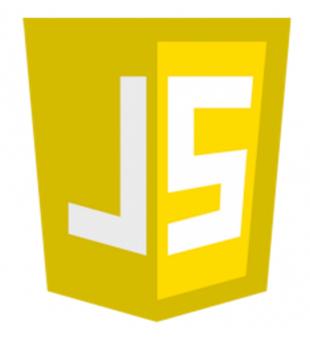
JavaScript



Ana Carolina N R Gracioso

carol.nrg@gmail.com

Agenda

- Arrays;
- Objetos: propriedades, métodos e classes.

- Assim como em outras linguagens de programação, o JavaScript dá suporte a manipular uma estrutura de dados que armazena uma coleção de elementos.
- Essa estrutura de dados também é conhecida como variável indexada, vetor (para arranjos unidimensionais) e matriz (para arranjos bidimensionais).

Criando Arrays:

```
var cars = new Array("Honda", "Volvo", "BMW");
ou simplesmente:
```

var cars = ["Honda", "Volvo", "BMW"];

Acessando Arrays:

```
document.write(cars[0]);
document.write(cars);
```

Em JavaScript, Arrays são um tipo particular de objeto:

```
var pessoa = ["Fulano", "de Tal", 28];
document.write("Nome: "+pessoa[0]+"<br>
document.write("Sobrenome: "+pessoa[1]+"<br>");
document.write("Idade: "+pessoa[2]+"<br>");
```

 Porém, é preferível acessar os objetos por nomes do que por índices

```
var pessoa = {
   primeiroNome:"Fulano",
   sobreNome:"de Tal",
   idade:28};

document.write("Nome:
"+pessoa.primeiroNome+"<br>");
document.write("Sobrenome:
"+pessoa.sobreNome+"<br>");
document.write("Idade: "+pessoa.idade+"<br>");
```

Elementos de um Array:

```
myArray[0] = Date.now; // objeto
myArray[1] = myFunction; // função
myArray[2] = myCars; // outro array
```

- Propriedades e Métodos:
 - length: tamanho do array;
 - sort(): ordena o array;
 - toString(): converte um array em string;
 - join(" * "): converte em string permitindo especificar o separador;

Métodos:

- push(): adiciona elementos no final do array e retorna o novo tamanho;
- pop(): remove elementos do final do array e retorna o elemento removido;
- shift(): remove elementos do início do array;
- unshifts(): adiciona elementos no início do array;

Excluindo elementos:

```
delete frutas[0];
```

Emendando um array:

```
var frutas= ["Banana", "Laranja", "Manga"];
frutas.splice(2, 0, "Limão", "Kiwi");
2 — posição de inserção
0 — quantidade de elementos a serem excluídos
```

Alterando o conteúdo de um elemento HTML

```
<!DOCTYPE html>
<html>
<body>
Click me to change my HTML content (innerHTML).
<script>
function myFunction() {
   document.getElementById("demo").innerHTML = "Paragraph"
changed!";
</script>
</body>
</html>
```

Criando elementos HTML

```
<!DOCTYPE html>
<html>
<body>
Click the button to create a P element with some text.
<button onclick="myFunction()">Try it</button>
<script>
function myFunction() {
    var e = document.createElement("P");
   var t = document.createTextNode("This is a paragraph.");
    e.appendChild(t);
   document.body.appendChild(e);
</script>
</body>
</html>
```

Criando elementos HTML

```
<!DOCTYPE html>
<html>
<body>
Click the button to create a P element with some text.
<button onclick="myFunction()">Try it</button>
<script>
function myFunction() {
    var e = document.createElement("P");
    var t = document.createTextNode("This is a paragraph.");
    e.appendChild(t);
    e.setAttribute("id", "meuParagrafo");
    e.setAttribute("style", "font-size: 20px; text-align:center;");
    document.body.appendChild(e);
</script>
</body>
</html>
```

Recuperando elementos pelo ID

```
<!DOCTYPE html>
<html>
<body>
       \langle t.r \rangle
                      Linha 1 Coluna 1
                      Linha 2 Coluna 2
              Linha 2 Coluna 1
                      Linha 2 Coluna 2
              <button onClick="alterarTexto()">Alterar Texto</button>
       <script>
              function alterarTexto(){
                      celulas = document.querySelectorAll('[id^=1]');
                      for (var i = 0; i < celulas.length; i++) {</pre>
              celulas[i].innerHTML = "Novo Texto";
       </script>
</body>
</html>
```

 Considerando o objeto carro, podemos dizer que todos os carros possuem propriedades semelhantes cujos valores podem diferenciarse. Todo carro executa ações semelhantes, mas em momentos diferentes.

Object	Properties	Methods
	car.name = Fiat	car.start()
	car.model = 500	car.drive()
	car.weight = 850kg	car.brake()
	car.color = white	car.stop()

Propriedades do objeto:

```
var pessoa = {
    nome:"Fulano",
    sobrenome:"de Tal",
    idade:28
};
```

document.write(pessoa.nome);

Métodos do objeto:

```
var pessoa = {
    nome:"Fulano",
    sobrenome:"de Tal",
    idade:28,
    nomeCompleto: function() {
       return this.nome + " " + this.sobrenome;
    }
};
```

document.write(pessoa.nomeCompleto());

 Classes: definição modelo das propriedades e métodos de um objeto

```
function Pessoa () {
    var nome;
    this.setNome = function(vNome){
         this.nome = vNome;
    this.getNome = function(){
         return this.nome;
```

```
var pessoa = new Pessoa();
pessoa.setNome("Fulano");
document.write(pessoa.getNome());
```

Atividade

1. Implemente um script com um array cujos valores sejam adicionados dinamicamente pelo usuário utilizando-se formulário. A cada nova inserção exiba todos os dados cadastrados em ordem alfabética em uma lista ordenada do html.

Referências

- http://www.w3schools.com/js/js_arrays.asp
- http://www.w3schools.com/js/js_objects.asp
- http://tableless.com.br/introducao-a-programacaoorientada-a-objetos-em-javascript/