

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  
2  
3  
4
```

```
5
6  var menu = {
    width: 800,
  }

  menu.height = 50;
  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
```

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'); // 'Joãoão'
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 conversão de números.

Arrays e Loops

Array

É um grupo de valores geralmente relacionados. Servem para guardarmos diferentes valores em uma única variável.

Array

```
1 var videoGames = ['Switch', 'PS5', 'XBox'];
2 videoGames[0] // Switch
3 videoGames[2] // Xbox
```

Métodos e Propriedades de uma Array

Methods

```
1 var videoGames = ['Switch', 'PS5', 'XBox'];
2
3 videoGames.pop(); // Remove o último item e retorna ele
4 videoGames.push('3DS'); // Adiciona ao final da array
5 videoGames.length; // 3
```

For Loop

O for loop possui 3 partes, **início**, **condição** e **incremento**

For

```
1 for (var numero = 0; numero < 10; numero++) {
2   console.log(numero);
3 }
4 // Retorna de 0 a 9 no console
```

While Loop

While

```
1 var i = 0;
2 while (i < 10) {
3   console.log(i);
4   i++;
5 }
6 // Retorna de 0 a 9 no console
```

Arrays e Loops

Loops em arrays

```
1 var videoGames = ['Switch', 'PS5', 'XBox', '3DS'];
2 for (var i = 0; i < videoGames.length; i++) {
3   console.log(videoGames[i]);
4 }
```

? O que acontece no código anterior?




retorna cada item do array no console

Break

Break

```
1 var videoGames = ['Switch', 'PS4', 'XBox', '3DS'];
2 for (var i = 0; i < videoGames.length; i++) {
3   console.log(videoGames[i]);
4   if(videoGames[i] === 'PS4') {
5     break;
6   }
7 }
```

```
6 }  
7
```

 O loop irá parar caso encontro e palavra break

forEach

forEach é um método que executa uma função para cada item da Array.

É uma forma mais simples de utilizarmos um loop com arrays (ou array-like)

for each

```
1 var videoGames = ['Switch', 'PS5', 'XBox', '3DS'];  
2 videoGames.forEach(function(item) {  
3     console.log(item);  
4 });  
5 // O argumento item será atribuído dinamicamente
```

Podemos passar os seguintes parâmetros `item`, `index` e `array`.



Exercício

- Crie uma array com os anos que o Brasil ganhou a copa
1958, 1962, 1970, 1994, 2002
- Interaja com a array utilizando um loop, para cada ano mostrar no console a seguinte mensagem:
O brasil ganhou a copa de ANO;
- Interaja com um loop nas frutas abaixo e pare ao encontrar Pera

```
var frutas = ['Banana', 'Maçã', 'Pera', 'Uva', 'Melância']
```

- Coloque a última fruta do array acima em uma variável, sem remover a mesma da array, depois imprima essa variável no console.