

JAVASCRIPT

Dr. Sóstenes P. Gomes

Por que estudar JavaScript?

- JavaScript é uma das três linguagens que todos os desenvolvedores *web* precisam aprender
 - **HTML** para definir o conteúdo das páginas
 - **CSS** para especificar o layout e a apresentação das páginas
 - **JavaScript** para programar o comportamento das páginas

Por que estudar JavaScript?

- JavaScript não é utilizado apenas em *web pages*
- Muitas aplicações *desktop* e programas de servidor também utilizam a linguagem
- Node.js é o mais conhecido
 - Angular.js, React, Vue.js

Introdução

Interação com o HTML

- JavaScript faz uso de métodos
- Um dos principais métodos HTML do JavaScript é **getElementById()**

JavaScript pode alterar conteúdo HTML

```
<!DOCTYPE html>
<html>
<body>

  <h2>What Can JavaScript Do?</h2>

  <p id="demo">JavaScript can change HTML content.</p>

  <button type="button" onclick='document.getElementById("demo").innerHTML = "Hello JavaScript!"'>Click Me!</button>

</body>
</html>
```

JavaScript pode alterar conteúdo HTML

```
<!DOCTYPE html>
<html>
  <body>

    <h2>What Can JavaScript Do?</h2>

    <p>JavaScript can change HTML attribute values.</p>

    <p>In this case JavaScript changes the value of the src (source) attribute of an image.</p>

    <button onclick="document.getElementById('myImage').src='pic_bulbon.gif'">Turn on the light</button>

    <button onclick="document.getElementById('myImage').src='pic_bulboff.gif'">Turn off the light</button>

  </body>
</html>
```

JavaScript pode mudar estilos CSS

```
<!DOCTYPE html>
<html>
  <body>

    <h2>What Can JavaScript Do?</h2>

    <p id="demo">JavaScript can change the style of an HTML element.</p>

    <button type="button" onclick="document.getElementById('demo').style.fontSize='35px'">Click Me!</button>

  </body>
</html>
```

JavaScript pode mudar estilos CSS

```
<!DOCTYPE html>
<html>
  <body>

    <h2>What Can JavaScript Do?</h2>

    <p id="demo">JavaScript can hide HTML elements.</p>

    <button type="button" onclick="document.getElementById('demo').style.display='none'">Click Me!</button>

  </body>
</html>
```


JavaScript pode mudar estilos CSS

```
<!DOCTYPE html>
<html>
  <body>

    <h2>What Can JavaScript Do?</h2>

    <p>JavaScript can show hidden HTML elements.</p>

    <p id="demo" style="display:none">Hello JavaScript!</p>

    <button type="button" onclick="document.getElementById('demo').style.display='block'">Click Me!</button>

  </body>
</html>
```

Onde escrever JavaScript?

- A tag **<script>**

```
<!DOCTYPE html>
<html>
  <body>

    <h2>JavaScript in Body</h2>

    <p id="demo"></p>

    <script>
      document.getElementById("demo").innerHTML = "My First JavaScript";
    </script>

  </body>
</html>
```

Onde escrever JavaScript?

- Pode ser escrito no **<head>** ou **<body>**

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function myFunction() {
        document.getElementById("demo").innerHTML = "Paragraph changed.";
      }
    </script>
  </head>
  <body>

    <h2>JavaScript in Head</h2>

    <p id="demo">A Paragraph.</p>

    <button type="button" onclick="myFunction()">Try it</button>

  </body>
</html>
```

Onde escrever JavaScript?

Em um arquivo externo.

Ex.: **myScript.js**:

```
function myFunction() {  
    document.getElementById("demo").innerHTML = "Paragraph changed.";  
}
```

No HTML:

```
<!DOCTYPE html>  
<html>  
  <body>  
  
    <h2>External JavaScript</h2>  
  
    <p id="demo">A Paragraph.</p>  
  
    <button type="button" onclick="myFunction()">Try it</button>  
  
    <p>(myFunction is stored in an external file called "myScript.js")</p>  
  
    <script src="myScript.js"></script>  
  
  </body>  
</html>
```

Onde escrever JavaScript?

- Para adicionar vários arquivos de script em uma página, use várias tags

```
<script src="myScript1.js"></script>  
<script src="myScript2.js"></script>
```

Onde escrever JavaScript?

- Scripts externos podem ser referenciados através de uma URL completa ou de um caminho de diretório no computador

```
<script src="https://www.w3schools.com/js/myScript1.js"></script>
```

```
<script src="/js/myScript1.js"></script>
```

```
<script src="myScript1.js"></script>
```

Vantagens de se utilizar JavaScript externo

- Separa HTML e código
- Faz o HTML e o JavaScript mais fácil de ler e de aplicar manutenção
- JavaScript em cache acelera o carregamento de páginas

Opções de saída no JavaScript

Usando innerHTML

```
<!DOCTYPE html>
<html>
  <body>

    <h2>My First Web Page</h2>
    <p>My First Paragraph.</p>

    <p id="demo"></p>

    <script>
      document.getElementById("demo").innerHTML = 5 + 6;
    </script>

  </body>
</html>
```

My First Web Page

My First Paragraph.

11

Opções de saída no JavaScript

Usando document.write()

```
<!DOCTYPE html>
<html>
  <body>

    <h2>My First Web Page</h2>
    <p>My first paragraph.</p>

    <p>Never call document.write after the document has finished loading.
    It will overwrite the whole document.</p>

    <script>
document.write(5 + 6); My First Web Page
    </script>

    My first paragraph.

    Never call document.write after the document has finished loading. It will ov
  </body>
</html>
```

Opções de saída no JavaScript

Usando window.alert()

```
<!DOCTYPE html>
<html>
  <body>

    <h2>My First Web Page</h2>
    <p>My first paragraph.</p>

    <script>
      window.alert(5 + 6);
    </script>

  </body>
</html>
```

Opções de saída no JavaScript

Usando o Console (pressione F12)

```
<!DOCTYPE html>
<html>
  <body>

    <h2>Activate debugging with F12</h2>

    <p>Select "Console" in the debugger menu. Then click Run again.</p>

    <script>
      console.log(5 + 6);
    </script>

  </body>
</html>
```

Instruções no JavaScript

- Valores
- Operadores
- Expressões
- Palavras reservadas
- Comentários
- Separadas por ponto e vírgula (;)

```
var x, y, z;    // Statement 1
x = 5;          // Statement 2
y = 6;          // Statement 3
z = x + y;      // Statement 4
```

```
document.getElementById("demo").innerHTML = "Hello Dolly.";
```

Variáveis em JavaScript

- Tipos de dados

```
var pi = 3.14;  
var person = "John Doe";  
var answer = 'Yes I am!';
```

- JavaScript não é linguagem tipada
- Strings podem ser escritas entre aspas duplas ou simples

Declarando variáveis

```
var person = "John Doe", carName = "Volvo", price = 200;
```

```
var person = "John Doe",  
    carName = "Volvo",  
    price = 200;
```

Operadores aritméticos e de atribuição

+

-

*

**

/

%

++

--

=

x = y

x = y

+=

x += y

x = x + y

-=

x -= y

x = x - y

*=

x *= y

x = x * y

/=

x /= y

x = x / y

%=

x %= y

x = x % y

**=

x **= y

x = x ** y

Operações com *string*

```
var txt1 = "John";  
var txt2 = "Doe";  
var txt3 = txt1 + " " + txt2;
```

John Doe

```
var txt1 = "What a very ";  
txt1 += "nice day";
```

What a very nice day

Operações com *string*

```
var x = 5 + 5;  
var y = "5" + 5;  
var z = "Hello" + 5;
```

10

55

Hello5

Operadores lógicos

Operator	Description
==	equal to
===	equal value and equal type
!=	not equal
!==	not equal value or not equal type
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
?	ternary operator

Operadores de tipo

typeof – Retorna o tipo de uma variável

instanceof – Retorna true se um objeto é uma instancia de um tipo de objeto

Funções

- Declaradas com a palavra reservada **function**

```
function toCelsius(fahrenheit) {  
    return (5/9) * (fahrenheit-32);  
}
```

Funções

- Crie uma função para a seguinte instrução


```
document.getElementById("demo").innerHTML = "A temperatura é "  
+ toCelsius(77) + " Celsius";
```

- Crie a seguinte função também

```
function toCelsius(fahrenheit) {  
    return (5/9) * (fahrenheit-32);  
}
```

- Faça com que a primeira função altere um parágrafo de ID = “demo” em um HTML

Objetos

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

```
var car = {type:"Fiat", model:"500", color:"white"};
```

Objetos

- Definindo atributos

```
var person = {  
  firstName: "John",  
  lastName: "Doe",  
  age: 50,  
  eyeColor: "blue"  
};
```

Objetos

- Definindo métodos

```
var person = {  
  firstName: "John",  
  lastName : "Doe",  
  id       : 5566,  
  fullName : function() {  
    return this.firstName + " " + this.lastName;  
  }  
};
```


Objetos

- Acessando atributos

```
document.getElementById("demo").innerHTML =  
person.firstName + " " + person.lastName;
```

- Acessando métodos

```
document.getElementById("demo").innerHTML = person.fullName();
```