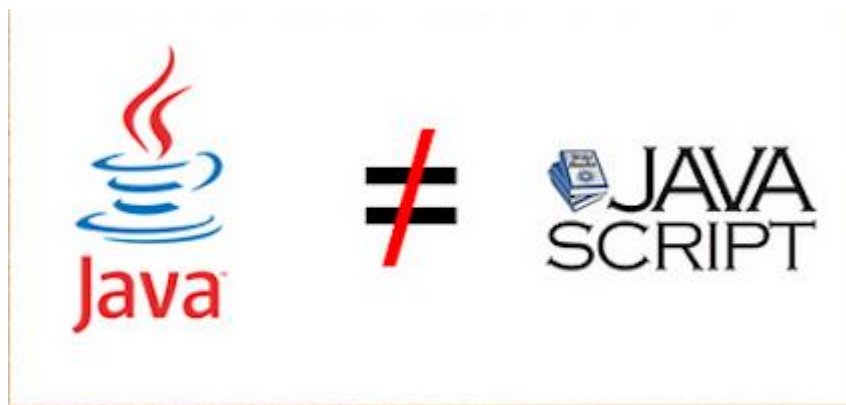


JS

Брендан Эйк



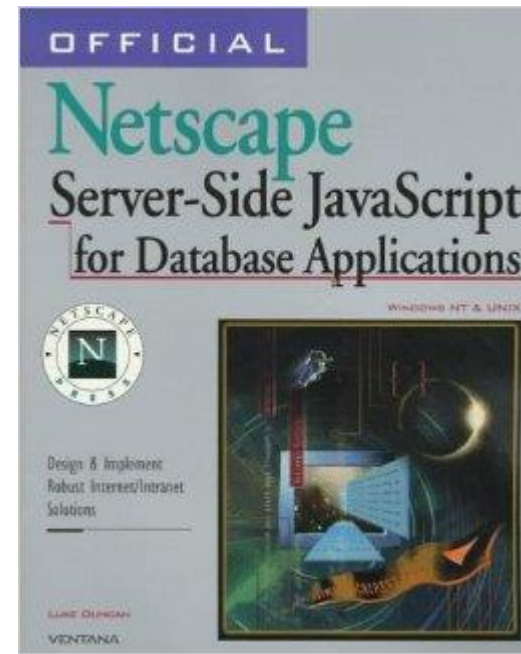
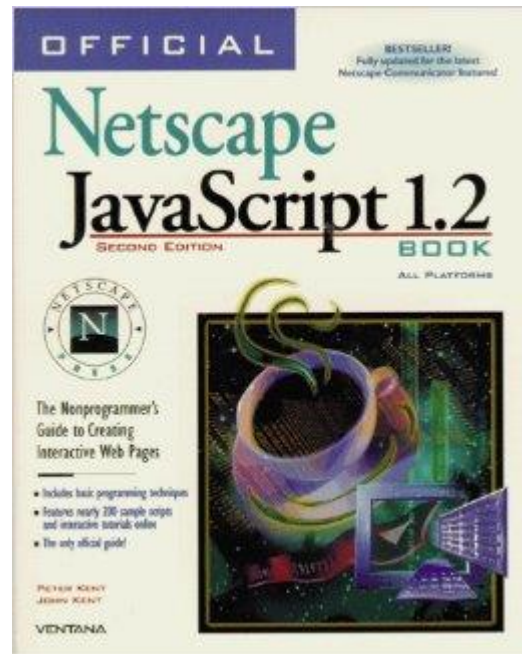
Май 1995



Айк, напиши за 10 дней!

- похож на Java (для простоты освоения)
- простой для начинающих
- «чтобы управлял в браузере Netscape всем, что там есть»

Netscape



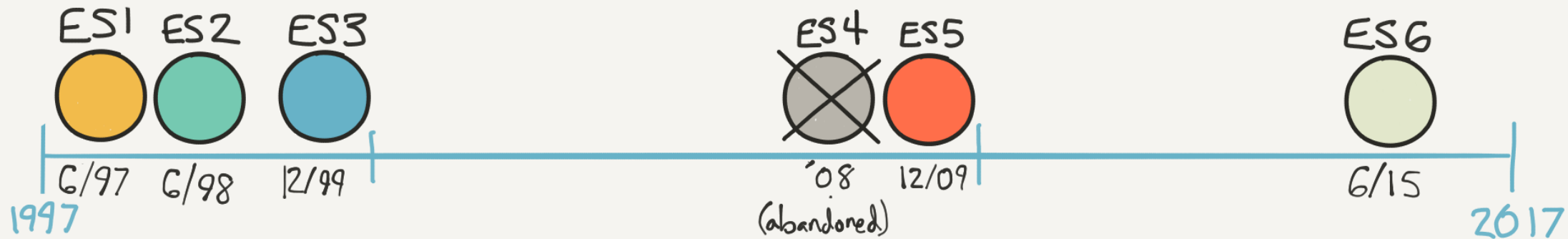
JS. Долгий путь.

- Mocha – LiveScript – JavaScript – ECMAScript
- ES3 – ES5 – ~~ES6~~ ES2015- ES2016- ES2017- ES2018 - ES2019 - ES2020

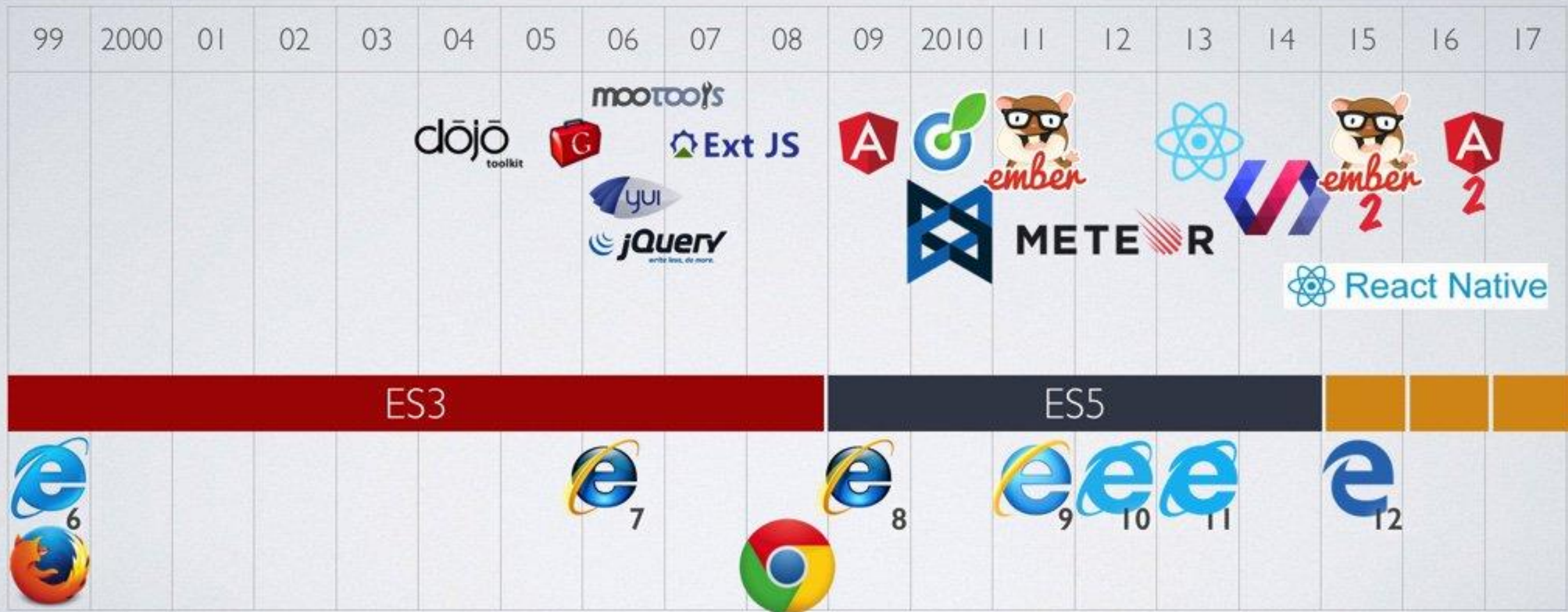


История

ECMAScript Releases

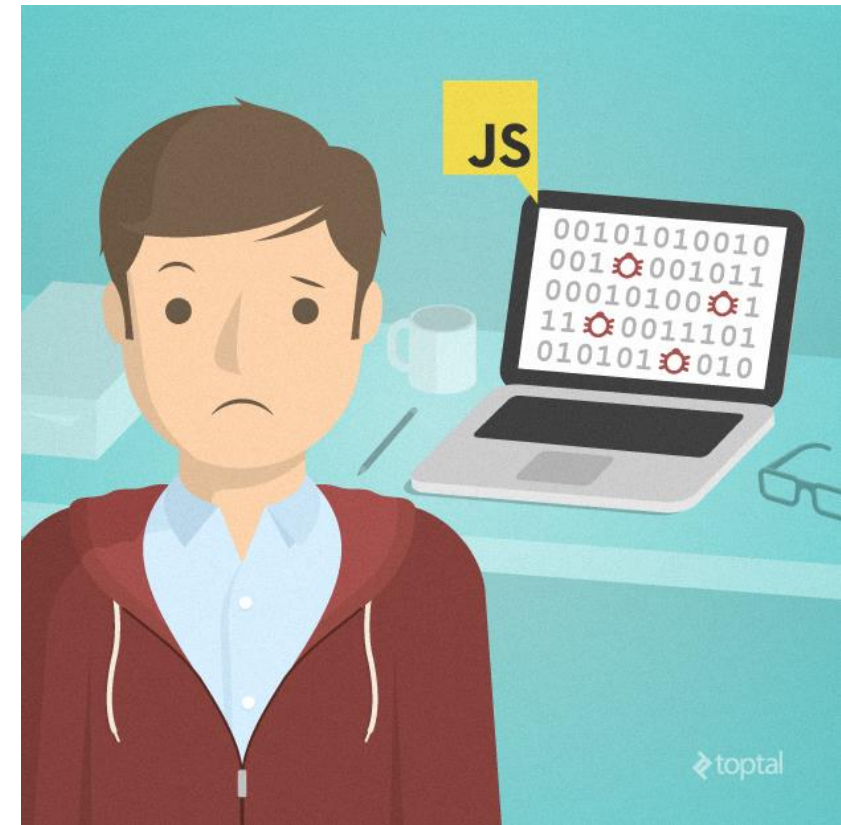


JS UI FRAMEWORKS



Недостатки

- ДИНАМИЧЕСКАЯ ТИПИЗАЦИЯ
- ПРОТОТИПНОЕ НАСЛЕДОВАНИЕ
- Неявные особенности
- Он кажется очень простым
- Открытый код
- (Условно) однопоточное исполнение (в браузерах)



«Достоинства»

- Он прикольный и модный *
- Он простой **
- Кроссплатформенный
- Неожиданно быстро работает
- Невероятно гибкий
- Быстро развивается
- Единственное решение для фронтенда
- Полная интеграция с HTML/CSS.

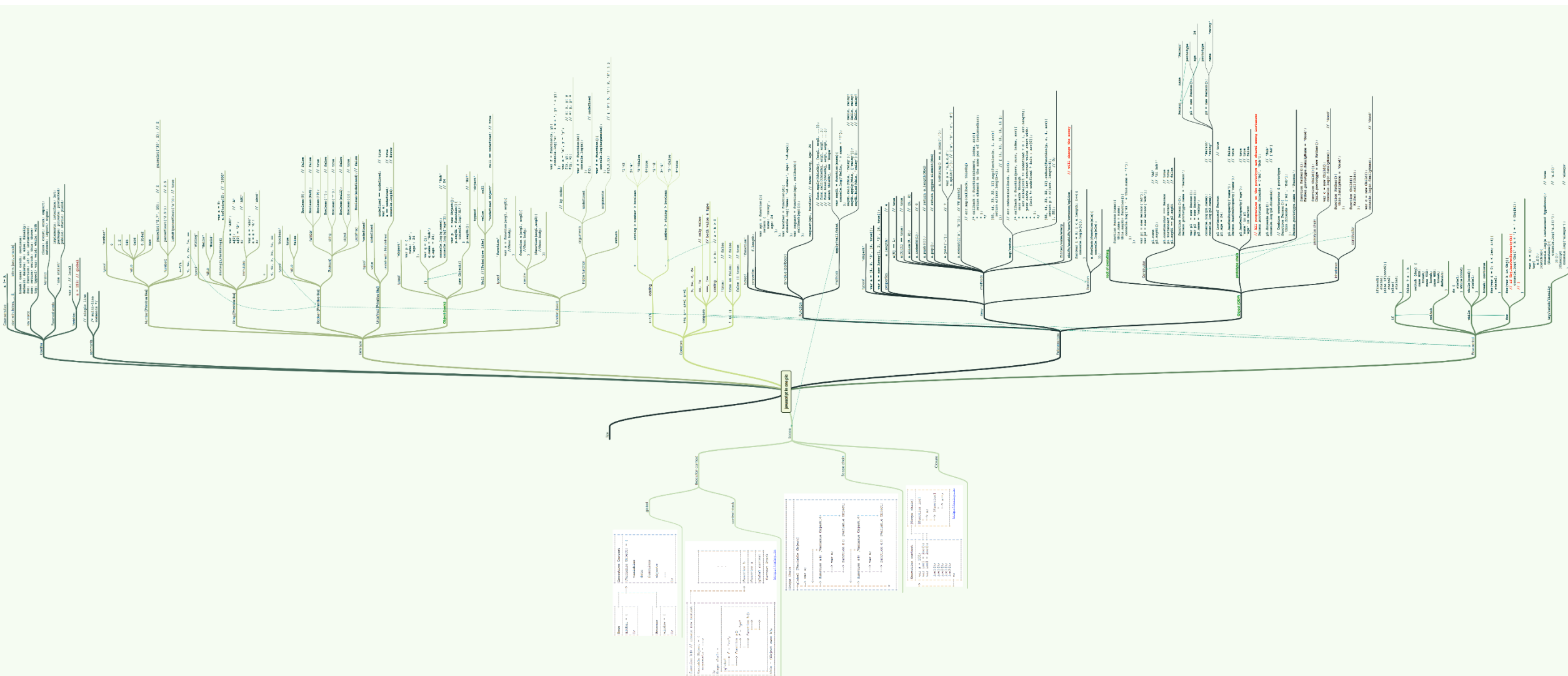
* - расхожее мнение в узких кругах

** - расхожее мнение в широких кругах



Что еще?

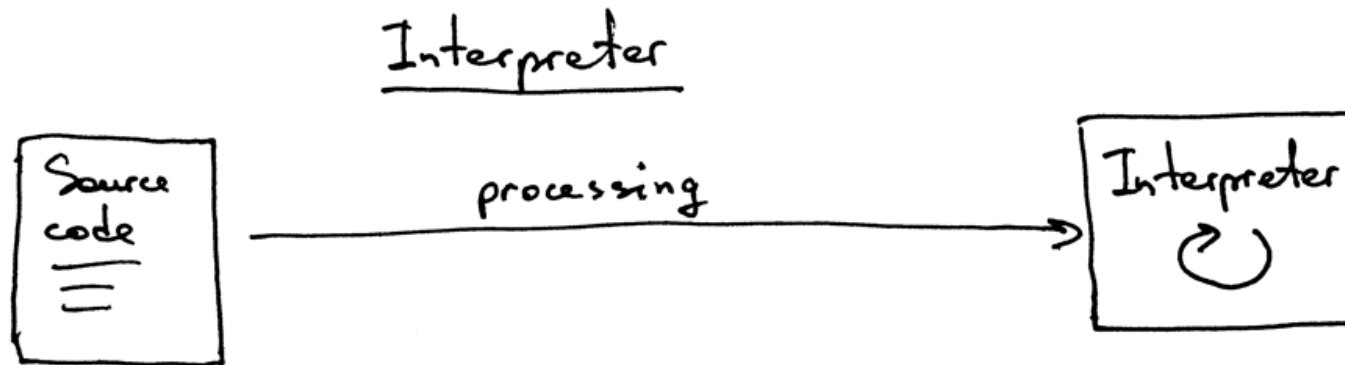
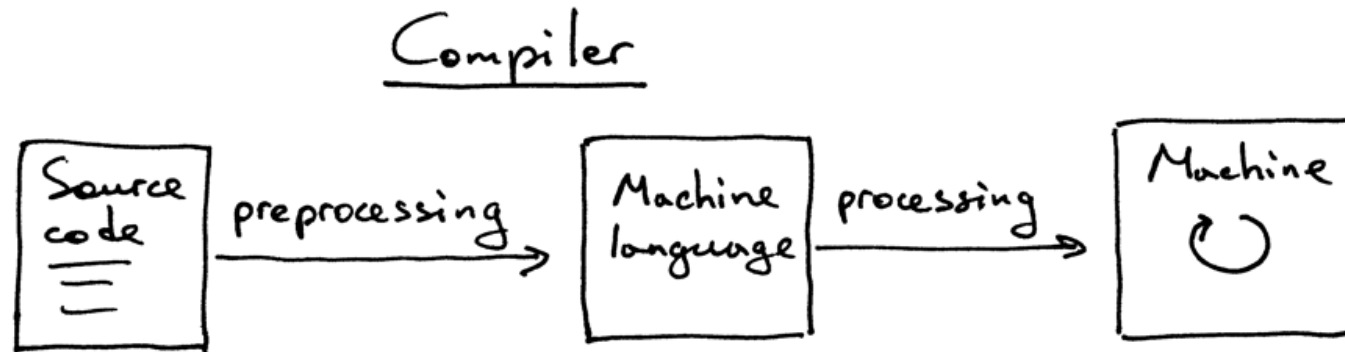
- **CoffeeScript** — «синтаксический сахар» (Ruby - сообщество)
- **Dart** — имеет независимую среду выполнения
- **TypeScript** — строгая типизация данных.



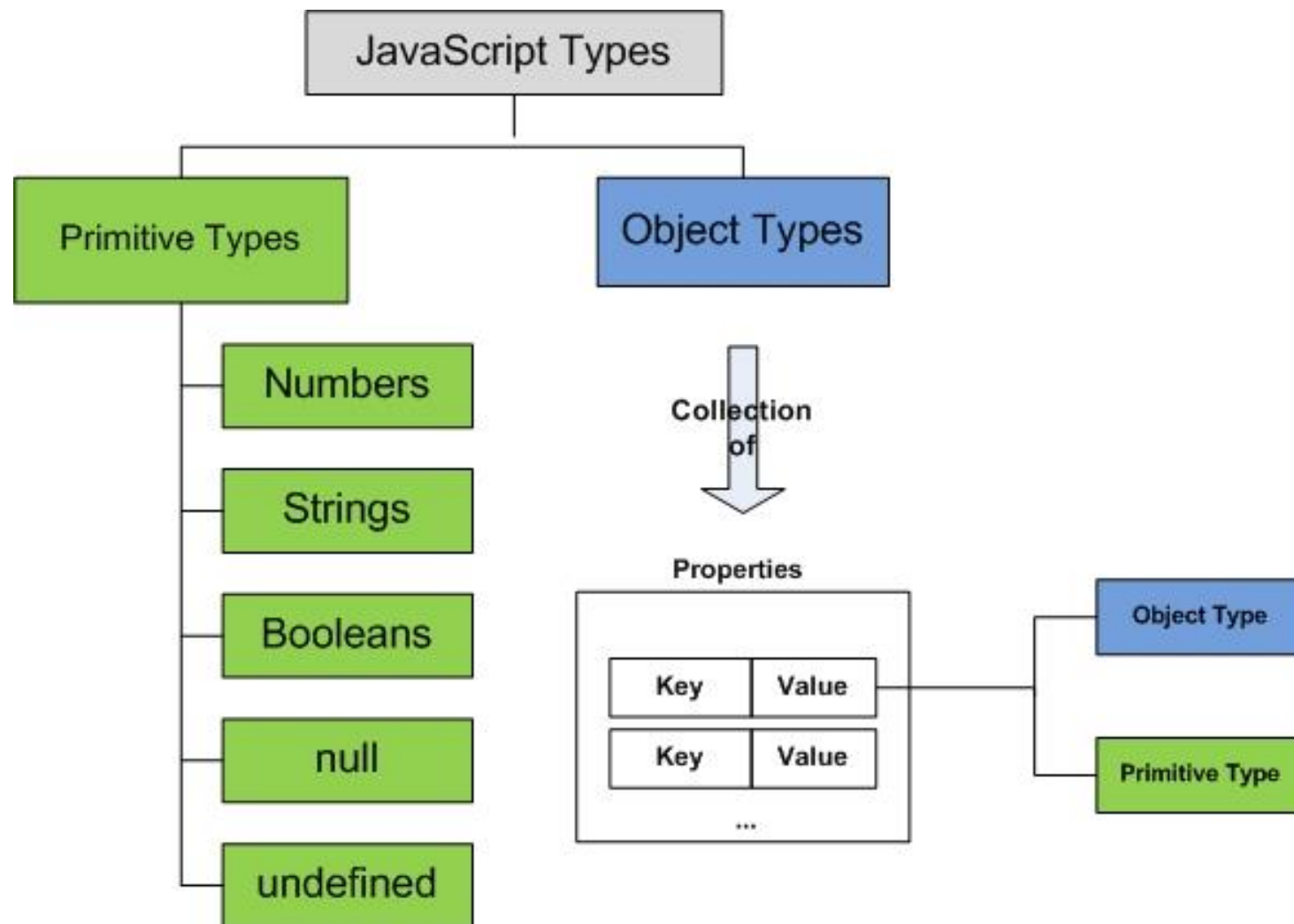


Давай. Вошли и вышли, приключение
на 20 минут.

Интерпретируемый



Типы данных



learn.javascript.ru

Основные моменты

- С-подобный синтаксис
- Типы `null` и `undefined`
- Приведение типа: к строке (+), к числу (==)
- Область видимости `var` – функция
- Объект – ассоциативный словарь
- Функция – объект, `function declaration` и `function expression`
- Замыкание
- Контекст вызова `this`
- Функция-конструктор
- Функциональное и прототипное ООП и классы

Замыкания

```
function makeCounter() {  
  var currentCount = 1;  
  
  return function() {  
    return currentCount++;  
  };  
}
```

Замыкания

```
var counter = makeCounter();
```

```
var counter2 = makeCounter();
```

```
alert( counter() );
```

```
alert( counter() );
```

```
alert( counter() );
```

```
alert( counter2() );
```



Функция-конструктор

```
1  function User(name) {  
2      this.name = name;  
3  
4      this.sayHi = function() {  
5          alert( "Моё имя: " + this.name );  
6      };  
7  }  
8  
9  var ivan = new User("Иван");  
10  
11 ivan.sayHi(); // Моё имя: Иван  
12
```

Функция-конструктор. Псевдо-класс

```
1  function User(firstName, lastName) {  
2    // вспомогательная переменная  
3    var phrase = "Привет";  
4  
5    // вспомогательная вложенная функция  
6    function getFullName() {  
7      return firstName + " " + lastName;  
8    }  
9  
10   this.sayHi = function() {  
11     alert( phrase + ", " + getFullName() ); // использование  
12   };  
13 }  
14  
15 var vasya = new User("Вася", "Петров");  
16 vasya.sayHi(); // Привет, Вася Петров
```

ООП. Базовый класс

```
1 // ----- Класс-Родитель -----  
2 // Конструктор родителя пишет свойства конкретного объекта  
3 function Animal(name) {  
4     this.name = name;  
5     this.speed = 0;  
6 }  
7  
8 // Методы хранятся в прототипе  
9 Animal.prototype.run = function() {  
10     alert(this.name + " бежит!");  
11 }
```

ООП. Наследник

```
13 // ----- Класс-потомок -----
14 // Конструктор потомка
15 function Rabbit(name) {
16     Animal.apply(this, arguments);
17 }
18
19 // Унаследовать
20 Rabbit.prototype = Object.create(Animal.prototype);
21
22 // Желательно и constructor сохранить
23 Rabbit.prototype.constructor = Rabbit;
24
25 // Методы потомка
26 Rabbit.prototype.run = function() {
27     // Вызов метода родителя внутри своего
28     Animal.prototype.run.apply(this);
29     alert( this.name + " подпрыгивает!" );
30 };
31
32 // Готово, можно создавать объекты
33 var rabbit = new Rabbit('Кроль');
34 rabbit.run();
```




Инструменты

- Текстовый редактор (IDE)
 - Блокнот
 - Sublime
 - WebStorm
 - Visual Studio
 - VS Code
 - Инструментарий разработчика Chrome/Mozilla
- NPM
- Модули Node
 - Webpack / GULP.JS / GRUNT.JS
 - YEOMAN.JS

NPM – node package manager

<https://www.npmjs.com/>

- `npm install gulp` – локально
- `npm install -g gulp` – глобально (прописывается в PATH)
- `npm install gulp --save` – в `package.json`

С третьей версии – плоская структура

