

Язык JavaScript, лекция 2: Типы данных и операции

Пожидаев М. С.

14 октября 2022 г.

Определение переменных

```
var i = 0; //просто число
const s = "Tomsk"; //неизменяемая строка
var e = /[0-9]{4}/; //регулярное выражение
var o = {name: "Anna"}; //объект
var a = ["Anna", "Maria"]; //массив
```

Зарезервированные значения

Специальные соглашения для пустого значения

- ▶ `undefined` — значение не существует;
- ▶ `null` — значение существует, но оно не совпадает ни с одним значением из допустимого множества.

Работа с переменными

```
const v = "Anna";  
const a = (!!v)?"Yes":"No"; //тернарный оператор  
const a = `Name is ${v}`; //вставить значение  
const a = typeof v; //получить тип
```

Операции со строками

- ▶ `.length` — получить длину строки;
- ▶ `.substring()` — получить подстроку;
- ▶ `.trim()` — получить значение строки без пробелов в начале и в конце;
- ▶ `.match()` — наложить на строку регулярное выражение;
- ▶ `.replaceAll()` — произвести подстановку в строке.

Массивы

```
const a = [];  
a[0] = "Anna";  
a.push("Maria");  
a["name"] = "Anna"; //ассоциированный массив  
delete a["name"]; //удалить значение
```

Операции с массивами

- ▶ `.length` — получить длину массива;
- ▶ `.push()` — добавить новый элемент в массив;
- ▶ `Array.isArray()` — проверить, что аргумент является массивом.

JSON

Пример:

```
{"name": "Anna", items: [15, 20]}
```

- ▶ `JSON.stringify()` — сгенерировать строку из объекта;
- ▶ `JSON.parse()` — построить объект из строки.

Циклы

Выполнение повторяющихся действий

Традиционный цикл:

```
for(let i = 0; i < count; i++)
```

Цикл по элементам массива:

```
for(i of v)
```

Цикл по ключам ассоциативного массива:

```
for(i in v)
```

Модули

Подключить модуль:

```
const config = require("./config");
```

Экспортировать объект:

```
module.exports = function() {  
};
```

Функции

```
function f(a) {  
  return a * a;  
}
```

```
const f = function(a){  
  return a * a;  
};
```

```
const f = (a)=>{  
  return a * a;  
};
```

Конструкторы

```
function Friend(name) {  
  this.name = name;  
  this.likePresentsForBirthdays = true;  
}  
  
const Anna = new Friend("Anna");  
console.log(Anna.name); //напишет "Anna"
```

Лямбда-выражения

и замыкания для них

```
var a = false;
```

```
const f = ()=>{  
  a = true;  
};
```

```
f();  
fooBar(f);
```

Прототипическое наследование

Подход к реализации объектной модели

```
const a = {  
  proba: ()=>{ console.log("proba"); }  
};  
const b = function() {};  
b.prototype = a;  
const c = new b();  
c.proba();
```

Спасибо за внимание!

Всё о курсе: <https://marigostra.ru/materials/js.html>

E-mail: msp@luwrain.org

Канал в Телеграм: <https://t.me/MarigostraRu>