

JavaScript

Ana Paula Ambrósio Zanelato
anapaula@unitoledo.br
Disciplina: Aplicações para Web

Linguagem de Programação

Conjunto de regras e convenções que especificam como transmitir informações entre pessoas e máquinas. É composta de um conjunto de símbolos (vocabulário) e a regras sintáticas de uso desses símbolos.

Script

- Script é uma sequência de instruções (ou até mesmo um programa) que são interpretadas por outro programa (navegador).
- Exemplos de Linguagens de Script:
 - JavaScript
 - JScript
 - VBScript

Script

- É interpretada no lado do cliente. Sendo assim, as respostas são mais rápidas.
- É altamente portátil e independente de hardware, dependendo apenas do browser.
- Usuário pode bloquear a sua execução!

Exemplo Script

```
JavaScript  
<script language="javascript">  
  var num;  
  num = 10;  
  
  if (num == 10)  
  {  
    alert("Valor 10");  
  }  
  
</script>
```

Script

- Os blocos de SCRIPT podem ser usados em qualquer parte de uma página HTML. Porém, é mais utilizado inserir o SCRIPT entre as tags HEAD.
- Estrutura do fonte HTML

```
<html>  
<head>  
  <title> titulo </title>  
  <script language="javascript">  
  </script>  
</head>  
  
<body>  
</body>  
</html>
```

Por que JavaScript ?

- JavaScript transforma uma página Web em uma experiência interativa.
- O JavaScript pode solucionar problemas como: “Como validar os campos de um formulário ? ”, “Como trocar dinamicamente as imagens em uma página?”.
- O JavaScript permite detectar praticamente qualquer ação que ocorre em uma página Web e adicionar uma funcionalidade a esta ação.

HTML: fornece estrutura;

CSS: adiciona o estilo;

JavaScript: injeta funcionalidade a partir de uma ação.

Adicionando JavaScript

Existem duas formas de inserirmos código JavaScript:

1) Na própria página HTML:

```
<html>
<head>
  <title></title>
  <script language="javascript">
    .....
  </script>
</head>
<body>
</body>
</html>
```

Adicionando JavaScript

Existem duas formas de inserirmos código JavaScript:

2) Criando um arquivo JavaScript (.js) e vinculando a página HTML:

```
<html>
<head>
  <title></title>
  <script language="javascript" src="funcoes.js">
  </script>
</head>
<body>
</body>
</html>
```

JavaScript

- Linguagem case-sensitive (Nome é diferente de nome).
- Comentários:
 - Uma única linha: // comentário
 - Bloco de texto: /* comentário */
- Aspas: Podem ser simples ou duplas;
 - Porém deve-se fechar com a mesma aspa que se abriu.

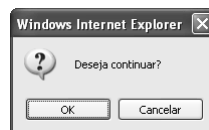
JavaScript – Caixa de Mensagens

- Alerta: exibe uma mensagem com um botão OK.
Sintaxe: alert(“Olá Mundo”);



JavaScript – Caixa de Mensagens

- Confirmação: apresenta a mensagem com opção de cancelar ou confirmar. Sintaxe:
if (confirm(“Deseja continuar?”)){
 }
else {
 }



JavaScript – Caixa de Mensagens

- Prompt: entrada de texto. Sintaxe:

```
var id;
```

```
id = prompt("Informe sua idade",0);
```



Valores em JavaScript

String (texto)

É uma cadeia de caracteres que pode incluir letras, sinais de pontuação e números.

Ex: "Aplicações para Web"
'JavaScript'

Uma string vazia é representada por "" ou "".

Valores em JavaScript

Número

Trabalha-se com dois tipos: *inteiro e ponto flutuante*

Inteiros: números não fracionários (...-2,-1,0,1,2...)

Ponto Flutuante: números fracionários (3.12, 0.5)

Os números em JavaScript podem variar de 10^{-307} a 10^{308} .

Valores em JavaScript

Booleano (lógico)

Só podem assumir dois valores:

Verdadeiro (*true*) e Falso (*false*)

Ex: variavel = true;

Variáveis

- Variáveis são nomes dados aos locais na memória do computador onde alguns dados são armazenados durante a execução de nossos scripts. Elas permanecem armazenadas até que a janela seja fechada.
- As variáveis podem ser declaradas ou então criadas dinamicamente.
- Para declarar uma variável, usamos a instrução **var**
Ex: **var** msg;
var x=1; //declaração com inicialização

Variáveis

- Algumas restrições da linguagem:
 - Podem ser usados caracteres;
 - O seu valor é *undefined* enquanto não inicializada.
 - O primeiro caracter não pode ser um número.
 - Não pode usar espaços.
 - Evite caracteres especiais.
 - Lembre-se: a linguagem JavaScript diferencia caracteres maiúsculos e minúsculos.

Operadores

- Atribuição (=)
- Adição ou concatenação de Strings (+)
- Subtração (-)
- Divisão (/)
- Multiplicação (*)
- Módulo (%) – resto da divisão
- Adiciona 1 (++)
- Subtrai 1 (--)

Operadores

- Verifica se dois valores são iguais (==)
- Verifica se dois valores são diferentes (!=)
- Verifica se um valor é maior que o outro (>)
- Verifica se um valor é maior ou igual ao outro (>=)
- Verifica se um valor é menor que o outro (<)
- Verifica se um valor é menor ou igual ao outro (<=)

Operadores

- Há três operadores lógicos:

E (and) - &&

if (a>2 && b==2)

Ou (or) - ||

if (a>2 || b==2)

Negação (not) - !

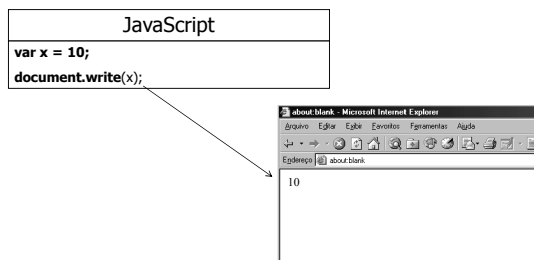
if (!a)

Conversores de Tipos

- Converte String para Inteiro
 - x = **parseInt**("10");
- Converte String para Ponto Flutuante (real)
 - x = **parseFloat**("9.5");
- Verifica se uma variável é numérica:

```
var x=1;  
if(!isNaN(x))  
  alert("É número");  
else ("Não é número");
```

Comandos de Saída



Estruturas de Seleção

	JavaScript
Seleção Simples	<pre>if (idade >= 18) alert("É maior");</pre>
Seleção Composta	<pre>if (idade >= 18) alert("É maior "); else alert("É menor ");</pre>

Estruturas de Seleção

JavaScript

Seleção
Encadeada

```
if ( idade >= 18 ) {  
  if (sexo = 'F')  
    alert("Mulher, maior.");  
}
```

Comando switch (Caso)

switch (variável)

```
{  
  case constante_1:  
    declaração_1;  
    break;  
  case constante_2:  
    declaração_2;  
    break;  
  case constante_n:  
    declaração_n;  
    break;  
  default:  
    declaração_default;  
}
```

É necessário incluir a instrução break; após a execução de cada condição, caso contrário o programa continuará executando.

Estruturas de Repetição

Existem três tipos de repetições:

- Com teste no início
- Com teste no final
- Com variável de controle

Estruturas de Repetição

- Iteração com teste no início

```
var x = 10;  
while (x < 20)  
{  
  alert("Valor de X = " + x);  
  x = x + 1;  
}
```

Sintaxe
while (expressão condicional)
{
 bloco de instruções
}

O laço será executado enquanto a expressão condicional for verdadeira.

Estruturas de Repetição

- Iteração com teste no final

```
var x = 10;  
do  
{  
  alert("Valor de X é " + x);  
  x = x + 1;  
}  
while (x < 20);
```

Sintaxe
do
{
 blocos de instruções;
}
while (expressão condicional);

Nesta iteração as instruções são executadas pelo menos uma vez.

O laço será executado até que a expressão condicional seja falsa ou enquanto for verdadeira.

Estruturas de Repetição

- Iteração com variável do controle

```
for (x=10; x<=20; x=x+1 )  
  alert("Valor de x é " + x);
```

Sintaxe
for (inicialização; expressão condicional; incremento)
{
 blocos de instruções;
}

Atribuições

- `x++;`
`(x = x + 1;)`
- `x--;`
`(x = x - 1;)`

Atribuições

- `x += 2;` `(x = x + 2;)`
- `x -= 2;` `(x = x - 2;)`
- `x *= 2;` `(x = x * 2;)`
- `x /= 2;` `(x = x / 2;)`

Depuração de Algoritmos (Erros)

- Todos os desenvolvedores estão sujeitos a criar códigos com erros. E em JavaScript os erros **NÃO** são fáceis de serem encontrados. Algumas ações a serem tomadas:
- Debug (passo-a-passo): FireBug <https://addons.mozilla.org/en-US/firefox/addon/1843>
- Firefox: melhor navegador de depuração.
- Procure exibir os valores das variáveis: `alert(var);`
- Procure comentar trechos de código e testar se o erro permanece, limitando a área do erro.

Bibliografia

- FREEMAN, Elisabeth; FREEMAN, Eric, **Use a cabeça!: HTML com CSS e XHTML**. Tradução: Betina Macêdo. Rio de Janeiro: Alta Books, c2006.
- OLIVIERO, Carlos A.; **Faça um site HTML 4.0: conceitos e aplicações**. 1. ed. São Paulo: Érica, 2007.
- MANZANO, José Augusto N. G.; TOLEDO, Suelly Alves de, **Guia de orientação e desenvolvimento de sites HTML, XHTML, CSS e JavaScript/Jsript**. 1. ed. São Paulo: Érica, 2008.
- MARCONDES, Christian Alfim; **HTML 4.0 Fundamental: a base da programação para web**. 2. ed. São Paulo: Érica, 2007.