MULTIMÍDIA

- JavaScript – Básico - Parte I

Agenda

- JavaScript – Básico - Parte I
- Exemplos
- Exercícios

História do JavaScript

- Início da Internet páginas eram pouco ou nada interativas;
- Navegar através de links e enviar informações através de formulários era basicamente tudo o que se podia fazer.

História do JavaScript

- Visando o potencial da Internet para o público geral e a necessidade de haver uma interação maior do usuário com as páginas, a Netscape, criadora do navegador mais popular do início dos anos 90, criou o Livescript, uma linguagem simples que permitia a execução de scripts contidos nas páginas dentro do próprio navegador.

História do JavaScript

- Aproveitando o iminente sucesso do **Java**, que vinha conquistando cada vez mais espaço no mercado de desenvolvimento de aplicações corporativas, a Netscape logo rebatizou o Livescript como **JavaScript** num acordo com a Sun para alavancar o uso das duas.

Como o JavaScript funciona?

- O **JavaScript** é uma linguagem de script multiplataforma, orientada a objetos.
- É uma linguagem Case Sensitive.
- Dentro de um ambiente de hospedagem, o **JavaScript** pode ser conectado aos objetos de seu ambiente para proporcionar um controle programático sobre elas.

Como o JavaScript funciona?

- O **núcleo do JavaScript** contém um **núcleo de objetos**, como **Array**, **Date**, e **Math**, e um **núcleo de elementos de linguagem** como **operadores, estruturas de controle**, e **declarações**.

- O núcleo do **JavaScript** pode ser extendido para uma variedade de propósitos complementando-o com objetos adicionais; por exemplo:

Como o JavaScript funciona?

- O **lado do cliente no JavaScript** extende o núcleo da linguagem fornecendo objetos para controlar um navegador (Navigator ou outro navegador web) e seu Document Object Model (DOM).

- **Por exemplo**, extensões para o lado do cliente permitem a uma aplicação colocar elementos em um formulário HTML e responder a eventos de usuários como cliques de mouse, entrada de dados e navegação na página.

Como o JavaScript funciona?

- O **lado do servidor no JavaScript** extende o núcleo da linguagem fornecendo objetos relevantes à execução de **JavaScript** no servidor.

- **Por exemplo**, extensões do lado do servidor permitem a uma aplicação comunicar-se com um banco de dados relacional, proporcionar continuidade de informação de uma invocação da aplicação para outra, ou executar manipulações de arquivos em um servidor.

Como o JavaScript funciona?

- Nas linguagens de scripting normalmente elas são **linguagens interpretadas**, ou seja, não dependem de compilação para serem executadas.

- Essa **característica** é **presente** no **JavaScript**: o código é **interpretado** e **executado** conforme é **lido** pelo **navegador**, linha a linha, assim como o HTML.

Engine Javascript

- Uma **engine Javascript** é um programa ou um interpretador que executa código Javascript.

- Lista de **projetos** populares que estão implementando uma **engine Javascript:**

  - **V8** — open source, desenvolvido pelo Google, escrito em C++;

  - **Rhino** — gerenciado pela Mozilla Foundation, open source, desenvolvido inteiramente em Java;

Engine Javascript

- Lista de projetos populares que estão implementando uma engine Javascript:
  - **SpiderMonkey** — a primeira engine Javascript, que um dia empoderou o Netscape Navigator, e hoje empodera o Firefox;
  - **JavaScriptCore** — open source, comercializado como Nitro desenvolvido pela Apple para o Safari;
  - **KJS** — KDE's engine originalmente desenvolvido por Harri Porten para o projeto KDE Konqueror web browser;

Engine Javascript

- Lista de projetos populares que estão implementando uma engine Javascript:
  - **Chakra** (JScript9) — Internet Explorer;
  - **Chakra** (JavaScript) — Microsoft Edge;
  - **JerryScript** — é uma engine leve para a internet das coisas(IOT).

Porque a engine V8 foi criada?

- A **engine V8** que é construída pelo **Google** é open source e escrito em C++.
- Essa **engine** é usada dentro do **Google Chrome**.
- Ao contrário do resto das engines, no entanto, a **V8** é também usado pelo **popular** runtime do **Node.js**.

JavaScript hoje

- A tecnologia evoluiu para atender às mais diversas demandas que surgiam com a evolução da internet.

- Atualmente, é possível não apenas desenvolver sites e aplicativos ricos, mas também aplicativos para smartphones e até mesmo programas desktop.

JavaScript hoje

- Algumas tecnologias que surgiram com a evolução do JavaScript.
  - **jQuery** (biblioteca Javascript)
  - **React** (biblioteca Javascript)
  - **React Native** (biblioteca Javascript)
  - **Angular JS** (framework JavaScript)
  - **Vue.js** (framework JavaScript)
  - **Electron JS** (framework JavaScript)
  - **Phaser JS** (criar jogos HTML5 para desktop e dispositivos móveis)
  - **Node.js** (interpretador de JavaScript)

Console do Navegador

- Existem várias formas de executar códigos JavaScript em uma página. Uma delas é executar códigos no que chamamos de **Console**.

- No Chrome, é possível chegar ao **Console** apertando **F12** e em seguida acessar a aba "**Console**".

JAVASCRIPT – BÁSICO - PARTE I

JavaScript – Básico - Parte I

- **Operadores Aritméticos**
  - Podemos somar, subtrair, multiplicar e dividir como em qualquer linguagem:

  - Teste algumas contas digitando diretamente no console:

```
> 12 + 13
< 25
> 14 * 3
< 42
> 10 - 4
< 6
> 25 / 5
< 5
> 23 % 2
< 1
```

JavaScript – Básico - Parte I

- **Variáveis (var)**

  - Declarações de variáveis, onde quer que elas ocorram, são processadas antes que qualquer outro código seja executado.

  - O escopo de uma variável declarada com **var** é seu contexto atual em execução, o qual é a função a qual pertence ou, para variáveis declaradas fora de qualquer função, o escopo é o global.

JavaScript – Básico - Parte I

- **Variáveis**
  - Para armazenar uma valor para uso posterior, podemos criar uma **variável**:

```
> var resultado = 50 / 2
< undefined
```

  - No exemplo acima, guardamos o resultado de 50 / 2 na variável **resultado**. O resultado de criar uma variável é sempre **undefined**. Para obter o valor que guardamos nela ou mudar seu valor podemos fazer o seguinte:

```
> var resultado = 50 / 2
< undefined
> resultado
< 25
> resultado = resultado + 10
< 35
> resultado
< 35
```

JavaScript – Básico - Parte I

- **Variáveis**
  - Também podemos alterar o valor da variável usando as operações básicas com uma sintaxe bem compacta:

```
> var idade = 10; // undefined
> idade += 10; // idade vale 20
> idade -= 5; // idade vale 15
> idade /= 3; // idade vale 5
> idade *= 10; // idade vale 50
```

JavaScript – Básico - Parte I

- **Variáveis (let)**
  - Declara uma variável local no escopo do bloco atual, opcionalmente iniciando-a com um valor.
  - Foi pensando em trazer o escopo de bloco (tão conhecido em outras linguagens) que o **ECMAScript 6** destinou-se a disponibilizar essa mesma flexibilidade (e uniformidade) para a linguagem.
  - Através da palavra-chave **let** podemos declarar variáveis com escopo de bloco.

JavaScript – Básico - Parte I

- **Exemplo – variaveis.html**

```
10  <body>
11      <script>
12          function exemplo() {
13              console.log(x);
14              for (var x = 0; x < 5; x++) {
15                  console.log(x);
16              }
17              console.log(x);
18          }
19
20          function exemplo2() {
21              //console.log(x); //daria erro
22              for (let x = 0; x < 5; x++) {
23                  console.log(x);
24              }
25
26              //console.log(x); //daria erro
27          }
28          exemplo();
29          console.log(" ");
30          exemplo2();
31      </script>
32  </body>
```

JavaScript – Básico - Parte I

- **Operadores de Comparação**

Operador	Descrição
Igual (`==`)	Retorna verdadeiro caso os operandos sejam iguais.
Não igual (`!=`)	Retorna verdadeiro caso os operandos não sejam iguais.
Estritamente igual (`===`)	Retorna verdadeiro caso os operandos sejam iguais e do mesmo tipo. Veja também `Object.is` e igualdade em JS.
Estritamente não igual (`!==`)	Retorna verdadeiro caso os operandos não sejam iguais e/ou não sejam do mesmo tipo.

JavaScript – Básico - Parte I

- **Operadores de Comparação**

Maior que (`>`)	Retorna verdadeiro caso o operando da esquerda seja maior que o da direita.
Maior que ou igual (`>=`)	Retorna verdadeiro caso o operando da esquerda seja maior ou igual ao da direita.
Menor que (`<`)	Retorna verdadeiro caso o operando da esquerda seja menor que o da direita.
Menor que ou igual (`<=`)	Retorna verdadeiro caso o operando da esquerda seja menor ou igual ao da direita.

JavaScript – Básico - Parte I

- **Operadores Lógicos**

Operador	Utilização
Logical AND (`&&`)	*expr1* `&&` *expr2*
Logical OR (`||`)	*expr1* `||` *expr2*
Logical NOT (`!`)	`!`*expr*

JavaScript – Básico - Parte I

- **Tipos de Dados**:
- O mais recente padrão ECMAScript define **sete tipos de dados**:
  - Seis tipos de dados são os chamados primitivos:
    - **Boolean** - true e false.
    - **null** - Uma palavra-chave que indica valor nulo. Devido JavaScript ser case-sensitive, null não é o mesmo que Null, NULL, ou ainda outra variação.
    - **Undefined** - Uma propriedade superior cujo valor é indefinido.
    - **Number** - 42 ou 3.14159.
    - **String** - “Rodrigo"
    - **Symbol** - (novo em ECMAScript 6). Um tipo de dado cuja as instâncias são únicas e imutáveis.
    - **Object**

JavaScript – Básico - Parte I

- **Tipos de dados**
  - **Number**

```
var pi = 3.14159;
var raio = 20;
var perimetro = 2 * pi * raio
```

JavaScript – Básico - Parte I

- **Tipos de dados**
  - **String**

```
> var empresa = "Empresa X"
< undefined
> alert(empresa)
```

- Para exibirmos o valor empresa fora do console, podemos usar o comando **alert**;

Nota: É possível omitir o ponto e vírgula no final de cada declaração. A omissão de ponto e vírgula funciona no JavaScript devido ao mecanismo chamado **automatic semicolon insertion (ASI)**.

JavaScript – Básico - Parte I

- **A TAG SCRIPT**
  - Para inserirmos um código JavaScript em uma página, é necessário utilizar a tag <script>. **exemplo1.html**:

```
1   <!DOCTYPE html>
2   <html lang="pt-br">
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>JavaScript - Exemplo 1</title>
7
8       <script>
9           alert("Olá Mundo");
10
11      </script>
12  </head>
13  <body>
14  </body>
15  </html>
```

JavaScript – Básico - Parte I

- **exemplo2.html**

```
1   <!DOCTYPE html>
2   <html lang="pt-br">
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>JavaScript - Exemplo 2</title>
7       <script>
8           alert("Olá Mundo");
9       </script>
10  </head>
11  <body>
12      <h1>JavaScript</h1>
13      <h2>Linguagem de Programação</h2>
14  </body>
15  </html>
```

JavaScript – Básico - Parte I

- **exemplo3.html**

```html
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript - Exemplo 1</title>
</head>
<body>
    <h1>JavaScript</h1>
    <h2>Linguagem de Programação</h2>

    <script>
        alert("Olá Mundo");
    </script>
</body>
</html>
```

JavaScript – Básico - Parte I

- **JavaScript em arquivo externo. exemplo4.html / exemplo4.js**

```html
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript - Exemplo 4</title>
</head>
<body>
    <h1>JavaScript</h1>
    <h2>Linguagem de Programação</h2>

    <script src="js/exemplo4.js"></script>
</body>
</html>
```

JS *exemplo4.js* ×

01-códigos > js > JS exemplo4.js

```
1  console.log("Olá Mundo");
```

JavaScript – Básico - Parte I

- **Estruturas de decisão (if...else)**
  - A condicional **if** é uma estrutura condicional que **executa** a **afirmação**, **dentro** do bloco, **se** determinada **condição** for **verdadeira**. **Se** for **falsa**, **executa** as **afirmações** dentro do **else**.

```
if (condição) {
   instrução1
} else {
   instrução2
}
```

```
if (condição1)
   instrução1
else if (condição2)
   instrução2
else if (condição3)
   instrução3
...
else
   instruçãoN
```

```
if (condição1)
   instrução1
else
   if (condição2)
      instrução2
   else
      if (condição3)
...
```

JavaScript – Básico - Parte I

- **exemplo5.html**

```html
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript - Exemplo 5</title>
</head>
<body>
    <h1>JavaScript - Básico - Parte I</h1>
    <h2>Estruturas de Decisão</h2>

    <input type="text" id="nota1" placeholder="Digite a nota 1:">
    <input type="text" id="nota2" placeholder="Digite a nota 2:">
    <button onclick="calcular()">Calcular</button>

    <script src="js/exemplo5.js"></script>
</body>
</html>
```

JavaScript – Básico - Parte I

- **exemplo5.js**

01-códigos > js > JS exemplo5.js > ...

```
1
2   function calcular() {
3       var nota1 = parseFloat(document.getElementById('nota1').value);
4       var nota2 = parseFloat(document.getElementById('nota2').value);
5       var media = (nota1 + nota2) / 2;
6       if (media>=6) {
7           document.write("<h1>Aluno Aprovado</h1>");
8       }
9       else{
10          document.write("<h1>Aluno Reprovado</h1>");
11      }
12  }
```

JavaScript – Básico - Parte I

- **exemplo6.html**

```html
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript - Exemplo 6</title>
</head>
<body>
    <h1>JavaScript - Básico - Parte I</h1>
    <h2>Estruturas de Decisão</h2>

    <input type="text" id="nome" placeholder="Digite o nome do aluno:">
    <input type="text" id="nota1" placeholder="Digite a nota 1:">
    <input type="text" id="nota2" placeholder="Digite a nota 2:">
    <button onclick="calcular()">Calcular</button>

    <script src="js/exemplo6.js"></script>
</body>
</html>
```

JavaScript – Básico - Parte I

- **exemplo6.js**

01-códigos > js > JS exemplo6.js > ...

```
1
2   function calcular() {
3       var nome = document.getElementById('nome').value;
4       var nota1 = parseFloat(document.getElementById('nota1').value);
5       var nota2 = parseFloat(document.getElementById('nota2').value);
6       var media = (nota1 + nota2) / 2;
7       if (media>=6) {
8           document.write("<h1>O aluno "+ nome + " está Aprovado</h1>");
9       }
10      else if (media>=4){
11          document.write("<h1>O aluno "+ nome + " terá que fazer SUB</h1>");
12      }
13      else{
14          document.write("<h1>O aluno "+ nome + " está Reprovado</h1>");
15      }
16  }
```

JavaScript – Básico - Parte I

- **A condicional Switch**
  - A condicional **switch** avalia uma expressão, combinando o valor da expressão para um cláusula **case**, e executa as instruções associadas ao case.

Sintaxe:

```
switch (expressão) {
  case valor1:
    //Instruções executadas quando o resultado da expressão for igual á valor1
    [break;]
  case valor2:
    //Instruções executadas quando o resultado da expressão for igual á valor2
    [break;]
  ...
  case valueN:
    //Instruções executadas quando o resultado da expressão for igual á valorN
    [break;]
  default:
    //Instruções executadas quando o valor da expressão é diferente de todos os cases
    [break;]
}
```

JavaScript – Básico - Parte I

exemplo7.html

```
1   <!DOCTYPE html>
2   <html lang="pt-br">
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>JavaScript - Exemplo 6</title>
7   </head>
8   <body>
9       <h1>JavaScript - Básico - Parte I</h1>
10      <h2>O condicional Switch</h2>
11
12      <h3>Digite o último dígito de sua placa: </h3>
13      <input type="number" name="placa" id="placa"
14      placeholder="Digite o último dígito de sua placa" min="0" max="9">
15      <button onclick="verificar()">Verificar</button>
16
17      <script src="js/exemplo7.js"></script>
18  </body>
19  </html>
```

JavaScript – Básico - Parte I

exemplo7.js

01-códigos > js > JS exemplo7.js > verificar

```
1
2   function verificar() {
3       var placa = parseInt(document.getElementById('placa').value);
4       //document.write(placa);
5       switch (placa) {
6           case 1:
7               document.write("<h2>Rodízio na Segunda-Feira</h2>");
8               break;
9           case 2:
10              document.write("<h2>Rodízio na Segunda-Feira</h2>");
11              break;
12          case 3:
13              document.write("<h2>Rodízio na Terça-Feira</h2>");
14              break;
15          case 4:
16              document.write("<h2>Rodízio na Terça-Feira</h2>");
17              break;
18          case 5:
19              document.write("<h2>Rodízio na Quarta-Feira</h2>");
20              break;
21          case 6:
22              document.write("<h2>Rodízio na Quarta-Feira</h2>");
23              break;
```

JavaScript – Básico - Parte I

exemplo7.js

```
24          case 7:
25              document.write("<h2>Rodízio na Quinta-Feira</h2>");
26              break;
27          case 8:
28              document.write("<h2>Rodízio na Quinta-Feira</h2>");
29              break;
30          case 9:
31              document.write("<h2>Rodízio na Sexta-Feira</h2>");
32              break;
33          case 0:
34              document.write("<h2>Rodízio na Sexta-Feira</h2>");
35              break
36          default:
37              document.write("<h2>Valor Inválido</h2>");
38      }
39  }
```

JavaScript – Básico - Parte I

- **Estruturas de repetição**

  - **Laços** oferecem um jeito fácil e rápido de executar uma ação repetidas vezes.

  - Os possíveis laços de repetição em JavaScript:
    - for
    - for...in
    - for...of
    - for...each
    - do...while
    - while

JavaScript – Básico - Parte I

- **Estruturas de repetição: for**
  - Um laço **for** é repetido até que a condição especificada seja falsa.

Sintaxe:

```
for ([expressaoInicial]; [condicao]; [incremento])
  declaracao
```

JavaScript – Básico - Parte I

exemplo8.html

```
10  <body>
11      <h1>JavaScript - Básico - Parte I</h1>
12      <h2>O laço for</h2>
13
14      <form name="selectForm">
15          <p>
16              <label for="tipoMusica">Escolha alguns tipos de música, em seguida,
17                  clique no botão abaixo:</label>
18              <select id="tipoMusica" name="tipoMusica" multiple="multiple">
19                  <option selected="selected">Rock</option>
20                  <option>Sertanejo</option>
21                  <option>Pagode</option>
22                  <option>Samba</option>
23                  <option>MPB</option>
24                  <option>POP</option>
25              </select>
26          </p>
27          <p><input id="btn" type="button" value="Quantos foram selecionados?" /></p>
28      </form>
29
30      <script src="js/exemplo8.js"></script>
31  </body>
32  </html>
```

JavaScript – Básico - Parte I

exemplo8.js

01-códigos > js > JS exemplo8.js > ...

```
1   function minhaFuncao(selectObject) {
2       var numeroSelecionadas = 0;
3       for (var i = 0; i < selectObject.options.length; i++) {
4           if (selectObject.options[i].selected) {
5               numeroSelecionadas++;
6           }
7       }
8       return numeroSelecionadas;
9   }
10
11  var btn = document.getElementById("btn");
12  btn.addEventListener("click", function () {
13      alert('Total de opções selecionadas: ' + minhaFuncao(document.selectForm.tipoMusica))
14  });
```

JavaScript – Básico - Parte I

- **Estruturas de repetição: do...while**
  - A instrução **do...while** repetirá até que a condição especificada seja falsa.

Sintaxe:

```
do
  declaracao
while (condicao);
```

JavaScript – Básico - Parte I

exemplo9.html

```
10  <body>
11      <h1>JavaScript - Básico - Parte I</h1>
12      <h2>O laço for</h2>
13
14      <h3>Digite um número: </h3>
15      <input type="text" name="num" id="num"
16      placeholder="Digite um número"/>
17
18      <button onclick="calcular()">Calcular</button>
19
20      <script src="js/exemplo9.js"></script>
21  </body>
22  </html>
```

JavaScript – Básico - Parte I

exemplo9.js

01-códigos > js > JS exemplo9.js > ...

```
1   function calcular(){
2
3       var num = parseFloat(document.getElementById('num').value);
4       var i = 0;
5
6       do {
7           //let tabuada = num * i;
8           //document.write("<h2>"+ num + " X " + i +" = "+tabuada+ "<br></h2>");
9           document.write("<h4>"+ `${i} X ${num} = ${i*num}`+"</h4>");
10          i++;
11      } while (i < 11);
12  }
```

Template String: `${ js }`

JavaScript – Básico - Parte I

- **Estruturas de repetição: while**
  - Uma declaração **while** executa suas instruções, desde que uma condição especificada seja avaliada como verdadeira.

Sintaxe:

```
while (condicao)
  declaracao
```

JavaScript – Básico - Parte I

exemplo10.html

```
10  <body>
11    <h1>JavaScript - Básico - Parte I</h1>
12    <h2>O comando while</h2>
13    <h2>Acumulador de 0 a 10</h2>
14
15    <script>
16      n = 0;
17      acumulador = 0;
18      while (n < 11) {
19        acumulador += n;
20        n++;
21        console.log(acumulador);
22      }
23      console.log("Total= " + acumulador);
24    </script>
25  </body>
26
27  </html>
```

JavaScript – Básico - Parte I

exemplo11.html

Use **break** para terminar laços.

```
10  <body>
11    <h1>JavaScript - Básico - Parte I</h1>
12    <h2>O uso do break</h2>
13
14    <script>
15      var n = 1;
16      while (n <= 100) {
17        if (n > 50) {
18          break; //sai do loop
19        }
20        console.log(n);
21        n++;
22      }
23    </script>
24  </body>
```

JavaScript – Básico - Parte I

exemplo12.html

A declaração **continue** pode ser usada para reiniciar uma instrução while, do-while ou for.

```
10  <body>
11    <h1>JavaScript - Básico - Parte I</h1>
12    <h2>O uso do continue</h2>
13
14    <script>
15
16      // exibindo os pares até 100
17      var n = 1;
18      while (n <= 100) {
19
20        if (n % 2 != 0) {
21          n++;
22          continue;
23        }
24        console.log(n);
25        n++;
26      }
27    </script>
28  </body>
```

JavaScript – Básico - Parte I

- **Estruturas de repetição: for...in**
  - O loop **for...in** executa iterações a partir de uma variável específica, percorrendo todas as propriedades de um objeto.

Sintaxe:

```
for (variavel in objeto) {
  declaracoes
}
```

JavaScript – Básico - Parte I

exemplo13.html

```
10  <body>
11    <h1>JavaScript - Básico - Parte I</h1>
12    <h2>A iteração for...in</h2>
13
14    <script>
15
16      //Objeto
17      var obj = { a: 1, b: 2, c: 3 };
18
19      //Para prop (propriedade) in obj (objeto) faça
20      for (var prop in obj) {
21        console.log("obj." + prop + " = " + obj[prop]);
22      }
23
24
25    </script>
26  </body>
27  </html>
```

JavaScript – Básico - Parte I

- **Estruturas de repetição: for...of**

  - O loop **for...of** percorre objetos iterativos (incluindo Array, Map e Set), chamando uma função personalizada com instruções a serem executadas para o valor de cada objeto distinto.

Sintaxe:

```
for (variavel of iteravel) {
  declaração
}
```

JavaScript – Básico - Parte I

exemplo14.html

```
10  <body>
11    <h1>JavaScript - Básico - Parte I</h1>
12    <h2>A iteração for...of</h2>
13
14    <script>
15
16      let valores = [10, 20, 30];
17
18      for (let value of valores) {
19        console.log(value);
20      }
21
22      console.log(" ");
23      let nome = "rodrigo";
24
25      for (let value of nome) {
26        console.log(value);
27      }
28
29    </script>
30  </body>
31  </html>
```

JavaScript – Básico - Parte I

- **exemplo15.html**
- **Estruturas de repetição:**
- **for...in X for..of**

```
10  <body>
11    <h1>JavaScript - Básico - Parte I</h1>
12    <h2>A diferença entre for...in x for...of</h2>
13
14    <script>
15
16      const numeros = [1, 2, 3, 4, 5];
17      for (let numero of numeros) {
18        console.log(numero);
19      }
20
21      console.log(" ");
22      console.log("Agora com o for...in")
23      for (let numero in numeros) {
24        console.log(numero);
25      }
26
27    </script>
28  </body>
29
30  </html>
```

JavaScript – Básico - Parte I

- **Arrays**
- **exemplo16.html**

```
10  <body>
11    <h1>JavaScript - Básico - Parte I</h1>
12    <h2>Arrays</h2>
13
14    <p id="demo"></p>
15    <p id="demo2"></p>
16    <p id="demo3"></p>
17    <p id="demo4"></p>
18    <p id="demo5"></p>
19
20    <script>
21      var carros = ["Onix", "Polo", "Argos"];
22      document.getElementById("demo").innerHTML = carros;
23
24      var carros2 = [
25        "Fusca",
26        "Brasília",
27        "Passat"
28      ];
29      document.getElementById("demo2").innerHTML = carros2;
```

JavaScript – Básico - Parte I

```
30
31      var carros3 = new Array("Belina", "Corcel", "Del Rey");
32      document.getElementById("demo3").innerHTML = carros3;
33
34      document.getElementById("demo4").innerHTML = "Carro na posição 0: " + carros3[0];
35
36      carros3[0] = "Parati";
37      document.getElementById("demo5").innerHTML = "Carro na posição 0 alterado: " + carros3[0];
38
39
40    </script>
41  </body>
42
43  </html>
```

JavaScript – Básico - Parte I

- **Arrays x Objetos**

- Arrays usam números para acessar seus "elementos".

- Objetos usam nomes para acessar seus "membros".

- Elementos da matriz podem ser objetos.

exemplo17.html

```
10  <body>
11    <h1>JavaScript - Básico - Parte I</h1>
12    <h2>Arrays x Objetos</h2>
13
14    <p id="demo"></p>
15    <p id="demo2"></p>
16    <p id="demo3"></p>
17    <p id="demo4"></p>
18
19    <script>
20      var pessoa = ["Rodrigo", "Martins", 41];
21      document.getElementById("demo").innerHTML = pessoa[0];
22
23      var pessoa2 = { primeiroNome: "Rodrigo", ultimoNome: "Martins", idade: 41 };
24      document.getElementById("demo2").innerHTML = pessoa2["idade"];
```

JavaScript – Básico - Parte I

exemplo17.html

```
25
26      document.getElementById("demo3").innerHTML = pessoa.length;
27
28      var frutas, texto, qtdEl, i;
29      frutas = ["Banana", "Laranja", "Maça", "Manga"];
30      qtdEl = frutas.length;
31
32      texto = "<ul>";
33      for (i = 0; i < qtdEl; i++) {
34        texto += "<li>" + frutas[i] + "</li>";
35      }
36      texto += "</ul>";
37
38      document.getElementById("demo4").innerHTML = texto;
39
40    </script>
41
42  </body>
43
44  </html>
```

JavaScript – Básico - Parte I

exemplo18.html

```
10  <body>
11    <h1>JavaScript - Básico - Parte I</h1>
12    <h2>Pop, Push, Shift, Unshift, Sort e Reverse</h2>
13
14    <h2>pop()</h2>
15
16    <p>O método pop() remove o último elemento do Array</p>
17
18    <p id="demo1"></p>
19    <p id="demo2"></p>
20
21    <h2>shift()</h2>
22
23    <p>O método shift() remove o primeiro elemento do Array</p>
24    <p id="demo3"></p>
25
```

JavaScript – Básico - Parte I

exemplo18.html

```
26  <h2>push()</h2>
27
28  <p>O método push() adiciona um elemento no Array</p>
29
30  <button onclick="adiciona()">PUSH</button>
31
32  <h2>Unshift()</h2>
33
34  <p>O método unshift() adiciona no início um novo elemento no Array</p>
35  <button onclick="adicionaInicio()">UNSHIFT</button>
36  <p id="demo3"></p>
37
38  <h2>Sort() e Reverse()</h2>
39
40  <p>O método sort() ordena um Array</p>
41  <p>O método reverse() classifica decrescente um Array</p>
42
43  <button onclick="ordena()">SORT</button>
44  <button onclick="decrescente()">REVERSE</button>
45
```

JavaScript – Básico - Parte I

exemplo18.html

```
60      function adicionaInicio() {
61        frutas.unshift("Abacaxi");
62        document.getElementById("demo3").innerHTML = frutas;
63      }
64
65      function ordena() {
66        frutas.sort();
67        document.getElementById("demo3").innerHTML = frutas;
68      }
69
70      function decrescente() {
71        frutas.reverse();
72        document.getElementById("demo3").innerHTML = frutas;
73
74      }
75
76    </script>
77
78  </body>
79
80  </html>
```

Exercícios Propostos

1. Elabore um script que solicite o peso e a altura de uma determinada pessoa. Após a digitação, exibir se esta pessoa está ou não com seu peso ideal, conforme tabela abaixo:

IMC (IMC = peso / altura²)	MENSAGEM
imc < 20	Abaixo do peso
20 > = imc < 25	Peso Ideal
IMC > = 25	Acima do Peso

2. Elaborar um script em que dada a idade de um nadador, classifique-o em uma das seguintes categorias: infantil A (de 5 a 7 anos), infantil B (de 8 a 10 anos), juvenil A (de 11 a 13 anos), juvenil B (14 a 17 anos) e senior (maior que 17 anos)

Exercícios Propostos

3. Construa um script que calcule o novo salário de um funcionário. Considere que o funcionário deverá receber um reajuste de 15% caso seu salário seja menor que 1000. Se o salário for maior ou igual a 1000, mas menor ou igual a 1500, o reajuste deve ser de 10%. Caso o salário seja maior que 1500, o reajuste deve ser de 5%.

Exercícios Propostos

4. Faça um programa em C++ que receba o número de horas trabalhadas e o valor do salário mínimo. Calcule e mostre o salário a receber seguindo as regras abaixo:

a. A hora trabalhada vale a metade do salário mínimo;
b. O salário bruto equivale ao número de horas trabalhadas multiplicado pelo valor da hora trabalhada
c. O imposto equivale a 3% do salário bruto;
d. O salário a receber equivale ao salário bruto menos o imposto.

5. Construa um programa em C++ que calcule o novo salário de um funcionário. Considere que o funcionário deverá receber um reajuste de 15% caso seu salário seja menor que 1000. Se o salário for maior ou igual a 1000, mas menor ou igual a 1500, o reajuste deve ser de 10%. Caso o salário seja maior que 1500, o reajuste deve ser de 5%.

Exercícios Propostos

6. Escreva um script que mostre o quadrado dos números inteiros no intervalo de 1 a 20.

7. Escreva um script que escreva todos os números múltiplos de 5, no intervalo de 1 a 500.

8. Em uma eleição presidencial existem dois candidatos. Os votos são informados através de códigos. Os dados utilizados para a contagem dos votos têm-se a seguinte codificação: 1,2 = voto para os respectivos candidatos; 3= voto nulo; 4= voto em branco; Elabore um script que leia o código do candidato em um voto. Calcule e escreva: (1) percentual de votos para cada candidato; (2) percentual de votos nulos; (3) percentual de votos em branco;

Exercícios Propostos

9. Em um cinema, certo dia, cada espectador respondeu a um questionário, que perguntava a sua idade (ID) e a opinião em relação ao filme (OP), seguindo os seguintes critérios:

Opinião (OP)	Significado
1	Ótimo
2	Bom
3	Regular
4	Ruim

Ao final da pesquisa será indicado quando a idade do usuário for informada como negativa (idade inexistente). Construa um programa em C++ que, lendo esses dados, calcule e apresente:

A. Quantidade de pessoas que respondeu a pesquisa

B. Média de idade das pessoas que responderam a pesquisa

C. Porcentagem de cada uma das respostas

Referências Bibliográficas

- https://developer.mozilla.org/pt-BR/docs/Web/API/Document
- https://medium.com/reactbrasil/como-o-javascript-funciona-dentro-da-engine-v8-5-dicas-sobre-como-escrever-c%C3%B3digo-otimizado-e05af6088fd5
- https://canaltech.com.br/internet/O-que-e-e-como-funciona-a-linguagem-JavaScript/
- https://developer.mozilla.org/pt-PT/docs/Web/JavaScript/Guia/Introdu%C3%A7%C3%A3o_ao_JavaScript
- CAELLUM (2020) – Material WD-43 da Caellum

FIM

Obrigado Rodrigo