

Introdução a JavaScript

Usabilidade, desenvolvimento web, mobile e jogos

Prof. Me. Gustavo Torres Custódio
gustavo.custodio@anhembi.br

Conteúdo

Introdução ao JavaScript

Programando com JavaScript

Eventos em JavaScript



Introducao a JavaScript

Introdução ao JavaScript

JavaScript

- O que é JavaScript?
 - Introduzido em 1995 para ser utilizado no navegador Netscape.
 - Permitia realizar alterações em uma página sem a necessidade de recarregá-la.
 - Desde então tem sido a linguagem adotada por todos os navegadores de interface gráfica.
- Ao contrários do HTML e CSS, JavaScript **é uma linguagem de programação.**

JavaScript



JavaScript

- Um código escrito em JavaScript pode ser acrescentado à sua página de duas maneiras:
 - Adicionando o código dentro de uma tag `<script>` no próprio HTML.
 - Criando um arquivo `.js` e depois adicionando ele na respectiva página HTML.

Adicionando Código na Tag Script

- Vamos criar uma página vazia e adicionar um código JavaScript que rode duas funções:
 - `console.log`.
 - `document.write`

Adicionando Código na Tag Script

```
document.write("Meu primeiro JavaScript");  
console.log("Meu primeiro JavaScript");
```

- O resultado da função `console.log` pode ser verificada olhando o console do navegador utilizado.

Importando Código do Arquivo Js

- Crie uma pasta chamada “js” no seu diretório.
 - Essa pasta vai conter todos os arquivos JavaScript.
- Assim como código adicionado dentro da página HTML, podemos incluir o conteúdo de um arquivo JavaScript na página usando a tag `<script>`.
 - Utilizamos a propriedade `src` para identificar o arquivo js importado.

Importando Código do Arquivo Js

- Crie o arquivo **primeiro.js**:

```
document.write("Vamos contar: ");  
for (i = 1; i <= 10; i++) {  
    document.write("Contando "+ i + "<br>");  
}
```

- Adicione a linha de código no final de `<body>`:
 - `<script src="js/primeiro.js"></script>`

Importando Código do Arquivo Js

Vamos contar: Contando 1
Contando 2
Contando 3
Contando 4
Contando 5
Contando 6
Contando 7
Contando 8
Contando 9
Contando 10



Introducao a JavaScript

Programando com JavaScript

Variáveis

- Variáveis em JavaScript são definidas pelas palavras chave **var** ou **let**.
 - **var** multiplicacao = 5 * 5.
 - Uma variável chamada “multiplicacao” é criada com valor 25.
- Variáveis podem conter qualquer nome que não seja uma palavra reservada.
 - ou seja, qualquer palavra que já assuma outra função na linguagem (var, por exemplo).

Variáveis

- Variáveis em JavaScript são dinamicamente tipadas.
 - Têm seu tipo inferido com base no valor e podem ter o tipo alterado durante o programa.
 - Motivo de erros frequentes.

```
var variavel = 1;  
console.log(typeof(variavel)); // Mostra number
```

```
variavel = "Olá";  
console.log(typeof(variavel)); // Mostra string
```

Arrays

- Em JavaScript *arrays* são estruturas de dados que armazenam um ou mais valores.
 - São delimitados por colchetes ([]).
 - Os elementos dentro do *array* podem ser de múltiplos tipos diferentes:
 - `var elementos = [1, "dois", 3];`

Objetos

- *Arrays* são um tipo especial de objeto.
 - A função `typeof` passando um *array* de JavaScript retorna `object`.
 - Neste caso, temos um objeto que pode ter seus elementos referenciados por um índice:
 - `elementos[1] = "dois"`.

Objetos

- *Arrays* têm seus valores acessados por meio de índices numéricos.
- Objetos têm seus valores referenciados por meio de chaves.
 - chave : valor.
 - Podem ser valores numéricos, *strings* e outros tipos de variáveis.
- Suponha que temos um objeto chamado *capitais*.

Objetos

```
var capitais = {  
  "Brasil": "Brasilia",  
  "Japão": "Toquio",  
  "Austrália": "Canberra"  
}  
  
console.log(capitais["Brasil"]);
```

- Será mostrado na tela “Brasilia”.
- A **chave** de referência é o nome do país e o **valor** correspondente é a capital.

Percorrendo Arrays e Objetos

- O JavaScript permite uso de diferentes tipos de for.
 - For para com uma variável sendo incrementada/decrementada.
 - For para percorrer arrays e objetos.

Percorrendo Arrays e Objetos

- Podemos percorrer um array utilizando o `for ... of`:

```
// Cria um menu com Home, Contato e Sobre
itens_menu = ["Home", "Contato", "Sobre"];

document.write("<ul>");
for (const item of itens_menu) {
    document.write("<li>" + item + "</li>");
}
document.write("</ul>");
```

- Home
- Contato
- Sobre

Percorrendo Arrays e Objetos

- E percorrer um objeto utilizando o `for ... in`:

```
const populacoes = {  
  "Sao Paulo": 10000000,  
  "Santo Andre": 700000,  
  "Guarulhos": 1300000,  
};  
  
for (const cidade in populacoes) {  
  console.log(`${cidade} tem ${populacoes[cidade]} habitantes.`);  
}
```

Funções

- Funções em JavaScript são declaradas utilizando o identificador `function`.
- Funções em JavaScript são objetos.
 - É possível passar funções como parâmetros e declarar variáveis que recebem funções.

Funções

```
function somaDois(x, y) {  
    return x + y;  
}  
  
let resultado = somaDois(2, 2);  
console.log(resultado);
```

```
// Aqui a função é declarada  
var somaDois = function(x, y) {  
    return x + y;  
}  
  
// Aqui a função é chamada  
var resultado = somaDois(1, 5);  
console.log(resultado);
```

Funções

- Neste exemplo, criamos duas funções:
 - Uma função declarada de forma convencional;
 - Uma função anônima atribuída a uma variável.
- Funções anônimas são passadas como parâmetros com frequência.
 - Por exemplo, um botão que recebe uma ação que é realizada quando ele é clicado.



Introducao a JavaScript

Eventos em JavaScript

Eventos

- Existe alguma forma de chamar uma função em JavaScript quando algo ocorre na página html?
 - Por exemplo, uma função é executada quando clicamos em um botão.
 - Ou quando passamos o mouse por cima de algum elemento.

Eventos

- A linguagem JavaScript admite um conjunto de eventos:
 - **click** - o usuário clica em um elemento HTML.
 - **mouseover** - o usuário move o mouse sobre um elemento HTML.
 - **mouseout** - o usuário move o mouse para longe de um elemento HTML.
 - **keydown** - o usuário pressiona uma tecla do teclado.
 - **load** - o navegador terminou de carregar a página.

Eventos

- Utilizando eventos, podemos chamar uma função JavaScript quando o usuário realiza alguma ação.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Alerta</title>
  </head>
  <body>
    <button onclick="alertar();">
      Clique em mim
    </button>
    <script src="enviaralerta.js">
    </script>
  </body>
</html>
```

```
var alertar = function() {
  alert("Você clicou no botão da
        página");
}
```

DOM - Document Object Model

- Quando um documento HTML é carregado em um navegador da web, ele se torna um objeto de documento (DOM).
- Por meio do DOM é possível acessar qualquer elemento em uma página HTML.
 - `document.getElementById(id)`: encontre um elemento pelo id do elemento;
 - `document.getElementsByTagName(name)`: encontre elementos pelo nome da tag;
 - `document.getElementsByClassName(name)`: encontre elementos pela classe.

DOM - Document Object Model

- Acessando objetos por meio do DOM permite alterar seu conteúdo.
 - Por exemplo, podemos criar uma `div` que muda de acordo com o que é digitado em um campo de texto.

DOM - Document Object Model

```
<html>
  <head>
    <title>Exemplo Inner HTML</title>
    <style type="text/css">
      #divtitulo {
        font-family: Helvetica;
        margin: auto;
        width: 50%;
        border: 3px solid green;
      }

      #divtexto {
        margin: auto;
        width: 50%;
      }
    </style>
  </head>
  <body>
    <div id="divtitulo">
      <h1>Digite um texto na caixa de texto e clique no botão</h1>
    </div>

    <div id="divtexto">
      <input type="text" name="texto" id="texto">
      <button id="botao">Mudar Div</button>
    </div>

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

```
document.getElementById("botao").onclick = mudarDiv;

function mudarDiv() {
  let texto = document.getElementById("texto").value;

  document.getElementById(
    "divtitulo").innerHTML = `<h1>${texto}</h1>`;
}
```

DOM - Document Object Model

Digite um texto na caixa de texto e clique no botão

Mudar Div

Eventos

- Podemos adicionar eventos em JavaScript também por meio do `addEventListener`.
 - `elemento.addEventListener("evento", funcao);`

Eventos

```
<html>
  <head>
    <title>Exemplo Event Listener</title>
    <style type="text/css">
      #botao {
        font-size: 40px;
        width: 300px;
        height: 100px;
      }
    </style>
  </head>
  <body>
    <button id="botao">Mouseout</button>

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

```
document.getElementById("botao").addEventListener(
  "mouseover", mouseEmCima);

document.getElementById("botao").addEventListener(
  "mouseout", mouseFora);

function mouseEmCima() {
  let elemento = document.getElementById("botao");
  elemento.style.cursor = "hand";
  elemento.style.backgroundColor = "cyan";
  elemento.innerHTML = "Mouseover";
}

function mouseFora() {
  let elemento = document.getElementById("botao");
  elemento.style.backgroundColor = "white";
  elemento.innerHTML = "Mouseout";
}
```

Eventos

- Neste exemplo também vemos que o JavaScript consegue mudar o CSS dos elementos da página.
- Mudamos a cor e o cursor quando passamos o mouse por cima do botão

Exercício

- Crie uma página contendo a imagem de uma lâmpada acesa/apagada.
 - Crie um botão de acender e um botão de apagar a lâmpada.
 - Altere a imagem conforme o botão apertado.
 - Altere a propriedade src da imagem para fazer isso.

Exercício



Acender

Apagar

Referências

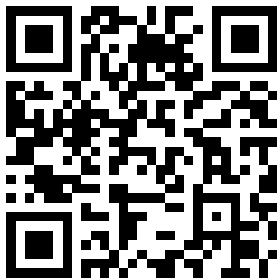


Haverbeke, M. (2018).

Eloquent javascript: A modern introduction to programming.

No Starch Press.

Conteúdo



<https://gustavotcustodio.github.io/usabilidade.html>

Obrigado

gustavo.custodio@anhembi.br