

**Мне  
ничего тебе  
сказать!**

Алексей Охрименко (IPONWEB)

# Алексей Охрименко

Twitter: @Ai\_boy



# IPONWEB (RTB)



# 5 min Angular



Пятиминутка Angular

Пятиминутка Angular - #1 - Angular 5 Beta 3,  
Angular CLI 1.3, CreateAngularComponents

3 days

# Technology



**<https://meetup.tinkoff.ru>**

**Telegram:** [https://t.me/angular\\_ru](https://t.me/angular_ru)

**Facebook:** <https://www.facebook.com/groups/1818344448397380/>

**VK:** <https://vk.com/club149556741>

**Twitter:** <https://twitter.com/AngularMoscow>

**Meetup:** <https://www.meetup.com/AngularMoscow/>

# Angular RU - Github

## Максим Иванов



The screenshot shows the GitHub profile of the Angular-RU repository. The repository name is "Angular-RU - russian-speaking community". It has a red logo featuring a white stylized letter 'A' with a small Russian flag on its top right. Below the name, a description reads: "More useful information on the topics of Angular, TypeScript, NativeScript, Dart, etc." It includes location "Russia", website "https://angular.su", and a telegram link "t.me/angular\_ru". The navigation bar shows "Repositories 36" (highlighted in red), "People 33", "Teams 12", "Projects 3", and "Settings".

### Pinned repositories

Customize pinned repositories

#### [≡ angular-universal-starter](#)

A simple Angular Universal repo with Angular 6+

TypeScript ★ 137 ⚡ 53

#### [≡ angular-cli-webpack](#)

Webpack configuration modifier for @angular/cli

TypeScript ★ 42 ⚡ 6

#### [≡ change-detection-tree](#)

Visual detecting changes in the component tree

TypeScript ★ 21 ⚡ 5

#### [≡ angular-ru-interview-questions](#)

Вопросы на собеседовании по Angular

#### [≡ angular-quick-starter](#)

Учебные материалы на русском

#### [≡ angular-awesome-list](#)

Список ресурсов по Angular на русском

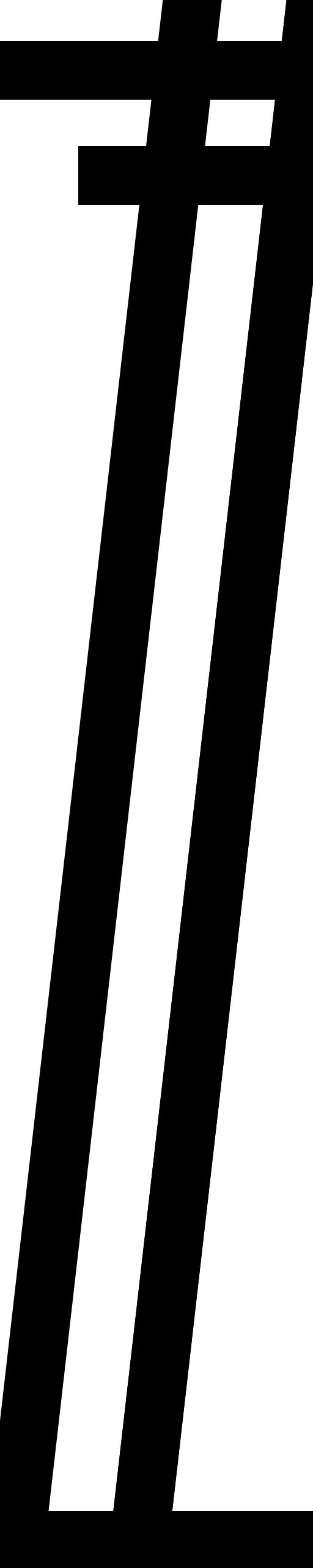
Angular Piter Telegram

Катерина Павленко, Александр  
Коротаев, Александр Духовняк

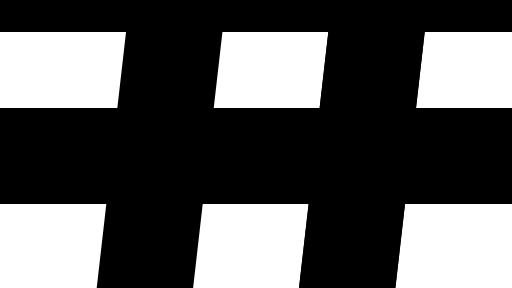
Angular Piter Meetup - Call For Speakers



**Давай на «ты»?**

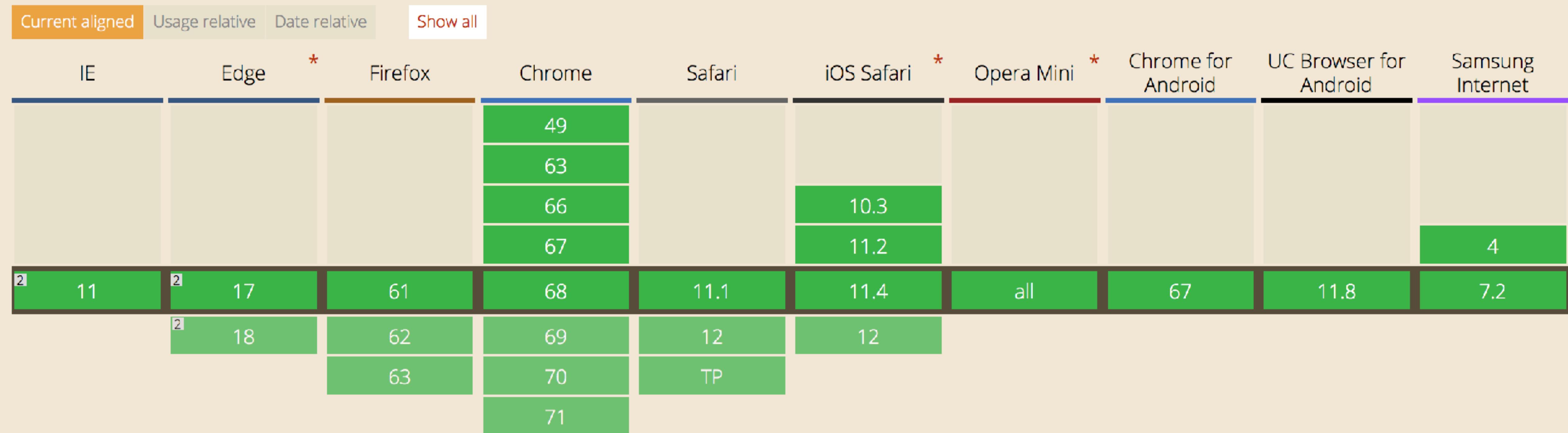


**Некоторые из  
существующих способов  
коммуникации с сайтами**

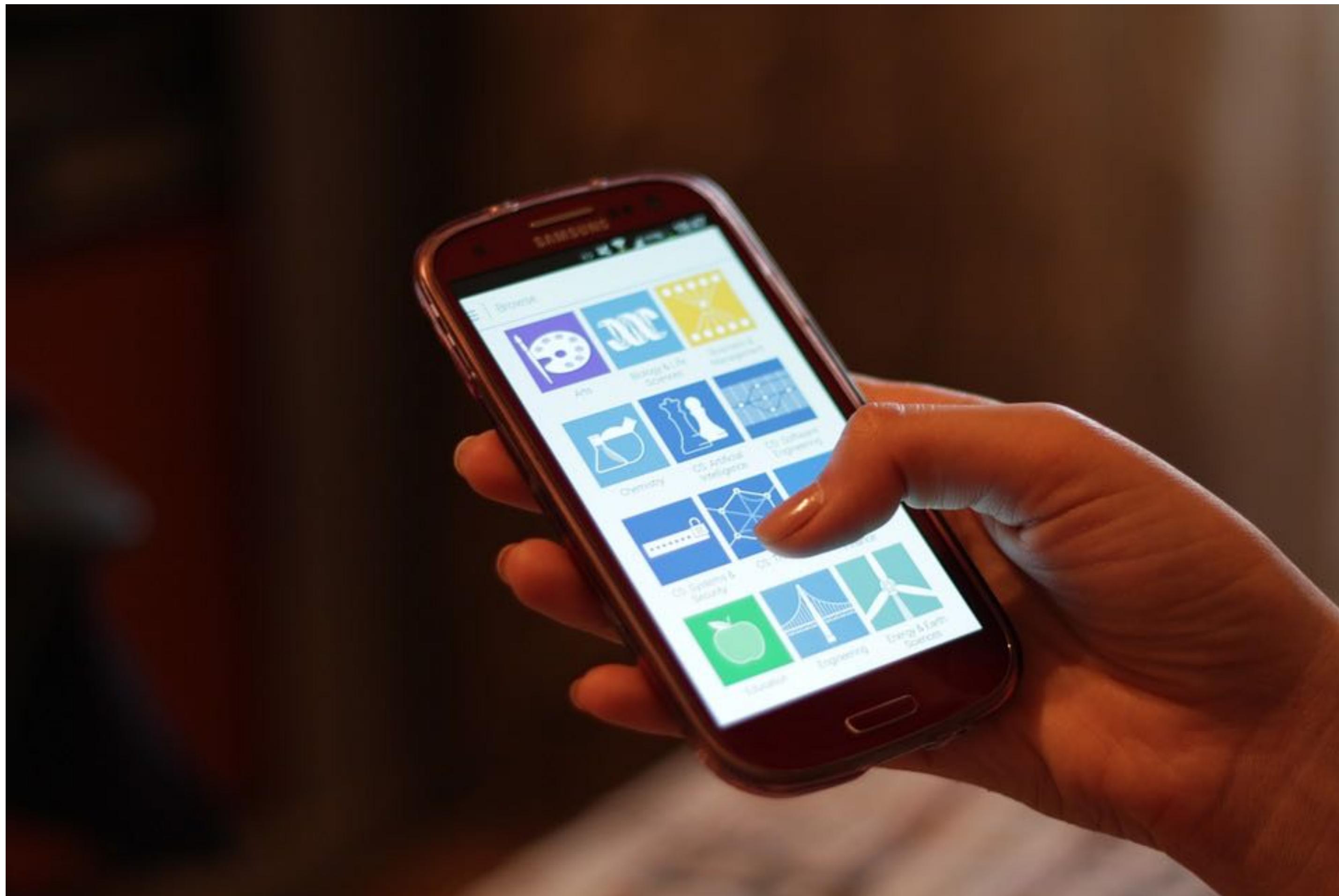


# Мышь





# Тач



## Touch events - REC

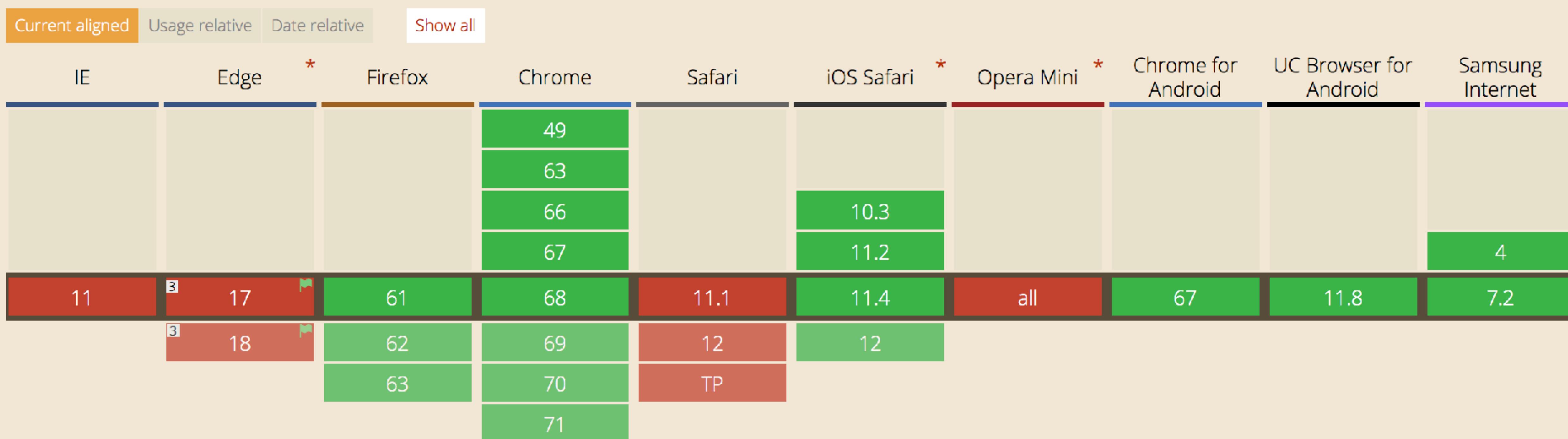
Usage

Global

% of all users

85.36% + 0.24% = 85.6%

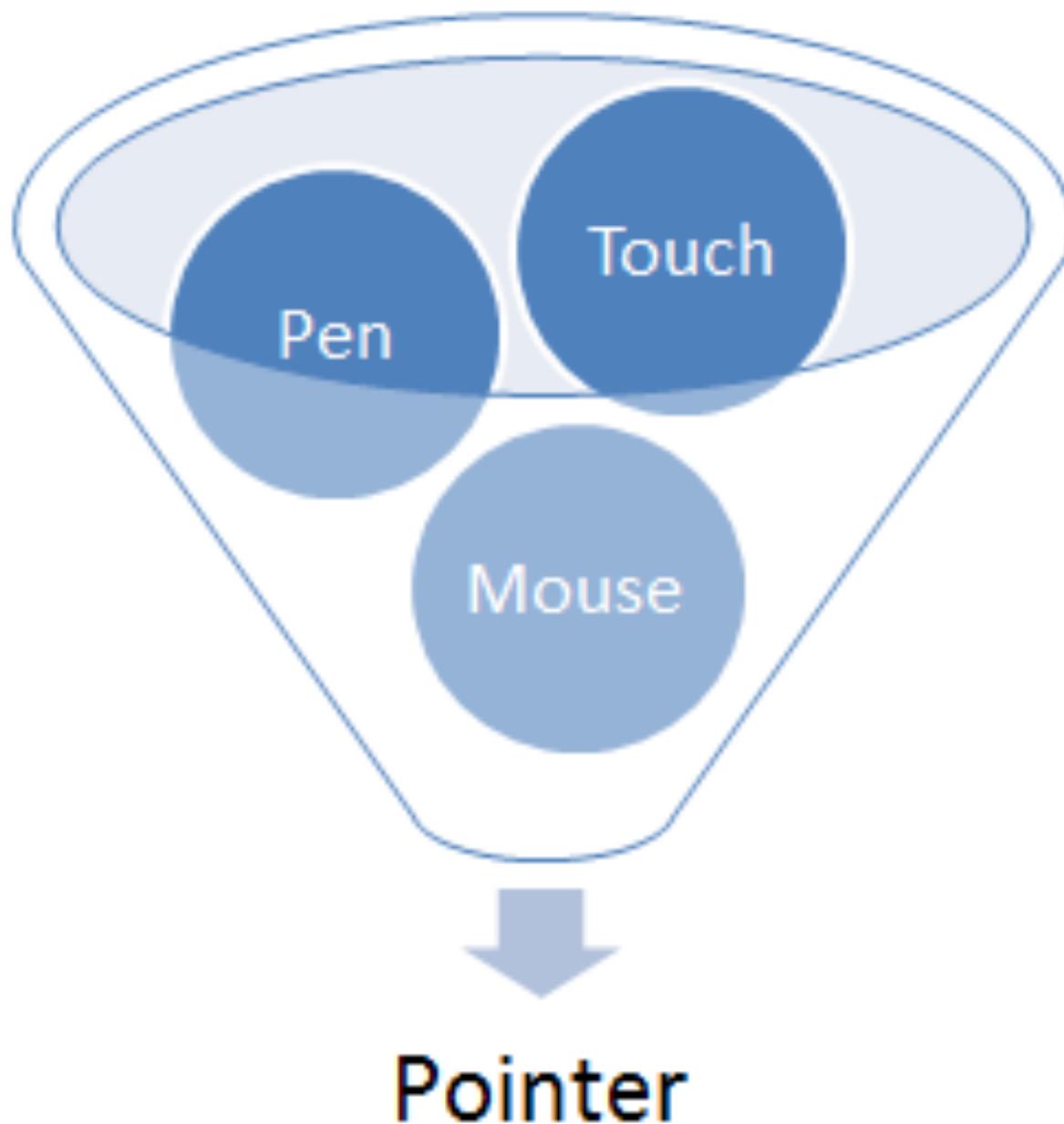
Method of registering when, where and how the interface is touched, for devices with a touch screen. These DOM events are similar to mousedown, mousemove, etc.



# Тач и стилус



# Тач и стилус Pointer Events



# Pointer Events vs Touch Events

Роман Дворнов



# Pointer events - REC

Usage

% of all users

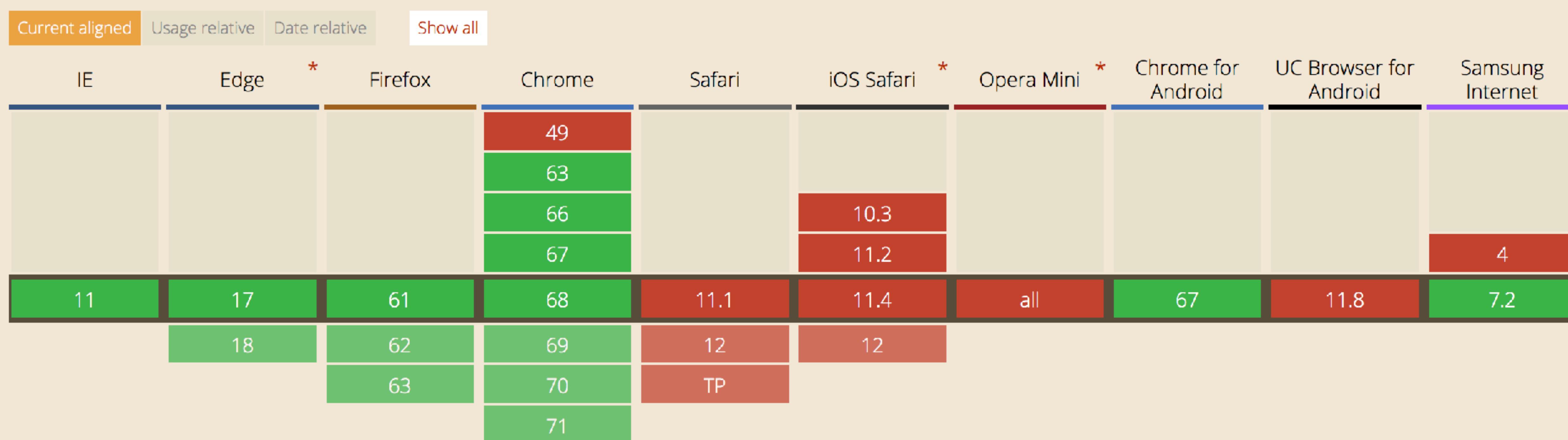
Global

67.93% + 0.12% = 68.05%

unprefixed:

67.93%

This specification integrates various inputs from mice, touchscreens, and pens, making separate implementations no longer necessary and authoring for cross-device pointers easier. Not to be mistaken with the unrelated "pointer-events" CSS property.



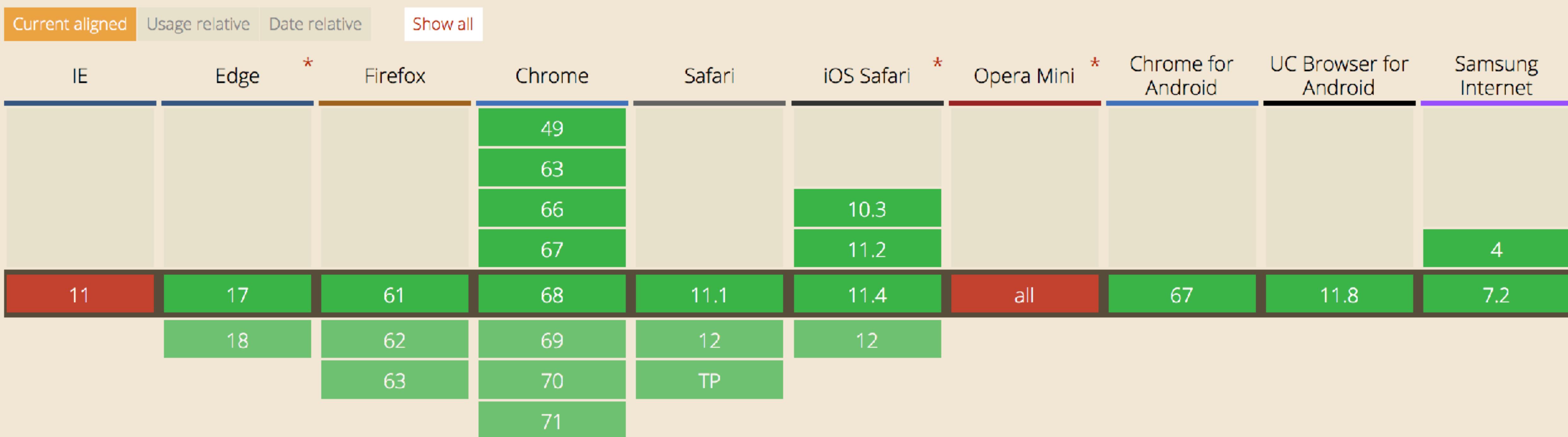
# Джойстик



## Gamepad API [- WD](#)

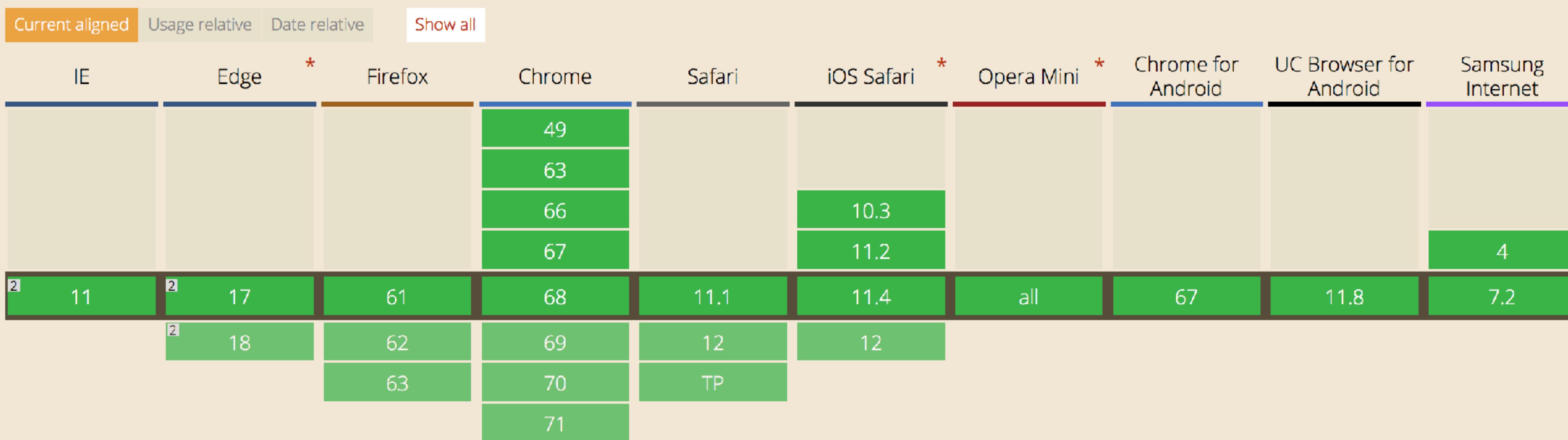
API to support input from USB gamepad controllers through JavaScript.

Usage	% of all users
Global	87.42%
unprefixed:	87.37%



# Клавиатура





# Клавиатура + скринридер

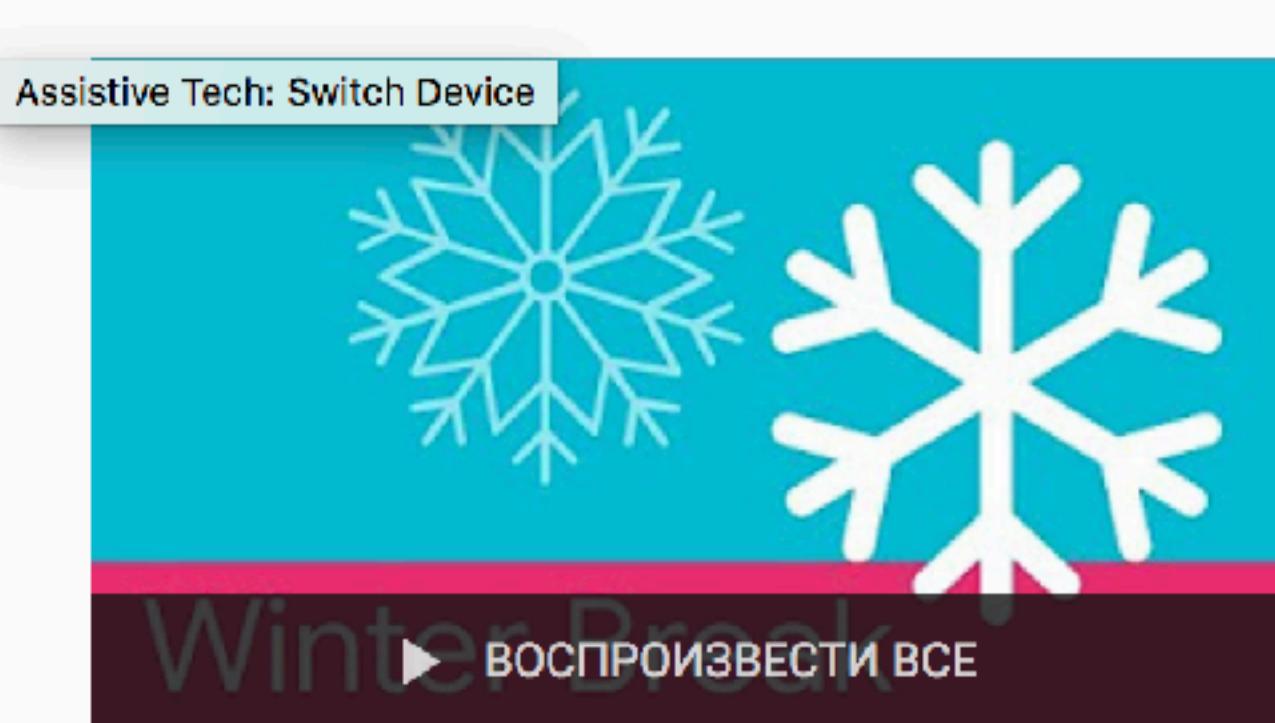


Людоедский интерфейс

**Вадим Макеев**

Accessibility Testing with a  
Screen Reader

**Сергей Кригер**



## A11ycasts with Rob Dodson

30 видео • 72 980 просмотров • Обновлен 9 янв.  
2018 г.



Google  
Chrome  
Developers

ПОДПИСАТЬСЯ 234 ТЫС.

Want to build accessible apps? Rod Dodson is teaching those fundamentals in his new series dedicated entirely to the art of accessibility. Meet A11ycasts!

A11ycasts is short for AccessibilityCasts (a11y is a common shortening of the term accessibility because there are 11 letters in between the "A" and the "Y"). The goal of A11ycasts is to teach developers how accessibility works all the way down at the platform level, while also demonstrating real world accessibility problems and solutions to fix them.

- 1 **A11ycasts Winter Break**  
Google Chrome Developers
- 2 **Assistive Tech: Switch Device**  
Google Chrome Developers
- 3 **Assistive Tech: VoiceOver on iOS**  
Google Chrome Developers
- 4 **Assistive Tech: TalkBack**  
Google Chrome Developers
- 5 **Testing Shadow DOM with aXe Coconut - A11ycasts #26**  
Google Chrome Developers
- 6 **How to build a toggle button - A11ycasts #25**  
Google Chrome Developers
- 7 **How to Label Custom Elements - A11ycasts #24**  
Google Chrome Developers

## A11ycasts Роба Додсона

**Ну зачем? Еще один  
доклад про а11у?**

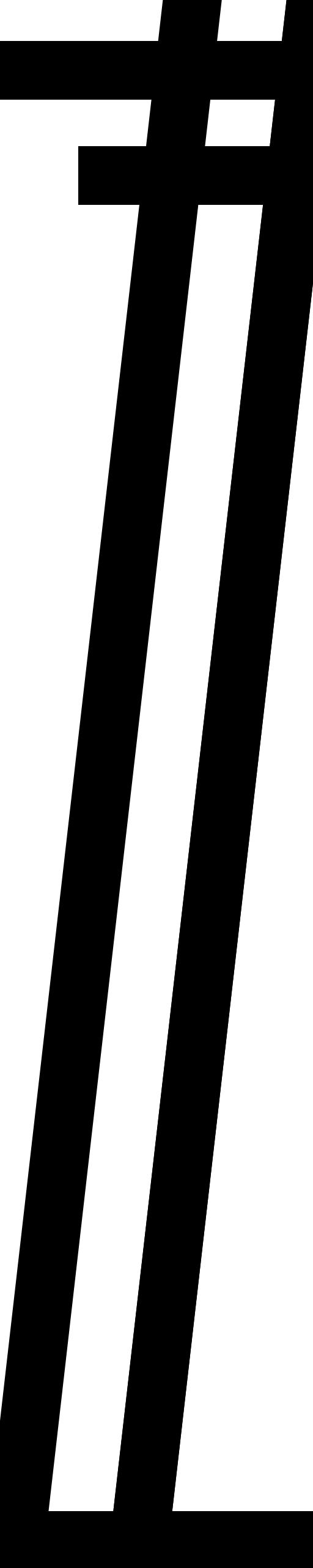
**Это не весело!  
я пришел за весельем!**

**Об этом говорят  
только девелоперы**

**Это не приносит**



**A11y это сложно**



Что если?

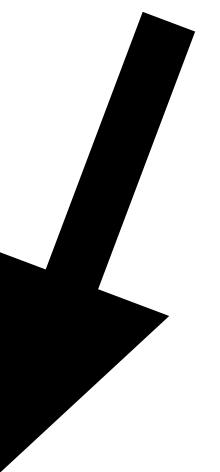
# **WebSpeech API**

**WebSpeech API - это не  
стандартный способ  
улучшения A11y**



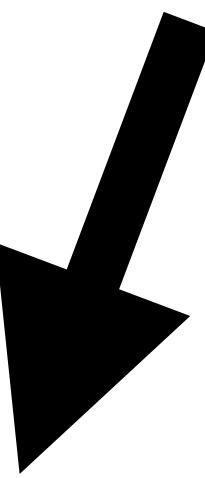
# WebSpeech API

# **WebSpeech API**

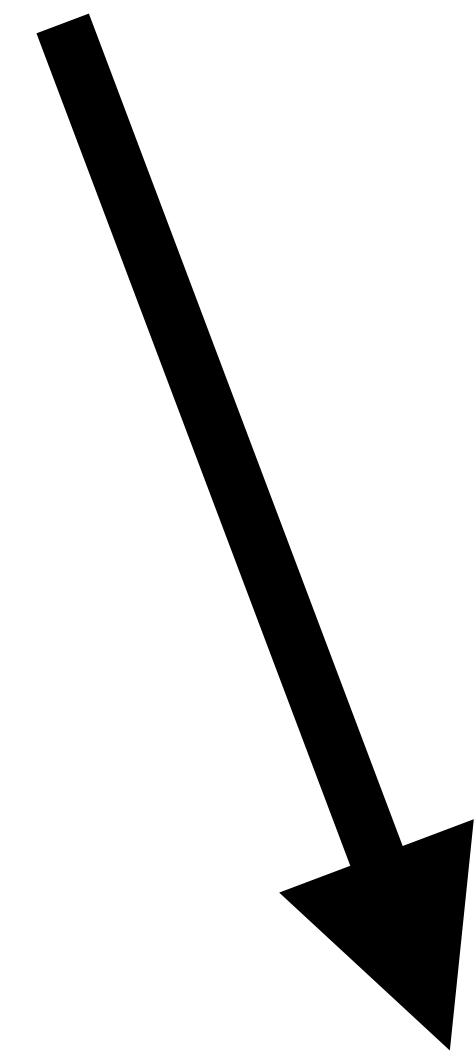


## **SpeechSynthesis**

# WebSpeech API



SpeechSynthesis



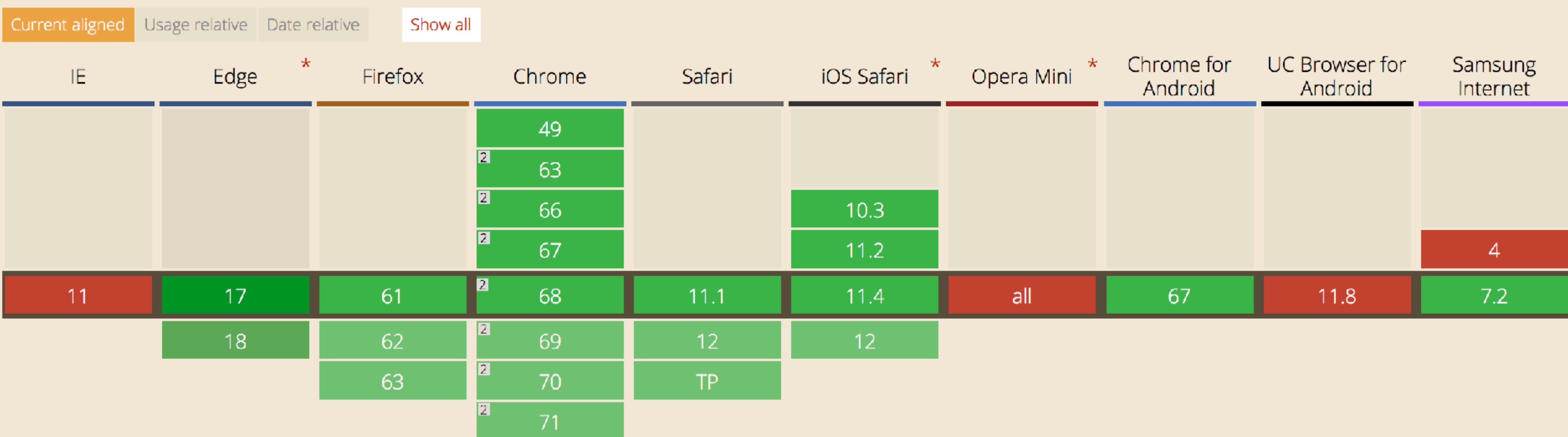
SpeechRecognition

# Speech Synthesis API

- UNOFF

Usage  
Global  
% of all users  
79.49%

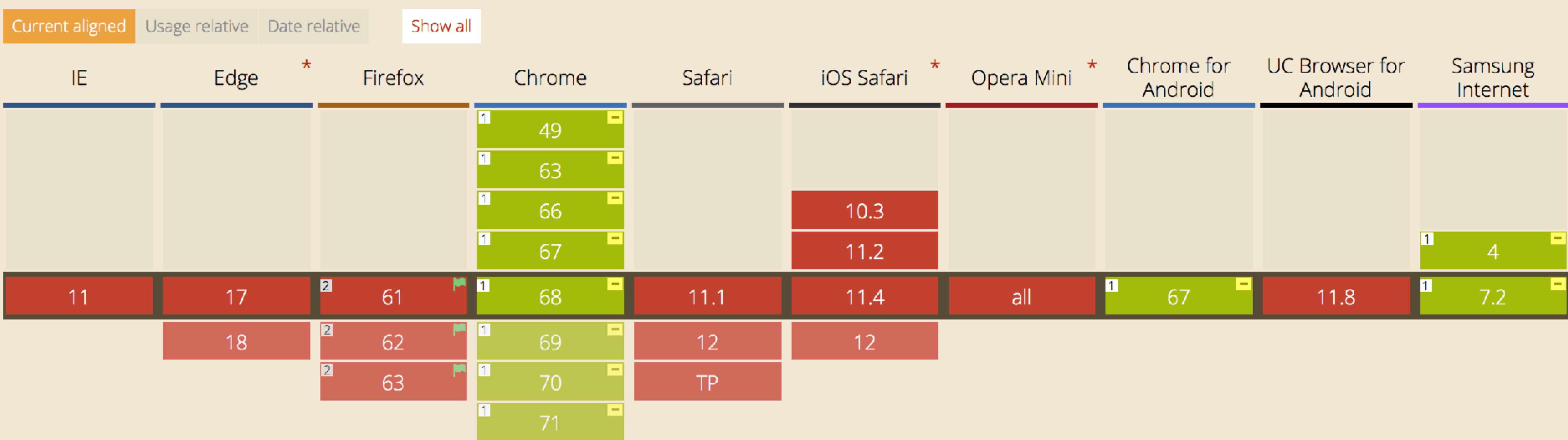
A web API for controlling a text-to-speech output.



## Speech Recognition API - UNOFF

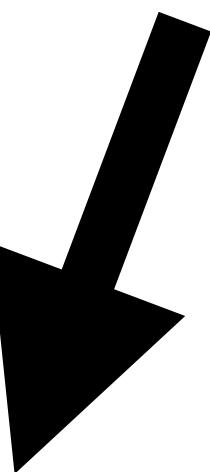
Method to provide speech input in a web browser.

Usage  
Global  
unprefixed:  
0% + 62.5% = 62.5%

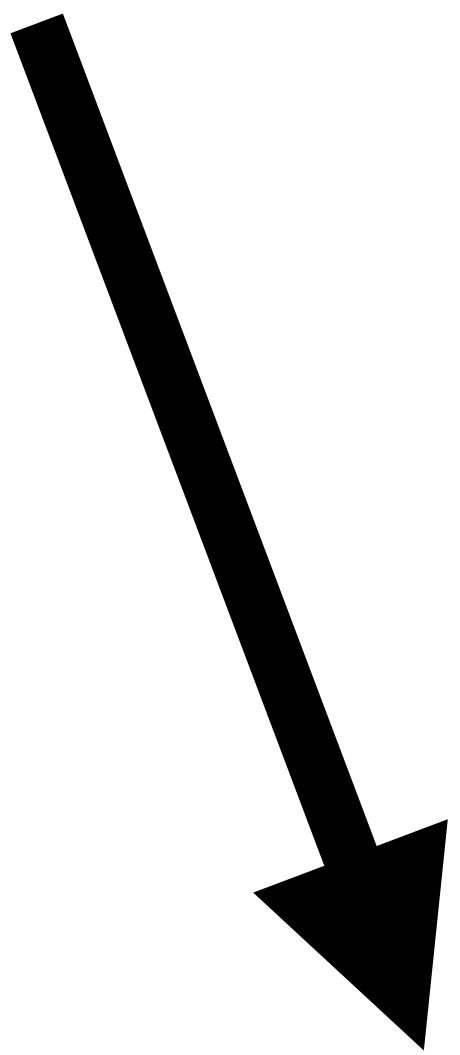


# SpeechSynthesis

# **SpeechSynthesis**

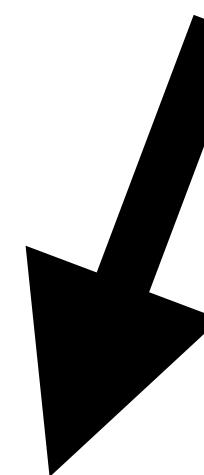


**SpeechSynthesisVoice**

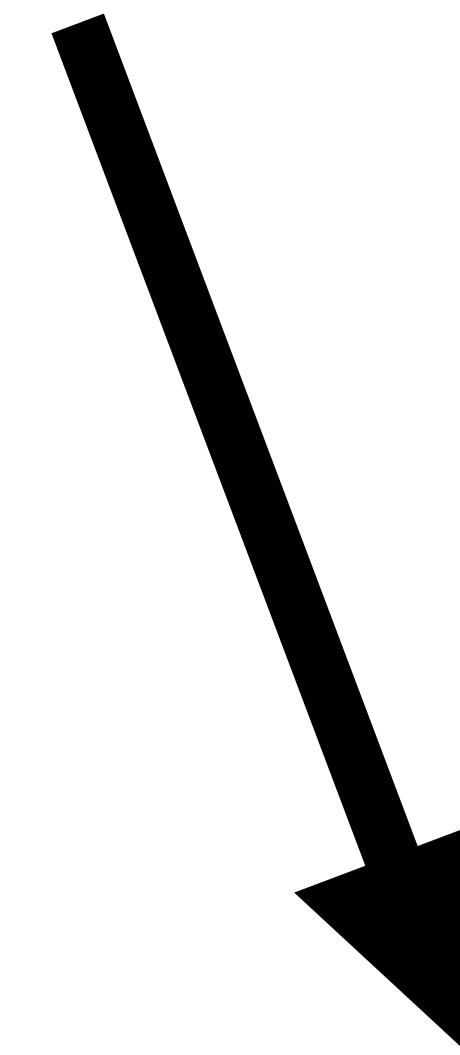


**SpeechSynthesisUtterance**

# SpeechSynthesis



**SpeechSynthesisVoice**



**SpeechSynthesisUtterance**

# SpeechSynthesis

```
if ('speechSynthesis' in window) {  
    const phrase = new SpeechSynthesisUtterance('Hello world');  
    speechSynthesis.speak(phrase);  
}
```

# SpeechSynthesis

```
if ('speechSynthesis' in window) {  
  const phrase = new SpeechSynthesisUtterance('Hello world');  
  speechSynthesis.speak(phrase);  
}
```

# SpeechSynthesis

```
if ('speechSynthesis' in window) {  
    const phrase = new SpeechSynthesisUtterance('Hello world');  
    speechSynthesis.speak(phrase);  
}
```

# **DEMO**

# **SpeechSynthesis**

# SpeechSynthesis

Pause, Resume, Speak, Cancel



# SpeechRecognition

# SpeechRecognition

```
if ('webkitSpeechRecognition' in window) {  
    var recognition = new webkitSpeechRecognition();  
    recognition.onresult = (event) => {  
        for (let i = 0; i < event.results.length; i++) {  
            let result = event.results[i];  
            let resultAlternative = result[0];  
            console.log(resultAlternative.transcript);  
        }  
    }  
    recognition.start();  
    // ... через некоторое время  
    recognition.stop();  
}
```

# SpeechRecognition

```
if ('webkitSpeechRecognition' in window) {  
    var recognition = new webkitSpeechRecognition();  
    recognition.onresult = (event) => {  
        for (let i = 0; i < event.results.length; i++) {  
            let result = event.results[i];  
            let resultAlternative = result[0];  
            console.log(resultAlternative.transcript);  
        }  
    }  
    recognition.start();  
    // ... через некоторое время  
    recognition.stop();  
}
```

# SpeechRecognition

```
if ('webkitSpeechRecognition' in window) {  
    var recognition = new webkitSpeechRecognition();  
    recognition.onresult = (event) => {  
        for (let i = 0; i < event.results.length; i++) {  
            let result = event.results[i];  
            let resultAlternative = result[0];  
            console.log(resultAlternative.transcript);  
        }  
    }  
    recognition.start();  
    // ... через некоторое время  
    recognition.stop();  
}
```

# SpeechRecognition

```
if ('webkitSpeechRecognition' in window) {  
    var recognition = new webkitSpeechRecognition();  
    recognition.onresult = (event) => {  
        for (let i = 0; i < event.results.length; i++) {  
            let result = event.results[i];  
            let resultAlternative = result[0];  
            console.log(resultAlternative.transcript);  
        }  
    }  
    recognition.start();  
    // ... через некоторое время  
    recognition.stop();  
}
```

# SpeechRecognition

```
if ('webkitSpeechRecognition' in window) {  
    var recognition = new webkitSpeechRecognition();  
    recognition.onresult = (event) => {  
        for (let i = 0; i < event.results.length; i++) {  
            let result = event.results[i];  
            let resultAlternative = result[0];  
            console.log(resultAlternative.transcript);  
        }  
    }  
    recognition.start();  
    // ... через некоторое время  
    recognition.stop();  
}
```

# SpeechRecognition

```
if ('webkitSpeechRecognition' in window) {  
    var recognition = new webkitSpeechRecognition();  
    recognition.onresult = (event) => {  
        for (let i = 0; i < event.results.length; i++) {  
            let result = event.results[i];  
            let resultAlternative = result[0];  
            console.log(resultAlternative.transcript);  
        }  
    }  
    recognition.start();  
    // ... через некоторое время  
    recognition.stop();  
}
```

# SpeechRecognition

```
if ('webkitSpeechRecognition' in window) {  
    var recognition = new webkitSpeechRecognition();  
    recognition.onresult = (event) => {  
        for (let i = 0; i < event.results.length; i++) {  
            let result = event.results[i];  
            let resultAlternative = result[0];  
            console.log(resultAlternative.transcript);  
        }  
    }  
    recognition.start();  
    // ... через некоторое время  
    recognition.stop();  
}
```

# SpeechRecognition

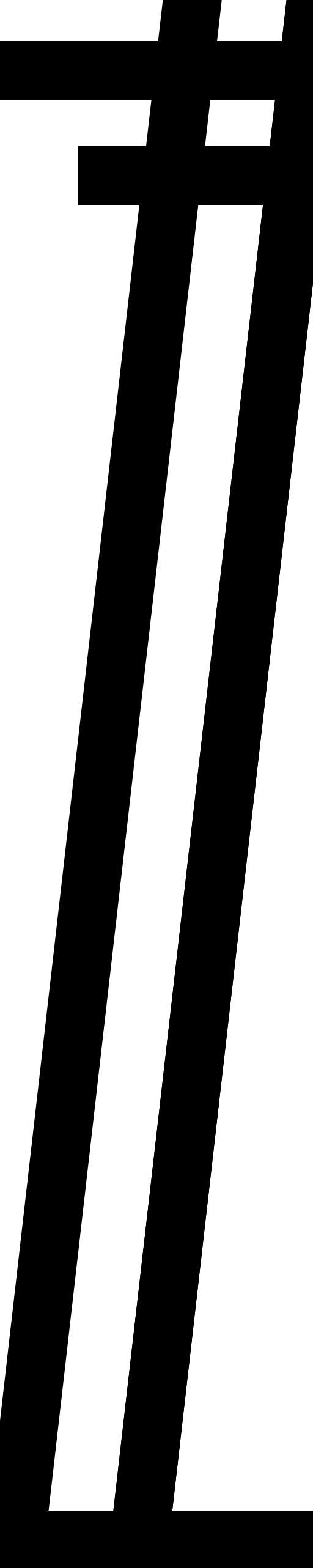
```
if ('webkitSpeechRecognition' in window) {  
    var recognition = new webkitSpeechRecognition();  
    recognition.onresult = (event) => {  
        for (let i = 0; i < event.results.length; i++) {  
            let result = event.results[i];  
            let resultAlternative = result[0];  
            console.log(resultAlternative.transcript);  
        }  
    }  
    recognition.start();  
    // ... через некоторое время  
    recognition.stop();  
}
```

# SpeechRecognition

```
if ('webkitSpeechRecognition' in window) {  
    var recognition = new webkitSpeechRecognition();  
    recognition.onresult = (event) => {  
        for (let i = 0; i < event.results.length; i++) {  
            let result = event.results[i];  
            let resultAlternative = result[0];  
            console.log(resultAlternative.transcript);  
        }  
    }  
    recognition.start();  
    // ... через некоторое время  
    recognition.stop();  
}
```

# **DEMO**

# **SpeechRecognition**



**Одна маленькая  
проблема**

# **NLP**

## **(Natural Language Processing)**

**Наука о том, как работать с  
языками алгоритмически**

# **20 лет**

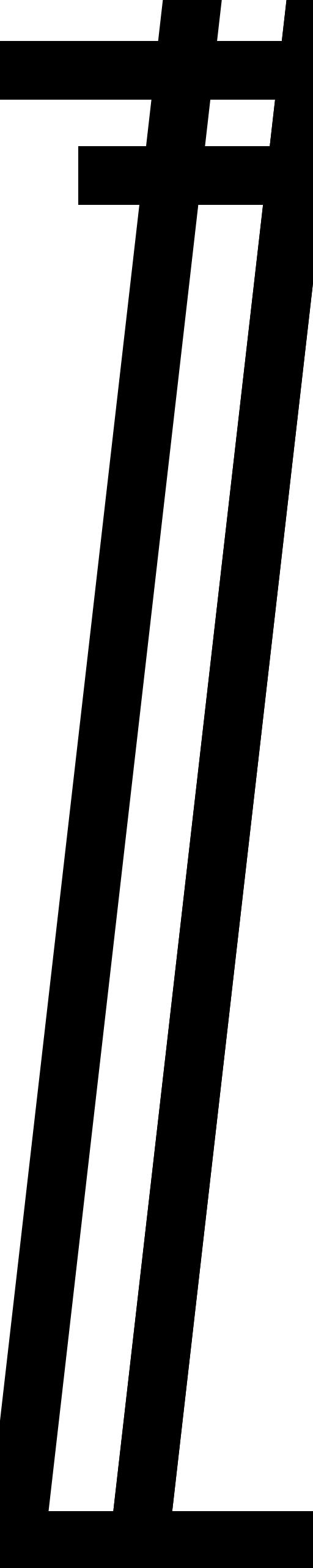
## **NLP**

### **(Natural Language Processing)**

**Наука о том, как работать с  
языками алгоритмически**

**Долго, муторно, много  
работы и исследований**

**Пока нет готовых  
библиотек**



Речь - сложная штука

**Есть один “грязный трюк”**

Amazon **Alexa**

Apple **Siri**,

**Google Assistant**,

Yandex **Алиса**

# **Последовательный диалог (Intent)**

- Привет! (приветствие)
- Привет!
- Завтра будет дождь? (запрос)
  - Завтра дождя не будет

- (приветствие)
  - (ответное приветствие)
- (запрос)
  - (ответ на запрос)

- Ok Google (приветствие)
  - (ответное приветствие)
- Ok Google (запрос)
  - (ответ на запрос)

```
commands = {  
    list: (model) => {},  
    search: (query) => {},  
    filterTime: (time) => {},  
    goTo: (pageTile) => {}  
}
```

# SpeechGrammar

# SpeechGrammar

JSpeech Grammar Format  
JSGF

# SpeechGrammar

JSpeech Grammar Format  
JSGF

Sun Microsystems / Oracle

# SpeechGrammar

JSpeech Grammar Format  
JSGF

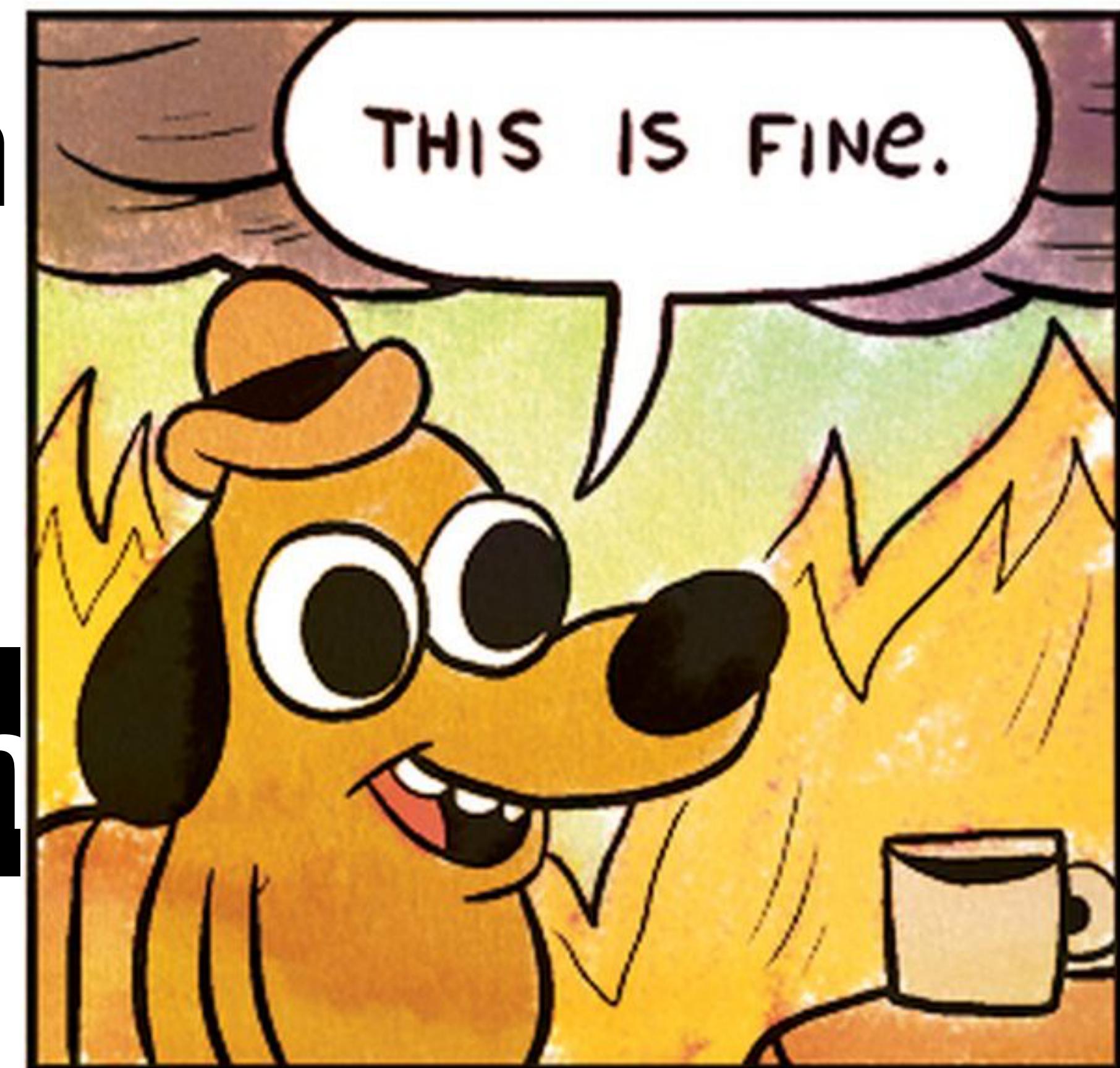
Sun Microsystems / Oracle



# SpeechGrammar

JSpeech Gramm  
JSGF

SunMicrosystem



# SpeechGrammar

```
var grammar = `#JSGF V1.0;
grammar colors;
public <color> = red | green ;`  
  
var recognition = new webkitSpeechRecognition();
var speechRecognitionList = new webkitSpeechGrammarList();
speechRecognitionList.addFromString(grammar, 1);
recognition.lang = 'en-US';
recognition.interimResults = false;
recognition.maxAlternatives = 1;
```

# SpeechGrammar

```
var grammar = `#JSRGF V1.0;
grammar colors;
public <color> = red | green ;`  
  
var recognition = new webkitSpeechRecognition();
var speechRecognitionList = new webkitSpeechGrammarList();
speechRecognitionList.addFromString(grammar, 1);
recognition.lang = 'en-US';
recognition.interimResults = false;
recognition.maxAlternatives = 1;
```

# SpeechGrammar

```
var grammar = `#JSGF V1.0;
grammar colors;
public <color> = red | green ;
`;

var recognition = new webkitSpeechRecognition();
var speechRecognitionList = new webkitSpeechGrammarList();
speechRecognitionList.addFromString(grammar, 1);
recognition.lang = 'en-US';
recognition.interimResults = false;
recognition.maxAlternatives = 1;
```

# SpeechGrammar

```
var grammar = `#JSGF V1.0;
grammar colors;
public <color> = red | green ;`  
  
var recognition = new webkitSpeechRecognition();
var speechRecognitionList = new webkitSpeechGrammarList();
speechRecognitionList.addFromString(grammar, 1);
recognition.lang = 'en-US';
recognition.interimResults = false;
recognition.maxAlternatives = 1;
```

# SpeechGrammar

```
var grammar = `#JSGF V1.0;
grammar colors;
public <color> = red | green ;`  
  
var recognition = new webkitSpeechRecognition();
var speechRecognitionList = new webkitSpeechGrammarList();
speechRecognitionList.addFromString(grammar, 1);
recognition.lang = 'en-US';
recognition.interimResults = false;
recognition.maxAlternatives = 1;
```

# SpeechGrammar

```
var grammar = `#JSGF V1.0;
grammar colors;
public <color> = red | green ;`  
  
var recognition = new webkitSpeechRecognition();
var speechRecognitionList = new webkitSpeechGrammarList();
speechRecognitionList.addFromString(grammar, 1);
recognition.lang = 'en-US';
recognition.interimResults = false;
recognition.maxAlternatives = 1;
```

# SpeechGrammar

```
var grammar = `#JSGF V1.0;
grammar colors;
public <color> = red | green ;`  
  
var recognition = new webkitSpeechRecognition();
var speechRecognitionList = new webkitSpeechGrammarList();
speechRecognitionList.addFromString(grammar, 1);
recognition.lang = 'en-US';
recognition.interimResults = false;
recognition.maxAlternatives = 1;
```

# SpeechGrammar

```
var grammar = `#JSGF V1.0;
grammar colors;
public <color> = red | green ;`  
  
var recognition = new webkitSpeechRecognition();
var speechRecognitionList = new webkitSpeechGrammarList();
speechRecognitionList.addFromString(grammar, 1);
recognition.lang = 'en-US';
recognition.interimResults = false;
recognition.maxAlternatives = 1;
```

# SpeechGrammar

```
var grammar = `#JSGF V1.0;
grammar colors;
public <color> = red | green ;`  
  
var recognition = new webkitSpeechRecognition();
var speechRecognitionList = new webkitSpeechGrammarList();
speechRecognitionList.addFromString(grammar, 1);
recognition.lang = 'en-US';
recognition.interimResults = false;
recognition.maxAlternatives = 1;
```

# SpeechGrammar

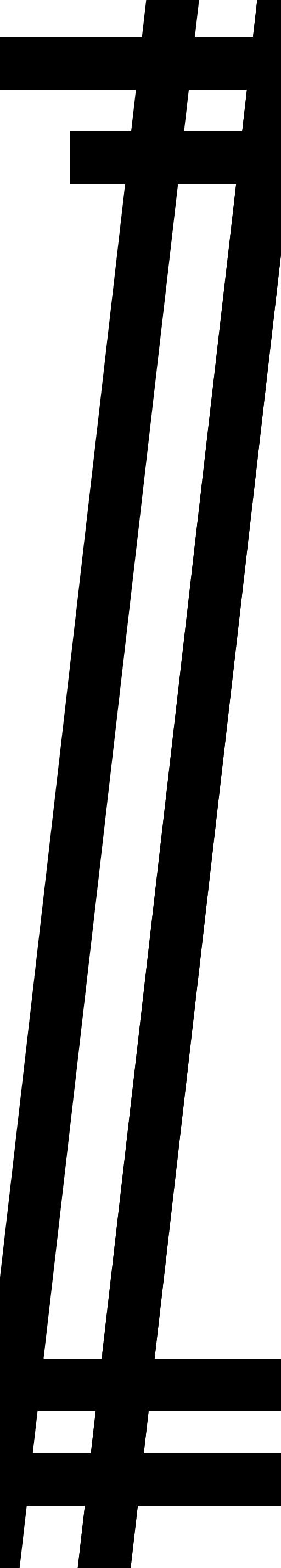
```
var grammar = `#JSGF V1.0;
grammar colors;
public <color> = red | green ;`  
  
var recognition = new webkitSpeechRecognition();
var speechRecognitionList = new webkitSpeechGrammarList();
speechRecognitionList.addFromString(grammar, 1);
recognition.lang = 'en-US';
recognition.interimResults = false;
recognition.maxAlternatives = 1;
```

# SpeechGrammar

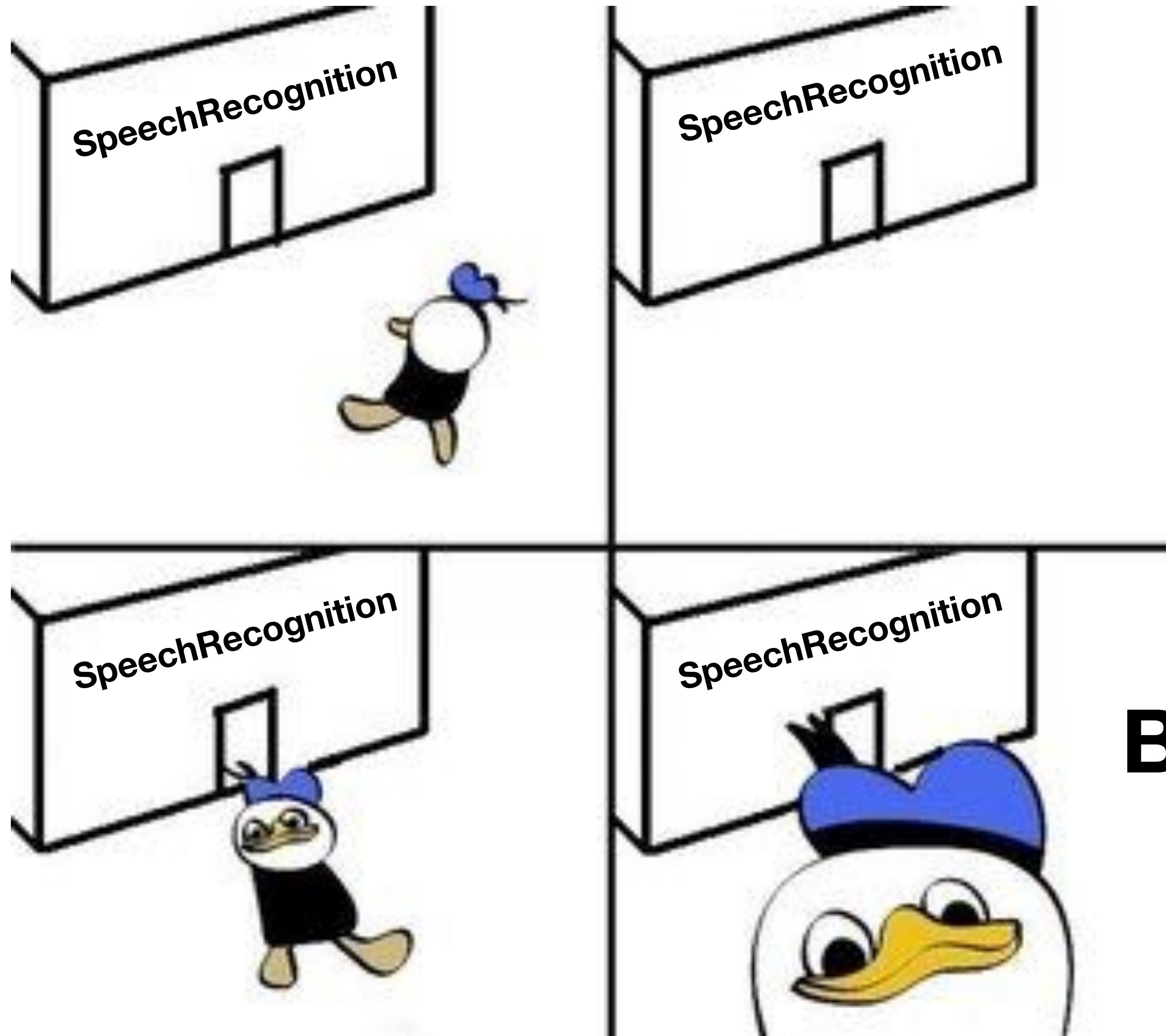
```
var grammar = `#JSGF V1.0;
grammar colors;
public <color> = red | green ;`  
  
var recognition = new webkitSpeechRecognition();
var speechRecognitionList = new webkitSpeechGrammarList();
speechRecognitionList.addFromString(grammar, 1);
recognition.lang = 'en-US';
recognition.interimResults = false;
recognition.maxAlternatives = 1;
recognition.grammars = speechRecognitionList;
```

# **DEMO**

# **SpeechGrammar**



**лучше...**

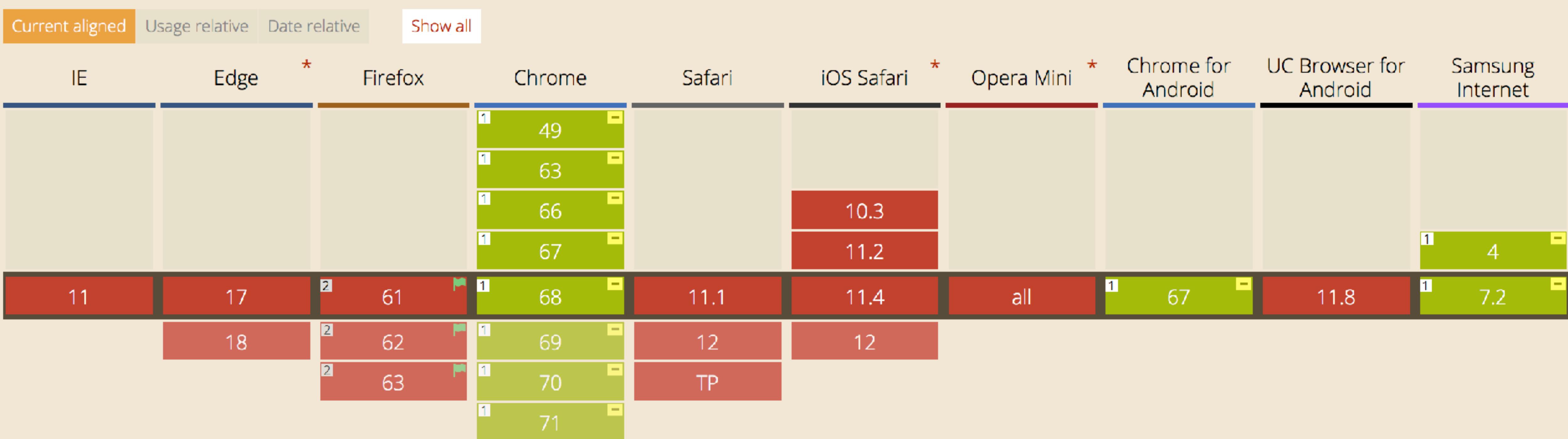


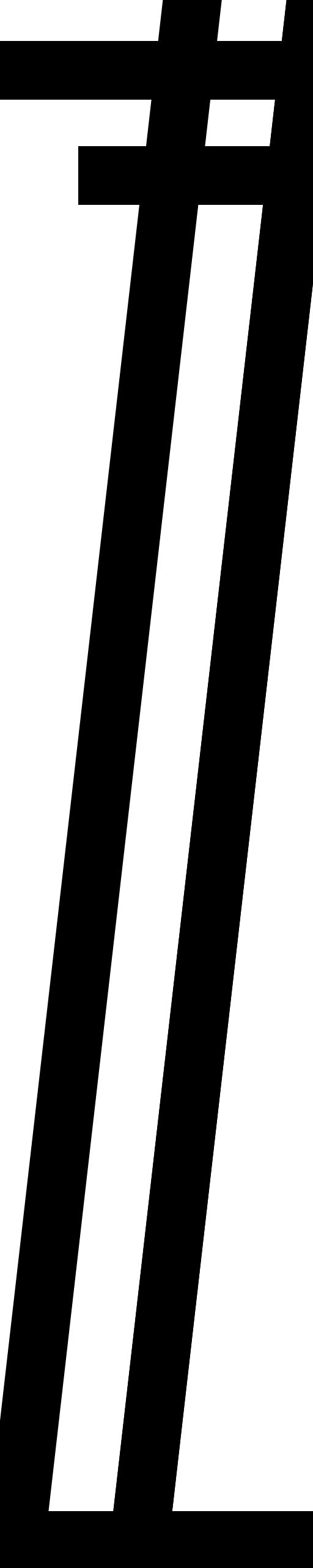
**Все очень  
плохо**

## Speech Recognition API - UNOFF

Method to provide speech input in a web browser.

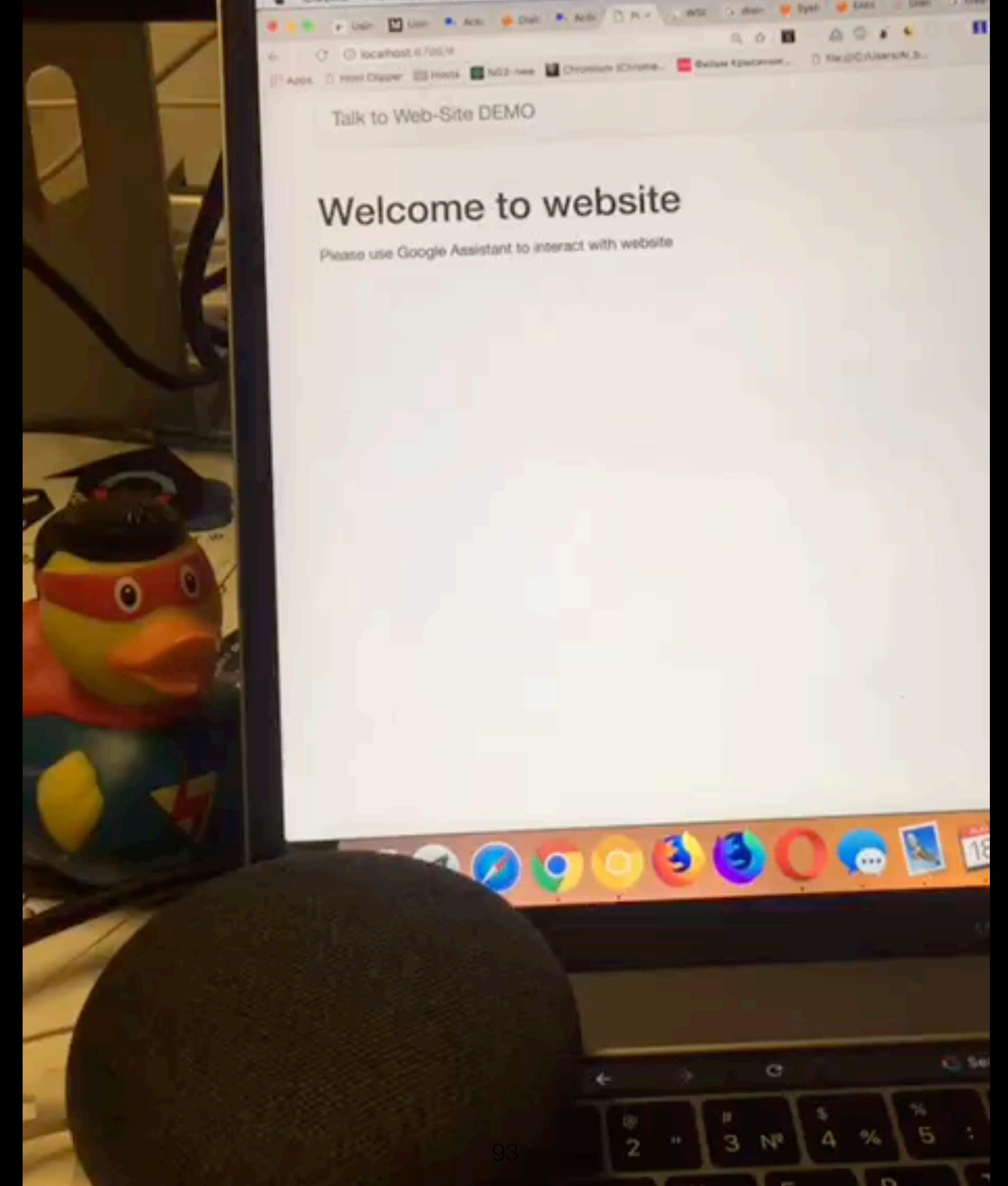
Usage  
Global  
unprefixed:  
0% + 62.5% = 62.5%

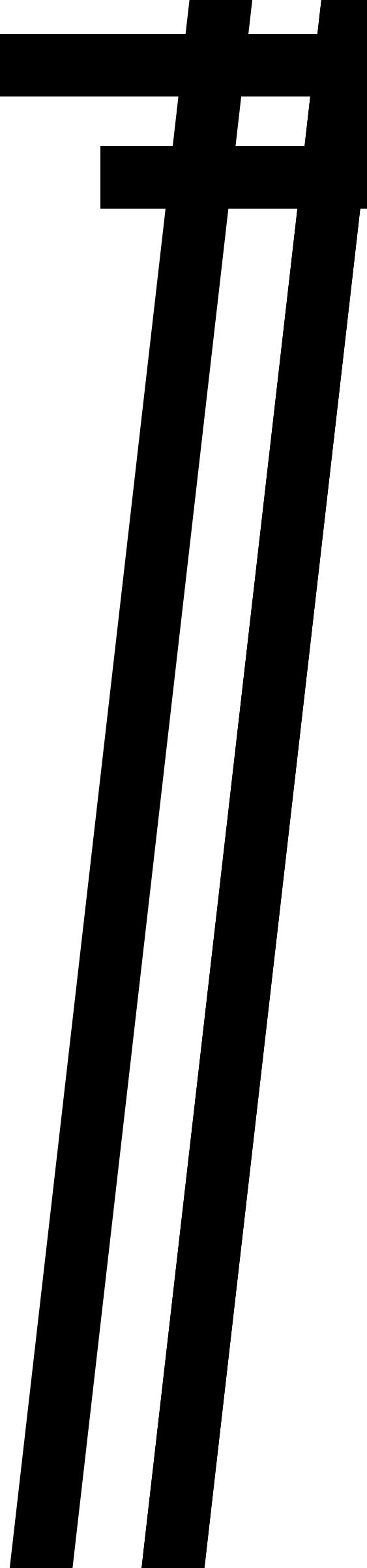




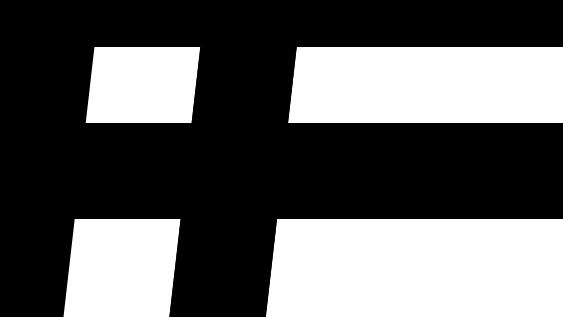
**Выхода нет?**

# **MAIN DEMO**





Kak?



**6 часов**

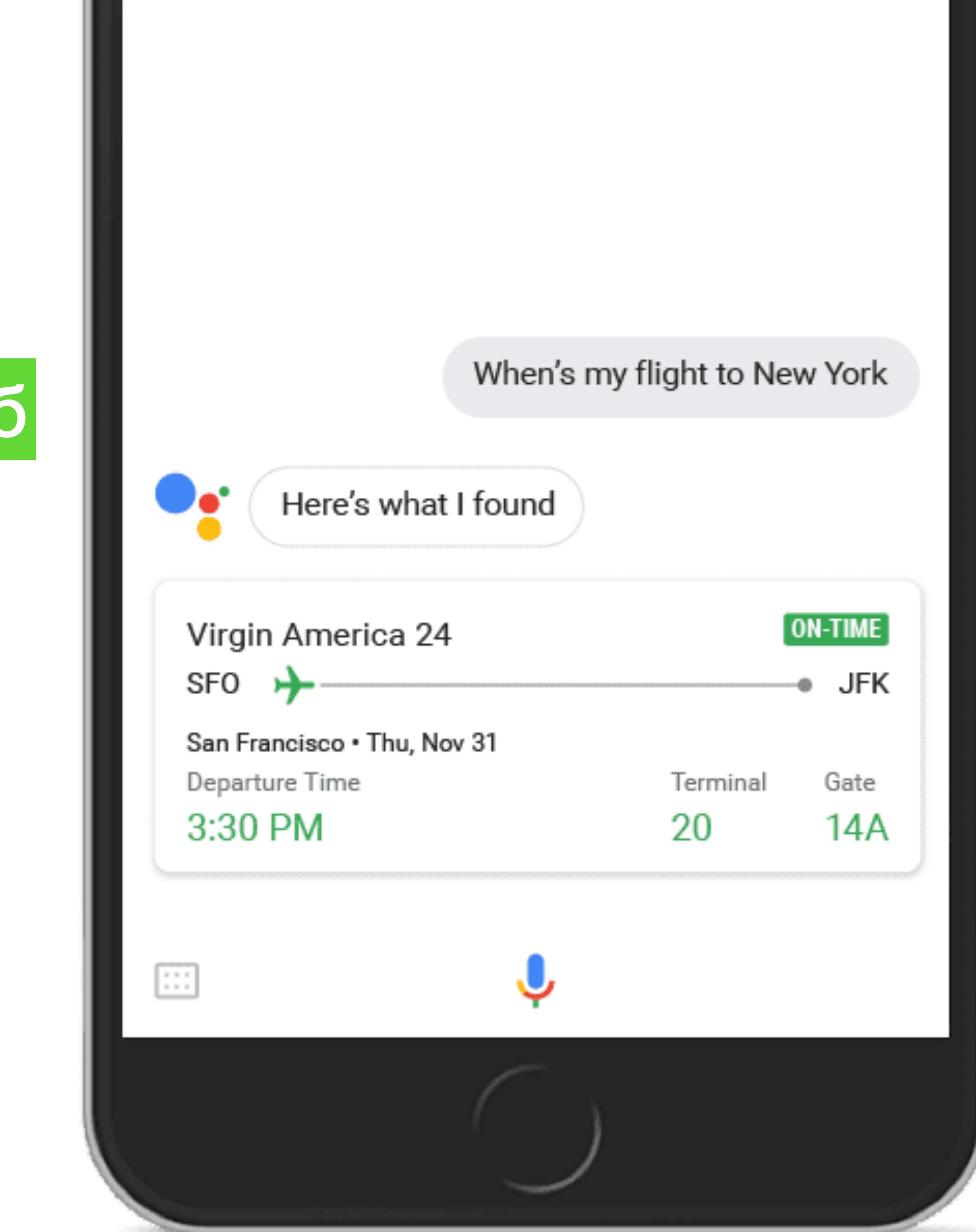
**Google Home**

~5000 руб



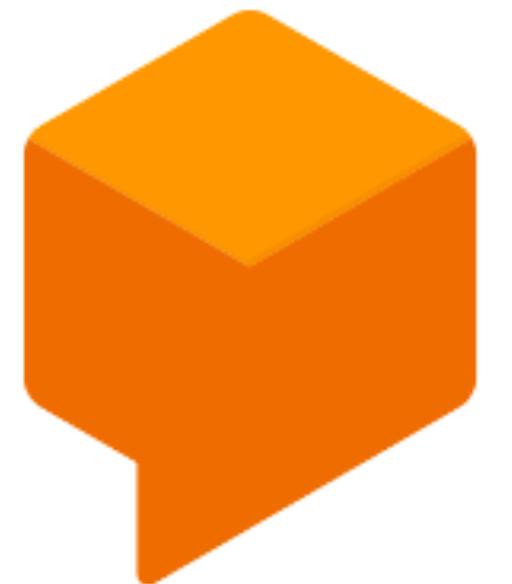
# Google Assistant ~0 руб

## Google Home



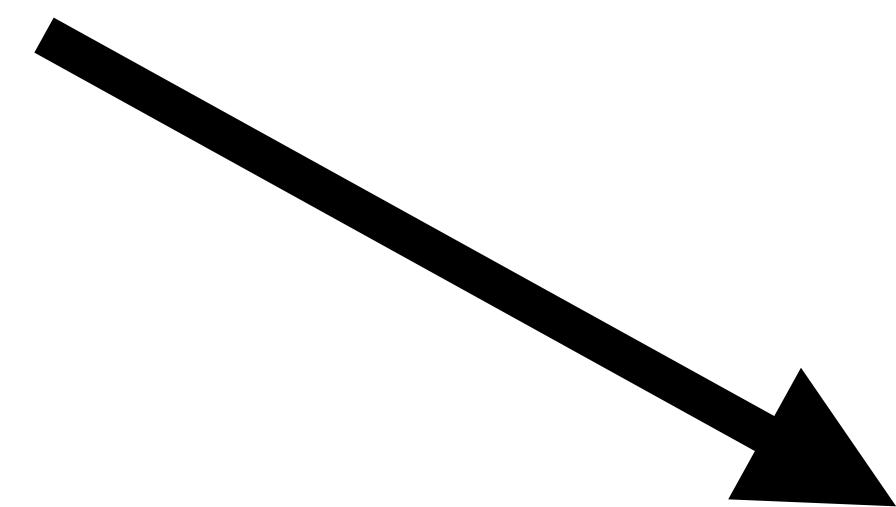
**Dialogflow** ~0 руб

**Google Assistant**  
**Google Home**



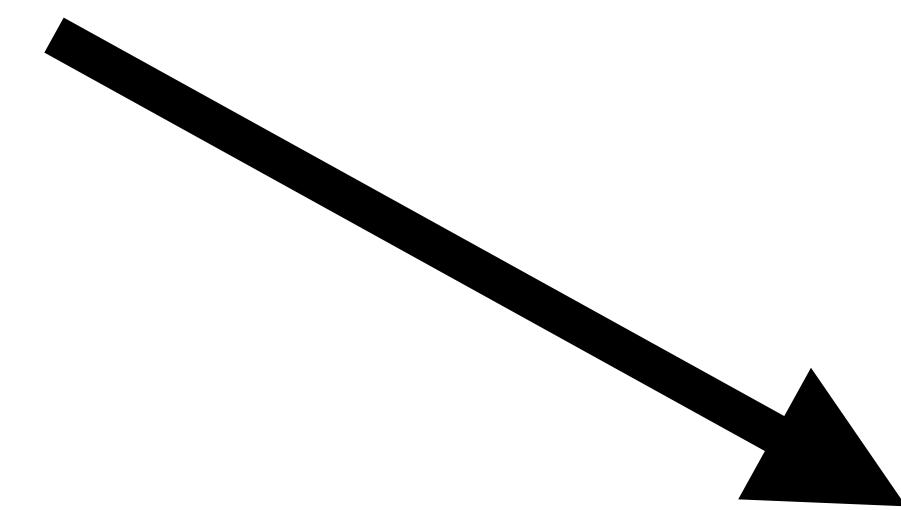
**Dialogflow**

**Dialogflow**  
**Google Assistant**  
**Google Home**



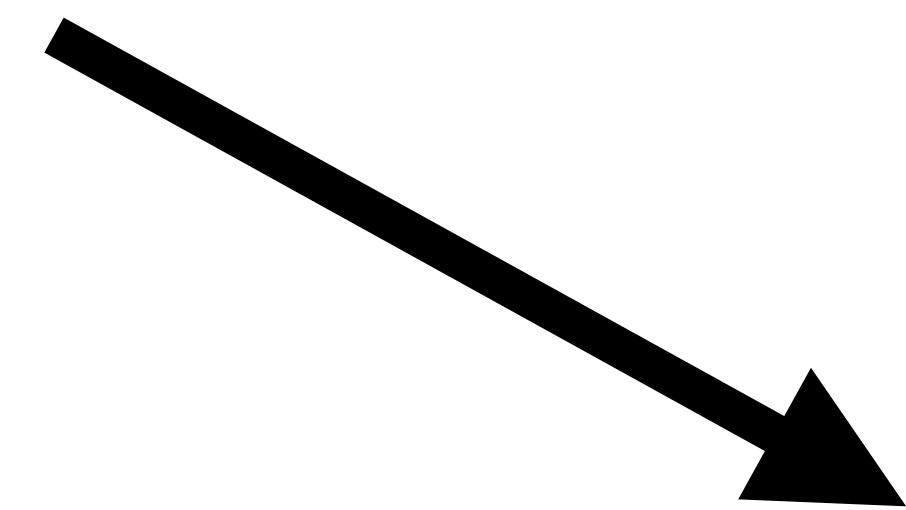
**Firebase**

**Dialogflow**  
**Google Assistant**  
**Google Home**



**Firebase**  
**Любой бэкенд**

**Dialogflow**  
**Google Assistant**  
**Google Home**

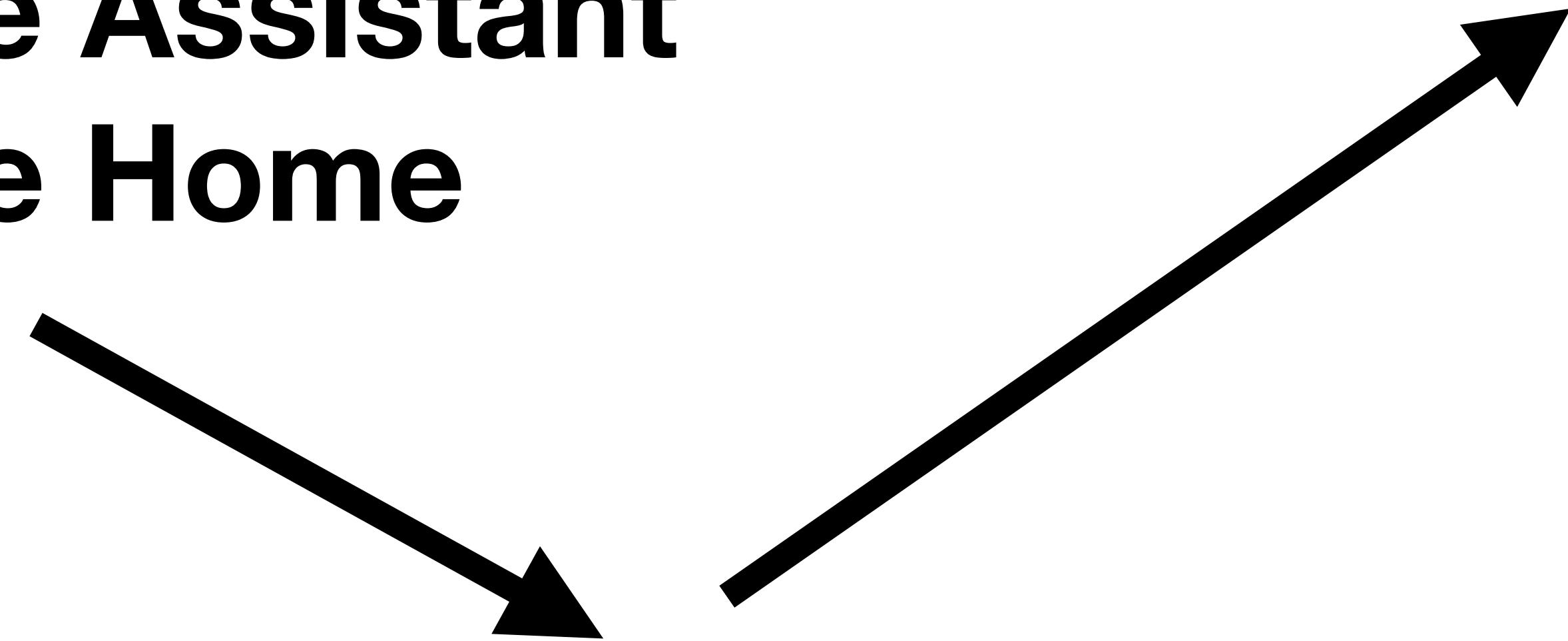


**Firebase** ~0 руб

**Любой бэкенд**

**Dialogflow  
Google Assistant  
Google Home**

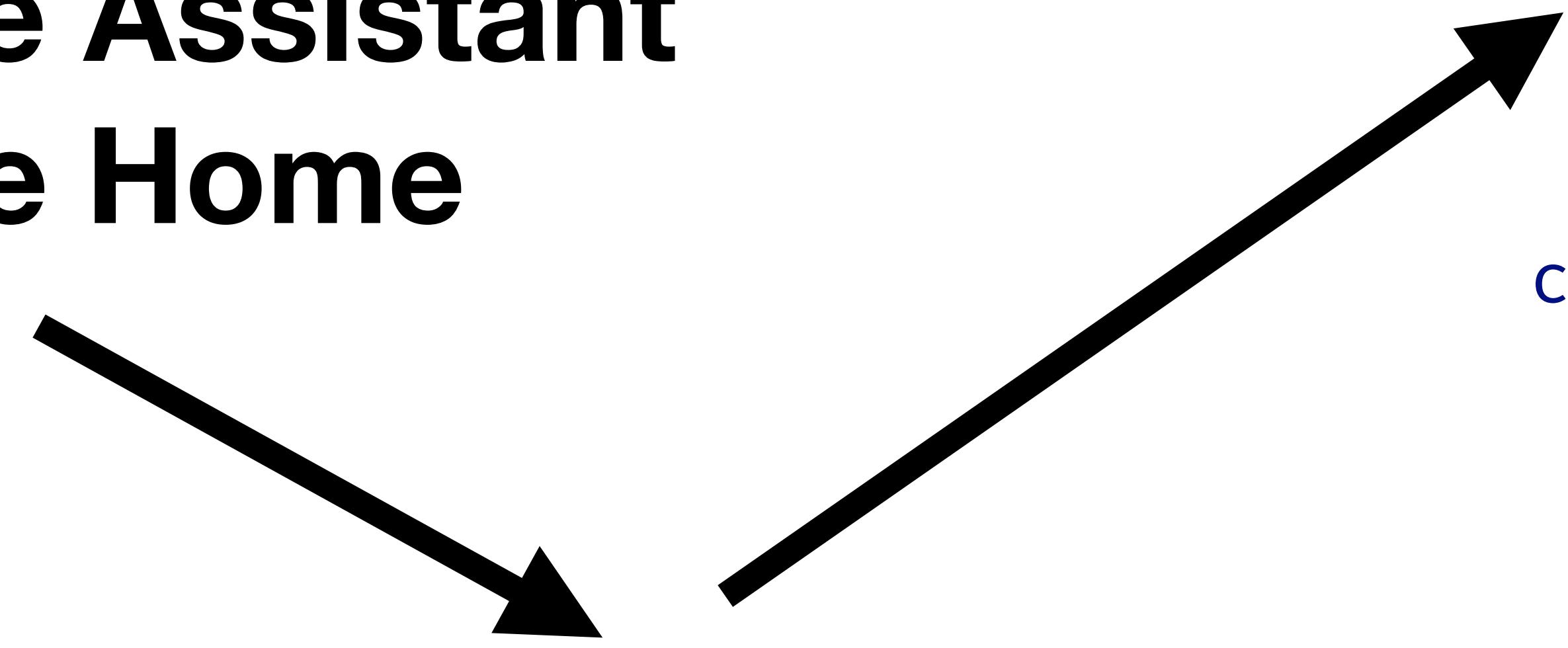
**JS-библиотека**



**Firebase**

**Любой бэкенд**

**Dialogflow  
Google Assistant  
Google Home**



**Firebase**  
**Любой бэкенд**

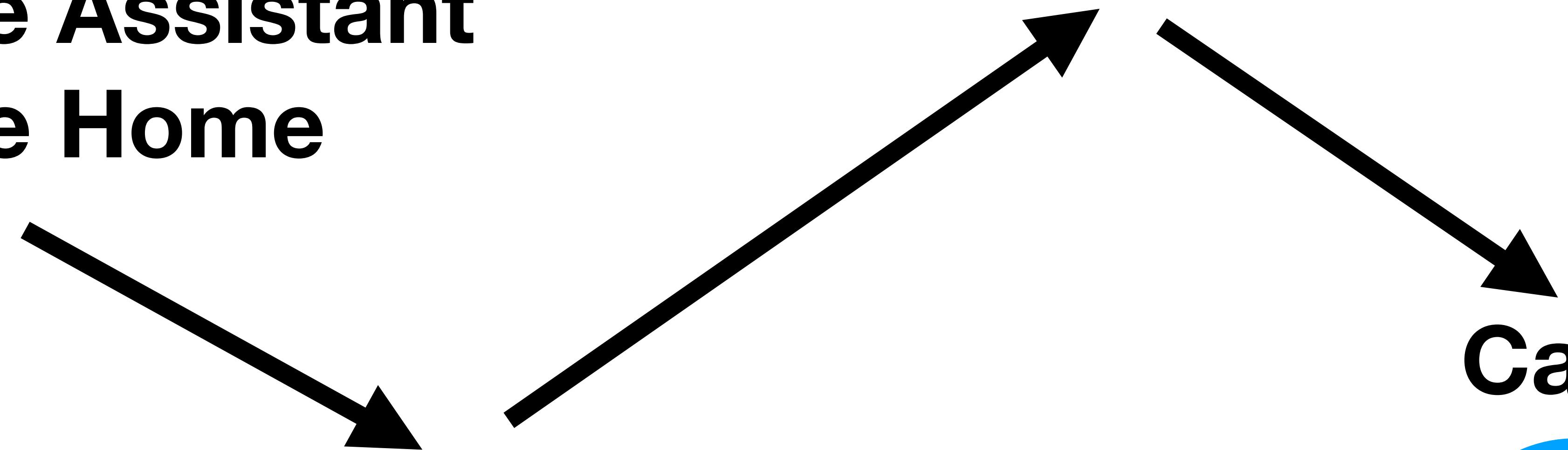
```
commands = {  
    list: (model) => {},  
    search: (query) => {},  
    filterTime: (time) => {},  
    goTo: (pageTile) => {}  
}
```

**Dialogflow  
Google Assistant  
Google Home**

**JS-библиотека**

**Сайт**

**Firebase  
Любой бэкенд**



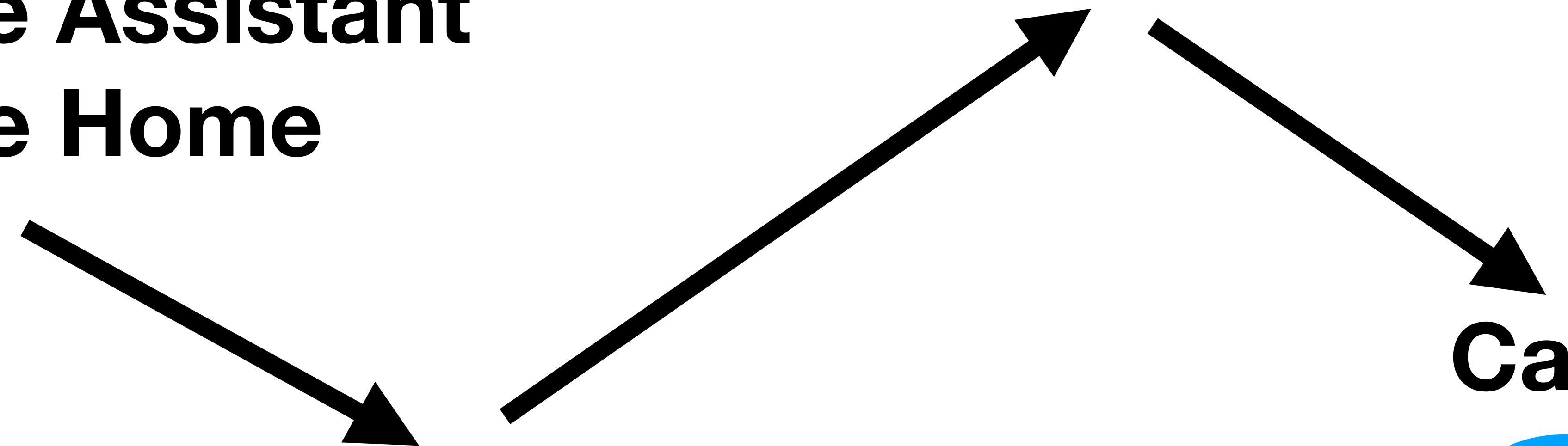
**Dialogflow  
Google Assistant  
Google Home**

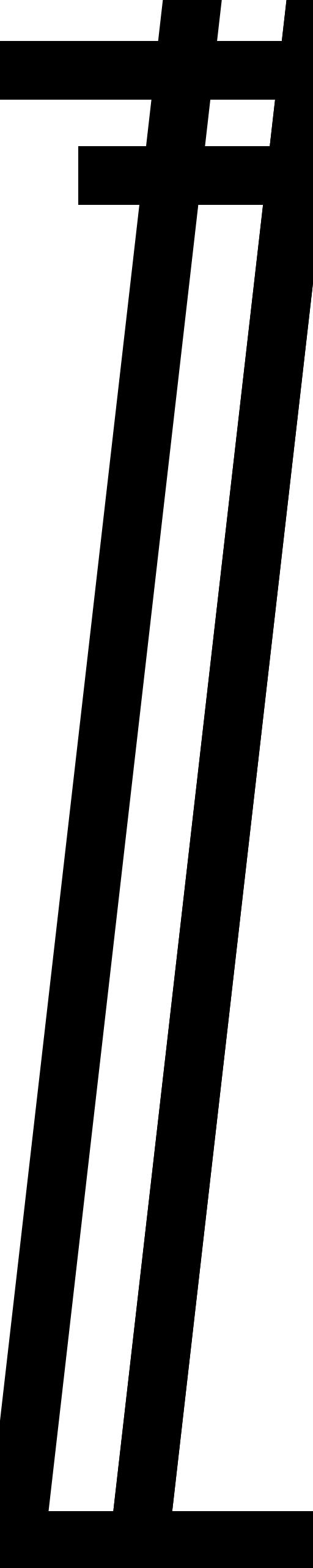
**JS-библиотека**

**Сайт**

**Firebase**

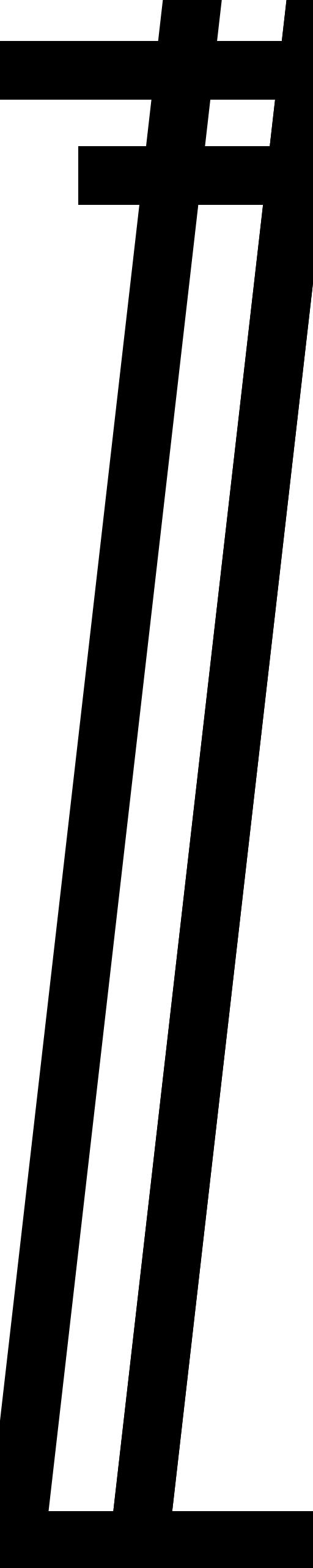
**Любой бэкенд**





Чему мы научились

- 1) WebSpeech API - SpeechSynthesis, SpeechRecognition, SpeechGrammar
- 2) Google Assistant / Dialogflow / backend / small js lib



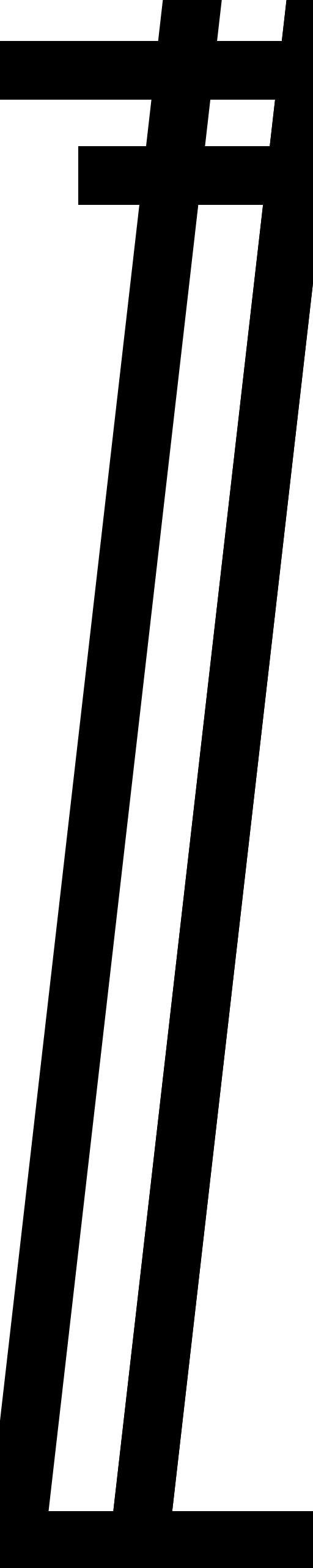
**Что дальше?**

**Голосовое взаимодействие  
с вебсайтами станет  
стандартом индустрии**

**Взаимодействие с  
сайтом исключительно  
голосом - как вариант  
(но не самый A11y)**

**А11у - это весело и  
удобно для всех**

**Сегодня ты пойдешь  
домой и поиграешься с  
этим API ;)**



**Мне нечего тебе  
сказать!**



**bit.ly/2MmuahZ**

Алексей  
Охрименко

Twitter: @Ai\_boy

