

# **Sistemas para Internet I**

## **JavaScript**

Profa. Diana F. Adamatti

Prédio do Centro de Ciências Computacionais – C3 1º. Andar  
– Sala 121

[dianaadamatti@furg.br](mailto:dianaadamatti@furg.br) [dianaada@gmail.com](mailto:dianaada@gmail.com)

# AGENDA

- Aula 1:
  - Introdução;
  - Variáveis;
  - Constantes;
  - Tipos;
  - Operadores;

# INTRODUÇÃO

- Linguagem de script utilizada para dar interatividade a páginas HTML;
- Criada em 1995 por Brendan Eich, quando ele trabalhava para o navegador Netscape;
- Padronizada pelo ECMAScript;

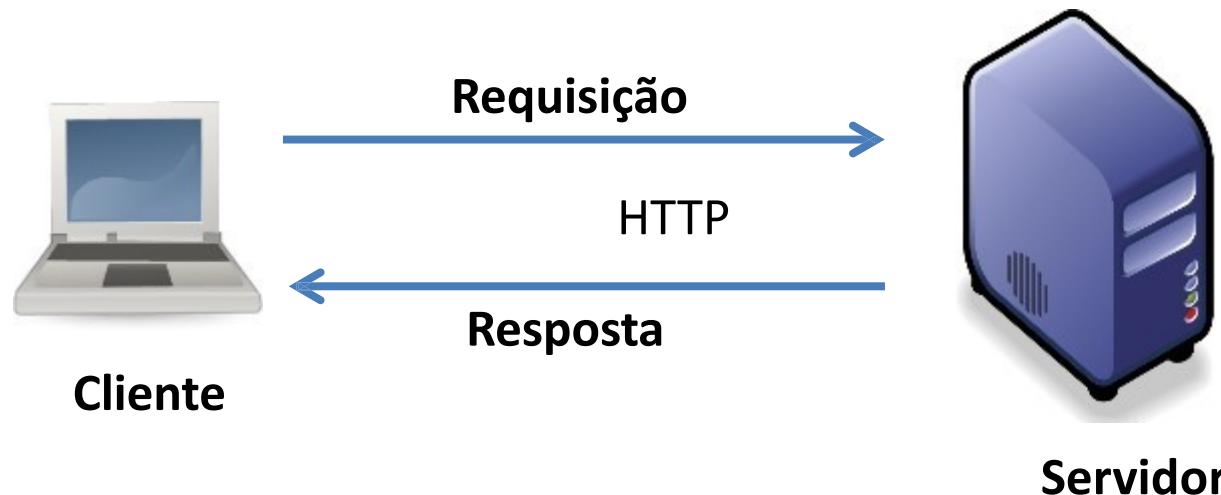
# INTRODUÇÃO

- Apesar dos nomes bem parecidos, JavaScript é **diferente** de Java;
- Java é uma linguagem de programação e JavaScript uma linguagem de Script;

# INTRODUÇÃO

- Diferente de linguagens como: PHP, ASP, JSP, etc, o JavaScript **executa no lado do cliente**;
- Também é possível executá-lo no lado do servidor, mas **não será abordado no contexto desta disciplina**.

# INTRODUÇÃO



# INTRODUÇÃO

- Existem três maneiras de incorporar JS ao HTML:

Inline: através dos manipuladores de evento: onclick, onmouseover, onload, etc

Incorporado: inserir o script na seção *head* do documento.

Externo: escreve o script em um arquivo externo e insere como um link na seção *head* do documento.

# INTRODUÇÃO

- **Inline**

```
<html>
  <head><title>JavaScript</title></head>
  <body>
    <input type="button" id="helloworld" value="Hello" onClick="alert('Hello World!');" />
  </body>
</html>
```

# INTRODUÇÃO

- Incorporado

```
<html>
    <head><title>JavaScript</title></head>
    <script>
        function funcao1() {
            alert("Eu sou um alert!");
        }
    </script>

    <body>
        <button type="button" onclick="funcao1() ">> xxx </button>
    </body>
</html>
```

# INTRODUÇÃO

- Externo

```
<html>
  <head>
    <title>JavaScript</title>
    <script type="text/javascript" language="javascript" src="hello.js"></script>
  </head>

  <body>
    <input type="button" id="helloworld" value="Hello" onClick="funcaoHello(); " />
  </body>
</html>
```

# INTRODUÇÃO

- Existem duas maneiras de fazer comentários em códigos JS:

# INTRODUÇÃO

- Comentário de linha (opção 1);

```
<html>
<head> <title> Javascript </title>
<script>
function funcao1() {
    // comentário em linha em javascript
    alert('Teste'); // exibir mensagem
    confirm('Você tem certeza?'); //caixa de confirmação
    prompt('Digite um valor'); //caixa de entrada de dados
}
</script>

</head>
<body>
<input type="button" id="teste" value="Tete - comentários" onclick="funcao1()" />
</body>

</html>
```

# INTRODUÇÃO

- Comentário de linha (opção 2);

```
<html>
  <head>
    <title>JavaScript</title>
    <script type="text/javascript" language="javascript" src="hello.js">
      <!-- comentário de linha em JS
          alert("Teste");           <!-- caixa de alerta
          confirm("Voce tem certeza?");  <!-- caixa de confirmação
          prompt ("Digite um valor");   <!-- caixa de dados
    &lt;/script&gt;
  &lt;/head&gt;

  &lt;body&gt;
    &lt;input type="button" id="x" value="Click" onClick="funcaoHello(); " /&gt;
  &lt;/body&gt;
&lt;/html&gt;</pre>
```

# INTRODUÇÃO

- Comentário de bloco;

```
<html>
  <head>
    <title>JavaScript</title>
    <script type="text/javascript" language="javascript" src="hello.js">
      /*
        comentário de linha em JS
        caixa de alerta
        caixa de confirmação
        caixa de dados
      */
      alert("Teste");
      confirm("Voce tem certeza?");
      prompt ("Digite um valor");
    </script>
  </head>

  <body>
    <input type="button" id="x" value="Click" onClick="funcaoHello(); " />
  </body>
</html>
```

# VARIÁVEIS

```
var x;           // Now x is undefined
var x = 5;       // Now x is a Number
var x = "John"; // Now x is a String
```

As variáveis em JS são:

- Fracamente tipadas e case sensitives (diferença entre minúsculas e maiúsculas);
- Podem começar com letra, cifrão e underline;
- Não podem **iniciar com número**, mas podem contê-los;
- Devem ser declaradas com a palavra-chave **var**;
- As variáveis tem um *tipo*, um *escopo* e um *identificador*;

# CONSTANTE

- As constantes em JS são definidas com a palavra reservada *const* na frente da variável;

```
const valor = 10;  
document.write("O valor constante é " + valor);
```

# TIPOS

- Undefined;
- Number;
- String;
- Array;
- Boolean;
- Object;
- Null;

# TIPOS

- Undefined;
- Number;
- String;
- Array;
- Boolean;
- Object;
- Null;

```
var x;                      // Undefined
var x = 5;                   // Number
var x = "SI";                // String

var carro=new Array();    //Array
cars[0]="Ferrari";
cars[1]="Volvo";
cars[2]="BMW";

var cars=new Array("Saab","Volvo","BMW");
```

# TIPOS

- Undefined;
- Number;
- String;
- Array;
- Boolean;
- Object;
- Null;

```
var x = true;           // Boolean
var y = false;          // Boolean

var person={           // Object
  firstname : "John",
  lastname  : "Doe",
  id        : 5566
};

pessoa    = null;       // Null
```

# TIPOS

```
name=person.lastname;           //acessando um objeto
name=person["lastname"];

/*
    Quando uma variável é declarada com a palavra 'new'
    a variável é declarada como um objeto;
*/
var name = new String;
var x     = new Number;
var y     = new Boolean;
```

# OPERADORES

- Operadores Aritméticos

Operador	Operação	Exemplo
+	Adição	$x+y$
-	Subtração	$x-y$
*	Multiplicação	$x*y$
/	Divisão	$x/y$
%	Módulo (resto da divisão inteira)	$x\%y$
-	Inversão de sinal	$-x$
++	Incremento	$x++$ ou $++x$
--	Decremento	$x--$ ou $--x$

# PRÉ-INCREMENTO E PÓS-INCREMENTO

- PRÉ-INCREMENTO
  - $++X$
  - PRIMEIRO REALIZA INCREMENTO E APÓS A OPERAÇÃO SEGUINTE
- PÓS-INCREMENTO
  - $X++$
  - PRIMEIRO REALIZA DETERMINADA OPERAÇÃO PARA APÓS REALIZAR O INCREMENTO
- \*\* O MESMO VALE PARA DECREMENTO

# EXEMPLO DE PRÉ-INCREMENTO E PÓS-INCREMENTO

```
<html>
  <head>
    <title>JavaScript</title>
    <script type="text/javascript">
      var x = 5;
      var y = 10;
      alert(++x);
      alert(y++);
      alert (y);
    </script>
  </head>

  <body>
  </body>
</html>
```

# OPERADORES

- Operadores de Comparaçõo

Operador	Operação	Exemplo
<code>==</code>	Igual a	<code>(x == y)</code>
<code>!=</code>	Diferente de	<code>(x != y)</code>
<code>==</code>	Idêntico (igual e mesmo tipo)	<code>(x === y)</code>
<code>!==</code>	Não idêntico a	<code>(x !== y)</code>
<code>&gt;</code>	Maior que	<code>(x &gt; y)</code>
<code>&gt;=</code>	Maior ou igual a	<code>(x &gt;= y)</code>
<code>&lt;</code>	Menor que	<code>(x &lt; y)</code>
<code>&lt;=</code>	Menor ou igual a	<code>(x &lt;= y)</code>

# OPERADORES

- Operadores de Atribuição

Operador	Operação	Equivalente
=	$x = 2$	-----
+=	$x += y$	$x = x + y$
-=	$x -= y$	$x = x - y$
*=	$x *= y$	$x = x * y$
/=	$x /= y$	$x = x / y$
%=	$x %= y$	$x = x \% y$
&=	$x \&= y$	$x = x \& y$

# getElementById – MANIPULANDO OBJETOS

- Javascript pode ser utilizado para manipular objetos HMTL
- Para acessar elementos HMTL com Javascript utilizamos o método `getElementById`
- `document.getElementById(id)`
- Id -> identificação de um elemento HTML

# getElementById – MANIPULANDO OBJETOS

- O resultado é um OBJETO
- Exemplo:
  - `var x = document.getElementById("v1");`
- x agora é uma variável do tipo objeto
- Uma de suas propriedades é o seu conteúdo => que pode ser acessado com a propriedade “value”
- Exemplo
  - `x.value` → corresponde ao conteúdo do objeto v1 de uma página HMTL

# getElementById – EXEMPLO

```
<html>
  <head>
    <title>JavaScript</title>
    <script type="text/javascript">
      function mostrar(){
        var x = document.getElementById("v1");
        alert("o valor digitado foi "+x.value);
      }
    </script>
  </head>
  <body>
    <form id="form">
      X: <input type="text" id="v1"> <br>
      <input type="button" id="resultado" value="resultado" onClick="mostrar();"/>
    </form>
  </body>
</html>
```

# EXERCÍCIOS

1. Em uma página HTML há dois *inputs* do tipo *text* e um *input* do tipo *button*. Cada *input* recebe um valor numérico e ao clicar no *button* os valores devem ser somados e exibido o resultado com um alert;

\*\* Você vai precisar da função de conversão de string para inteiro → parseInt(variavel)

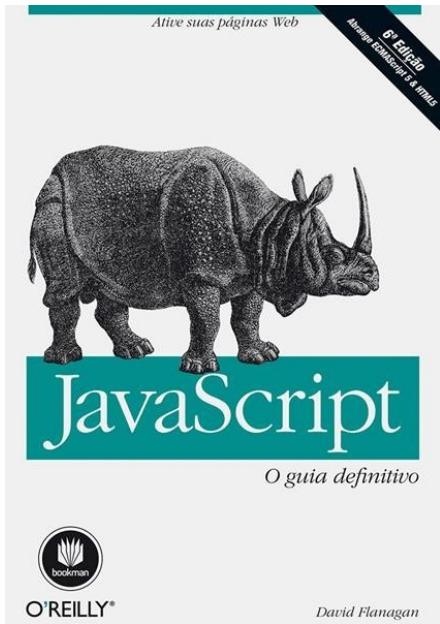
# ESTRUTURA DE CONTROLE

- **If:** use esta declaração para executar algum código só se uma condição especificada é verdadeira;
- **If ... else:** use esta declaração para executar algum código se a condição for verdadeira e outro código se a condição for falsa;
- **If ... else if ... else:** use esta instrução para selecionar um dos muitos blocos de código a ser executado;
- **Switch:** usar esta declaração para selecionar um dos muitos blocos de código a ser executado;

# Próxima Aula

- Estruturas de Controle;
- Estruturas de Repetição;
- Funções;
- Vetores;
- Objetos;

# REFERÊCIAS



Javascript – O Guia Definitivo,  
David Flanagan;



<http://www.w3schools.com/js>