

Java Script

JS

Carla Plantier Message – cpmessage@unoeste.edu.br

Tamiris Malacrida – tamiris@unoeste.edu.br

js... (história)

- Tecnologia desenvolvida pela Netscape em 1995
- Objetivos: incorporar conteúdo dinâmico em páginas HTML estáticas
- Move algum processamento lógico para o lado do cliente
- Linguagem de *script* de alto nível, baseada em objeto

js... (para que serve?)

- Foi projetada para aumentar interatividade das páginas web:
 - Validação de campos de formulários, interação com o usuário (p.ex., tratamento de cliques de botões), detecção de navegadores, conexão entre sistemas, etc
- Seu processamento pode ser feito tanto na máquina cliente (*browser*) quanto no servidor
- Não gera programa executável do tipo arquivo .exe

js... (como usar?)

A programação JavaScript deve vir dentro da tag script

```
<script type="text/javascript">
```

programação

```
</script>
```

js... (como usar?)

- Uma tag `<script>` pode ser definida numa seção **head**, numa seção **body** e também pode ser definida **externamente**:
 - Na seção **head**, os scripts são executados quando são chamados ou quando algum evento ocorre;
 - Na seção **body**, os scripts são executados na carga da página web
 - De forma **externa**, um arquivo com a extensão **".js"**

js... (exemplo)

□ Exibindo uma mensagem

A mensagem será executada na carga da página

```
...  
<body>  
  <script type="text/javascript">  
    alert("Olá Mundo");  
  
  </script>  
</body>  
...
```



js... (como usar?)

- A forma mais habitual é definir a codificação *javascript* em um arquivo externo (extensão **.js**) que será acionado no html.

Diretório onde o
arquivo foi salvo

```
<script type="text/javascript" src="scripts/testes.js" >
```

```
</script>
```

js... (primeiros exemplos)

- Método para imprimir um texto em uma página html

```
...  
<body>  
<script type="text/javascript">  
  
    document.write("Texto inserido com instruções JavaScript");  
  
</script>  
</body>
```


js... (primeiros exemplos)

O script abaixo lê o nome do usuário e dá boas-vindas.

...

```
<body>
```

```
<script type="text/javascript">
```

```
  /* Script de Boas-Vindas */
```

```
  nome = prompt ("Entre com seu nome: : , "Digite-o aqui: ");
```

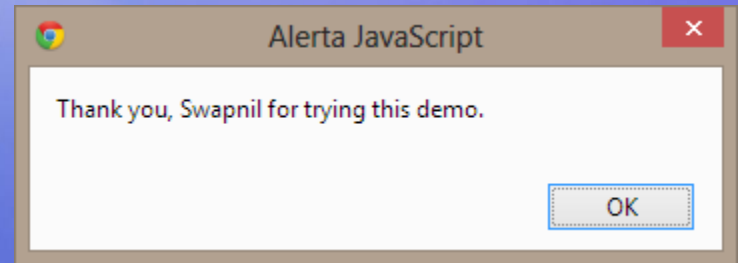
```
  document.write ("Oi " + nome );
```

```
</script>
```

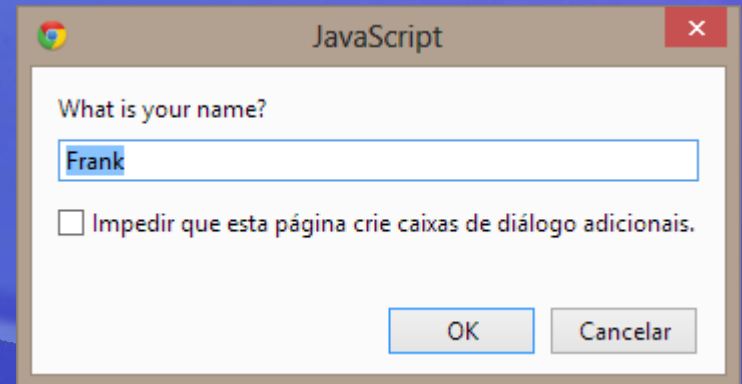
```
</body>
```

js... (janelas de interação)

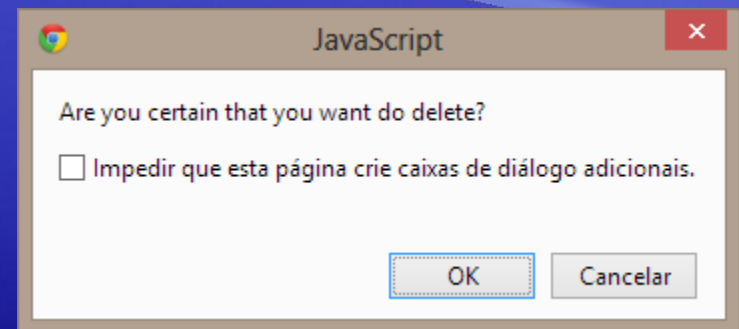
```
alert("Thank you, Swapnil for  
trying this demo.");
```



```
var name;  
name = prompt("What is your  
name?", "Frank");
```



```
var r = confirm("Are you certain that  
you want do delete?");  
if (r==true)  
    document.write("yes!!!");
```



js... (considerações iniciais)

- Javascript é case sensitive.
 - Certo: `document.write("Deu certo");`
 - Errado: `Document.Write("Deu errado");`
- Todo final de instrução deve ser finalizado com o caracter “.”.
`document.write("texto 1");`
- O código Javascript pode ser agrupado em blocos dentro da página (objetos-funções);
 - `{ ... }`: Delimitador de blocos de comando.
 - `//`: Comentário de linha.
 - `/* ... */`: Comentário de bloco.

js... (variáveis)

- Os nomes de variáveis são case sensitive (y e Y são variáveis diferentes).
- Não existe um sessão pré-definida para declaração das variáveis.
- Uso da palavra reservada **var** (não obrigatório, mas deve ser considerado para evitar problemas das variáveis globais).
- É uma linguagem que usa **tipagem dinâmica de dados**, ou seja, não se define inicialmente o tipo de dado.

```
var x;  
var curso;  
  
var x=5;  
var curso="Informática";  
  
x=5;  
curso="Informática";
```

js... (convertendo variáveis)

Em Javascript temos duas funções globais:

parseInt(): Converte uma string para um número inteiro.

parseFloat(): Converte uma string para um número real.

isNaN(): Valida se o valor recebido não é número

```
var v1="042";
```

```
var v2="42.33";
```

```
num1=parseInt(v1); //resultado = 42
```

```
num2=parseInt(v2); // resultado = 42
```

```
valor=parseFloat(v1); //resultado = 42
```

```
valor=parseFloat(v2); //resultado = 42.33
```

```
isNaN("oi"); // resultado = true (não é numérico)
```

js... (como usar?)

Usando valores numéricos

```
n1 = prompt("Digite o 1º valor" , "Digite aqui: ");
```

```
n2 = prompt("Digite o 2º valor" , "Digite aqui: ");
```

```
res = parseInt (n1) + parseInt (n2);
```

```
Document.Write ("Resultado = ", res);
```

js... (operadores)

▣ Aritméticos: igual a Java, C e C#

Operador	Função	Exemplo	Resultado
+	Adição	$x = y + 2$	$x = 7$
-	Subtração	$x = y - 2$	$x = 3$
*	Multiplicação	$x = y * 2$	$x = 10$
/	Divisão	$x = y / 2$	$x = 2.5$
%	Módulo (resto da divisão)	$x = y \% 2$	$x = 1$
++	Incremento	$x = ++y$	$x = 6$
--	Decremento	$x = --y$	$x = 4$

Operador	Exemplo	Mesmo que	Resultado
=	$x = y$		$x = 5$
+=	$x += y$	$x = x + y$	$x = 15$
-=	$x -= y$	$x = x - y$	$x = 5$
*=	$x *= y$	$x = x * y$	$x = 50$
/=	$x /= y$	$x = x / y$	$x = 2$
%=	$x \% = y$	$x = x \% y$	$x = 0$

js... (operadores)

▣ Relacionais: igual a Java, C e C#

Operador	Descrição	Mesmo que	Resultado
==	É igual	x==8	False
!=	Diferente	x!=8	True
>	Maior que	x>8	False
<	Menor que	x<8	True
>=	Maior ou igual que	x>=8	False
<=	Menor ou igual que	x<=8	True

js... (operadores)

- Lógicos: igual a Java, C e C#

Operador	Descrição	Exemplo
&&	And (e)	(x < 10 && y > 1) é true
	Or (ou)	(x==5 y==5) é false
!	Not (não)	!(x==y) é true

&&/and	/or	!/not
V && V = V	V V = V	!V = F
V && F = F	V F = V	!F = V
F && V = F	F V = V	
F && F = F	F F = F	

js... (estruturas de decisão)

Decisão Simples:

```
if <(condição)>{  
    /*Instruções para condição verdadeira*/  
}
```

Decisão Composta:

```
if <(condição)>{  
    /*Instruções para condição verdadeira*/  
}  
else{  
    /*Instruções para condição falsa*/  
}
```

Símbolo

Significado

||

Ou (OR)

&&

E (AND)

!

Não (NOT)

js... (estruturas de decisão)

```
var d = new Date();
var time = d.getHours();
if (time<=11)
{
    document.write("<b>Bom dia!</b>");
}
else if (time>=12 && time<18)
{
    document.write("<b>Boa tarde!</b>");
}
else
{
    document.write("<b>Boa noite!</b>");
}
```

js... (estruturas de decisão)

Switch:

```
switch (dia_semana)
{
    case 1: document.write("Janeiro");
        break;
    case 2: document.write("Fevereiro");
        break;
    ...
}
```

js... (estruturas de decisão)

```
var d=new Date();
dia=d.getDay();
switch (dia)
{
case 5:
    document.write("Finalmente sexta-feira!");
    break;
case 6:
    document.write("Super sábado!");
    break;
case 0:
    document.write("Domingo, descansar ");
    break;
default: //Não é obrigatório
    document.write("Cadê o final de semana?");
}
```

js... (estruturas de repetição)

for

```
for(condicaoInicial;condicaoFinal;incremento)

{

    executa bloco de código;

}
```

```
for (i=0; i<= 10; i++)
{
    document.write('Linha '+i);
}
```

js... (estruturas de repetição)

while

```
while (condicao) {  
    bloco de operação  
}
```

```
while (var1 <= 10)  
{  
    document.write('linha '+var1);  
    var1++;  
}
```

js... (estruturas de repetição)

do ... while

do

{

bloco de operacao

}

while (condicao);

Strings

Usado para manipular cadeias de caracteres, é imutável, ou seja, seus métodos não atualizam seu próprio conteúdo.

Declaração e uso:

```
palavra = new String("javascript");  
palavra = "javascript";
```

Strings

Propriedade:

length: retorna a quantidade de caracteres

```
palavra = "javascript";
```

```
palavra.length;    → 9
```

Strings (Métodos) http://www.w3schools.com/jsref/jsref_obj_string.asp

charAt(index)	Retorna o caracter de um determinado índice
concat(s1,s2,...sn)	Retorna a concatenação de strings
indexOf(value,start)	Retorna a posição de primeira ocorrência de uma substring
lastIndexOf(val,start)	Retorna a posição de última ocorrência de uma substring
replace(value,newvalue)	Realiza busca e troca na string. Retorna o valor modificado
split(separator,limit)	Divide a string em várias partes
<pre>data = "sexta-feira, 07 de Julho de 1777" dia = data.split(","); document.write(dia[0]); // sexta-feira document.write(dia[1]); // 07 de Julho de 1777</pre>	
substr(start,end)	Retorna uma parte da string
substring(from,to)	Retorna uma parte da string
toLowerCase()	Retorna o conteúdo da string em minúsculos
toUpperCase()	Retorna o conteúdo da string em minúsculos
trim()	Remove os espaços em branco do início e fim da string

Strings

Métodos de formatação:

link(): ` texto `

small(): `<small> texto </small>`

big(): `<big> texto </big>`

strike(): `<strike> texto </strike>`

sub(): `_{texto}`

sup(): `^{texto}`

italics(): `<i> texto </i>`

bold(): ` texto `

fontcolor(): ` texto `

fontsize(): ` texto `

Array

Usado para representar coleções de qualquer tipo na forma de vetores indexados.

```
elementos = new Array(4);
```

```
elementos = new Array("ford","gm","kia","bmw");
```

O tamanho de um array pode aumentar ou diminuir:

```
elementos.length++;
```

```
elementos.length--;
```

Quando insere, o valor do elemento adicionado é undefined

Quando remove, o valor do último elemento é perdido

Array

Métodos:

```
elementos = new Array("ford","gm","kia","bmw");
```

reverse()

```
elementos.reverse(); → bmw,kia,gm,ford
```

sort()

```
elementos.sort(); → bmw,ford,gm,kia
```

Array

Matrices:

```
matriz = new Array(new Array("a","b"),  
                    new Array("c","d"));
```

```
matriz = new Array(new Array(2),new Array(3));  
matriz[0][0]="a";  
matriz[0][1]="b";  
matriz[0][2]="c";  
matriz[1][0]="d";  
matriz[1][1]="e";  
matriz[1][2]="f";
```

```
matriz.length           → 2
```

```
matriz[0].length        → 3
```


Array

Exemplo de Matriz

```
<script>
```

```
var tam_linha = 3;
```

```
var quant_linha = 2;
```

```
m = new Array(new Array(quant_linha),  
               new Array(tam_linha));
```

```
for(i=0;i < quant_linha; i++)
```

```
  for(j=0;j < tam_linha ;j++)
```

```
    m[i][j] = 2*i + 3*j;
```

```
for(i=0;i < quant_linha; i++)
```

```
  for(j=0;j < tam_linha;j++)
```

```
    document.write("<b> Posição("+i+", "+j+"):</b>" +  
                   m[i][j]+"<br>");
```

```
</script>
```

Posição(0,0):0

Posição(0,1):3

Posição(0,2):6

Posição(1,0):2

Posição(1,1):5

Posição(1,2):8

Date

Usado para representar datas.

```
dataqualquer = new Date();
```

- Cria um objeto com a data atual.
- A data utilizada é a data do sistema do usuário

Construtores:

```
data = new Date(ano,mês,dia);
```

```
data = new Date(ano,mês,dia,hora,minuto,segundo);
```

Date

Métodos

- getDate()
- getDay()
- getHours()
- getMinutes()
- getMonth()
- getSeconds()
- getTime()
- getTimezoneOffset()
- getYear()
- setDate(dia)
- setDay(dia)
- setHours(hora)
- setMinutes(minuto)
- setMonth(mês)
- setSeconds(segundo)
- setTime(tempo)
- setYear(ano)

js... (eventos)

- São fatos que ocorrem durante a execução do sistema, a partir dos quais o programador pode definir ações a serem realizadas pelo programa.
- Segue exemplo...

js... (eventos)



```
<html>
<head>
  <title>Clicar no botão</title>
</head>
<body>
  <input type="submit"
        value="Clique aqui"
        onclick='alert("Ola");' />
</body>
</html>
```

Quando for dado um clique no botão é executado o código (em JavaScript), neste caso:



js... (eventos)

- Um evento é gerado como resultado de uma ação:
 - Um clique,
 - Um movimento do mouse,
 - Uma seleção de texto,
 - O abandono da página
 - etc.
- O tratamento de *eventos* é feito por partes de código associados a eventos específicos

js... (eventos)

onload - Ocorre na carga do documento. Ou seja, só ocorre no BODY do documento.

onunload - Ocorre na saída do documento. Também só ocorre no BODY.

onchange - Ocorre quando o objeto perde o foco e houve mudança de conteúdo.

Válido para os objetos Text, Select e Textarea.

js... (eventos)

onblur - Ocorre quando o objeto perde o foco, independente de ter havido mudança.

Válido para os objetos **Text, Select e Textarea**.

onfocus - Ocorre quando o objeto recebe o foco.

Válido para os objetos **Text, Select e Textarea**.

onclick - Ocorre quando o objeto recebe um Click do Mouse.

Válido para os objetos **Button, Checkbox, Radio, Link, Reset e Submit**.

onselect - Ocorre quando o objeto é selecionado.

Válido para os objetos **Text e Textarea**.

onsubmit- Ocorre quando um botão tipo Submit recebe um click do mouse.

Válido apenas para o **Form**.

js... (+ eventos)

Eventos de mouse

Gerados quando o usuário:

onmouseover

Move o ponteiro do mouse para dentro da área de um elemento.

onmouseout

Move o ponteiro do mouse para fora da área de um elemento.

onmousedown

Pressiona qualquer um dos botões do mouse.

onmouseup

Libera qualquer um dos botões do mouse.

onmousemove

Move o mouse dentro da área de um elemento.

ondblclick

Efetua um duplo clique com o botão esquerdo do mouse sobre um elemento.

js... (+ eventos)

Eventos de teclado

Gerados quando o usuário:

onkeypress

Pressiona e solta uma tecla (o ciclo completo).

onkeydown

Pressiona uma tecla (ainda com a tecla abaixada).

onkeyup

Solta uma tecla.

js... (exercícios)

- 1) Crie uma página HTML que tenha a funcionalidade de uma calculadora, para isso você terá que requisitar ao usuário dois números e um carácter de operação (+, -, *, /) utilizando a função `prompt`, após isso, exiba no corpo do HTML o resultado da operação entre os dois números digitados utilizando a função `document.write`.
- 1) Faça uma página HTML que requirite ao usuário um número utilizando a função `prompt` e retorne ao usuário uma mensagem (`alert` ou `document.write`) indicando se o número que ele digitou é ímpar ou par.