

JavaScript

UMA INTRODUÇÃO

Elementos

Variáveis

Literais

Operadores

Estruturas de Controle

Funções

Objetos

Variáveis

Não é tipada!

Pode ser declarada com a palavra reservada var:

- var foo;
- Quando se usa o var para declarar a variável é local.

Pode ser criada automaticamente com a atribuição de um valor:

- foo=1; blah="Hi Dave";
- Quando não se usa o var para declarar a variável é global.

Literais

As mais comuns:

- Numbers 17 123.45
- Strings "Hello Dave"
- Boolean: true false
- Arrays: [1,"Hi Dave",17.234]



Arrays podem conter qualquer coisa!

Operadores

Aritméticos, comparação, atribuição, boolean, bitwise, boolean (como em C).

+ - * / % ++ -- == != > <

&& || ! & | << >>

Estruturas de Controle

Todos da linguagem C:

- if if-else ?: switch
- for while do-while

E mais alguns:

- for (var in object)
- with (object)

Comentários JS e HTML

<SCRIPT>

<!--

document.write("Hello World");

document.bgColor="BLUE";

-->

function mensagem()

{ /* comentário */ JAVASCRIPT comment

}

</SCRIPT>

HTML comment

Funções JavaScript

Uma função é um grupo de linha(s) de código de programação destinado uma tarefa bem específica e que podemos de necessário, utilizar várias vezes. A utilização de funções melhora bastante a leitura do script.

A palavra reservada `function` é usada para definir uma função (subrotina):

```
function add(x,y) {  
  ◦   return(x+y);  
}
```


Declaração de Funções

Para declarar ou definir uma função, utiliza-se a palavra **function**.
A sintaxe de uma declaração de função é a seguinte:

```
function nome_da_função(argumentos) {  
    ... código de instruções ...  
}
```

O nome da função segue as mesmas regras das variáveis (número de caracteres indefinido, começado por uma letra pode incluir números...).
Todos os nomes de funções num um script devem ser únicos.
A menção dos argumentos é facultativo, mas os parênteses são obrigatórios. É alias graças aos parênteses que o interpretador Javascript distinga as variáveis das funções.

Chamada de função

A invocação de uma função se faz através do nome da função com parênteses.

```
nome_da_função();
```

Onde declarar funções

Preferencialmente no <head>

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="Javascript">
      function mensagem() {
        document.write("Bem vindo a minha página");
      }
    </SCRIPT>
  </HEAD>

  <BODY onLoad="mensagem()">

  </BODY>
</HTML>
```

Passando valores para funções

Para passar um parâmetro a uma função, fornece-se um nome de uma variável dentro da declaração da função. Na declaração da função, escreve-se:

```
function Escreva(X) {  
    alert(X);  
}
```

O nome da variável X é definida como um parâmetro da função.

Na invocação da função, fornece-se o texto a ser escrito:

```
Escreva("Bom dia a todos");
```

Retornando valores

Para enviar um resultado, basta fornecer o nome da variável ou valor após o comando return.

```
function quadrado (numero) {  
  var resposta = numero*numero;  
  return resposta;  
}
```

Eventos JavaScript

JavaScript suporta manipulação de eventos :

- Executando comandos javascript quando alguns eventos ocorrem.
- Outras vezes o valor resultante de um comando determina uma ação do navegador

Botões

É possível associar os botões dos formulários HTML aos eventos do JavaScript

```
<FORM>  
<INPUT TYPE=BUTTON  
valor="Don't Press Me"  
onClick="alert('now you are in trouble!') " >  
</FORM>
```

Alguns Eventos

onUnLoad

onLoad

onClick

onMouseUp

onMouseDown

onDbIClick

onMouseOver

Window events

```
graph LR; WE[Window events] --> UL[onUnLoad]; WE --> LO[onLoad]; BE[Button events] --> CL[onClick]; BE --> MU[onMouseUp]; BE --> MD[onMouseDown]; BE --> DC[onDbIClick]; LE[Link events] --> MO[onMouseOver];
```

Button events

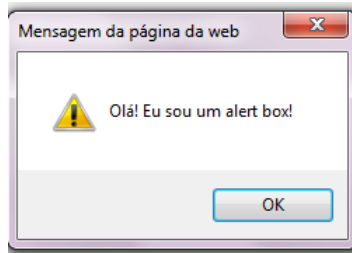
Link events

Descrição dos eventos

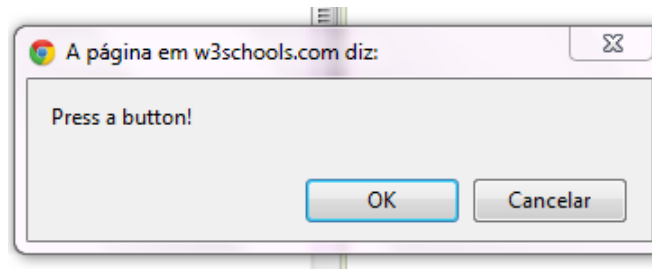
Evento	Descrição
onblur	Script é executado quando o elemento perde foco
onclick	Script to be run on a mouse click
ondblclick	Script to be run on a mouse double-click
onfocus	Script é executado quando o elemento obtem foco
onmousedown	Script to be run when mouse button is pressed
onmousemove	Script to be run when mouse pointer moves
onmouseout	Script to be run when mouse pointer moves out of an element
onmouseover	Script to be run when mouse pointer moves over an element
onmouseup	Script to be run when mouse button is released
onkeydown	Script to be run when a key is pressed
onkeypress	Script to be run when a key is pressed and released
onkeyup	Script to be run when a key is released

Mensagens

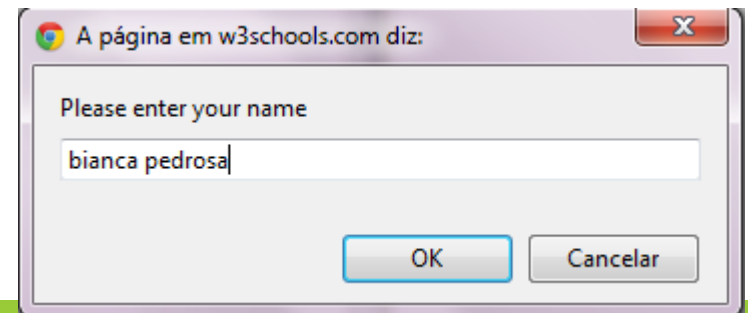
Alert box



Confirm box



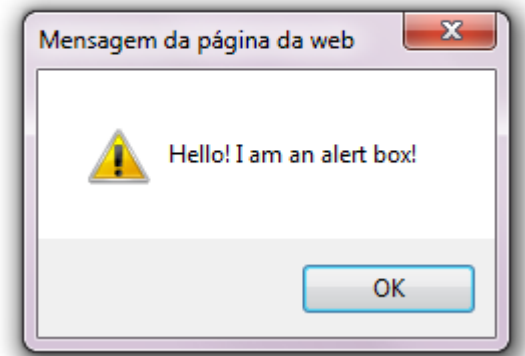
Prompt box



Alert box

```
1 <html>
2 <head>
3 <script type="text/javascript">
4   function sayHello()
5   {
6     alert("Hello! I am an alert box!");
7   }
8 </script>
9 </head>
10 <body>
11
12     <input type="button" onclick="sayHello()" value="Say Hello" />
13
14 </body>
15 </html>
```

Say Hello

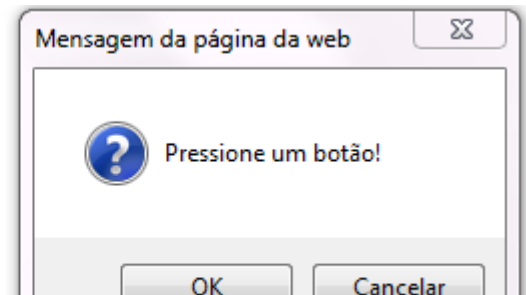


Confirm box

```
1 <html>
2 <body>
3   <p>Click aqui para ver um confirm box.</p>
4   <button onclick="confirma()">Confirma?</button>
5   <p id="demo"></p>
6   <script type="text/javascript">
7     function confirma()
8     {
9       var x;
10      var r=confirm("Pressione um botão!");
11      if (r==true)  x="Você pressionou OK!";
12      else          x="Você pressionou Cancel!";
13      document.getElementById("demo").innerHTML=x;
14    }
15  </script>
16 </body>
17 </html>
```

Click aqui para ver um confirm box.

Confirma?



Prompt box

```
1 <html>
2 <body>
3
4 <p>Click para ver um prompt box.</p>
5 <button onclick="informe()">clique aqui</button>
6 <p id="demo"></p>
7 <script type="text/javascript">
8   function informe()
9   {
10    var x;
11    var name=prompt("Informe seu nome","Bianca Pedrosa");
12    if (name!=null)
13    {
14      x="<p>Hello " + name + "! How are you today?</p>";
15      document.getElementById("demo").innerHTML=x;
16    }
17  }
18 </script>
19
20 </body>
21 </html>
```

Objetos Pré-definidos

String

- Métodos para manipulação de strings

Math

- funções matemáticas: `log()`, `cos()`, `random ()`, etc

Date

- funções para conversão de datas

Number

- limites, conversão para string

Funções Matemáticas

Função	Descrição
<u>abs(x)</u>	Retorna o valor absoluto de x
<u>ceil(x)</u>	Retorna x, arredondado para cima
<u>cos(x)</u>	Retorna o cosseno de x, em radianos
<u>exp(x)</u>	Retorna o valor de E^x
<u>floor(x)</u>	Retorna x, arredondado para baixo
<u>log(x)</u>	Retorna o logaritmo natural (base E) de x
<u>max(x,y,z,...,n)</u>	Retorna o maior valor
<u>min(x,y,z,...,n)</u>	Retorna o menor valor
<u>pow(x,y)</u>	Retorna o valor de x elevado a y
<u>random()</u>	Retorna um número randômico entre 0 e 1
<u>round(x)</u>	Arredonda x para o inteiro mais próximo
<u>sin(x)</u>	Retorna o seno de x, em radianos
<u>sqrt(x)</u>	Retorna a raiz quadrada de x
<u>tan(x)</u>	Retorna a tangente de x

Funções para manipular strings

Função	Descrição
<u>charAt(x)</u>	Retorna o <i>character</i> na posição x
<u>charCodeAt(x)</u>	Retorna o Unicode do <i>character</i> na posição x
<u>concat(a,b)</u>	Junta duas ou mais <i>strings</i> e retorna uma cópia delas combinadas
<u>fromCharCode()</u>	Converte Unicode em <i>characters</i>
<u>indexOf('a')</u>	Retorna a posição da 1ª ocorrência de 'a' na <i>string</i>
<u>lastIndexOf('a')</u>	Retorna a posição da última ocorrência de 'a' na <i>string</i>
<u>match()</u>	Busca pela existência de uma expressão na <i>string</i> e retorna suas múltiplas ocorrências
<u>replace()</u>	Searches for a match between a substring (or regular expression) and a string, and replaces the matched substring with a new substring
<u>search('xxx')</u>	Busca pela ocorrência da expressão 'xxx' na <i>string</i> , e retorna a posição da ocorrência

Funções para manipular strings

Função	Descrição
<u>slice()</u>	Extraí uma parte de uma string e retorna a nova string
<u>split()</u>	Divide uma string em um array de substrings
<u>substr(x, y)</u>	Extraí os caracteres de uma <i>string</i> , começando em posição x, seguindo por y posições
<u>substring(x, y)</u>	Extraí os characters de uma <i>string</i> , entre os índices x e y
<u>toLowerCase()</u>	Converte a <i>string</i> str em letras minúsculas
<u>toUpperCase()</u>	Converte a <i>string</i> str em letras maiúsculas
<u>valueOf()</u>	Retorna o valor primitivo de uma <i>string</i>

Objetos

Objetos tem atributos e métodos.

Muitos tipos de objetos pré-definidos

Uso dos objetos segue sintaxe C++/Java:

- `objectname.attributename`
- `objectname.methodname()`

Objetos Pré-definidos

JavaScript inclui alguns objetos que são automaticamente criados, estão sempre disponíveis

- document
- navigator
- screen
- window

O Objeto Document

Muitos atributos do documento corrente podem ser manipulados via document object:

- Title Referrer
- URL Images
- Forms Links
- Colors

Métodos do Document

`document.write()` funciona como um `print` – a saída vai para o documento HTML.

`document.write("O título deste HTML é" + document.title);`



Concatenação de string!

Exemplo JavaScript

```
<HEAD>
```

```
<TITLE>JavaScript is Javalicious</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
  <H3>
```

```
    I am a web page and here is my name:
```

```
  </H3>
```

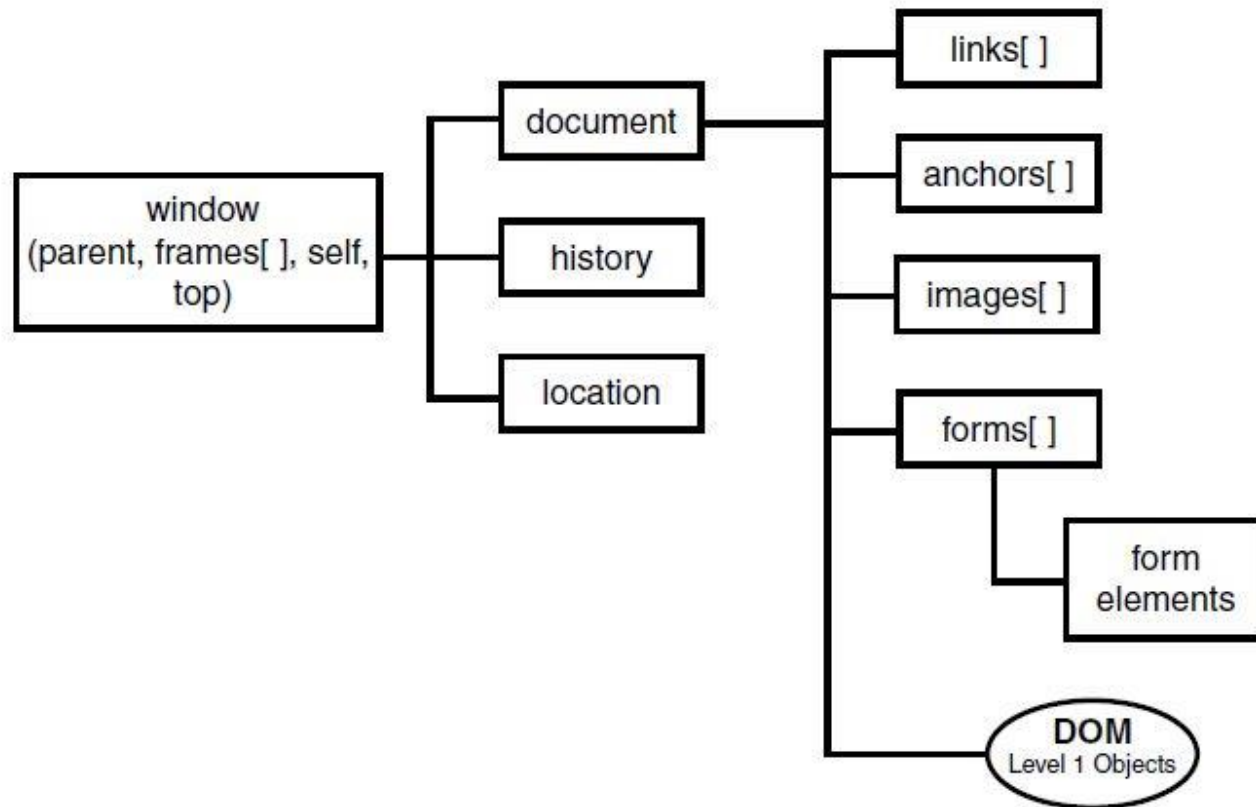
```
<SCRIPT>
```

```
  document.write(document.title);
```

```
</SCRIPT>
```

```
<HR>
```

Modelo de Objetos do Document



Formulário Cálculo Peso Ideal

```
30 <body>
31 <form name="ficha" action="nenhum.php" method="post">
32   <h1> Cálculo do Peso Ideal </h1>
33
34   Informe sua altura:
35   <INPUT TYPE='TEXT' NAME='altura'><BR>
36   Informe seu sexo:
37   <SELECT NAME='sexo'>
38     <option value='feminino'>feminino</option>
39     <option value='masculino'>masculino</option>
40   </select>
41   </br>
42   <input type="button" value="envia" onclick="calculapeso()" />
43 </FORM>
44 </body>
```

Cálculo do Peso Ideal

Informe sua altura:

Informe seu sexo: feminino ▼


```

4 <script type="text/javascript">
5   function calculapeso()
6   {
7       var a=document.forms["ficha"]["altura"].value;
8       var s=document.forms["ficha"]["sexo"].value;
9       if (a==' ')
10      {
11          alert("Vc precisa informar a altura!");
12          return false;
13      }
14      if(s==' ')
15      {
16          alert("Vc precisa informar o sexo!");
17          return false;
18      }
19      if (s=='masculino')
20          peso= (72.7*a)-58;
21      else
22          peso=(62.1*a)- 44.7;
23      document.write("<h2>Seu peso ideal é ", Math.round(peso), "</h2>");
24      return true;
25  }
26 </script>

```

Validação de Formulário

JavaScript pode ser usado para validar as informações digitadas pelo usuário

Quando o botão submit é pressionado o script checa os valores de todos os campos.

A validação de formulário é uma boa prática para verificar se o usuário preencheu todos os campos do formulário antes de submeter

Pode evitar que usuários pulem o seu formulário e criem solicitações de páginas manualmente

Não confie somente no javascript do lado do cliente para validar campos de formulários!

Mais detalhes sobre loops

Loops ou laços de repetição

Todo laço de repetição tem 4 partes:

1. o valor inicial
2. o valor final ou condição de teste
3. a ação ou código a ser executado
4. o incremento

Valor Inicial

Um valor inicial é atribuído a uma variável, normalmente chamado de *i* (mas você pode chamá-lo de qualquer coisa que você quiser). Esta variável atua como contador para o loop

Ação ou Código a ser Executado

Você digita um bloco de código uma vez e ele vai ser executado o número de vezes entre o valor inicial e o valor final do loop

Valor Final ou Teste

O circuito precisa de um limite a ser definido: ou um número definido (loop 5 vezes) ou uma condição de verdade (loop até que essa condição é avaliada como true). Caso contrário, você corre o risco de desencadear um loop infinito . Isso é muito ruim: é uma repetição interminável do mesmo código que impede os navegadores dos usuários de responder.

Incremento

Esta é a parte que move o ciclo para a frente: o contador que inicializar tem que se mover para cima (ou para baixo no caso de você optar por looping para trás). Enquanto o circuito não atingir o valor final ou a condição de teste não é satisfeito, o contador é incrementado (ou decrementado). Isso geralmente é feito usando operadores matemáticos.

O loop For

// Loop para o número de vezes entre o valor inicial

// e o valor final. Incrementar o valor cada vez que

// o limite não foi atingido.

for (var = início; var <= fim; var = incremento)

{

 // Código a ser executado

}

Exemplo For

```
1 <html>
2 <head>
3   <script type="text/javascript">
4     function myFunction()
5     {
6       var x="",i;
7       for (i=0;i<5;i++)
8       {
9         x += "The number is " + i + "<br />";
10      }
11      document.write(x);
12    }
13  </script>
14 </head>
15
16 <body>
17   <p>Clique no botão para executar um loop for.</p>
18   <button onclick="myFunction()">Tente!</button>
19 </body>
20 </html>
```

O loop While

1. // enquanto o valor inicial não atingir o valor final
2. // se a condição for avaliada como falsa desde o início,
3. // o código nunca será executado
4. while (variável <=valor final)
5. {
6. // Código a ser executado
7. }

Exemplo do loop while

```
1 <html>
2 <head>
3   <script type="text/javascript">
4     function myFunction()
5     {
6       var x="";
7       var i=1;
8       while(i<=5)
9       {
10        x += i + "<br/>";
11        i++;
12      }
13      document.write(x);
14    }
15  </script>
16 </head>
17
18 <body>
19   <p>Clique no botão para executar um loop while.</p>
20   <button onclick="myFunction()">Tente!</button>
21 </body>
22 </html>
```