

PROGRAMAÇÃO WEB Parte VI

Prof. Me. Felipe Diniz Dallilo



Javascript

- **Não é java !!!**
- Criado para o Netscape 2.0 em setembro de 1995.
- Linguagem de programação **interpretada**.
- Execução no "cliente/local" .
 - Tipagem dinâmica.
 - Todo comando é finalizado por um ";"

Olá mundo!

- `<html>`
- `<head>`
- `<script>`
- `alert('olá mundo');`
- `</script>`
- `</head>`
- `<body>`
- `</body>`
- `</html>`

Declarando variáveis

- Para declarar uma variável no javascript é necessário apenas definir o nome e atribuir um valor a mesma como ilustrado nos exemplos abaixo:

```
Variavel_Int = 1;
```

```
Variavel_Str = "Olá Mundo"
```

Olá mundo 2!

- Adicionem o texto do "Olá mundo" em uma variável e mostrem no alert!

Operações numéricas

Alguns exemplos abaixo para efetuar algumas operações no javascript:

Soma

Soma = 1 + 1 ;

Multiplicação

Mult = 1 * 1;

Divisão

Div = 1 / 1;

Subtração

Sub = 1 - 1;

Concatenar

Concatenar é a união de uma variável de texto com outra(s) variáveis (texto, numérico, data). Exemplo:

```
A = 'texto ' + ' digitado ';
```

```
B = A + ' aqui ';
```

```
C = 2;
```

```
D = B + C + 'x !'
```

Olá mundo 3!

Crie uma soma e mostre com o alert o resultado da seguinte forma:

'O resultado da sua soma é: **x**'

*x é o valor da sua soma concatenado com a frase.

Função

- Instrução ou conjunto de instruções que pode ser chamada(s) pelo fluxo de execução do script. Exemplo:
function nome_da_funcao() {
- // instruções
- }

```
function soma()  
{  
    A = 1 + 1;  
    alert(a);  
}
```

Atribuir função ao botão

Após declarada, para efetuar uma chamada a função javascript através do clique do botão é necessário efetuar a seguinte chamada:

```
<input type="submit" value="Somar"  
onclick="somar()"/>
```

Ola mundo 4!

- Criem um botão que ao ser clicado, mostre a mensagem "Olá mundo!" Pelo alert através de uma função!

Coletar valor de entrada

- Com a função **document.getElementById** é possível coletar valores pelo ID do componente HTML.
- Campo de texto:
- `<input type="text" id="txtID" />`
- Coletando valor digitado no javascript:
- `var s = document.getElementById('txtID').value;`

Ola mundo 5!

```
<html>
<script>
    function teste()
    {
        var s  = document.getElementById('txtID').value;
        var s2 = document.getElementById('txtNome').value;
        result = s + s2;
        alert(result);
    }
</script>
<body>
    <input type="text" id="txtID" />
    <input type="text" id="txtNome" />
    <input type="submit" value="teste" onclick="teste()"/>
</body>
</html>
```


Var

- Foi visto anteriormente que é possível declarar uma variável no javascript apenas definido o nome da variável e atribuindo o valor, como ilustrado no exemplo abaixo:
- `valor = "abcde"`

Então, qual a diferença entre a declaração acima e a abaixo?

`Var valor = "mais de 8 mil"`

Var

- A diferença entre as mesmas está associada ao **escopo** da variável!! Se declarado com a palavra chave **“var”**, a variável é criada no escopo corrente (com visibilidade apenas dentro de uma function por exemplo)
- Se criado **sem o uso da palavra chave**, a variável assume o escopo de uma **variável global** (mesmo que criada dentro da function)

Por outro lado, **uma variável declarada sem o “var”, pode ser apagada por comandos js** e uma variável global criada com a utilização da **palavra chave “var” não pode** (a verdadeira variável global).

“delete valor;” (Funciona apenas sem o var)

“valor = null; ” (Criado com o var)

Const e let

- Para criar uma constante (apenas leitura), utilizar a palavra chave “**const**” como ilustrado abaixo:
- **const** i = 13;

Para criar uma variável local de escopo, utilizar a palavra chave “**let**”:

let valor = “teste”;

Diferença entre let e var

Talvez saiba que sempre deve usar o `var` para tornar o escopo local, ou seja ele vale apenas dentro da função onde ela foi declarada (pode ser global se não está dentro de função).

Mas isto não era suficiente, precisa ter um um escopo de bloco. O `let` foi criado, e está disponível nas versões mais novas da linguagem justamente para prover este escopo mais limitado.

Exemplos:

```
function exemplo() {  
  //x poderia ser acessado aqui  
  for(var x = 0; x < 5; x++) {  
    //x existe aqui  
  };  
  //x está visível aqui novamente  
};  
  
function exemplo() {  
  //x não existe aqui  
  for(let x = 0; x < 5; x++ ) {  
    //x existe aqui  
  };  
  //x não está visível aqui novamente  
};
```

StackOverflow

Conversão de tipos

- Algumas possibilidades de conversão de dados String para numérico no javascript:
- parseInt()
- parseFloat()
- `A = parseInt('1') + parseInt('2');`
- Numérico para String:
- `var a = 1;`
`var n = a.toString();`

Estrutura Condicional

- Para criar uma condição no Javascript é necessário utilizar o seguinte código:

```
if (condição) {  
    instruções que serão executadas caso a condição seja satisfeita  
}
```

Estrutura Condicional

- Caso seja necessário tomar duas ações distintas (uma quando satisfaz a condição e a outra quando não satisfaz) utilizar a estrutura abaixo:
- ```
if (condição) {
 instruções quando satisfaz a condição
} else {
 instruções quando não satisfaz
}
```

# Estrutura Condicional

Estrutura condicional aninhada:

```
if (condicao 1) {
 instruções quando satisfaz a condição 1
} else if (condição2) {
 instruções quando satisfaz a condição 2
} else {
 instruções quando não satisfaz nenhuma condição
}
```

# Condição

- Basicamente, uma condição é um valor booleano que pode ser obtido de diversas formas como:

## ➤ Comparando valores:

| Operador | Descrição               |
|----------|-------------------------|
| ==       | igual                   |
| >        | maior que               |
| <        | menor que               |
| >=       | maior igual             |
| <=       | menor igual             |
| !=       | Diferente               |
| !==      | Valor ou tipo diferente |

# Condição

## ➤ Associando condições com operadores lógicos

| Operador | Descrição |
|----------|-----------|
| &&       | E         |
|          | Ou        |
| !        | Não       |

**Ex:**

if( condição1 **&&** condicao2)



# Coletar valor de combo

- Já foi visto como definir valores inalteráveis ao usuário, no entanto, não foi visto como coletar estes valores e utilizar no javascript. Segue abaixo como fazer:

```
<select id="opcao">
 <option value="branco">branco</option>
 <option value="preto">preto</option>
 <option value="verde">verde</option>
 <option value="vermelho">vermelho</option>
</select>
```

```
function teste()
{
 var s = document.getElementById('opcao').value;
 alert(s);
}
```

## Visualização

Selecione a cor:  teste

- branco
- preto
- verde
- vermelho

preto

OK

## Referências

- Lemay, Laura; Colburn, Rafe; Tyler, Denise. Aprenda a Criar Páginas Web com HTML e XHTML em 21 Dias. Editora Pearson. 2013. 1ª Edição. ISBN: 9788534614283
- Chak, Andrew. Como Criar Sites Persuasivos. Editora Pearson. 2012. 1ª Edição. ISBN: 9788534615112
- Fábio Flatschart; Clécio Bachini; Cesar Cusin. Open Web Platform. Editora Pearson. 2013. 1ª Edição. ISBN: 9788574526140