Лекция 3

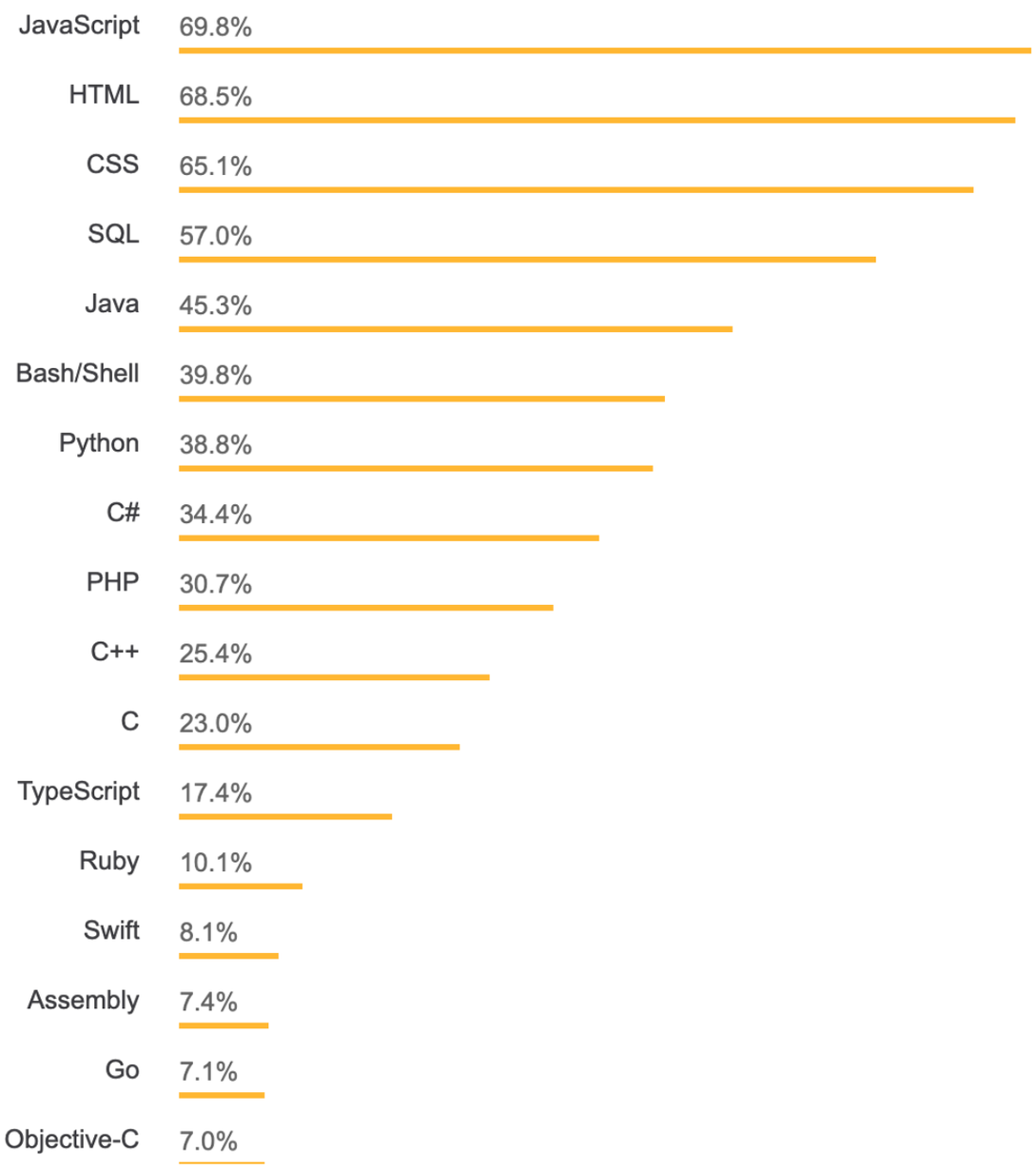
Основы Javascript

Ответы на вопросы

Это - примерный список технологий, которые преподаются на курсе.

JavaScript

All Respondents | Professional Developers



История

- 1995 - год на свет появляется LiveScript (Брэнд Айк)
- 1997 - ECMAScript начало стандартизации (Netscape, ECMA)
- 1999 - ECMAScript3 продолжении
- 2005 - … появляется множество lib (JQuery, MooTools …)
- 2009 - ECMAScript 5
- 2009 - 2010 Angular Node
- 2011 - React
- 2012 - TypeScript (Microsoft)
- 2013 - 2015 Electron
- 2015 - ECMAScript 6
- 2016 - ECMAScript 7

Основы

Javascript - интерпретируемый язык программирования

Куда писать код?

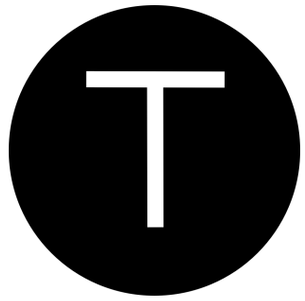
- Файлы с расширением -.js

- В файл с расширением -.html в тег <script> some code </script>

<script>

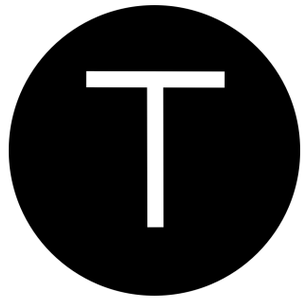
```
<html>

<body>
    <script>
        console.log('Первый');
        console.log('Второй');
        console.log('Первый, расчет окончен');
    </script>
</body>

</html>
```

.js



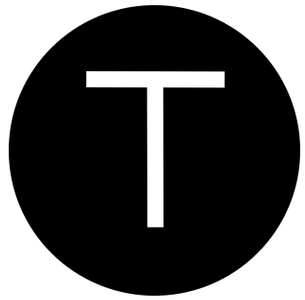
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script type="text/javascript" async="" src="https://www.google-analytics.com/analytics.js"></script>
    <title>Document</title>
</head>
<body>

</body>
</html>
```

Среда исполнения JavaScript



- Браузер;
- Веб-сервер;

Переменные



Переменная – это «именованное хранилище» для данных. Мы можем использовать переменные для хранения товаров, посетителей и других данных.

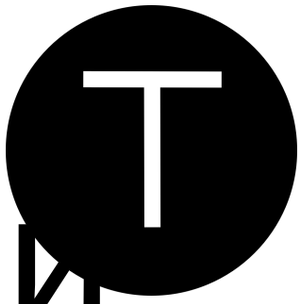
Для создания переменной в JavaScript используйте ключевое слово - let.

Ключевые слова для объявления переменных:

- var
- let
- const

https://learn.javascript.ru/variables

Отличия в объявлении



Ключевые слова для объявления переменных:

- var
- let
- const

const - создает константу.

var, let - создает обычную переменную.

Отличия var и let:

- var - всплывает (hoisting) на 1 стадии интерпретации кода.
- var - каждый раз пересоздает переменную в новой ячейке памяти при изменении значения.
- var - можем создать несколько переменных с одним именем, с let так не получится.
- var - имеет функциональную область видимости, а let - блочную.

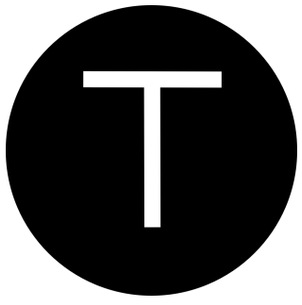
https://learn.javascript.ru/var

Пример

```
let THE_BEST_TEACHERS = ‘Ruben and Yury’;
console.log(THE_BEST_TEACHERS);
```

```
let THE_BEST_TEACHERS;
// THE_BEST_TEACHERS === undefined

THE_BEST_TEACHERS = ‘Ruben and Yury’;
console.log(THE_BEST_TEACHERS);
```

Типы в JavaScript



- number
- string
- boolean
- undefined
- null
- object
- Symbol (ES6)
- BigInt (ES7)

Примитивы

- Number
- String
- Boolean
- Null
- Undefined

Ссылочный тип данных

```
const obj = {

    a: 5,

    str: ‘строка’

}
```

Обращение к объекту

```
const obj = {

    a: 5,

    str: ‘строка’

}
```

```
console.log(obj.a);

console.log(obj.str);

console.log(obj[‘str’]);
```

Преобразование типов

Преобразование в число:

- `Number('713')` `// 100`
- `Number.parseInt('713')` `// 713`
- `+'713'` `// 713`
- `Number.parseFloat('713.713')` `// 713.713`

Преобразование типов

- Строковые: String(true) // ‘true’
- Численные: происходит в случае использования математических функциях и выражениях. Number(value)

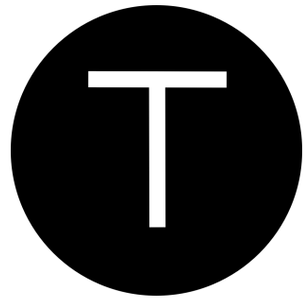
Значение	Преобразуется в...
`undefined`	`NaN`
`null`	`0`
`true / false`	`1` / `0`
`string`	Пробельные символы по краям обрезаются. Далее, если остаётся пустая строка, то получаем `0`, иначе из непустой строки «считывается» число. При ошибке результат `NaN`.

- Логические: Boolean(value)

Правило преобразования:

- Значения, которые интуитивно «пустые», вроде 0, пустой строки, null, undefined и NaN, становятся false.
- Все остальные значения становятся true.

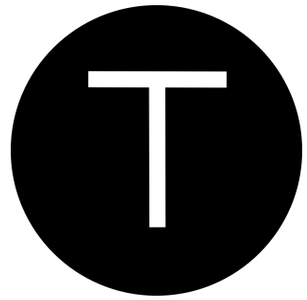
Методы чисел



```
Number.isNaN()
Number.isFinite()
Number.isInteger()
Number.isSafeInteger()
Number.toInteger()
Number.parseFloat()
Number.parseInt()
Number.toFixed()
Number.toLocaleString()
Number.toPrecision()
Number.toSource()
Number.toString()
Number.valueOf()
```

https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Number

Методы строк

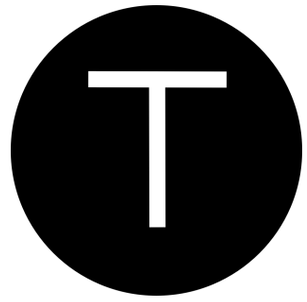


```
string.fromCharCode()
string.fromCodePoint()
string.raw()
string.charAt()
string.charCodeAt()
string.codePointAt()
string.concat()
string.includes()
string.endsWith()
string.indexOf()
string.lastIndexOf()
string.localeCompare()
string.match()
string.normalize()
string.quote()
```

```
string.repeat()
string.replace()
string.search()
string.slice()
string.split()
string.startsWith()
string.substr()
string.substring()
string.toLocaleLowerCase()
string.toLocaleUpperCase()
string.toLowerCase()
string.toSource()
string.toString()
string.toUpperCase()
string.trim()
string.trimLeft()
string.trimRight()
string.valueOf()
```

https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/String

Методы объекта



```
Object.assign()

Object.create()

Object.defineProperty()

Object.defineProperties()

Object.freeze()

Object.getOwnPropertyDescriptor()

Object.getOwnPropertyNames()

Object.getOwnPropertySymbols()

Object.getPrototypeOf()

Object.is()
```

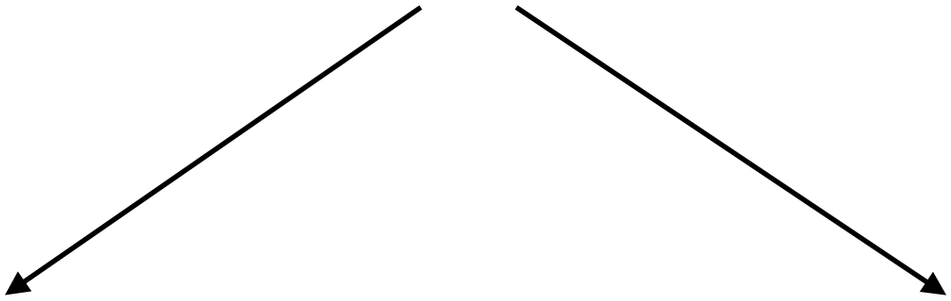
https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Object

Шаблонные строки

```
const name = ‘Виталик’

const str = `${name} пошел учить Frontend`

// Виталик пошел учить Frontend
```

```
+ мультилайновый текст

const str =

`asdasdasd

asdasdasd`;
```

Операторы сравнения



Нестрогое равенство - сравнение только по значению.

```
'1' == 1; // true

true == 1; // true

false == ''; // true
```

Строгое равенство - по значению и по типу.

```
'1' === 1; // false

true === 1; // false

false === ''; // false
```

Оператор if

```
const NUMBER = 5;

if (NUMBER > 2) {

      console.log(NUMBER);
}
```

if - преобразует условие в скобках к логическому типу, и в зависимости от преобразования либо выполняет код в фигурных скобках либо нет.

Оператор if/else

```
const NUMBER = 1;

if (NUMBER > 2) {
    console.log(NUMBER);
} else if (NUMBER <= 2) {
    console.log('Меньше двух');
} else {
    console.log('NaN');
}
```

Тернарный оператор

- упрощенная версия оператора if

Синтаксис: условие ? true : false

А еще есть switch - изучите сами.

https://learn.javascript.ru/ifelse
https://learn.javascript.ru/switch

Логические операторы

```
// И | AND
a && b;

// ИЛИ | OR
a || b;

// НЕ | NOT
!a;
```

Циклы

- while - цикл по условию

```
let i = 0;
while (i < 10) {
    console.log(i);
    i++;
}
```

- for - цикл по счетчику

```
for (let i = 0; i < 10; i++) {
    console.log(i);
}
```

https://learn.javascript.ru/while-for#%D1%86%D0%B8%D0%BA%D0%BB-while

Массивы

- это структура данных, которая является упорядоченным объектом, где используются ключи (индексы) от 0 до длины массива (length).

```
const arr = [1,2,3,4,5,6,7,8];

console.log(arr[5]); // 6

console.log(arr.length); //
```

```
const arr = [1, 2, 3, 4, 5];

for (let i = 0; i < arr.length; i++) {
    console.log(arr[i]);
}
```

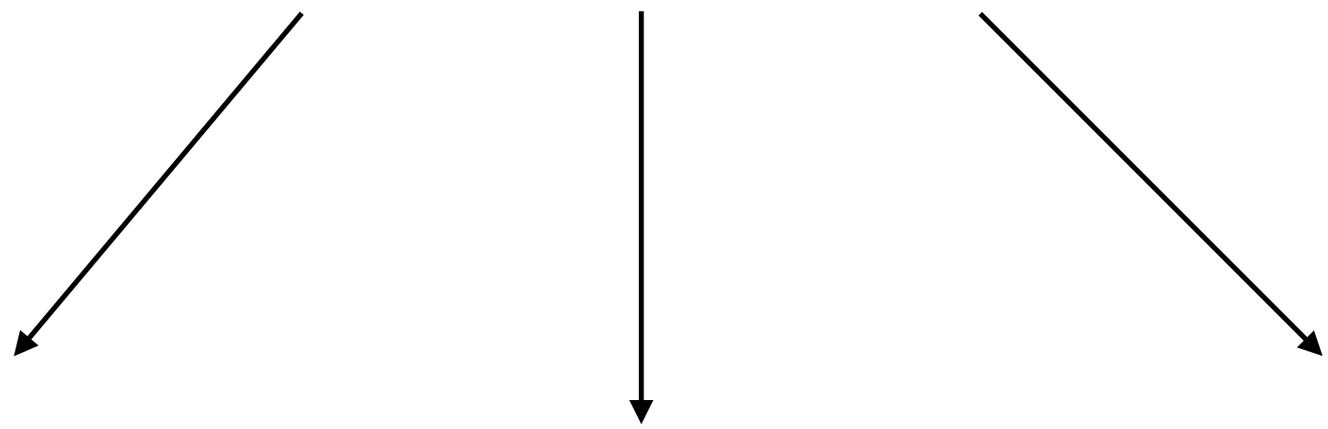
Массивы

```
const arr = [1, 2, 3, 4, 5];

for (let i = 0; i < arr.length; i++) {
    console.log(arr[i]);
}
```

```
// 1
// 2
// 3
// 4
// 5
```

Функции

- это объект, использующийся для того, чтобы сложить один и тот же алгоритм, и вызывать в случае частой необходимости. Предназначен для уменьшения одинакового кода, а также разделения логики.



Function Declaration Function Expression Arrow Function

https://learn.javascript.ru/function-basics

Функции

Синтаксис FD:

```
function sum(a,b) {
    return a+b;
}
sum(1,4) // 5
```

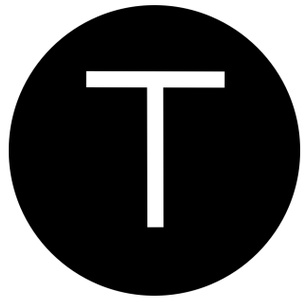
Синтаксис FE:

```
const sum = function(a,b) {
    return a+b;
}
sum(1,4) // 5
```

Синтаксис AF:

```
const sum = (a,b) => {
    return a+b;
}
sum(1,4) // 5
```

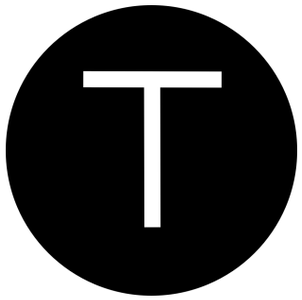
Функции



В теле функции мы можем делать ВСЕ тоже что и просто в файле с расширением .js

Более подробно поговорим о функциях на следующей лекции.

Что нужно посмотреть самим?



Все ссылки находящиеся в презентации.

Лабораторная работа 3

- Доделать лабораторную работу 2.
- Решить 3 задачки из папки lab3 в нашем репозитории. Разумеется предварительно спулить себе все изменения.
- Добиться полного прохождения тестов в Pull Request-е.
- Заполнить отчет по 3-ей лабораторной работе.