



**Семирульник Валентин**  
**@7rulnik**

**USE** TECH

**Альфа Банк**



Открытие и ведение счёта Кредиты Депозиты Эквайринг Стать партнёром ВЭД

# Счёт для бизнеса за 5 минут

полностью онлайн



## Бесплатно

открытие и обслуживание

## До 5% кэшбэк

за бизнес-расходы по карте

## Эквайринг 1%

терминалы бесплатно

Открыть счёт

Тарифы

Возможности

Как открыть счёт

## Прозрачные тарифы для бизнеса

### Все тарифы включают

0₽

- интернет-банк и мобильный банк
- платежи в бюджет и бизнес-клиентам Альфа-Банка

Frontend  
and  
Curse of Backward Compatibility

**ESNext → ES5**

а через месяц уже 2020 год

10

?

|

**Internet Explorer 11 will stop being supported by Microsoft in :**

**5 Years 10 Months 27 Days 08 Hours 53 Minutes 27 Seconds**



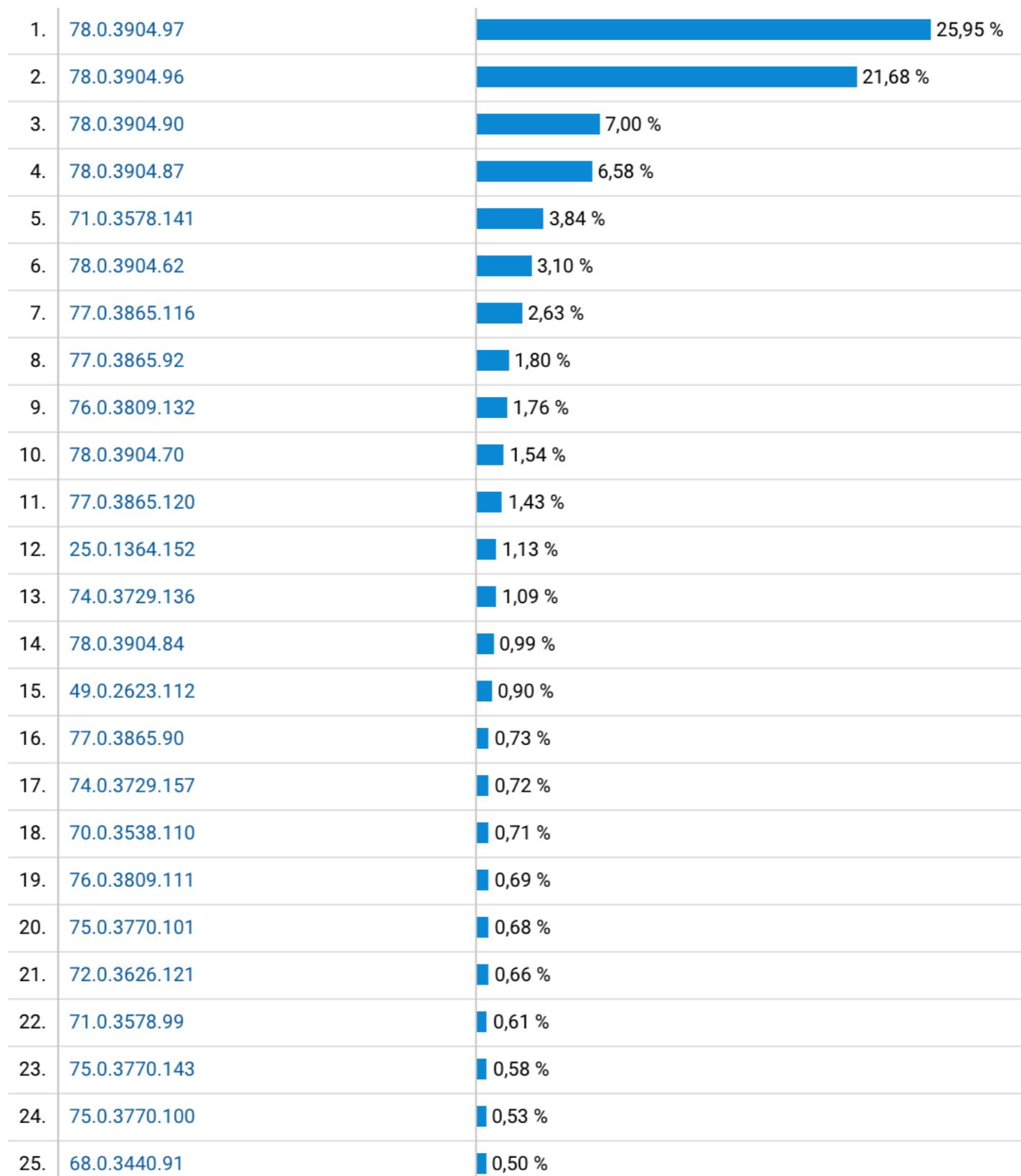
► Why you should stop supporting IE11

► Who stopped supporting IE11

**Всё ещё есть  
пользователи**

**Не только IE**

# Chrome





**Или по глупости**

**«Мы можем поддерживать 2 браузера:  
Chrome и IE?»**

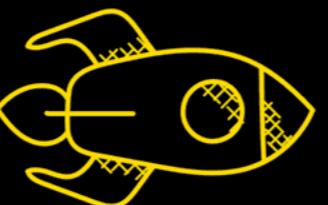
**Или просто  
из-за привычки**

А есть ли  
проблема?

# Web Performance 101

Ivan Akulov

@iamakulov



[3perf.com/talks/web-perf-101](http://3perf.com/talks/web-perf-101)

# Performance?

```
1 class MyCoolComponent extends React.Component {  
2   render() {  
3     return <div>Hello world!</div>  
4   }  
5 }
```

101 Bytes

```
1 "use strict";  
2  
3 function _instanceof(left, right) { if (right != null && typeof Symbol  
!== "undefined" && right[Symbol.hasInstance]) { return  
!!right[Symbol.hasInstance](left); } else { return left instanceof  
right; } }  
4  
5 function _typeof(obj) { if (typeof Symbol === "function" && typeof  
Symbol.iterator === "symbol") { _typeof = function _typeof(obj) {  
return typeof obj; }; } else { _typeof = function _typeof(obj) { return  
obj && typeof Symbol === "function" && obj.constructor === Symbol &&  
obj !== Symbol.prototype ? "symbol" : typeof obj; }; } return  
_typeof(obj); }  
6  
7 function _classCallCheck(instance, Constructor) { if  
(!_instanceof(instance, Constructor)) { throw new TypeError("Cannot  
call a class as a function"); } }  
8  
9 function _defineProperties(target, props) { for (var i = 0; i <  
props.length; i++) { var descriptor = props[i]; descriptor.enumerable =  
descriptor.enumerable || false; descriptor.configurable = true; if  
("value" in descriptor) descriptor.writable = true;  
Object.defineProperty(target, descriptor.key, descriptor); } }  
10  
11 function _createClass(Constructor, protoProps, staticProps) { if  
(protoProps) _defineProperties(Constructor.prototype, protoProps); if  
(staticProps) _defineProperties(Constructor, staticProps); return  
Constructor; }  
12  
13 function _possibleConstructorReturn(self, call) { if (call &&  
(_typeof(call) === "object" || typeof call === "function")) { r2.8kBn  
call; } return _assertThisInitialized(self); }
```

babeljs.io/repl

**Чем меньше, тем быстрее**  
загрузка и парсинг





OFF THE JACKS

Testing in Chrome 78.0.3904 / Mac OS X 10.15.0

	Test	Ops/sec
<b>class property babel</b>	<code>new classPropertyBabel().method()</code>	59,469,426 ±1.03% 25% slower
<b>class property</b>	<code>new ClassProperty().method()</code>	7,784,783 ±1.05% 90% slower
<b>bound method babel</b>	<code>new BoundMethodBabel().method()</code>	78,976,068 ±0.84% fastest
<b>bound method</b>	<code>new BoundMethod().method()</code>	78,407,236 ±6.88% 6% slower
<b>class property babel loose</b>	<code>new ClassPropertyLoose().method()</code>	63,959,552 ±0.98% 19% slower

[jsperf.com/class-properties-vs-bound-methods/2](https://jsperf.com/class-properties-vs-bound-methods/2)

Медленнее ≠ Медленно  
и когда-нибудь починят

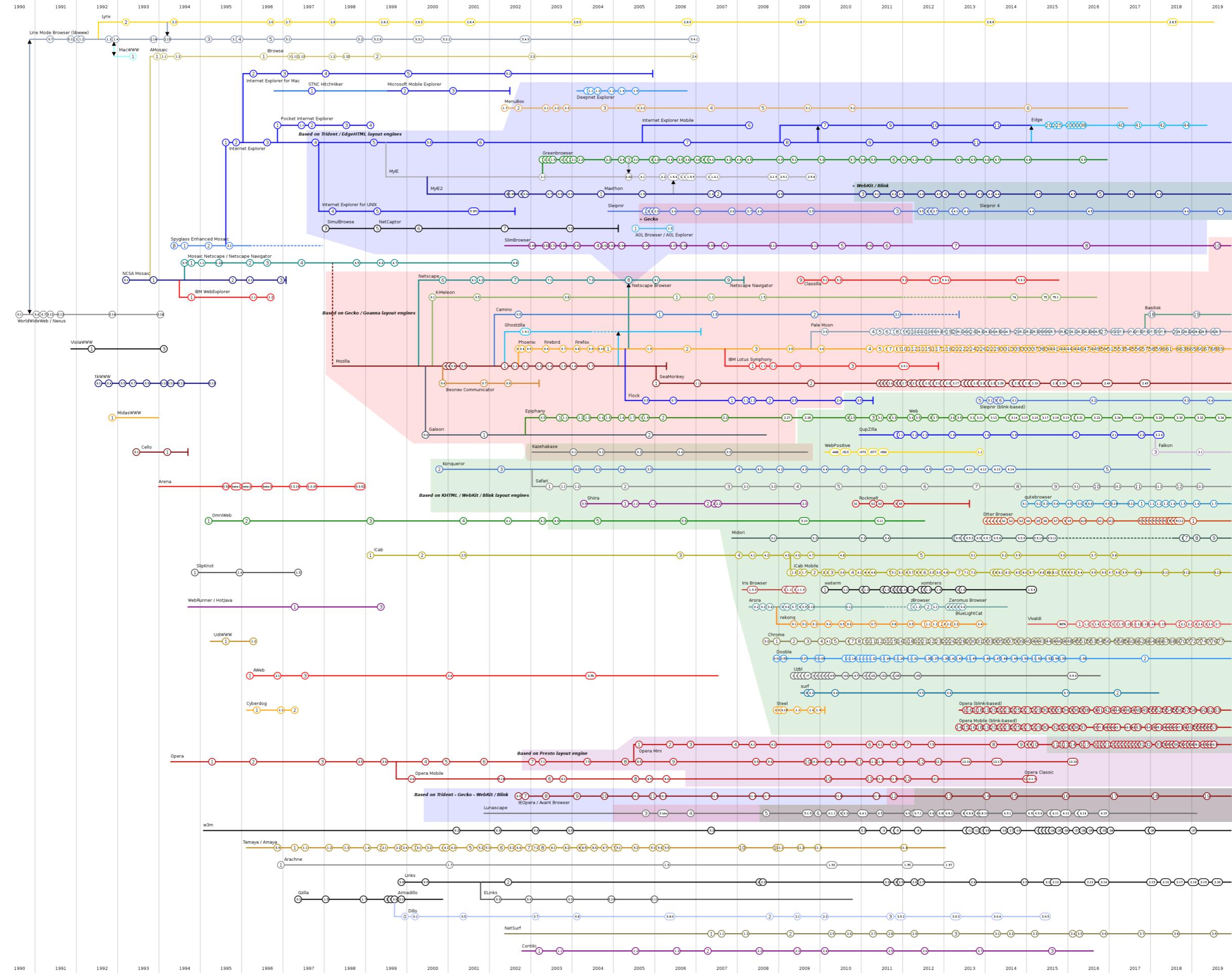
**Как же так вышло?**





**ES5 – это надёжно**

А сколько  
браузеров то?



# А сколько браузеров то?

- Chromium: Chrome, Opera, Edge  $\geq 76$
- Safari
- Firefox
- IE
- Edge

A close-up photograph of dense evergreen foliage. The leaves are small, sharp, and pointed, creating a textured, layered appearance. The color is a vibrant, dark green.

Evergreen

# Evergreen ли?



Браузеры работают по-разному  
И нам нужно их унифицировать



Как работают  
полифили?

```
1 [1, 2, 3].map((n) => n ** 3)
2
3
4 [1, 2, 3].map(function (n) {
5   return Math.pow(n, 3)
6 })
```

```
1 if (!String.prototype.includes) {  
2     String.prototype.includes = function(search, start) {  
3         'use strict';  
4         if (typeof start !== 'number') {  
5             start = 0  
6         }  
7  
8         if (start + search.length > this.length) {  
9             return false  
10        } else {  
11            return this.indexOf(search, start) !== -1  
12        }  
13    }  
14 }  
15  
16 'Valentin Semirulnik'.includes('Valentin')
```

А нужен ли мне  
полифил?

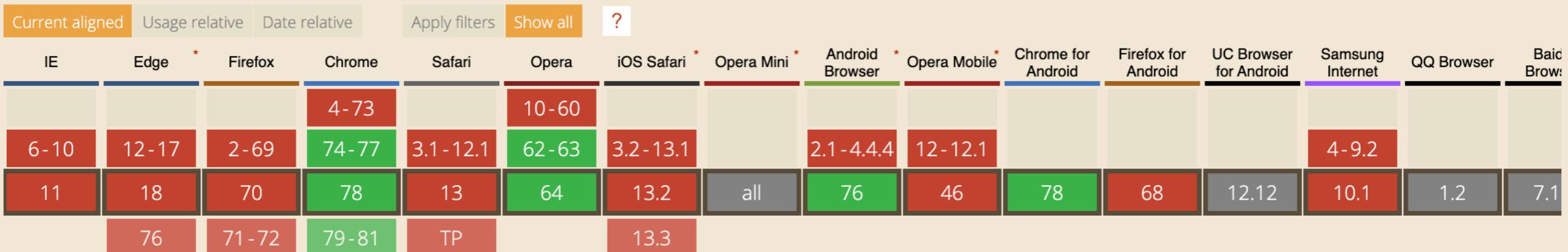
# Can I use private class fields

?  Settings

1 result found

## # JavaScript classes: Private class fields

Usage % of all users    
Global 64.05%



Notes Feedback

See full reference on [MDN Web Docs](#).

Support data for this feature provided by:

 [MDN browser-compat-data](#)

Sort by **Engine types**  Show obsolete platforms  Show unstable platforms

& webbedspace & zlojrock

781

**Какие инструменты  
используем?**

BABEL

`@babel/preset-env`

core-js



browserslist

last 1 version

> 1%

IE 10

Ts

# Варианты

1. Выбирать target либо lib в tsconfig.json самостоятельно: ES2015, ES2016.Array.Include, ESNext.
2. Компилировать в последний стандарт (например, target=ES2020) и после прогонять через @babel/preset-env
3. Компилировать через @babel/preset-typescript и @babel/preset-env



А как доставить код?

А мож ну её,  
эту обратную совместимость?

**Отказ от обратной  
совместимости это привилегия**

# 2 бандла

## Legacy и Modern

**Modern?**

# Native way

.mjs

# Can I use

# modules

?  Settings

3 results found

## # JavaScript modules via script tag - LS

Usage

Global

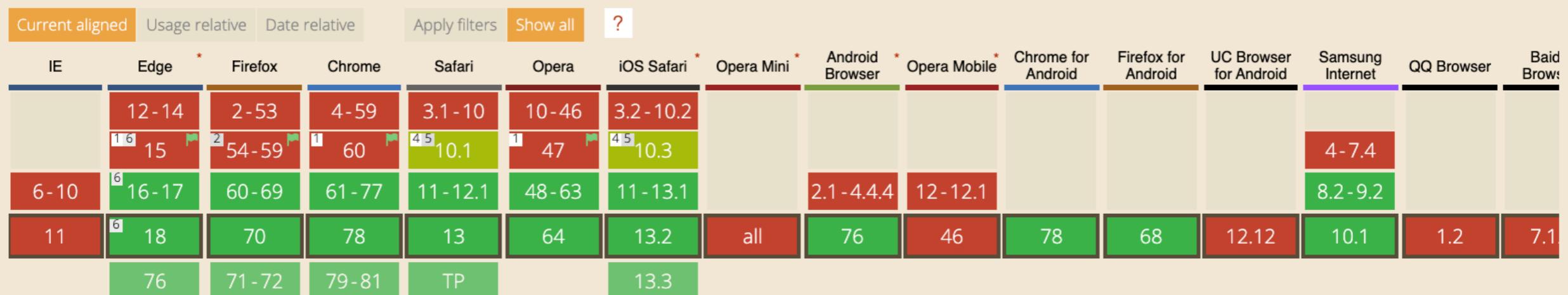
% of all users



87.83% + 0.27% = 88.1%

Loading JavaScript module scripts using `<script type="module">`

Includes support for the `nomodule` attribute.



Notes

Sub-features (1)

Known issues (2)

Resources (13)

Feedback

<sup>1</sup> Support can be enabled via `about:flags`

<sup>2</sup> Support can be enabled via `about:config`

<sup>4</sup> Does not support the `nomodule` attribute

<sup>5</sup> A [polyfill is available](#) for Safari 10.1/iOS Safari 10.3

<sup>6</sup> `nomodule` scripts are not executed, but they're still fetched

# .babelrc

```
1 {
2   "presets": [
3     [
4       "@babel/preset-env",
5       {
6         "targets": {
7           "esmodules": true
8         }
9       }
10      ]
11    ]
12 }
```

## &gt; SETTINGS

## &gt; PRESETS

## ✓ ENV PRESET

 Enabled

## BROWSERS

safari 10.1

## ELECTRON

1,8



## NODE

8,9



## BUILT-INS

Usage



## SPEC



## LOOSE



## SHIPPED PROPOSALS



## FORCE ALL TRANSFORMS



## &gt; PLUGINS

1 let foo = 2

1 "use strict";

2

3 var foo = 2;

```
<script type="module" src="modern.js"></script>  
  
<script nomodule src="legacy.js"></script>
```

# Проблемы

1. Скачивает и выполняет оба бандла:  
**Safari 10.1**
2. Скачивает оба бандла: **IE11, Firefox 59, Edge 15-16**
3. Скачивает старый и дважды новый  
бандлы: **Edge 18**

# **Стандарты**

# **Имплементация**

**Стандарт из 2015 устарел**  
а язык то развивается



# User Agent

```
// на клиенте  
navigator.userAgent  
  
// или на сервере  
req.headers['user-agent']
```

# User Agent

Mozilla/**5.0** (**Macintosh; Intel Mac OS X 10\_15\_0**) **AppleWebKit/537.36** (**KHTML, like Gecko**) **Chrome/78.0.3904.70** **Safari/537.36**

Mozilla/[version] ([system and browser information]) [platform] ([platform details]) [extensions]

# Let's parse

Как выглядит  
среднестатистическая  
библиотека для парсинга?

~~отвратительно~~

**browserslist-useragent-regexp**

/((CPU[ +]OS|iPhone[ +]OS|CPU[ +]iPhone|  
CPU IPhone OS)[ +]+(11[\_\.](3|4)|12[\_\.]  
(0|1))(?:\_\.\d+)?|(OperaMini(?:\vatt)?  
\v?(?\d+)?(?:\_\.\d+)?(?:\_\.\d+)?|(Opera\v.  
+Opera Mobi.+Version\v46\.0)|(Opera\v.  
46\.0.+Opera Mobi)|(Opera Mobi.+Opera(?:  
\v|\s+)46\.0)|(SamsungBrowser\v(8|9)\.2)|  
(Edge\v(17|18)(?:\_\.\d+)?|(HeadlessChrome(?:  
\v(72|73)\.0\.\d+)?|((Chromium|Chrome)\v  
(72|73)\.0(?:\_\.\d+)?)|  
(IEMobile[\v]11\.0)|(Version\v12\.(0|1)  
(?:\_\.\d+)?.\*Safari\v)|(Trident\v7\.0)|  
(Firefox\v(65|66)\.0\.\d+)|(Firefox\v(65|  
66)\.0(pre|[ab]\d+[a-z]\*))|((( [MS]?IE)  
11\.0)/

# На сервере

```
1 import path from 'path'
2 import { ChunkExtractor } from '@loadable/server'
3 import { getUserAgentRegExp } from 'browserslist-useragent-regexp'
4
5 const statsFileLegacy = path.resolve('../dist/loadable-stats-legacy.json')
6 const statsFileModern = path.resolve('../dist/loadable-stats-modern.json')
7
8 const browsers = ['Chrome >= 76', 'Safari >= 12.1', 'Firefox >= 68']
9 const regex = getUserAgentRegExp({
10   browsers,
11   allowHigherVersions: true
12 });
13
14 const isModernBrowser = regex.test(req.headers['user-agent'])
15 const statsFile = isModernBrowser ? statsFileModern : statsFileLegacy
16
17 const extractor = new ChunkExtractor({ statsFile })
18
19 const jsx = extractor.collectChunks(<YourApp />)
20 const html = renderToString.jsx)
21 const scriptTags = extractor.getScriptTags()
22 const html = `<html>
23   <body>
24     <div id="root">${html}</div>
25     ${scriptTags}
26   </body>
27 </html>`
```

# В браузере

```
1 import fs from 'fs';
2 import { getUserAgentRegExp } from 'browserslist-useragent-regexp';
3
4 import statsFileLegacy from '../dist/loadable-stats-legacy.json'
5 import statsFileModern from '../dist/loadable-stats-modern.json'
6
7 const browsers = ['Chrome >= 76', 'Safari >= 12.1', 'Firefox >= 68',]
8 const regex = getUserAgentRegExp({
9   browsers,
10  allowHigherVersions: true
11 });
12
13 function prepareJsAssets(stats) {
14   return stats.entrypoints.main.asset.filter(asset =>
15     asset.endsWith('.js')).map(asset => `${publicPath}/${asset}`
16   )
17 }
18
19 const legacyAssets = prepareJsAssets(statsFileLegacy)
20 const modernAssets = prepareJsAssets(statsFileModern)
21
22 const assets = {
23   legacyAssets,
24   modernAssets
25 }
26
27 const html = `<html>
28   <body>
29     <div id="root"></div>
30     <script>
31       var assets = ${JSON.stringify(assets)}
32       var isModernBrowser = ${regex}.test(navigator.userAgent)
33       var assetsForInjection = isModernBrowser ? assets.modernAssets :
34         assets.legacyAssets
35       assetsForInjection.forEach(function (asset) {
36         var tag = document.createElement('script');
37         tag.src = asset;
38         tag.defer = true;
39         document.body.appendChild(tag)
40       })
41     </script>
42   </body>
43 </html>`;
44
45 fs.writeFileSync('index.html', html)
```

```
1 import fs from 'fs';
2 import { getUserAgentRegExp } from 'browserslist-useragent-regexp';
3
4 import statsFileLegacy from '../dist/loadable-stats-legacy.json'
5 import statsFileModern from '../dist/loadable-stats-modern.json'
6
7 const browsers = ['Chrome >= 76', 'Safari >= 12.1', 'Firefox >= 68', ]
8 const regex = getUserAgentRegExp({
9     browsers,
10    allowHigherVersions: true
11 });
12
13 function prepareJsAssets(stats) {
14     return stats.entrypoints.main.asset
15         .filter(asset => asset.endsWith('.js'))
16         .map(asset => `${publicPath}/${asset}`)
17 }
18
19 const legacyAssets = prepareJsAssets(statsFileLegacy)
20 const modernAssets = prepareJsAssets(statsFileModern)
21
22 const assets = {
23     legacyAssets,
24     modernAssets
25 }
```

```
1 const html = `<html>
2   <body>
3     <div id="root"></div>
4     <script>
5       var assets = ${JSON.stringify(assets)}
6       var isModernBrowser = ${regex}.test(navigator.userAgent)
7       var assetsForInjection = isModernBrowser
8         ? assets.modernAssets
9         : assets.legacyAssets
10
11      assetsForInjection.forEach(function (asset) {
12        var tag = document.createElement('script');
13        tag.src = asset;
14        tag.defer = true;
15        document.body.appendChild(tag)
16      })
17    </script>
18  </body>
19 </html>`;
20
21 fs.writeFileSync('index.html', html)
```

**CSS  
IS  
AWESOME**

**Но доклад не об этом**

**Какие браузеры  
выбрать?**

**Изучайте свою аудиторию  
и решайте сами**

\* мы выбираем 70%

**На 7% меньше**

# Резюме

- Делайте 2 бандла: ES5 и Bleeding edge
- Используйте User agent и browserslist-useragent-regexr
- Переодически обновляйте browserslist для актуальных браузеров или используйте last n versions

# ЭПИЛОГ

- Когда-нибудь не evergreen браузеры умрут (IE 11)
- Всегда будут пользователи со старыми версиями браузеров
- Мы продолжим использовать babel (или похожий инструмент)

**А что дальше?**

# Динамические полифилы на основе User Agent

[polyfill.io](https://polyfill.io)

[github.com/zloirock/core-js/issues/638](https://github.com/zloirock/core-js/issues/638)

```
<script
  type="module"
  srcset="2018.mjs 2018, 2019.mjs 2019"
  src="2017.mjs"
>
```

[github.com/whatwg/html/issues/4432](https://github.com/whatwg/html/issues/4432)

Don't break the Web

# Вопросы?

Семирульник Валентин  
@7rulnik