



# Лекция 12

## Meta Frontend

Мартин Комитски



ЦРИТО  
Центр Развития  
ИТ-Образования



# План на сегодня

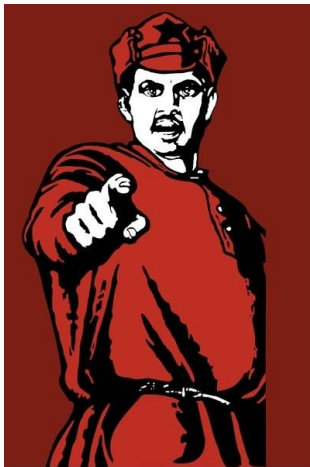


- Новые WEB API
- В чем нужно разбираться в 2020
- Где получать знания?
- Server-side
- Storage
- Desktop
- Mobile
- OS, TV

# Минутка бюрократии



- Внимание
- Отметки о посещении занятий
- Обратная связь о лекциях





# Развитие web- технологий

# Meta. Новые WEB API



- Payment Request API — cross-browser standard for Payment transaction process
- Web Bluetooth API — communications over GATT with nearby user-selected Bluetooth devices
- MediaRecorder API — MediaStream Recording
- WebRTC — Real-time Peer-to-peer Communications Between Browsers
- Credential Management API — store and retrieve user, federated , and public key credentials
- Feature Policy — mechanism to allow and deny the use of browser features
- Trusted-types API — allows applications to lock down DOM XSS injection sinks to only accept non-spoofable, typed values in place of strings
- Page Lifecycle API — supports browsers' ability to manage lifecycle of web pages

# Meta. Payment Request API



## Payment Request API - CR

Usage

% of

Global

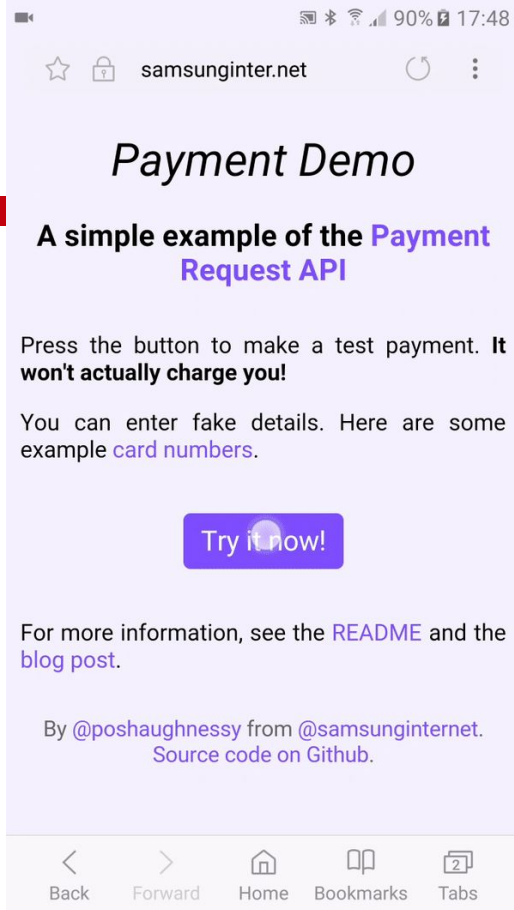
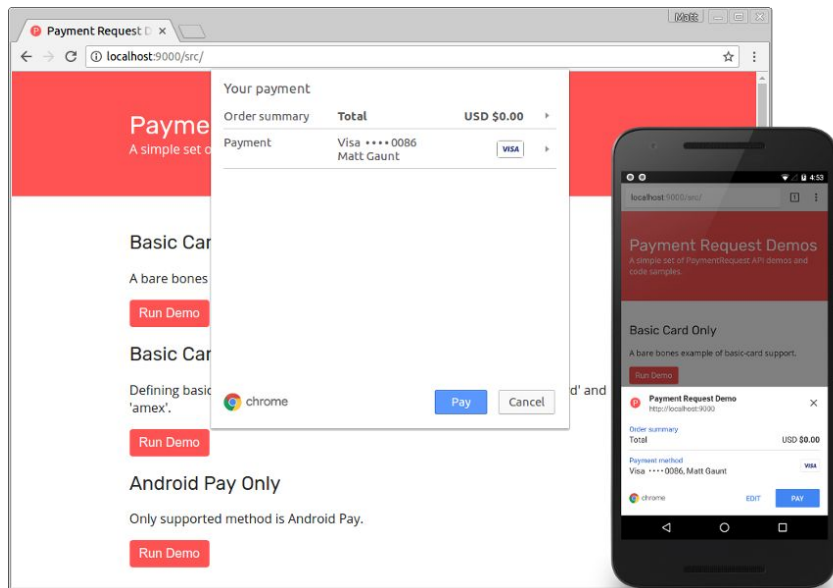
44.14% + 45.62% = 89.76%

Payment Request is a new API for the open web that makes checkout flows easier, faster and consistent on shopping sites.

Current aligned	Usage relative	Date relative	Apply filters	Show all	?								
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet
			4-52										
	12-13		1 53-58	3.1-9.1	10-39	3.2-9.3							
	2 14	2-54	4 59-60	3 10-11	1 40-47	3 10-11.2							
	7 15-18	6 55-65	7 61-77	7 11.1-12	7 48-65	7 11.3-12.1							4
6-10	79-80	8 66-74	78-80	12.1-13	66-67	12.2-13.3		2.1-4.4.4	12-12.1				7 5-10.1
11	81	3 75	81	13.1	68	13.4	all	81	46	5 81	68	12.12	7 11.1
		8 76-77	83-85	TP									

# Meta. Payment Request API

- Payment Request API — позволяет организовать оплату на сайте, используя нативный UI устройства, с указанием деталей карты, вариантами доставки, адресами и контактными данными — [developers.google.com](https://developers.google.com)



# Meta. Web Bluetooth API



- Web Bluetooth API — предоставляет методы взаимодействия браузера с удалёнными Bluetooth LE устройствами — [developers.google.com](https://developers.google.com/web/updates/2018/07/web-bluetooth)

## Web Bluetooth - UNOFF

Usage % of all users  ?  
Global 67.38%

Allows web sites to communicate over GATT with nearby user-selected Bluetooth devices in a secure and privacy-preserving way.





# Meta. MediaRecorder API



- MediaRecorder API — позволяет получать доступ к камере и микрофону компьютера через JavaScript API — [developers.google.com](https://developers.google.com)

## MediaRecorder API - WD

Usage % of all users  ?  
Global 72.4%

The MediaRecorder API (MediaStream Recording) aims to provide a really simple mechanism by which developers can record media streams from the user's input devices and instantly use them in web apps, rather than having to perform manual encoding operations on raw PCM data, etc.

Current aligned	Usage relative	Date relative	Apply filters	Show all	?										
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ	
			4-46		10-33										
	12-18	2-28	1 47-48	3.1-12	1 34-35	3.2-11.4							4		
6-10	79-80	29-74	49-80	2 12.1-13	36-67	3 12-13.3		2.1-4.4.4	12-12.1				5-10.1		
11	81	75	81	13.1	68	3 13.4	all	81	46	81	68	12.12	11.1		
		76-77	83-85	TP											



- WebRTC (RTCPeerConnection API) — API, предоставляемое браузером и позволяющее организовать P2P соединение и передачу данных напрямую между браузерами — [codelabs.developers.google.com](https://codelabs.developers.google.com)

## WebRTC Peer-to-peer connections - WD

Usage % of all users ?  
Global 90.32% + 2.09% = 92.41%  
unprefixed: 85.51% + 2.09% = 87.6%

Method of allowing two users to communicate directly, browser to browser using the RTCPeerConnection API.

Current aligned	Usage relative	Date relative	Apply filters	Show all	?										
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ	
	12-14	2-21	4-22		10-17										
	15-18	22-43	23-55	3.1-10.1	18-42	3.2-10.3									
6-10	79-80	44-74	56-80	11-13	43-67	11-13.3		2.1-4.4.4	12-12.1				4-10.1		
11	81	75	81	13.1	68	13.4	all	81	46	81	68	12.12	11.1		
		76-77	83-85	TP											

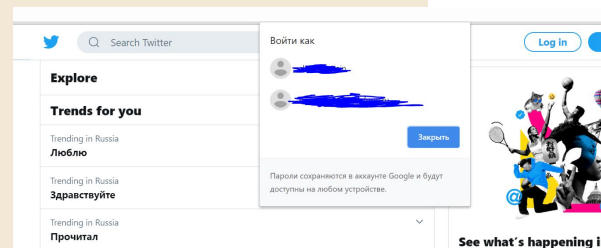
# Meta. Credential Management API



- Credential Management API — позволяет сохранять и получать данные для аутентификации — [developers.google.com](https://developers.google.com/credential-management/)

## Credential Management API 📄 - WD

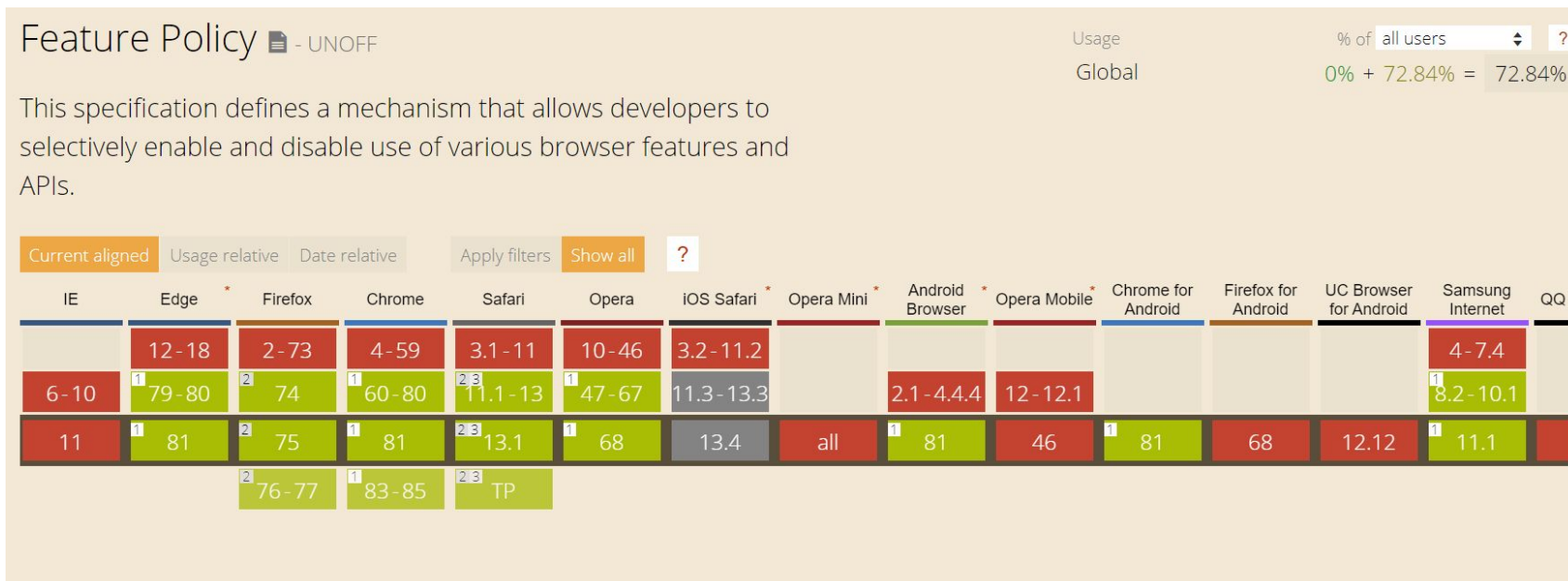
API that provides a programmatic interface to the browser's credential manager. In short, an origin can request a user's credentials to sign them in, or can ask the browser to save credentials on the user's behalf. Both of these requests are user-mediated.



# Meta. Feature-Policy



- HTTP-заголовок Feature-Policy — позволяет разрешать или запрещать браузеру выполнение определенных функций и API для улучшения приватности — [developers.google.com](https://developers.google.com)



# Meta. Feature-Policy



Feature-Policy: <feature> <allow list origin(s)>

Standardized Features:

accelerometer, ambient-light-sensor, autoplay, camera, document-domain,  
fullscreen, gyroscope, magnetometer, microphone, midi, payment,  
picture-in-picture, sync-xhr, usb, wake-lock, xr2

# Meta. Feature-Policy



Feature-Policy: geolocation 'none'

```
> navigator.geolocation.getCurrentPosition(position => {  
  console.log(position.coords.latitude, position.coords.longitude);  
}, err => {  
  console.error(err.message);  
});
```

<> undefined

⚠ ▶ Geolocation access has been blocked because of a Feature Policy applied to the current document. See <https://goo.gl/EuHzyy> for more details.

```
document.featurePolicy.allowedFeatures(); // список разрешенных фич  
// ["geolocation", "midi", "camera", "usb", "autoplay",...]
```

```
document.featurePolicy.allowsFeature('geolocation');  
// true
```

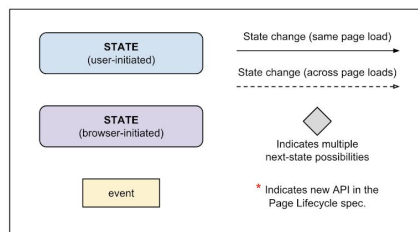
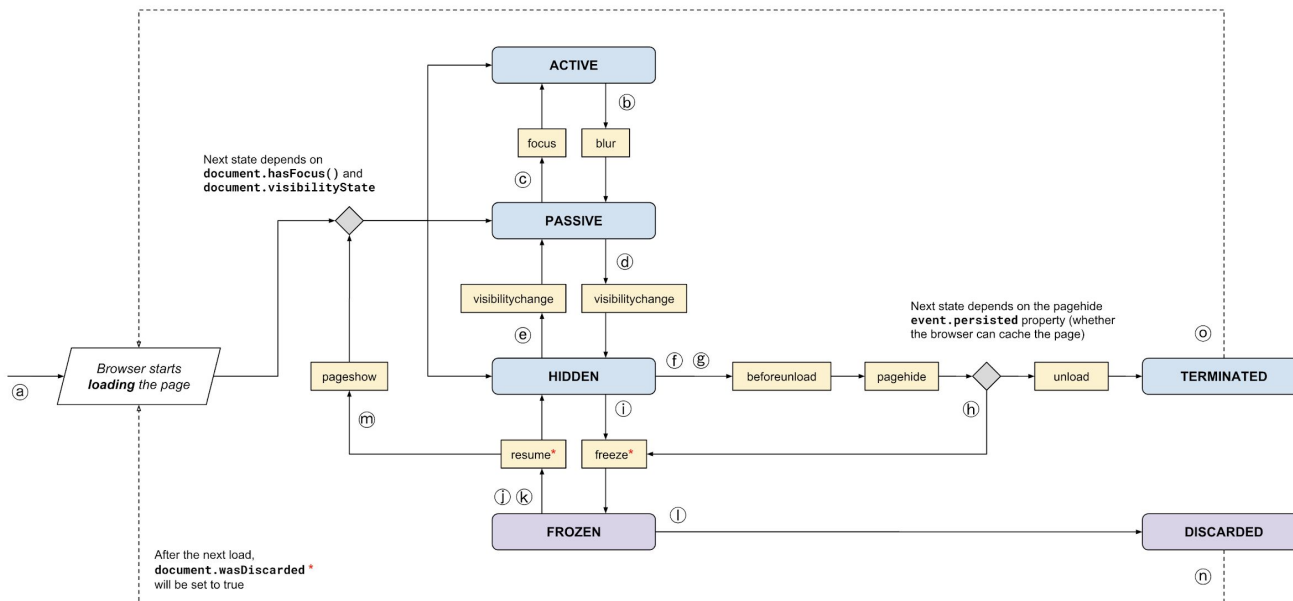
```
// список разрешенных фич, независимо от того, действуют ли они  
document.featurePolicy.features();  
// ["geolocation", "midi", "camera", "usb", "autoplay",...]
```

# Meta. Page Lifecycle API



- Page Lifecycle API — позволяет отслеживать жизненный цикл страницы и реагировать на его изменения— [developers.google.com](https://developers.google.com)

# Meta. Page Lifecycle API



- a User navigates to the new page
- b User focuses a page in another tab, window, or app
- c User re-focuses the page
- d User switches tabs away from the page
- e User switches tabs back to the page
- f User navigates to another page
- g User closes the tab, window, or app
- h System puts page into the page cache
- i System freezes the page to conserve CPU
- j System resumes the frozen page
- k User switches tabs back to the frozen page
- l System discards the page to conserve memory
- m User navigates back/forward to the cached page
- n User switches tabs back to the discarded page
- o User navigates back to a terminated page



# Meta. Page Lifecycle API



```
1. document.addEventListener('freeze', (event) => {
2.   // страница "заморожена"
3. });
4.
5. document.addEventListener('resume', (event) => {
6.   // страница "разморожена"
7. });
8.
9. // Типы событий:
10. // focus, blur, visibilitychange, freeze,
11. // resume, pageshow, pagehide, beforeunload, unload
12.
```

## Meta. Как находить новые API



- <https://developers.google.com/web/> - официальный сайт Google для web-разработчиков
- <https://github.com/tc39> - комитет, принимающий изменения в Ecma-262
- <https://wicg.io> - объединение, которое создает и публикует предложения по новым API



**В чем нужно  
разбираться в 2020**



- Знать про протоколы WEB'a: HTTP/1 (HTTP/1.1), HTTP/2, HTTP/3, HTTPS, WebSocket
- Знание JavaScript — причём, полезно знать как es6+, так и es5. TypeScript
- Знание возможностей браузеров (HTML5, CSS), уметь в кроссбраузерность, понимать, как браузеры работают внутри
- Вопросы, связанные с безопасностью web-приложений (что такое XSS, как работают cookie, что такое CORS...)
- Понимать вопросы, связанные с инфраструктурой, деплоем статики
- Иметь представление о UX
- Иметь представление о server-side
- Всё, что связано с программированием в целом
  - АиСД
  - ООП
  - ...

# Meta. Инструменты



- Инструменты разработки: Node.js
- Инструменты сборки: Webpack, grunt, gulp
- Инструменты работы с CSS: PostCSS
- Модульные системы: ES6-модули, AMD, UMD, commonjs
- DevTools браузеров
- Инструменты виртуализации, контейнеризации

# Meta. Библиотеки/фреймворки



- Библиотеки для сборки
  - require.js
  - SystemJS
- Библиотеки и Фреймворки
  - React
  - Angular
  - VueJS
  - Prototype
  - Svelte

## Meta. Новые подходы/концепции



- rx.js — реактивное программирование на js
- rambda.js — функциональное программирование на js



**Где получать знания?**



# Meta. Как развиваться?



- Много практики — нужно пробовать новое, развиваться вширь
- Читать статьи по тегам JavaScript, Frontend
  - Habrahabr
  - Medium
  - dev.to
- Посещать конференции:
  - Веб-стандарты
  - Фронтенд-субботники от Яндекса
  - HolyJS
  - MoscowJS
  - Другие более узкие митапы

# Meta. Как развиваться?



- Социальные сети
  - Каналы в tg ([netology.ru](https://netology.ru), [geekbrains](https://geekbrains))
  - Профили Instagram ([developerspace](https://developerspace), [richwebdeveloper](https://richwebdeveloper))
  - Паблики в VK ([forwebdev](https://forwebdev), [webstandards\\_ru](https://webstandards_ru), [webtackles](https://webtackles))
  - Профессиональные сообщества в LinkedIn
- Пользоваться ресурсами
  - StackOverflow
  - MDN
  - Подкасты
  - Новостные рассылки (<https://css-weekly.com/>, <https://ecmascript-daily.github.io/>)
  - Online-учебники ([frontend-handbook/2019](https://frontend-handbook/2019), [JavaScript-Garden](https://javascript-garden), [You-Dont-Know-JS](https://you-dont-know-js))
  - <https://stateofjs.com/>

# Meta. Как развиваться?



- Полноценное обучение
  - Дополнительные курсы и каналы на Youtube ([DevTips](#), [webDev](#))
  - Codeacademy
  - HTML Academy
  - javascript.ru
  - udemy
  - egghead.io



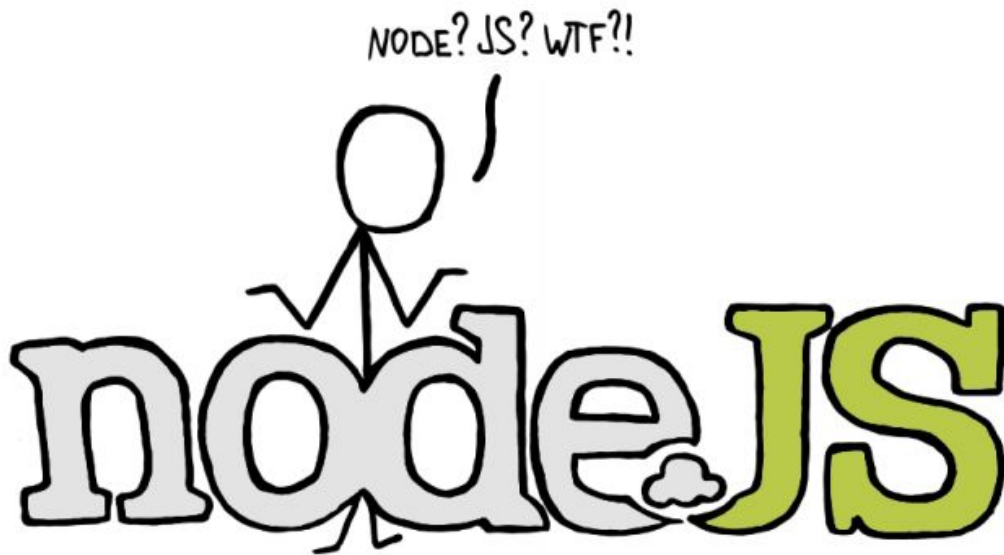
**Frontend-разработка  
— это не только web**

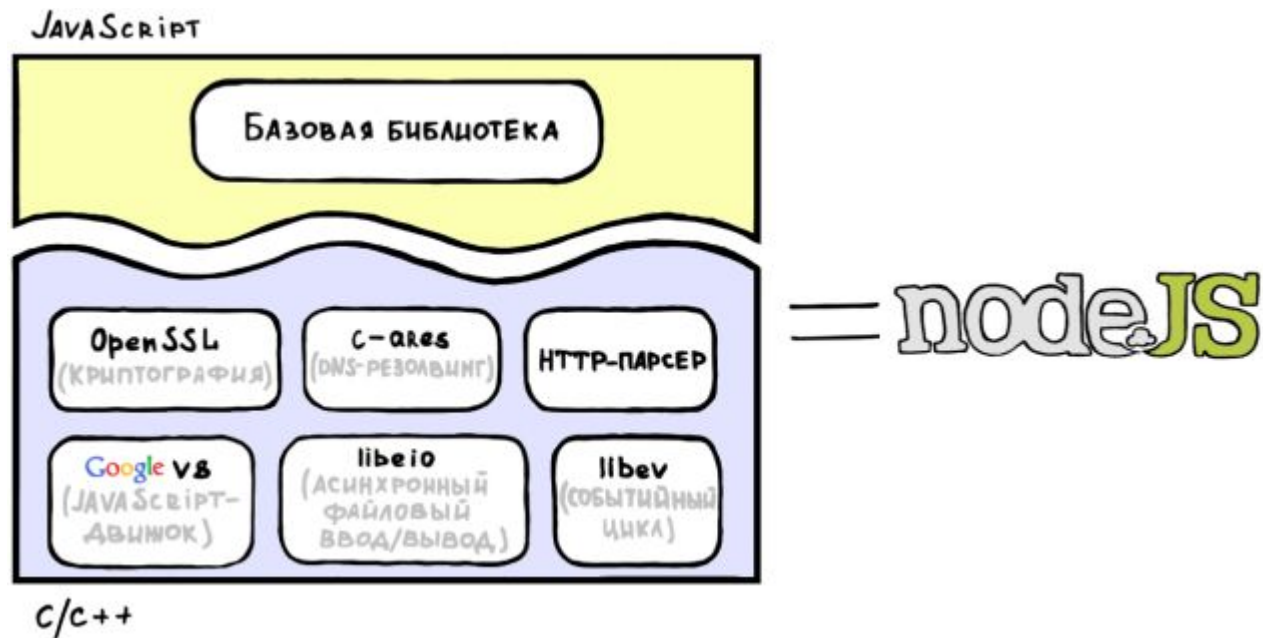


**Server-side**



- Node или Node.js — программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения







- Локальные и консольные утилиты
  - Сборщики, трансляторы, компиляторы
  - Скриптинг, CLI, генерация документации, тесты
- Серверы
  - API, dev-сервера, бекенды для SPA
  - Highload & real-time системы: игры, чаты
  - Заплаты в узкие места уже готовых систем
- Клиентские приложения
  - Краулеры, сборщики данных, анализаторы логов, статистики
  - Desktop приложения (nw.js, Electron)
- Железо
  - Работа на микроконтроллерах (Tessel, Espruino...)



# Meta. Node.JS

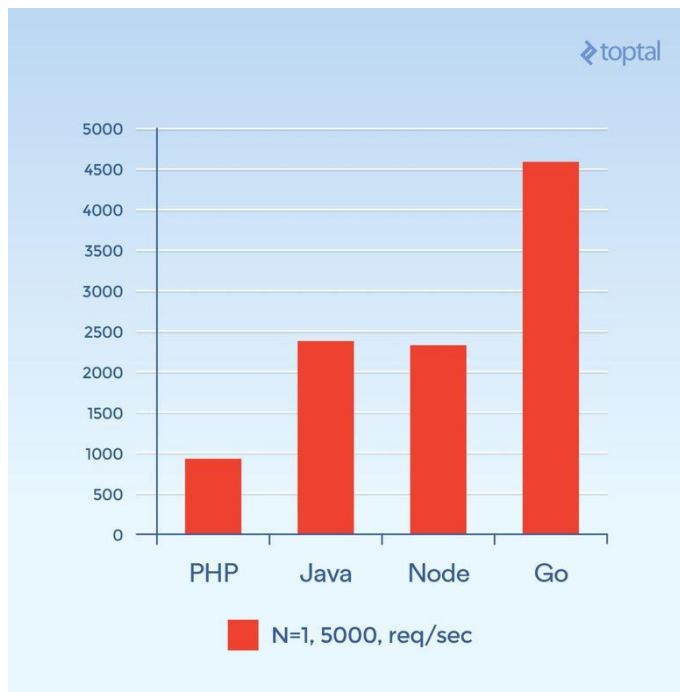


- <https://node.green/>

# Meta. Node.JS. Производительность



- <https://habr.com/ru/company/mailru/blog/329258/>





- `os` — информация о ОС, в которой запускается программа
- `child_process` — методы для запуска и управления дочерними процессами
- `assert` — набор простых методов тестирования
- `process` — информация о текущем процессе
- `net` (`dgram`) — создание tcp (и udp) серверов и клиентов
- `http` (`http2`, `https`) — модули для работы с http
- `fs` — работа с файловой системой
- `path` — методы для парсинга и работы с путями в file system
- `url` — методы для парсинга и работы с url-адресами
- `crypto` — модуль, предоставляющий криптографические методы
- `stream` — модуль для потоковой обработки файлов
- `cluster` — модуль для запуска многопроцессорных приложений



*npm* — менеджер пакетов, входящий в состав Node.js

- *express* — веб-сервер
- *koa* — веб-сервер
- *babel* — бабель
- *webpack* — вебпак
- *pm2* — production-демон для запуска node.js приложений
- *typescript* — компилятор typescript
- *debug* — модуль для логирования node.js приложений
- *socket.io* — библиотека для работы с веб-сокетами
- и многие другие...



**Storage**



```
{
  _id: <ObjectId>,
  username: "123xyz",
  contact: {
    phone: "123-456-7890",
    email: "xyz@example.com"
  },
  access: {
    level: 5,
    group: "dev"
  }
}
```

Embedded sub-document

Embedded sub-document



Collection



```
db.users.insert (  ← collection
{
  name: "sue",      ← field: value
  age: 26,          ← field: value
  status: "A"       ← field: value
}                  } document
)
```





**Desktop**

## Meta. Desktop

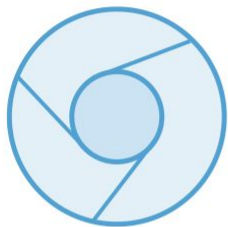


- Chromium — веб-браузер с открытым исходным кодом, разрабатываемый сообществом The Chromium Authors, компанией Google и некоторыми другими компаниями

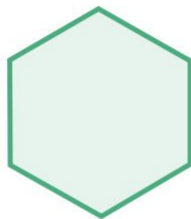
# Meta. Desktop



- NW.js
- Electron



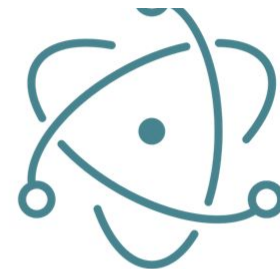
+



+



=



**Chromium**  
for making  
web pages

**Node.js**  
for filesystems  
and networks

**Native APIs**  
for three  
systems

**ELECTRON**

# Meta. Desktop



- Несмотря на то, что NW.js существует дольше Electron, на Electron разработано гораздо больше популярных приложений
- Atom
- Slack
- Visual Studio Code
- GitHub Desktop
- Hyper
- Discord
- Spotify



**Mobile**

# Meta. Mobile



- PWA
- TWA
- React Native
- Flutter
- Native Script
- ReactXP
- PhoneGap (Apache Cordova)
- и многие другие...

# Meta. Mobile



1<sup>st</sup>  
Gen

## Platform Proprietary



Objective-C / Swift



Android Java



Windows .NET

Native UX  
High performance  
Multi-platform  
Unified codebase  
Hardware & platform access

2<sup>nd</sup>  
Gen

## Hybrid HTML & JavaScript Frameworks



CORDOVA



Native UX  
High performance  
Multi-platform  
Unified codebase  
\*Hardware & platform access

3<sup>rd</sup>  
Gen

## Cross- Platform Native



Xamarin



React Native



NativeScript

Native UX  
High performance  
Multi-platform  
\*Unified codebase  
Hardware & platform access



**OS, TV**





- WebOS — веб-приложение, организующее платформу (операционную среду с набором готовых функций API) для выполнения других веб-приложений
- Tizen — открытая операционная система на базе ядра Linux, предназначенная для широкого круга устройств, включая смартфоны, интернет-планшеты, компьютеры, автомобильные информационно-развлекательные системы, «умные» телевизоры и цифровые камеры
- Chrome OS — ОС от компании Google. Главной особенностью является доминирование веб-приложений над обычными функциями ОС

# Вопросы



Вопросы?





“

**Перерыв! (10 минут)**

*Препоd (с)*



**Практика**

## Полезные ссылки



- Все было выше

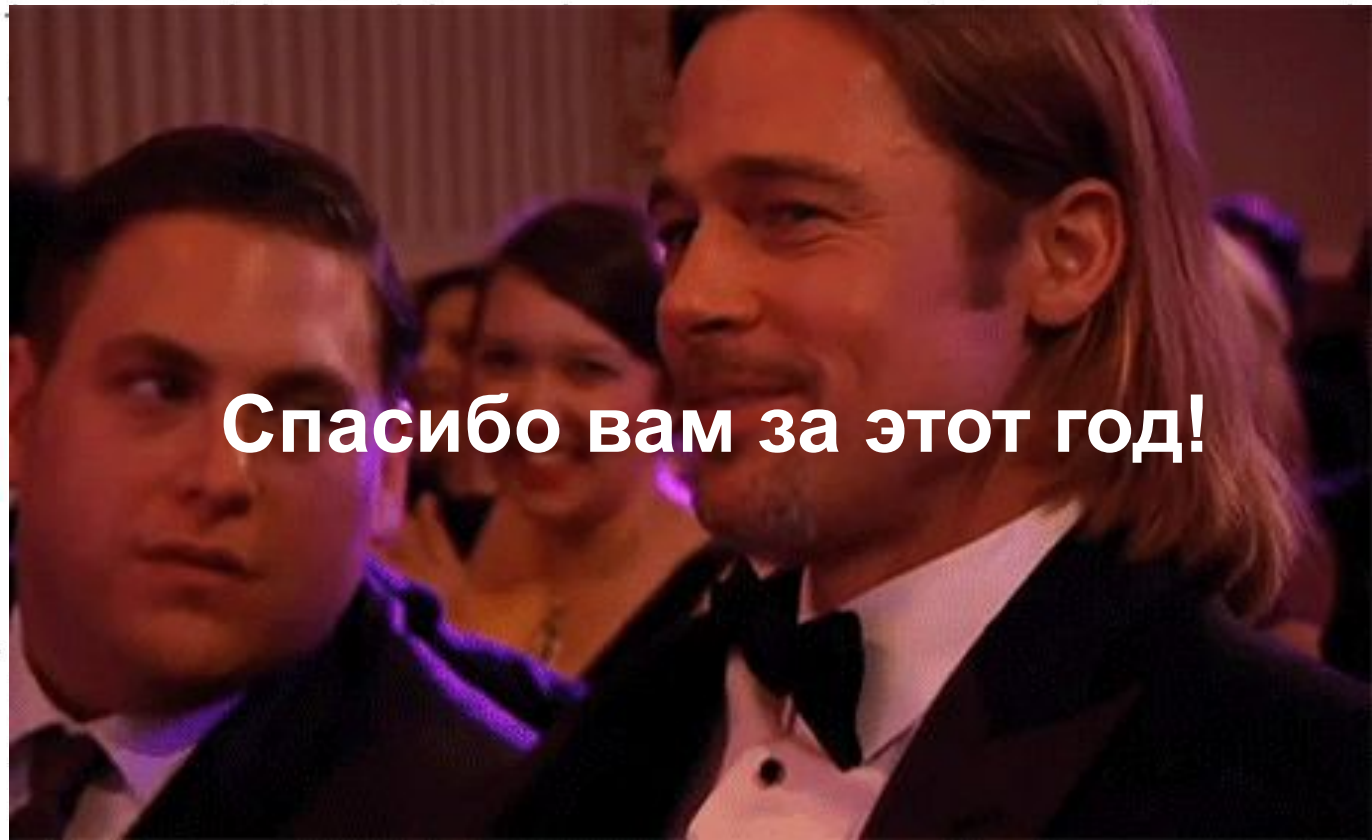
## Домашнее задание № 9



1. Подготовиться к экзамену
  - а. Теория за весь фронтенд
  - б. Практика React, typescript
2. Подумать о карьере frontend разработчика

**Срок сдачи**

*7 мая - Экзамен*



**Спасибо вам за этот год!**