




JavaScript

Введение. Основы языка.

Жигалов Сергей


Откуда   Булочная № 5 


Новопесчаная улица...

ул. Новопесчаная, 2,...

На ближайшее время 



Требования (1) 

+79120409889 

Тарифы

Комментарий к заказу

Эконом от 199 руб 

Для расчёта приблизительной стоимости

☐ Такси с жёлтыми номерами

☐ Некурящий водитель

☒ Детское кресло...

Детское кресло...

Детское кресло (2-5 лет)

Бустер (от 4 лет)

☐ Перевозка велосипеда

☐ Кондиционер

☐ Перевозка животного

☐ Квитанция об оплате



7 минут — среднее
время подачи такси




ЗАГРУЗИТЕ НА
Google play



Загрузите в
App Store



Загрузите в
Windows Phone Store

 Такси онлайн



Java

- написать код
- скомпилировать
- упаковать в апплет
- подключить

JavaScript

- написать код
- ~~скомпилировать~~
- ~~упаковать в апплет~~
- подключить

- Синтаксис → C, C++, Java
- Динамическая типизация → Perl
- Ссылки на функции → Lisp
- Наследование через прототипы → Self

Брэндон Аик

разработал JavaScript для Netscape 1995



Брэндон Айк

... он должен был быть написан
за 10 дней, а иначе мы бы
имели что-то похуже JS...

Брэндон Айк

... В то время мы должны были
двигаться очень быстро, т.к.
знали, что Microsoft идет за
нами...

Брэндон Айк

... JS был обязан «выглядеть как Java», только поменьше, быть эдаким младшим братом-простаком для Java...

JSON

JavaScript Object Notation

Дуглас Крокфорд, 2001

~~XML~~

```
<!--?xml version="1.0" encoding="UTF-8" ?-->
<coffee_shop>
  <name>Works</name>
  <cashlesspayment>true</cashlesspayment>
  <capacity>3</capacity>
  <barista>
    <persone>
      <name>Лёша</name>
      <favourite>cappuccino</favourite>
    </persone>
    <persone>
      <name>Лиза</name>
      <favourite>tea</favourite>
    </persone>
  </barista>
</coffee_shop>
```

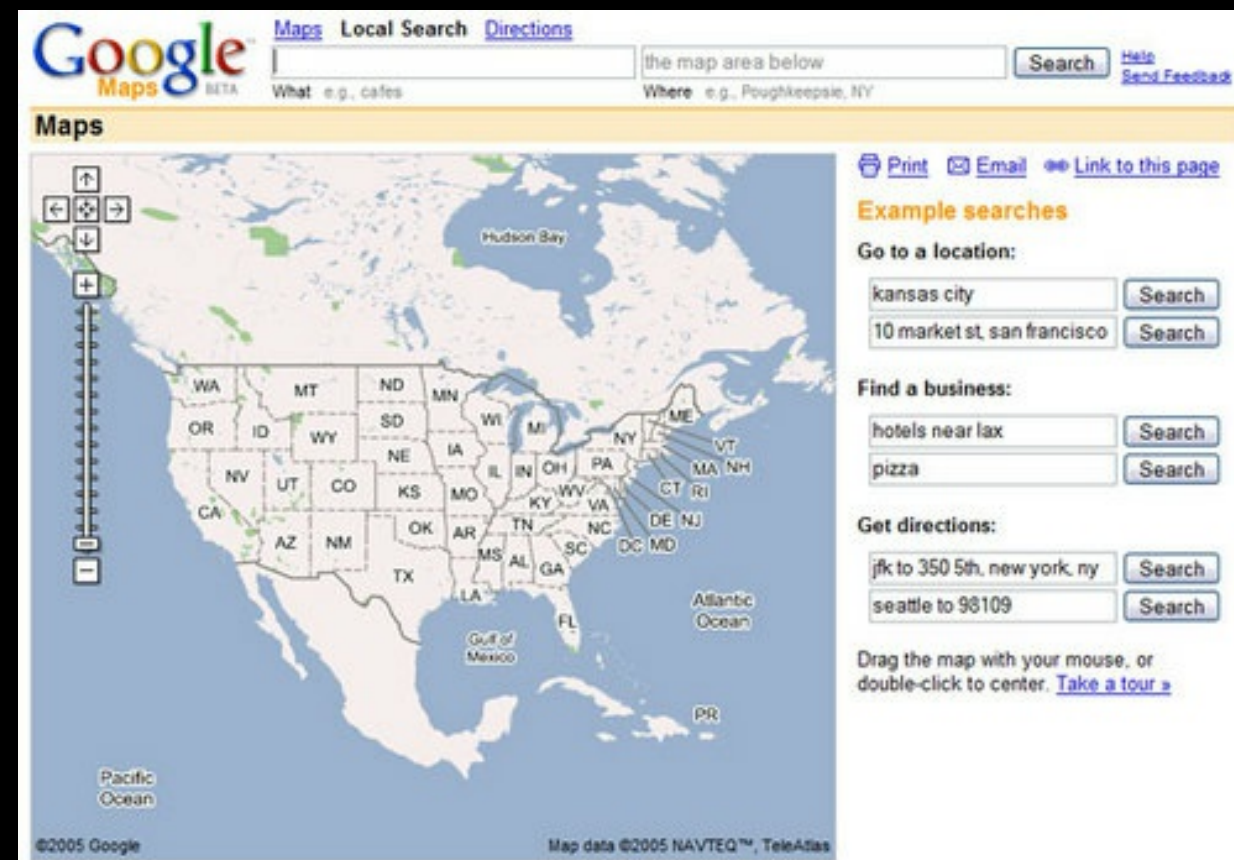
JSON

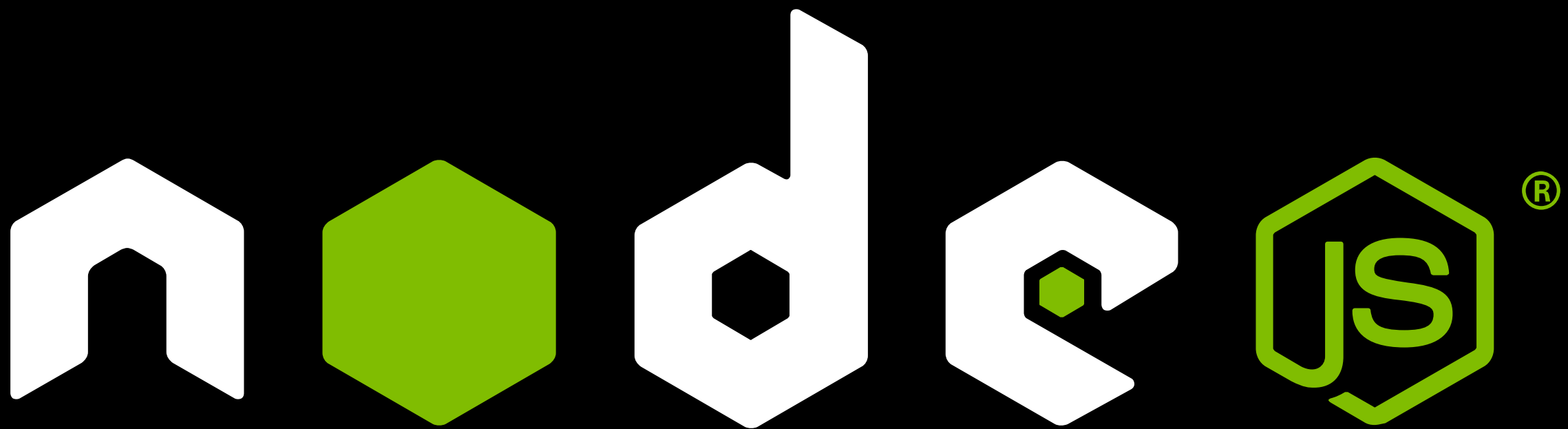
```
{
  "name": "Works",
  "cashlessPayment": true,
  "capacity": 3,
  "barista": [
    {
      "name": "Лёша",
      "favourite": "cappuccino"
    },
    {
      "name": "Лиза",
      "favourite": "tea"
    }
  ]
}
```

AJAX

(Asynchronous JavaScript and XML)

Джеймс Гаррет, 2005





NodeJS

- Linux / Windows / MacOS / ...
- Единая архитектура
- Один поток*
- Node Package Manager



JavaScript

- Браузер
- Сервер
- Базы данных



Типы данных

Типы данных

- числа
- строки
- булевы величины
- неопределённые величины
- объекты и массивы
- функции

Числовой

123 // 123

12.3 // 12.3

Числовой. Системы счисления

0x11 // 17

0b11 // 3

0o11 // 9

Числовой. Экспонента

```
2^64 = 18446744073709552000
```

```
Number.MAX_SAFE_INTEGER  
// 9007199254740991
```

```
2.998e8  
// 2.998 × 10^8  
// 299800000
```

Числовой. Бесконечность

```
Infinity  
-Infinity
```

```
Infinity - 1 === Infinity // true
```


Числовой. Не число

NaN

```
0/0 // NaN
```

```
Infinity - Infinity // NaN
```

```
'один' / 'два' // NaN
```

Строковый

```
' строка текста'  
"строка текста"  
'中文 español русский \' '
```

Логический

```
true  
false
```

Неопределенное значение

```
undefined  
null
```

```
var a;  
console.log(a); // undefined
```

typeof

```
typeof 0;  
// 'number'
```

```
typeof '0';  
// 'string'
```

Преобразование к числу

```
parseInt(string, radix);
```

```
parseInt('17', 10); // 17  
parseInt('123');    // 123  
parseInt('11', 2);   // 3  
parseInt('a1', 10);  // NaN  
parseInt('2b', 10);  // 2
```



```
parseFloat(строка)
```

```
parseFloat('3.14'); // 3.14
```

```
parseFloat('314e-2'); // 3.14
```

```
parseFloat('a1'); // NaN
```

Переменные

Переменные

```
var studentsCount;  
studentsCount = 98;
```

```
studentsCount; // 98;
```

Переменные

```
var studentsCount = 98;
```

```
var studentsCount,  
    language = 'JavaScript';
```

Именованные переменные

oneMoment

a-z _ \$

1moment

Именованние переменных

soSo

A-Z a-z 0-9 _ \$

so-so

Именованние переменных

```
// Переменная  
var currentTime;
```

```
// Константа  
var MILLISECONDS_IN_DAY;
```


Зарезервированные слова

break	do	try	while
case	else	new	with
catch	finally	return	
continue	for	switch	
debugger	function	this	
default	if	throw	
delete	in	instanceof	
typeof	var	void	

class	enum	extends	super
const	export	import	

Именованные переменные

```
spisokDruzey; // x  
tsena;        // x
```

```
friends;      // ok  
price;        // ok
```

Именованные переменные

```
h, w; // x  
friendsListWithNameAndAge; // x
```

```
height, width; // ok  
myFriends; // ok
```

Именованние переменных

```
my_friends;    // x
```

```
myFriends;    // ok
```

Именованные переменные

```
isCorrect = true;  
totalCount = 47;  
friends = [];
```

Комментарии

```
// это короткий комментарий
```

```
/* а это длинный комментарий  
написанный в несколько строк */
```

Комментарии

```
/* ах этот длинный комментарий ... */  
var weather = 'cold';  
  
console.log(weather); // cold
```

Комментарии

```
var weather = 'sunny';  
  
/*/* ах этот длинный комментарий ... */  
var weather = 'cold';*/  
  
console.log(weather);  
// SyntaxError: Unexpected token *
```


Операторы

Унарные

++ (инкремент)
-- (декремент)
+ (унарный плюс)
- (унарный минус)
! (логическое НЕ)

Инкремент

```
var a = 1;  
var b = a++; // b === 1, a === 2
```

```
var a = 1;  
var b = ++a; // b === 2, a === 2
```

Бинарные

Арифметические

* (умножение)

/ (деление)

% (остаток от деления)

+ (сложение)

– (вычитание)

+ (сложение строк)

Сложение

2 + 3 = 5

'«JavaScript — это простой, но ' +
'изящный язык, который является ' +
'невероятно мощным для решения ' +
'многих задач» © Джон Резиг'

Сравнения

< (меньше)

<= (меньше или равно)

> (больше)

>= (больше или равно)

== (проверка на равенство)

!= (проверка на неравенство)

=== (проверка на идентичность)

!== (проверка на неидентичность)

Логические

&& (И)

|| (ИЛИ)

Присваивание

= (присваивание)

*=, /=, +=, -=, &=, ^=, |=
(присваивание с операцией)

```
var a = 1;  
a = a + 1;  
a += 1;
```

Условные операторы

```
if (language === 'JavaScript') {  
    likes = likes + 1;  
} else {  
    likes = likes - 1;  
}
```

```
likes = language === 'JavaScript' ?  
    likes + 1 :  
    likes - 1;
```

Условные операторы

```
switch (language) {  
    case 'JavaScript':  
        likes++;  
        break;  
    case 'C++':  
    case 'Java':  
        break;  
    default:  
        likes--;  
}
```

Точка с запятой

Точка с запятой

```
function getTrue() {  
    return true;  
}  
  
getTrue();    // true
```

Точка с запятой

```
function getTrue() {  
    return  
    true;  
}  
  
getTrue();    // undefined
```

Строгий режим

Строгий режим

```
'use strict';  
  
// этот код будет работать  
// по современному стандарту ES5
```

Спецификация


```
text = 'hello';
```

```
text; // 'hello'
```

```
'use strict';
```

```
text = 'hello'; // ReferenceError:  
                // text is not defined
```

Пример запуска

```
console.log('Hello, world!');
```

В браузере

```
// Ctrl + Alt + J
```

```
console.log('Hello, world!');
```

<https://nodejs.org>

NodeJS

```
// index.js  
console.log('Hello, world!');
```

```
$ node index.js
```