



Introdução ao JavaScript

**Tecnólogo em Análise e Desenvolvimento de
Sistemas - Faculdades SENAC/PE**

Professor: Danilo Farias

Prof. Vilson Junior



Prof. Vilson Heck Junior

Início

Técnico em Informática

Introdução a Programação
Arquitetura de Computadores
Estrutura de Dados
Programação para a Internet

Ciência da Computação

Matemática Discreta
Prog. Orientada a Objetos
Linguagens e Paradigmas de
Programação
Sistemas Operacionais
Grafos
Computação Gráfica

Atenção

O novo espaço docente, com materiais atualizados de disciplinas deve ser acessado [AQUI](#). Esta página será desativada em breve.



[Instituto Federal de Santa Catarina - Campus Lages](#)
vilson.junior@ifsc.edu.br

O conteúdo publicado nesta página é de responsabilidade exclusiva do docente e não representa necessariamente a opinião do Instituto Federal de Santa Catarina (IFSC)

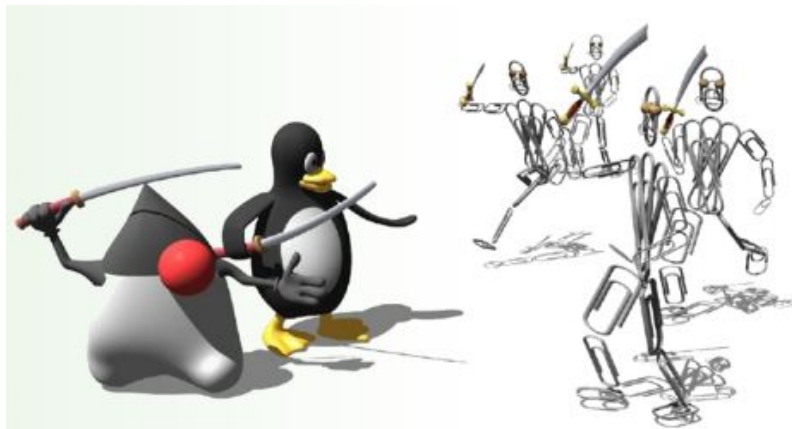
- **JavaScript:**

- Também chamada de JS, é a linguagem de criação de scripts para a Web;
- É utilizado por bilhões de páginas para:
 - Adicionar funcionalidades;
 - Verificar formulários;
 - Comunicar com servidores;
 - E muitos mais.
 - <https://www.w3schools.com/js/default.asp>

- Originalmente criada na Netscape por Brendan Eich em 1994;
- Disputa: Netscape vs Microsoft:
 - Netscape se aliou com Java da Sun (potencial crescimento);
 - Javascript
 - Java para programadores não profissionais!
 - Microsoft -> Visual Basic;
 - Visual Basic -> VB Script;



- Java e JavaScript são “coisas” completamente distintas e desconexas;
- Compartilham apenas um passado de “disputa territorial” contra a Microsoft;



- JavaScript não permite a criação de applets nem de aplicativos;
- JavaScript reside dentro de documentos HTML e pode prover diferentes níveis de interatividades não suportados pelo HTML sozinho;



- Diferenças chaves em relação ao Java:
 - Java é uma linguagem de programação;
 - JavaScript é uma linguagem de script;
 - Aplicativos Java são executados pela máquina virtual Java;
 - Scripts JavaScript são executados pelos browsers;
 - Java é compilado;
 - JavaScript é texto puro;
 - Cada tecnologia requer um plug-in diferente.

- Atualmente, o maior mantedor da linguagem é a **Fundação Mozilla**;
- Encontramos ótimos materiais e tutoriais sobre JavaScript na W3School, mas também encontramos referência completa do JavaScript no site do Mozilla:
 - <https://developer.mozilla.org/en-US/docs/web/javascript>
 - <https://www.w3schools.com/js/default.asp>



- Com o tempo, muitas **funcionalidades** foram criadas em forma de Script para os **browser** e foram “**incorporadas**” ao **JavaScript**:
 - **JavaScript hoje é um conjunto de funcionalidades** e, até mesmo, diferentes padrões.



- Os **principais padrões** a destacar são:
 - A Linguagem Núcleo:
 - [ECMAScript](#) (Versão 7, de Junho de 2016);
 - Padrão mantido por ECMA International Associação Industrial de padronização de tecnologias da Informação e Comunicação;
 - DOM:
 - Document Object Model;
 - Define a Interface da Linguagem com o Browser;
 - Padrão mantido por [W3C](#);

- Para inserir códigos **JavaScript**, iremos fazê-lo em uma Tag HTML apropriada:
 - **<script>**
 - **</script>**
- O JavaScript é **orientado a objetos**;

- Da mesma forma como nos arquivos CSS, podemos deixar funções e comandos **JavaScript em arquivos externos**:
 - Estes arquivos devem ter a extensão .JS
 - Para importar:
 - `<script src="meuScript.js"></script>`

A tag JavaScript - Hello World



- Crie uma nova página HTML;
- Dentro da seção <body> insira o trecho:

```
<script>  
    'document.write("Hello World!");  
</script>
```

- Neste caso, o trecho “escrito” pelo JavaScript, será incorporado ao HTML apenas em sua construção;

- **Propriedades** de Exemplo:
 - title – Define ou Retorna o Título da Página;
 - URL – Retorna o URL completo da página;
- **Métodos** de Exemplo:
 - write() – Escreve texto no documento;
 - writeln() – Escreve uma linha de texto no documento;

- Na página criada anteriormente;
- Defina um título: **"JavaScript Hello World!"**;
- Altere o script:

```
<script>
```

```
    document.write("<h2>" + document.title + "</h2>");
```

```
</script>
```

- Neste caso, o trecho “escrito” pelo JavaScript, será incorporado ao HTML apenas em sua construção;

- É possível disparar scripts a partir de diversos tipos de **eventos**;
- O primeiro que iremos estudar é o de um clique em um botão:
 - **Tag:** `<button>Clique Aqui!</button>`
 - **Atributos:**
 - `type="button";`
 - `onclick="alert('Bem vindo!')"`

`<button type="button" onclick="alert('Bem vindo!')"> Clique Aqui!</button>`

- Atributo global **ID**:

```
<p id="paragrafo">Olá!</p>
<script>
    function mFuncao() {
        var x = document.getElementById("paragrafo");
        x.innerHTML="Hello!";
    }
</script>
<button type="button" onclick="mFuncao();"> Translate</button>
```

JavaScript - Alterando um Atributo



```
<script>
  function trocaImagem() {
    var elemento=document.getElementById("myimage");
    if (elemento.src.match("bulbon")) {
      elemento.src="pic_bulboff.gif";
    } else {
      elemento.src="pic_bulbon.gif";
    }
  }
</script>
```

```
<p></p>
<p>Clique na lâmpada para ligar/desligar a luz</p>
```

JavaScript é uma linguagem de tipagem dinâmica e fraca:

- Não é necessário declarar o tipo de uma variável;
- Todas as variáveis são objetos (referência);
- Números são todos reais de 64bits;
- A variável irá “alterar” o seu tipo de dado conforme os valores forem atribuídos:
 - Tipo de dado dinâmico:

```
var x;           // x é indefinido
x = 5;           // x é um número
x = "John";      // x é uma string
x = true;        // x é um valor lógico
x = null;        // x é indefinido
```

JavaScript - Usando as Variáveis



- JavaScript:

```
function Soma()  
{  
    var e1 = parseInt(document.getElementById("v1").value);  
    var e2 = parseInt(document.getElementById("v2").value);  
    document.getElementById("res").innerHTML = "Resposta: " + (e1 + e2);  
}
```

Conversão de texto para inteiro!

- HTML:

```
<input type="text" id="v1"><br>  
<input type="text" id="v2">  
<p id="res">Resposta: </p>  
<button type="button" onclick="Soma()">Soma</button>
```

- Arrays são construídos através de um construtor e possuem tamanho dinâmico:
 - `var nomes = new Array();`
 - `//var nomes = [];`
 - `nomes[0] = "Fulano de Tal";`
 - `nomes[1] = "Beutrano";`
 - `nomes.push("Ciclano");`
 - https://www.w3schools.com/js/js_arrays.asp

- Objetos possuem atributos e métodos:
 - Assim como em outras linguagens orientadas a objetos, separados por ponto;
 - Para criar um objeto carro:
 - `var carro={placa:" ABC-1234", ano:2013}`
 - Para usar os atributos:
 - `carro.ano = 2012;`
 - `document.write(carro.placa);`
 - Para adicionar novo atributo:
 - `carro.cor = "verde";`
- https://www.w3schools.com/js/js_object_definition.asp

- Javascript é orientada a objetos por prototipação, não possui exatamente o conceito de classes;
- É possível construir uma função que gera novas variáveis e, assim, simular o comportamento de uma classe;
- https://www.w3schools.com/js/js_class_intro.asp

```
function Carro(ano, placa) { //Construtor
  this.ano = ano; //Atributo
  this.placa = placa;
  var nCor = Math.random() * Carro.cores.length;
  this.cor = Carro.cores[Math.floor(nCor)];
  this.alterarAno = function(novoAno) { //Método
    this.ano = novoAno;
  };
}
Carro.cores = ["Azul", "Branco", "Vermelho"]; //Atributo estático
Carro.adicionarCor = function (novaCor) { //Método estático
  Carro.cores.push(novaCor);
};
```


JavaScript - Operadores



| Operador | Descrição |
|----------|--|
| + | Efetuar soma de números ou Concatenação de strings |
| - | Efetuar subtração de números |
| * | Efetuar multiplicação de números |
| / | Efetuar divisão de números (Sempre divisão real) |
| % | Resto da divisão |
| ++ | Incremento |
| -- | Decremento |

JavaScript - Operadores de Comparação



| Operador | Descrição |
|-------------------|--|
| == | Valor igual. (5 == "5") retorna true |
| === | Valor e tipo iguais. (5 === "5") retorna false |
| != | Valor diferente. (5 != "5") retorna false |
| !== | Valor e tipos diferentes. (5 !== "5") retorna true |
| > | Maior |
| < | Menor |
| >= | Maior ou Igual |
| <= | Menor ou Igual |
| && | E (and) |
| | OU (or) |
| ! | NÃO (not) |

- **if else** funciona igual em C/Java:

```
if (condição) {  
    código para quando retornar true  
} else {  
    código para quando retornar false  
}
```

- **Obs.: switch** case também funciona igual
- https://www.w3schools.com/js/js_if_else.asp
- https://www.w3schools.com/js/js_switch.asp

- **for**, **while** e **do while** funcionam da mesma forma que em C/Java;
 - Incluindo os comandos `continue` e `break`;

```
for (x=0;x<10;x++) { }
```

```
while (x < 10) { }
```

```
do {  
  } while (x < 10);
```

- https://www.w3schools.com/js/js_loop_for.asp

- **Erros irão sempre acontecer:**
 - Erros de sintaxe (muitas vezes de digitação);
 - Recursos inexistentes (diferentes browsers);
 - Entrada de dados errada;
 - E muitas outras coisas misteriosas do além.

- **Exceções:**

- Delimitar área que será verificada:

```
try {  
    //Código passivo de erro  
}
```

- Capturar um eventual erro:

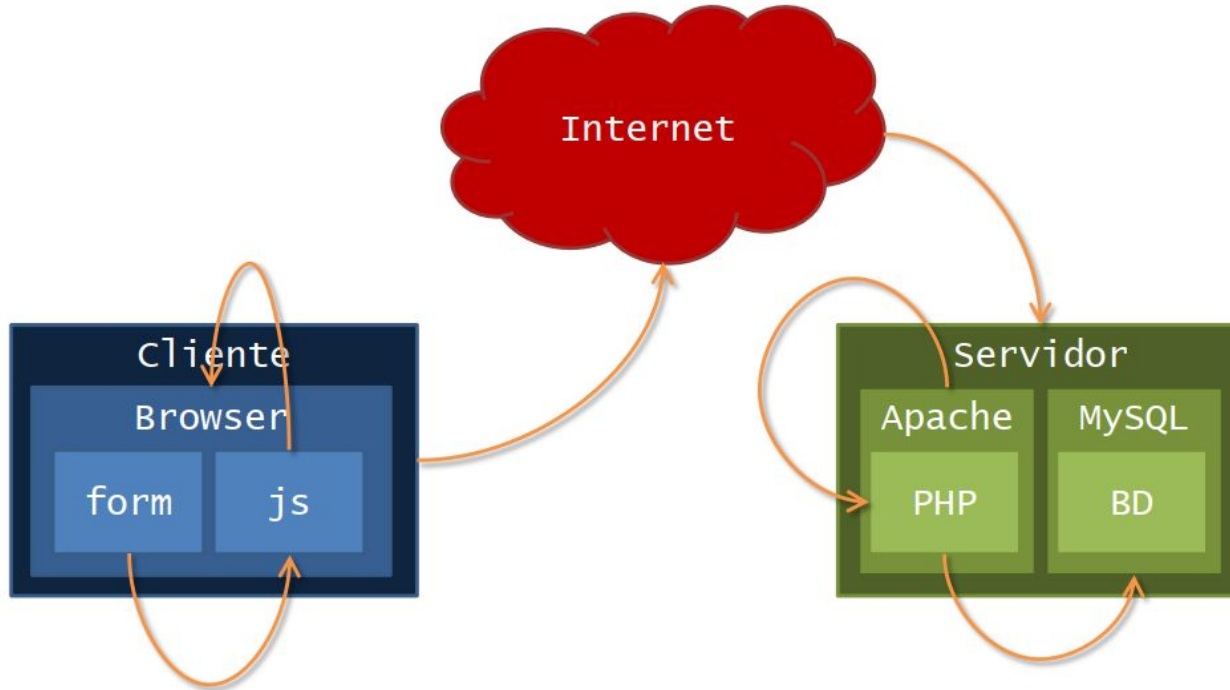
```
catch (erro) {  
    //Tratamento para eventual erro capturado  
}
```

JavaScript - Tratamento de Exceções



```
<script>
function verificar() {
  try {
    var x=document.getElementById("valor").value;
    if (x == "")      throw "Campo vazio";
    if (isNaN(x))     throw "não é um número válido";
    if (x > 10)       throw "Alto demais";
    if (x < 5)        throw "Baixo demais";
    document.getElementById("mesg").innerHTML = "Número aceito!";
  } catch(err) {
    var y=document.getElementById("mesg");
    y.innerHTML="Erro: " + err + ".";
  }
}
</script>
<h1>Exceções</h1>
<p>Digite um número entre 5 e 10:</p>
<input id="valor" type="text">
<button type="button" onclick="verificar()">Testar</button>
<p id="mesg"></p>
```

JavaScript - Formulário



- Ao desenvolver aplicativos para internet, dados serão informados pelo usuário;
- Antes de enviar estes dados ao servidor, é possível validar/verificar se eles têm coerência em relação ao que é solicitado:
 - O usuário esqueceu campos em branco?
 - O e-mail digitado é válido?
 - A data digitada é válida?
 - Num campo numérico, foi digitado um número?

JavaScript - Validação de Formulários



(Validando formulário: 1/2)

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function validarForm() {
        var val = document.getElementById("valido");
        try {
          var x = document.forms["meuForm"]["nome"].value;
          if (x == null || x == "") {
            throw "O Nome deve ser preenchido!";
          }
          var y = document.forms["meuForm"]["email"].value;
          var atpos = y.indexOf("@");
          var dotpos = y.lastIndexOf(".");
          if (atpos < 1 || dotpos < atpos + 2 || dotpos + 2 >= y.length){
            throw "Digite um e-mail válido!";
          }
          return true;
        }
      }
    </script>
  </head>
  <body>
    <div id="valido">
    </div>
  </body>
</html>
```

JavaScript - Validação de Formulários



(Validando formulário: 2/2)

```
    } catch (err) {  
        val.style.color = "#FF0000";  
        val.innerHTML = "Erro: " + err;  
        return false;  
    } //Fim catch  
} //Fim function  
</script>  
</head>
```

```
<body>  
    <form name="meuForm" action= "recebe.php"  
        onsubmit="return validarForm();" method="post">  
        Nome: <input type="text" name="nome">  
        e-mail: <input type="text" name="email">  
        <input type="submit" value="Enviar">  
    </form>  
    <p id="valido">Preencha o formulário e clique em Enviar</p>  
</body>  
</html>
```



- É possível adicionar novos elementos HTML;
- Qualquer tipo de elemento, definindo qualquer propriedade;
- Tudo através do JavaScript;

```
<body id="corpo">
  <h1>Adicionar Elementos</h1>
  <p>Digite o texto: <input type="text" id="texto"></p>
  <p><button onclick="adicionar()">Adicionar</button></p>
  <script>
    function adicionar() {
      var texto = document.getElementById("texto").value;
      var para = document.createElement("p");
      para.innerHTML = texto;
      var corpo = document.getElementById("corpo");
      corpo.appendChild(para);
    }
  </script>
</body>
```

- É possível remover elementos HTML;
- Qualquer tipo de elemento, com a condição de que conheçamos também o seu **pai**;

```
<body id="corpo">
  <h1>Remover Elemento</h1>
  <p><button onclick="remover()">Remover</button></p>
  <p id="texto">Texto que será removido...</p>
  <script>
    function remover() {
      var pai = document.getElementById("corpo");
      var filho = document.getElementById("texto");
      pai.removeChild(filho);
    }
  </script>
</body>
```

- Cookies são **variáveis** que ficam **armazenadas no browser** do visitante;
- Basicamente elas permitem que **a página lembre de qualquer informação ou interação** que já teve com o mesmo **browser/usuário** que esta acessando novamente:
 - Datas;
 - Dados Pessoais;
 - Login;
 - Senha;
 - Informações de Seção de Conexão;
 - Etc...

- Cookies são **variáveis** que ficam **armazenadas no browser** do visitante;
- Basicamente elas permitem que **a página lembre de qualquer informação ou interação** que já teve com o mesmo **browser/usuário** que esta acessando novamente:
 - Datas;
 - Dados Pessoais;
 - Login;
 - Senha;
 - Informações de Seção de Conexão;
 - Etc...

```
<!DOCTYPE HTML>

<html>
<head>
  <title>Teste com Cookies</title>
</head>
<body onload="checkCookie()">
  <h1>Sistema de biscoitos</h1>
  <p id="msg"></p>
  <script src="cookie.js"></script>
</body>
</html>
```



```
function checkCookie() {  
    var user = getCookie("user");  
    var msg = document.getElementById("msg");  
    if (user != null && user != "") {  
        msg.innerHTML = "Bem-vindo de volta " + user;  
    } else {  
        user = prompt("Digite seu nome:", "");  
        if (user != null && user != "") {  
            setCookie("user", user, 365);  
            msg.innerHTML="Bem-vindo " + user;  
        }  
    }  
}
```

```
function setCookie(c_name, value, exdays) {  
    var exdate = new Date();  
    exdate.setDate(exdate.getDate() + exdays);  
    var c_value = escape(value) +  
        ((exdays == null) ? "" : "; expires="+exdate.toUTCString());  
    document.cookie = c_name + "=" + c_value + "; path=/";  
}
```

```
function setCookie(c_name, value, exdays) {  
    var exdate = new Date();  
    exdate.setDate(exdate.getDate() + exdays);  
    var c_value = escape(value) +  
        ((exdays == null) ? "" : "; expires="+exdate.toUTCString());  
    document.cookie = c_name + "=" + c_value + "; path=/";  
}
```

JavaScript - Cookies



```
function getCookie(c_name) {  
    var c_value = document.cookie;  
    var c_start = c_value.indexOf(" " + c_name + "=");  
    if (c_start == -1) {  
        c_start = c_value.indexOf(c_name + "=");  
    }  
    if (c_start == -1) {  
        c_value = null;  
    } else {  
        c_start = c_value.indexOf("=", c_start) + 1;  
        var c_end = c_value.indexOf(";", c_start);  
        if (c_end == -1) {  
            c_end = c_value.length;  
        }  
        c_value = unescape(c_value.substring(c_start,c_end));  
    }  
    return c_value;  
}
```

Dúvidas?





Introdução ao JavaScript

Tecnólogo em Análise e Desenvolvimento de
Sistemas - Faculdades SENAC/PE

Professor: Danilo Farias