

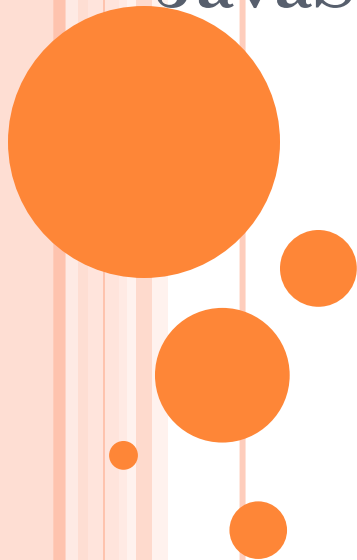
INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA
CURSO TÉCNICO INTEGRADO EM INFORMÁTICA
CAMPUS CURRAIS NOVOS

Desenvolvimento Web

JavaScript – aula II – *variáveis (cont.) e diálogos*

Professor: Bruno Gomes

2010



INTRODUÇÃO

- Na aula anterior:
 - Introdução à tecnologia Java Script,
 - Inserção de Java Script na página e
 - Declaração de variáveis

- Na aula de hoje:
 - Tipos de variáveis (continuação)
 - Janelas de diálogo



TIPOS DE VARIÁVEIS EM JAVASCRIPT

- Números (Inteiros e ponto-flutuante)
- **Strings**
- Booleanos
- *Arrays*
- Objetos



TIPO STRING

- Pode ser inseridas utilizado aspas simples ou duplas, mas nunca ambas
 - **var** nome = "Bruno Gomes";
 - **var** nome 'Bruno Gomes';
 - ERRADO: **var** nome "Bruno';
- Também pode ser declarado com o operador *new*:
 - var str = **new** String ("Olá")
- Strings podem ser concatenadas com o operador +
 - var nome = "Bruno" + " Gomes" // "Bruno Gomes"
- Funções de conversão:
 - *parseInt* – converte uma *String* com um número inteiro em um inteiro
 - *parseFloat* – converte uma *String* em com um número real em um *float*



ALGUMAS FUNÇÕES DE STRING

- **charAt(i)** Retorna o caractere no índice especificado
- **concat(s, s2,..., sn)** Concatena duas ou mais *Strings* e retorna uma cópia da string concatenada
- **indexOf(str)** Retorna a posição da primeira ocorrência encontrada do valor especificado no argumento na *string*
- **lastIndexOf(str)** Retorna a posição da primeira ocorrência encontrada do valor especificado no argumento na string
- **slice(ini, fim)** Extrai parte de uma *String* e retorna a nova string
- **split(sep, lim)** Quebra a String em um *array* de *Substrings* (dado um *separador*)
- **substr(ini, fim)** Extrai uma *substring*, começando de uma posição inicial (ini) até a posição final (fim)
- **substring (ini, fim)** Extrai os caracteres de uma String entre os índices especificados de início e fim (subconjunto da *string*)
- **toLowerCase()** Converte a String para letras minúsculas
- **toUpperCase()** Converte a string para letras maiúsculas



TIPOS DE VARIÁVEIS

○ *Arrays*

- declaração é feita com um []
- **var** i = []
- **var** numeros = [1, 2, 3, 4, 5]
- **var** pessoas = ["joao", "maria", 'jose']
- Índice do *array* começa de 0



TIPOS BOOLEANOS E OBJETOS

○ Booleanos

- Dois valores possíveis (*true* – verdadeiro ou *false* - *false*)
- **var** isNumber = false;

○ Objetos

- **var** Robot = new Object();
- **var** str = new String('oi');



NOMES DE VARIÁVEIS

- O identificador (nome) de uma variável deve seguir certas regras:
 - Qualquer combinação de letras, números, sublinhados (`_`) e cifrão (`$`)
 - Não deve iniciar com um dígito e não deve conter espaços em branco
 - Deve ser diferente de palavras-chave da linguagem (como *if*, *var*, *while*, etc.)
- Lembre-se que nomes de variáveis em *JavaScript* diferenciam letras maiúsculas e minúsculas
 - Ex.: “Pessoa”, “pessoa” e “PESSOA” são três variáveis diferentes



EXERCÍCIO

- Quais declarações abaixo são válidas?
 - nome\$
 - Rua
 - _1num
 - 10p
 - p&b
 - __\$valor
 - a:b
 - \$valor\$
 - #nota1
 - Kl/h




SEQUÊNCIAS DE ESCAPE

- Caracteres especiais que podem ser utilizados dentro de *Strings*
- `\n`, `\t` e `\r` na tabela a seguir alteram a exibição de texto dentro de caixas de diálogo
 - não necessariamente a alteram a exibição de XHTML (a não ser que se use a *tag* `<pre>`)

Caractere	Descrição
<code>\n</code>	Nova linha
<code>\t</code>	Tabulação horizontal
<code>\r</code>	Retorna o cursor para o começo da linha atual
<code>\\</code>	Caractere de barra invertida
<code>\"</code>	Caractere de aspas duplas em uma String de aspas duplas. Ex.: <code>window.alert ("\"entre aspas\");</code>
<code>\'</code>	Caractere de aspas simples em uma String de aspas simples. Ex.: <code>window.alert ('\entre aspas');</code>

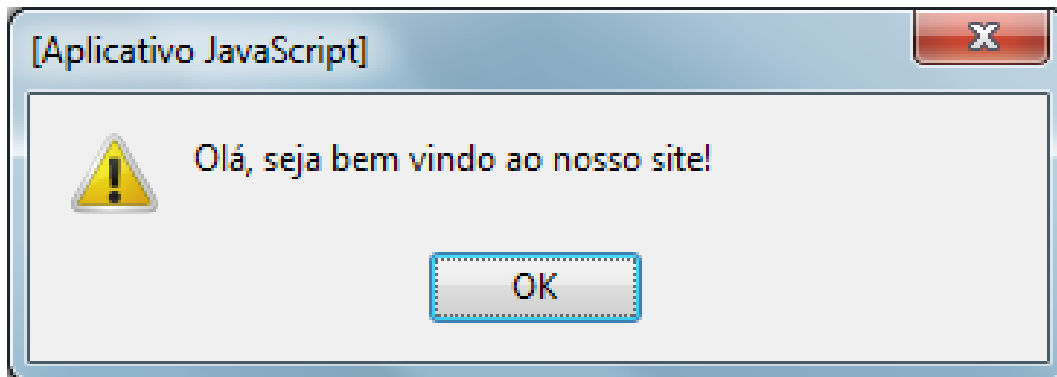


JANELAS (OU CAIXAS) DE DIÁLOGO

- Três tipos de janela para interação com o usuário
 - **Alert**
 - Exibe uma mensagem para o usuário
 - **Confirm**
 - Quando se quer uma resposta do usuário para tomar alguma decisão. Ele pode responder “OK” ou “Cancelar”
 - **Prompt**
 - Quando se quer receber alguma entrada de dados do usuário, normalmente ao abrir uma página
 - Abusar do uso de Janelas pode não ser uma boa idéia
 - Irritam o usuário
 - Deve-se preferir na maioria dos casos o uso de elementos de interação que estejam presentes em XHTML, como os elementos de formulário
- 

JANELA ALERT

```
<head>  
<title>Diálogo Alert</title>  
  
  <script type="text/javascript">  
    <!--  
      window.alert ("Olá, seja bem vindo ao nosso site!");  
    // -->  
  </script>  
</head>
```



JANELA CONFIRM

- Dois botões, OK e Cancelar
 - OK retorna o booleano “*true*”
 - Cancelar retorna o booleano “*false*”
- *Para testar esta janela, coloque a função abaixo em um arquivo externo:*

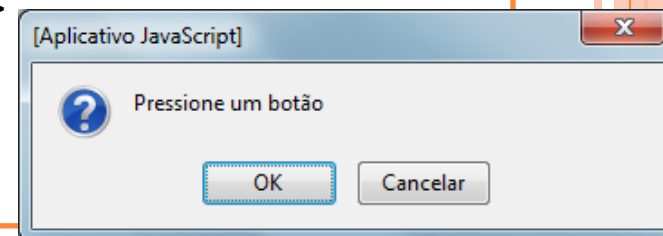
```
function show_confirm()
{
  var r = window.confirm ("Pressione um botão");
  if (r == true) {
    window.alert("Você pressionou 'OK!'");
  }
  else {
    window.alert("Você pressionou 'Cancelar!'");
  }
}
```



JANELA CONFIRM

- Documento XHTML que usa a função definida no slide anterior
 - OBS: *dialogs.js* foi o nome escolhido por mim para o arquivo *JavaScript* externo.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Janela Confirm!</title>
    <script type="text/javascript" src="dialogs.js">
    </script>
  </head>
  <body>
    <p>
      <input type="button" onclick="show_confirm()"
        value="Exibir janela de confirmação" />
    </p>
  </body>
</html>
```



JANELA PROMPT

- A função abaixo é um exemplo do uso da janela *prompt* para pegar a entrada de dados de um usuário

```
function show_prompt()
{
  var name = window.prompt ("Por favor, digite o seu nome",
                              "<Seu Nome aqui!>");

  if (name != null && name != "")
  {
    window.alert("Olá " + name + "! Como você está?");
  }
}
```



JANELA PROMPT

- XHTML utilizando a função do slide anterior que exibe uma Janela do tipo “*Prompt*”

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>Janela Prompt</title>
    <script type="text/javascript" src="dialogs.js">
    </script>
  </head>
  <body>
    <p>
      <input type="button" onclick="show_prompt()"
        value="Exibir Janela Prompt!" />
    </p>
  </body>
</html>
```

