

# Objetos

Object

```
1  var pessoa = {
2    nome: 'Joãozinho',
3    idade: 28,
4    profissao: 'Desenvolvedor',
5    frequente: true
6  }
7
8  pessoa.nome; // 'Joãozinho'
9  pessoa.frequente; // true
```

Conjunto de variáveis e funções, que são chamadas de propriedades e métodos.

Propriedades e métodos consistem em nome (chave) e valor

## Métodos

Method

```
1  var quadrado = {
2    lados: 4,
3    area: function(lado) {
4      return lado * lado;
5    },
6    perimetro: function(lado) {
7      return this.lados * lado;
8    }
9  }
10
11 quadrado.lados; // 4
12 quadrado.area(5); // 25
13 quadrado.perimetro(5); // 20
```

É uma propriedade que possui uma função no local do seu valor.

Method ES6

```
1  var quadrado = {
2    lados: 4,
3    area(lado) {
4      return lado * lado;
5    },
6    perimetro(lado) {
7      return this.lados * lado;
8    }
9  }
```

Abreviação de area: `function() {}` para `area() {}`, no ES6+

## Organização

Objetos servem para organizar o código em pequenas partes reutilizáveis.

Organização

```
1  Math.PI; // 3.14
2  Math.random(); // número aleatório
3
4  var pi = Math.PI;
5  console.log(pi); // 3.14
```

## Criar um Objeto

Create

```
1  var carro = {};
2  var pessoa = {};
3
4  console.log(typeof carro); // 'object'
```

Um objeto é criado utilizando as chaves `{}`.

## Dot Notation Get

Get

```
1 var menu = {
2   width: 800,
3   height: 50,
4   backgroundColor: '#84E'
5 }
6
7 var bg = menu.backgroundColor; // '#84E'
```

Acesse propriedades de um objeto utilizando o ponto `.`

## Dot Notation Set

Substitua o valor de uma propriedade utilizando `.` e o `=` após o nome da mesma.

Set

```
1 var menu = {
2   width: 800,
3   height: 50,
4   backgroundColor: '#84E',
5 }
6
7 menu.backgroundColor = '#000';
8 console.log(menu.backgroundColor); // '#000'
```

## Adicionar Propriedades e Métodos

Basta adicionar um novo nome e definir o valor.

Add

```
1 var menu = {
2   width: 800,
3 }
4
5 menu.height = 50;
6 menu.position = 'fixed';
```

## Palavra-chave this

`this` irá fazer uma referência ao próprio objeto.

this

```
1 var height = 120;
2 var menu = {
3   width: 800,
4   height: 50,
5   halfHeight() {
6     return this.height / 2;
7   }
8 }
9
10 menu.halfHeight(); // 25, sem o this, seria 60
```

## Protótipo e Herança

O objeto herda propriedades e métodos do objeto que foi utilizado para criar o mesmo.

Prototipo

```
1 var menu = {
2   width: 800,
3 }
4
```

```
5 menu.hasOwnProperty('width') // true
6 menu.hasOwnProperty('height') // false
```

`hasOwnProperty` é um método de `Object`

## Tudo é Objeto

Strings, Números, Boolean, Objetos e mais, possuem propriedades e métodos. Por isso são objetos.

```
String

1 var nome = 'Joãozinho';
2 nome.length; // 9
3 nome.charAt(1); // 'o'
4 nome.replace('inho', 'ão'); // 'Joãozão'
5 nome; // 'Joãozinho'
```

Uma string herda propriedades e métodos do construtor `String()`

```
Number

1 var altura = 1.8;
2
3 altura.toString(); // '1.8'
4 altura.toFixed(); // '2'
```

O número, temporariamente, é envolvido em um **Objeto (coerção)**, tendo acesso assim as suas propriedades e métodos

## Funções

```
Function

1 function areaQuadrado(lado) {
2     return lado * lado;
3 }
4
5 areaQuadrado.toString();
6
7 // "function areaQuadrado(lado) {
8 //   return lado * lado;
9 // }"
10
11 areaQuadrado.length; // 1
```

## Elementos do DOM

```
<a class="btn">Clique</a>
```

```
DOM

1 var btn = document.querySelector('.btn');
2 btn.classList.add('azul') // adiciona a classe azul
3 btn.innerText; // 'Clique'
4 btn.addEventListener('click', function() {
5     console.log('Clicou')
6 })
```

 **Exercício**

- nomeie 3 propriedades ou métodos de strings
- nomeie 5 propriedades ou métodos de elementos do DOM
- nomeie 3 métodos para fazer converão de números.