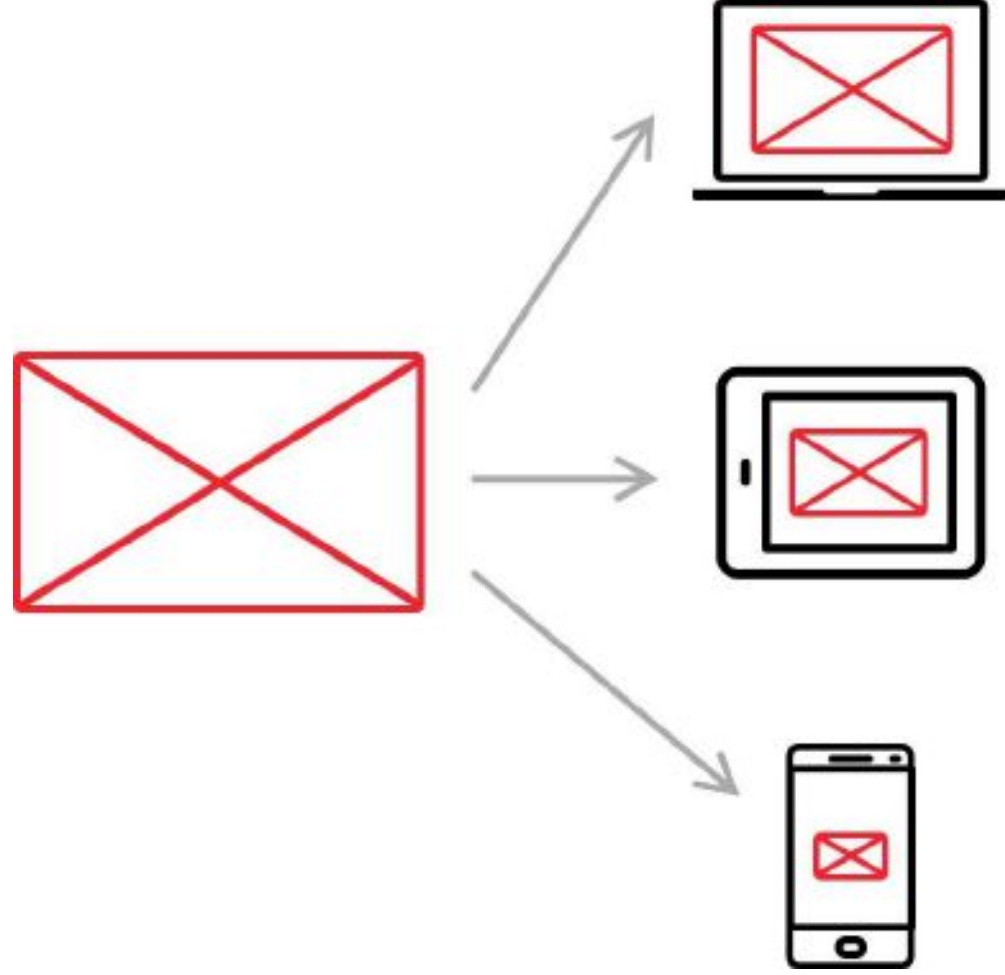


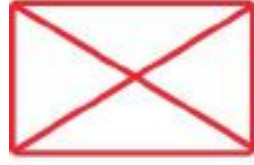
<picture>

2 Используйте современные форматы изображений

1,05 s ^

Для изображений в форматах JPEG 2000, JPEG XR и WebP используется более эффективное сжатие, поэтому они загружаются быстрее и потребляют меньше трафика, чем изображения PNG и JPEG. [Подробнее...](#)





<picture>

```
<picture class="wrapper__picture picture">
  <source type="image/webp" media="(min-width: 960px)" srcset="./img/panorama-desktop@1x.webp 1x,
    ./img/panorama-desktop@2x.webp 2x">
  <source type="image/webp" media="(min-width: 660px)" srcset="./img/panorama-tablet@1x.webp 1x,
    ./img/panorama-tablet@2x.webp 2x">
  <source type="image/webp" srcset="./img/panorama-mobile@1x-min.webp 1x, ./img/panorama-
    mobile@2x.webp 2x">
  <source media="{min-width: 960px}" srcset="./img/panorama-desktop@1x.jpg, ./img/panorama-
    desktop@2x.jpg 2x">
  <source media="{min-width: 660px}" srcset="./img/panorama-tablet@1x.jpg, ./img/panorama-
    tablet@2x.jpg 2x">
  
</picture>
```

← fallback

Can I use Picture ~ 88%

Current aligned Usage relative Date relative Apply filters Show all ?												
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android
		2-33	4-36		10-23							
	12	34-37	37	3.1-9	24	3.2-9.2						
6-10	13-17	38-64	38-71	9.1-11.1	25-56	9.3-11.4		2.1-4.4.4	7	12-12.1		
11	18	65	72	12	57	12.1	all	67	10	46	71	64
		66-67	73-75	12.1-TP		12.2						

WebP

Преимущества	Недостатки
Меньший размер файла	Плохая поддержка браузерами
Улучшенный алгоритм сжатия	Искажение имеет пластиковый вид
Более качественные переходы цвета	Неудобный интерфейс экспорта
Маска альфа-канала	



Can I use WebP ~ 75%

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android
			4-8		10.1							
			9-22		11.5			2.1-3				
			23-31		12.1-18			4-4.1				
6-10	12-17	2-64	32-71	3.1-11.1	19-56	3.2-11.4		4.2-4.4.4	7	12-12.1		
11	18	65	72	12	57	12.1	all	67	10	46	71	64
		65-67	73-75	12.1-TP		12.2						

Вставка видео с YouTube используя Iframe



[YouTube](#)

Все	HTML	CSS	JS	XHR	Шрифты	Изображения	Медиа
Статус	Метод	Домен	Фа...	Причина			
200	GET	www.youtub...	ww... script				
200	GET	www.youtub...	ba... script				
200	GET	www.google....	i6f... script				
200	GET	www.youtub...	ca... script				
200	GET	www.youtub...	en... script				
200	GET	www.youtub...	re... script				

Принцип работы



[GitHub](#)

```
<div class="video">
  <a class="video__link" href="https://youtu.be/{video_ID}">
    <picture>
      <source
        srcset="https://i.ytimg.com/vi_webp/{video_ID}/maxresdefault.webp"
        type="image/webp"
      />
      
    </picture>
  </a>
  <button class="video__button" type="button" aria-label="Запустить видео">
    <svg width="68" height="48" viewBox="0 0 68 48">
      <path
        class="video__button-shape"
        d="M66.52,7.74c-0.78-2.93-2.49-5.41-5.42-6.19C55.79,.13,34,0,34,0S12.21,.13,6.9,1.55 C3.97,2.33,2.
      ></path>
      <path class="video__button-icon" d="M 45,24 27,14 27,34"></path>
    </svg>
  </button>
</div>
```


Вставка видео с YouTube

Результаты оптимизации:

ДЛЯ МОБИЛЬНЫХ

ДЛЯ КОМПЬЮТЕРОВ



Оценка скорости загрузки основана на данных, полученных методом имитации загрузки сайта с помощью инструмента Lighthouse

Затраченное время: 10.03.2019, 14:58:23

Scale: ■ 90–100 (высокая скорость) ■ 50–89 (средняя скорость) ■ 0–49 (низкая скорость)

Render-blocking resources

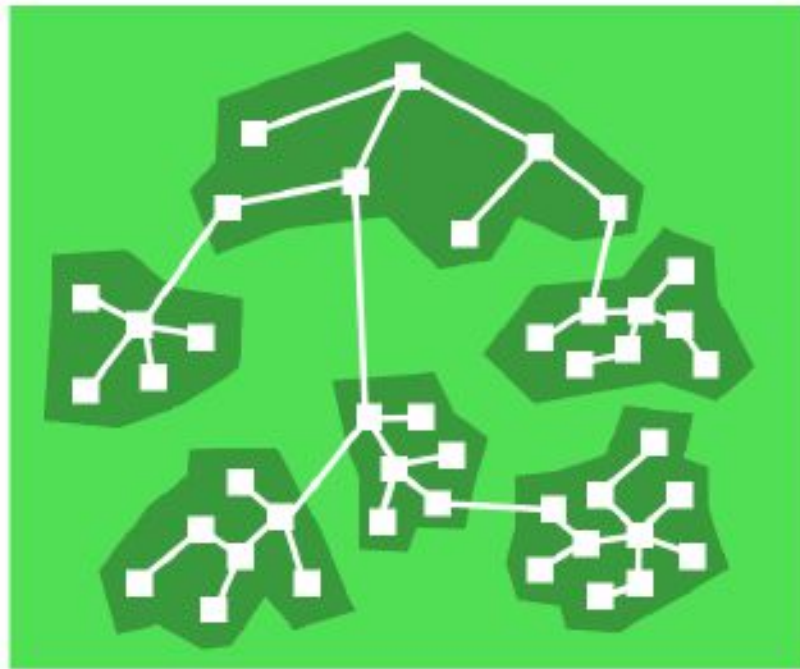
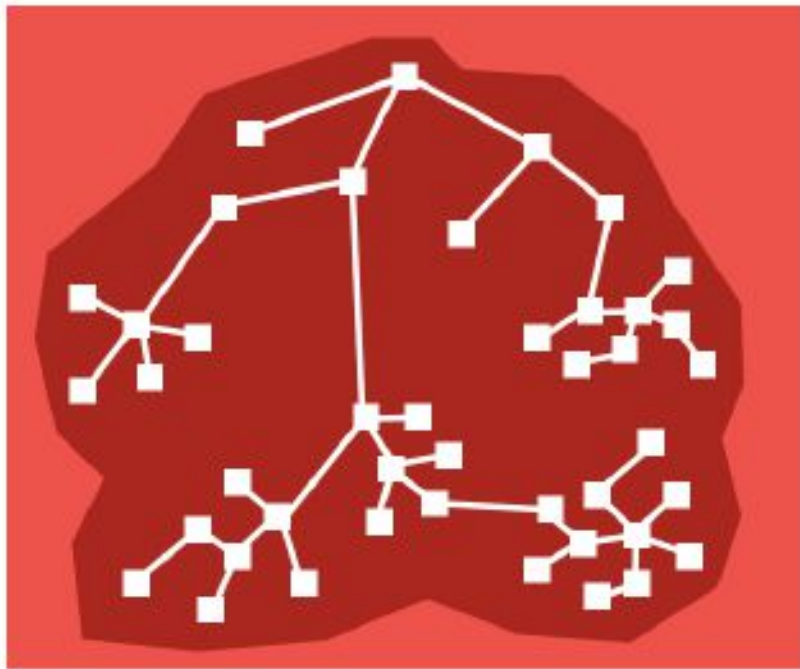
▲ Сократите время выполнения кода JavaScript — 4,2 сек.



Рекомендуем сократить время на анализ, компиляцию и выполнение скриптов JS. Для этого вы можете уменьшить размер фрагментов кода JS. [Подробнее...](#)

URL	Общее процессорное время	Время оценки скриптов	Время анализа скриптов
/metrika/tag.js (mc.yandex.ru)	2 100 ms	1 986 ms	42 ms
Other	1 382 ms	418 ms	6 ms

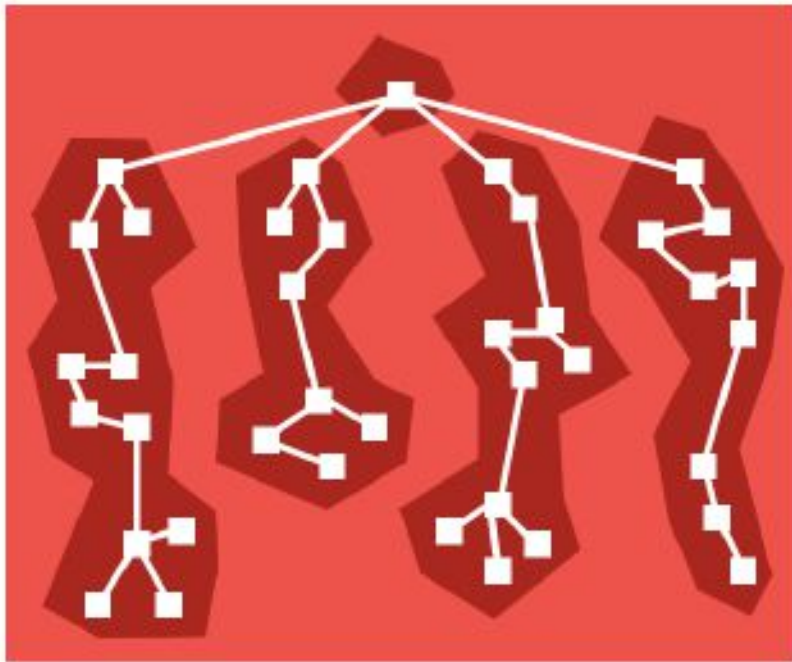
Code splitting



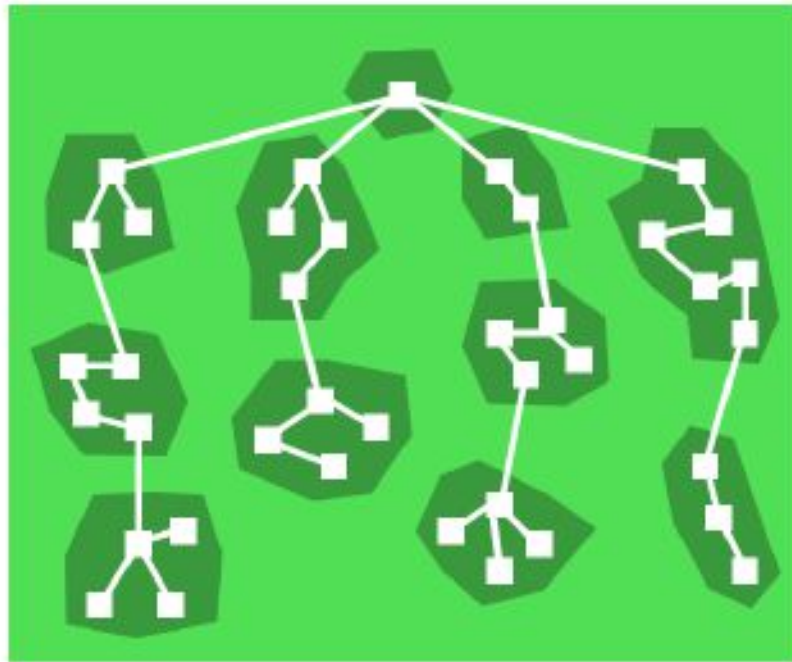


FoamTree

Роут-ориентированный подход



Компонент-ориентированный подход



```
class MyComponent extends React.Component {  
  state = {  
    AnotherComponent: null  
  };  
  
  componentWillMount() {  
    import('./another-component').then(AnotherComponent => {  
      this.setState({ AnotherComponent });  
    });  
  }  
  
  render() {  
    let {AnotherComponent} = this.state;  
    if (!AnotherComponent) {  
      return <div>Loading...</div>;  
    } else {  
      return <AnotherComponent/>;  
    }  
  }  
}
```




```
import AnotherComponent from './another-component';

class MyComponent extends React.Component {
  render() {
    return <AnotherComponent/>;
  }
}
```

React loadable

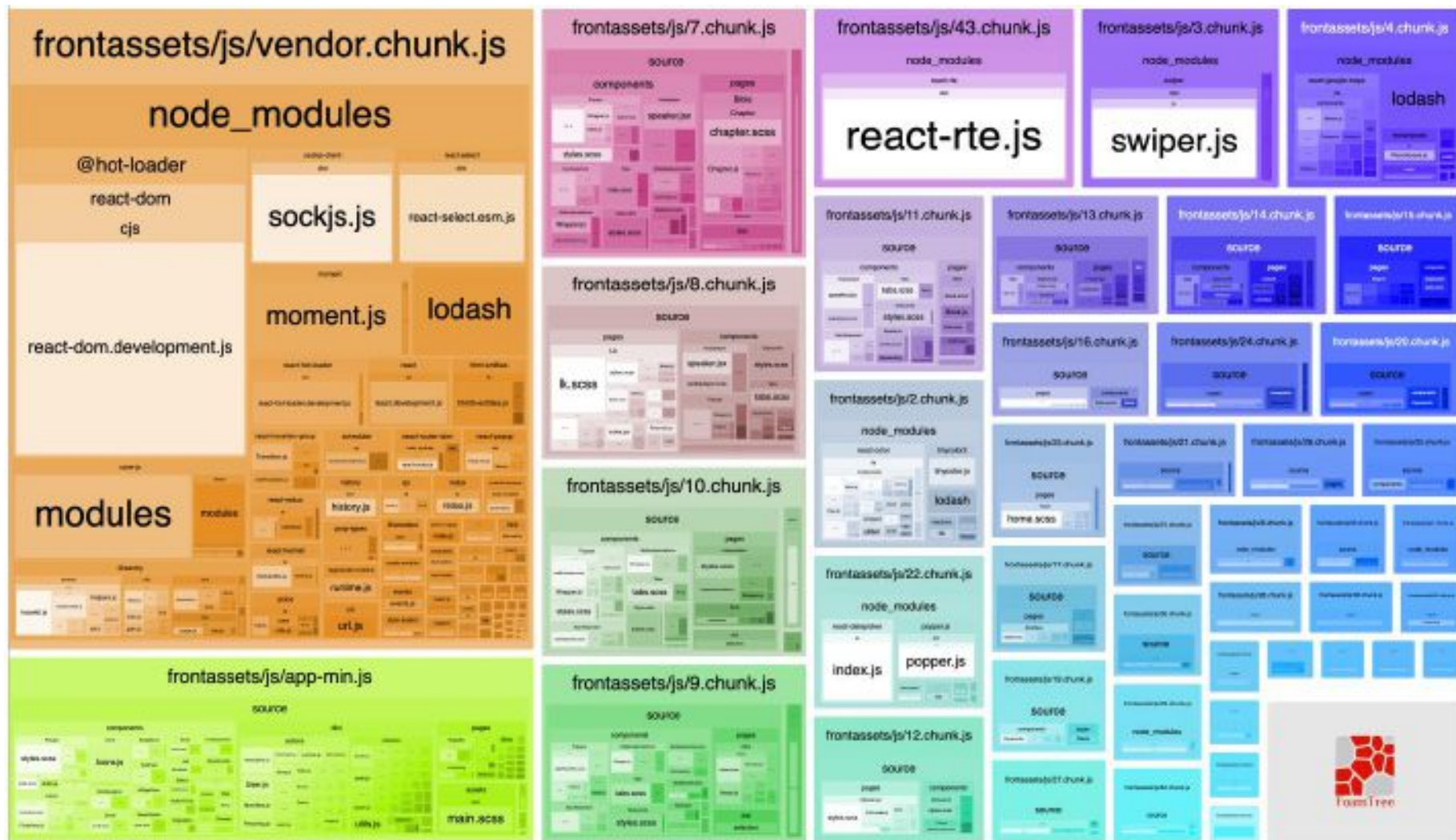
Route-centric code splitting, component-centric splitting is cool as SHIT

– Ken Wheeler... probably

```
import Loadable from 'react-loadable';
import Loading from './my-loading-component';

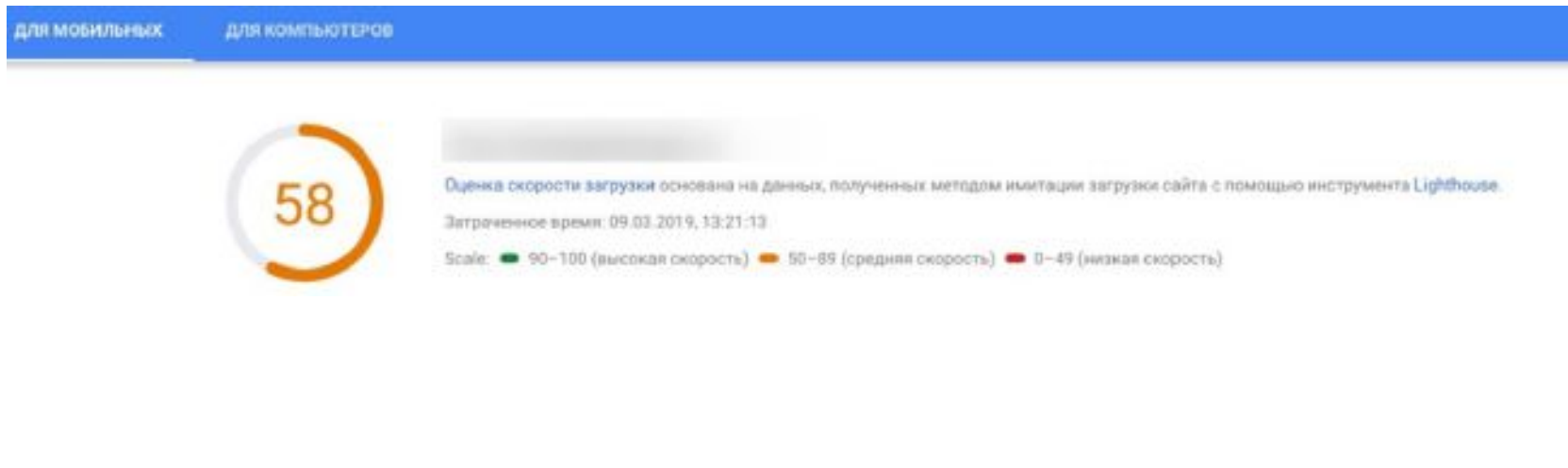
const LoadableComponent = Loadable({
  loader: () => import('./my-component'),
  loading: Loading,
});

export default class App extends React.Component {
  render() {
    return <LoadableComponent/>;
  }
}
```



Code splitting + Lazy loading

Результаты оптимизации:





Оценка скорости загрузки основана на данных, полученных методом имитации загрузки сайта с помощью инструмента [Lighthouse](#).

Затраченное время: 17.02.2019, 14:37:20

Scale: ● 90–100 (высокая скорость) ● 50–89 (средняя скорость) ● 0–49 (низкая скорость)



для мобильных

для компьютеров



Оценка скорости загрузки основана на данных, полученных методом имитации загрузки сайта с помощью инструмента [Lighthouse](#).

Затраченное время: 04.03.2019, 15:53:31

Scale: ■ 90–100 (высокая скорость) ■ 50–89 (средняя скорость) ■ 0–49 (низкая скорость)

для мобильных

для компьютеров



Оценка скорости загрузки основана на данных, полученных методом имитации загрузки сайта с помощью инструмента [Lighthouse](#).

Затраченное время: 04.03.2019, 20:05:25

Scale: ■ 90–100 (высокая скорость) ■ 50–89 (средняя скорость) ■ 0–49 (низкая скорость)



Спасибо за внимание

Крохмаль Даниил

tchernyavsky.daniil@yandex.ru

| @Tchernyavsky