

1. Основы Javascript

Типы данных, объявление переменных,
циклы, операторы, функции

Объявление переменных

```
var x = 6;  
let u = 89;  
const r = 4;
```

Типы данных

Примитивы

- **String:** `let y = '78;;6y ';`
- **Number:** `y = 56;`
- **Boolean:** `y = true;`
- **Undefined:** `y = undefined;`
- **Null:** `y = null;`
- **Symbol:** `y = Symbol();`

Типы данных

Комплексные

Object

```
y = {  
  ty: 'foo',  
  cy: '[[['  
};
```

Array

```
y = [  
  'ty',  
  67,  
  89,  
  { f: 34 }  
];
```

Типы данных

Как узнать тип данных?

```
typeof '56'; // string
```

```
typeof 56; // number
```

```
typeof null; // null
```

```
typeof undefined; // undefined
```

```
typeof [67,67]; // array
```

```
typeof {y: 56, p: 34}; // object
```

```
typeof true; // boolean
```

Типы данных

Как узнать тип данных?

```
typeof '56'; // string
typeof 56; // number
typeof null; // object
typeof undefined; // undefined
typeof [67,67]; // object
typeof {y: 56, p: 34}; // object
typeof true; // boolean
```

Array.isArray([67,45,89])

Операторы

Сравнение

> <
== !=
<= >=

Присваивание

= +=
*= /=
-= %=

Строгое сравнение

!== ===

Арифметические

++ --
+ -
/ %
*

Сравнение

Простая запись if/else

```
let y = 6;  
  
if (y > 9) {  
    console.log(y);  
} else {  
    console.log(2);  
}
```


Сравнение

Триадная запись

```
(y > 9) ? console.log(y) : console.log(2);
```

Присваивание и триада

```
let t = 2;  
let y = (t > 9) ? 5 : false;
```

Приведение типов

Строковое

```
"gfh" + "456"  
45 + '78'  
true + '7'  
true + ''
```

Логическое

Интуитивно «пустые»
всегда равны false

```
if (undefined) {  
  
    // your code  
}
```

```
!!NaN // false  
!!0 // false  
!!' ' // true  
!!5 // true
```

Численное

```
+ "56" // 56  
+ [ 'dfgdf' ] // Nan  
+ true // Nan  
+ null // 0  
+ undefined // Nan
```

```
let y = undefined;  
  
if (y == 0) {  
  
    // code, that  
    // you'll never reach.  
    // Mwahaha  
}
```

При сравнении с
помощью оператора
происходит численное
приведение

Приведение ТИПОВ

```
let k = undefined;
```

```
k == undefined; // true
```

```
let k = null;
```

```
k == null; // true
```

```
let v = +'force'; // NaN
```

```
isNaN(v); // true  
v == NaN; // false
```

```
let e = 7 / 0; // Infinity
```

```
isFinite(e); // false  
e == Infinity; // true
```

```
if ( +"f" ) {  
    // some code  
    // u can't  
    // reach  
}
```

ЦИКЛЫ

For

```
for (let k = 0; k < 5; k++) {  
    console.log(k);  
}
```

For in

```
let obj = {y: 67, ui: 4};  
for (let key in obj) {  
    console.log(key);  
}
```

For of

```
let arr = [45, 67, 7, 89];  
for(let item of arr) {  
    console.log(item);  
}
```

Циклы

While

```
let t = 6;  
while(t < 10) {  
  
    console.log(t);  
    t++;  
}
```

Do ... while

```
var i = 0;  
do {  
  
    console.log(i);  
    i++;  
} while (i < 3);
```

ЦИКЛЫ

Конструкции внутри циклов

Остановка выполнения цикла

```
let arr = [45, 67, 7, 89];
for (let item of arr) {
    console.log(item);
    if (item < 10) {
        break;
    }
}
```

Переход на следующую итерацию

```
let arr = [45, 67, 7, 89];
for (let item of arr) {
    if (item < 10) {
        continue;
    }
    console.log(item);
}
```

ЦИКЛЫ

Конструкции внутри циклов

Метка для break/continue

```
outer: for (var i = 0; i < 3; i++) {  
    for (var j = 0; j < 3; j++) {  
        var input = prompt('Значение в координатах ' + i + ', ' + j, '');  
  
        // если отмена ввода или пустая строка –  
        // завершить оба цикла  
        if (!input) break outer; // (*)  
    }  
}  
alert('Готово!');
```

Конструкция CASE

```
let k = 7;

switch(k) {

    case 5: console.log(5); break;

    case 8:

    case 3:

    case 2: console.log(true); break;

    case 7: console.log(1000); break;

    default: console.log('some string');

}
```


Функции

Именованная

```
function mayTheForce(padavan) {  
    return 'be with you,' + padavan + '!';  
}
```

Анонимная

```
var mayTheForce = function (padavan) {  
    return 'be with you,' + padavan + '!';  
}
```

Мгновенного исполнения

```
(function mayTheForce(padavan) {  
    return 'be with you,' + padavan + '!';  
})('tyeer')
```

ES6

```
var mayTheForce = padavan => `be with you, ${padavan}!`;
```

Задачи

1. Есть массив функций. Каждая выводит в консоль строку. Сделать вызов всех функций с помощью перебора массива. For, For .. of.
2. Есть объект со случайными значениями. Вывести в консоль только значения-строки.
3. Тот же объект - вывести только результат выполнения функций - если они есть в объекте.
4. Есть массив. Заполнен числами. Сделать перебор массива. При 3 - выводить в консоль «There is threeeee!», при 6 и 89 - «I don't know wha to do», при 100 и других - возводить в квадрат и выводить в консоль.
5. Клонировать массив и вывести в консоль его клон.

Читать и пользоваться



<https://learn.javascript.ru/>

<https://www.w3schools.com/js/default.asp>