

JavaScript





- Среда выполнения JavaScript, основанная на движке V8 из браузера Chrome
- Использует неблокирующую модель ввода-вывода
- Пакетная экосистема Node.js и npm является самой большой экосистемой библиотек в мире

JS

JavaScript

- Динамически
типизированный
- Слабо типизированный



// Statically typed

```
int a = 10;
```

```
string b = 'asdf';
```

// Dynamically typed

```
let a = 10;
```

```
a = 'asdf';
```



// Weakly typed

```
let a = 1 + '2'; // '12'
```

Сравнения



```
const num = 1;  
const strNum = '1';  
  
const sum = num + strNum; // 11  
  
if (num == strNum) {...} // true  
  
if (num === strNum) {...} // false
```


Primitive values (immutable)

- Undefined (`undefined`), used for unintentionally missing values.
- Null (`null`), used for intentionally missing values.
- Booleans (`true` and `false`), used for logical operations.
- Numbers (`-100`, `3.14`, and others), used for math calculations.
- Strings (`"hello"`, `"abracadabra"`, and others), used for text.
- Symbols (uncommon), used to hide implementation details.
- BigInts (uncommon and new), used for math on big numbers.

Objects and Functions (mutable)

- Objects (`{ }` and others), used to group related data together.
- Functions (`x => x * 2` and others), used to refer to code.

Объявление переменных

```
const var1 = 1234;  
let var2 = {  
  key1: 'value1',  
  key2: ['massive', 4321]  
};
```

Объявление функций



```
function foo(arg1, arg2) {  
    const result = arg1 + arg2;  
  
    return result;  
}
```

Что будет?



```
const num = 10;
```

```
function double(n) {  
    return n * 2;  
}
```

```
double(n);  
console.log(n);
```

Что будет?



```
const obj = { name: 'Valera' };  
  
function getWork(item) {  
    item.work = 'Student';  
}  
  
getWork(obj);  
console.log(obj);
```

Что будет?




```
const arr = ['Tom', 'Jerry'];
```

```
arr[0] = 'Tomas';
```

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

Что будет?



```
const str = 'Yikes';  
  
str[0] = 'L';  
  
console.log(str);
```




```
const original = {
  name: 'doc',
  meta: {
    title: 'Hello world'
  }
};

function makeCopy(orig) {
  const copy = {
    name: orig.name,
    meta: orig.meta
  };

  copy.meta.title = 'New World';

  return copy;
}

const copied = makeCopy(original);
```

Truthy, falsy значения

Falsy:

- false
- 0
- -0
- '' - empty string
- null
- undefined
- NaN

Truthy:

- Все остальное

Циклы и логические операторы

```
for (let i = 0; i < 10; ++i) {  
  console.log('Hello world!');  
  if ((i > 5 && true) || false) {  
    break;  
  }  
}
```

Стрелочные функции



```
const arrowFunction = (arg1, arg2) => {  
  const result = arg1 + arg2;  
  
  return result;  
}
```



```
const arrowFunction = arg1 => {  
  const result = arg1 * 2;  
  
  return result;  
}
```



```
const arrowFunction = arg1 => arg1 * 2;
```

Перебирающие методы массива

```
const even = [1, 2, 3, 4].filter(item => item % 2 === 0);  
  
const squared = [1, 2, 3, 4].map(item => item * item);  
  
const sum = [1, 2, 3, 4].reduce((accumulator, item) =>  
  accumulator + item, 0);
```