



```
addressLines (FreeFormText)>  
GlobalCountryCode (#PCDATA)>  
NationalPostalCode (#PCDATA)>  
RegionName (FreeFormText)>  
Document (FreeFormText)>  
ListRequirements (FreeFormText)>  
ProductLineItem (shipFrom+, ProductQuan  
GlobalLocationIdentifier)>  
Quantity (#PCDATA)>  
Order (#PCDATA)>  
ProductPackageDescription  
PackageDescription (ProductIdent  
Identification (GlobalProductIde  
ProductIdentifier (#PCDATA)>  
ProductIdentification (GlobalPar  
ClassificationCode (#PCDAT  
ProductIdentifier (#PCDATA)>  
FillmentRequestCode (#P  
GlobalCountryCode)>  
DateStamp)>
```

Javascript

Javascript

- Script, descrição de instruções passo a passo para serem executadas
- O javascript desenvolvido pela Netscape Corporation
- A linguagem de script mais utilizada na Web
- Pode ser incorporada directamente em documentos HTML

Scripts

- Para adicionar javascript a uma página utiliza-se o tag SCRIPT
- Ex.

```
<HTML>
  <HEAD>
    <TITLE>Documento simples de HTML</TITLE>
  </HEAD>
  <H1>Seja bem vindo a minha página!</H1>
    <SCRIPT LANGUAGE="JavaScript">
      document.write("Olá Browser!");
    </SCRIPT>
  </BODY>
</HTML>
```

Localização dos Scripts

- No corpo do documento
- No cabeçalho da página
- Dentro de uma tag HTML, o Event Handler
- Num ficheiro com a extensão js que contenha os scripts

Operadores Lógicos

- Igual ==
- Diferente !=
- Maior >
- Menor <
- Maior ou igual >=
- Menor ou igual <=
- E &&
- OU ||

Operadores Matemáticos

- Adição e concatenação de strings +
- Subtracção de valores -
- Multiplicação de valores *
- Divisão de valores /
- Resto de uma divisão %

Controles especiais

- \b backspace
- \f form feed
- \n new line
- \r carriage return
- \t tab
- // comentário

Comandos condicionais

```
if (condicao)  
    {Acção}
```

```
else  
    {Acção}
```

```
for (inicialização;condição;incremento)  
    {acção}
```

```
while (condição)  
    {acção}
```


Eventos

- onload, qd o documento é carregado
- onunload, qd o documento é descarregado
- onchange, qd o objecto perde o focus e houve mudança de conteúdo, para Text, Select e Textarea
- onblur, qd o objecto perde o focus, para Text, Select e Textarea
- onfocus, qd o objecto recebe o focus , para Text, Select e Textarea
- onclick, qd o objecto recebe um click, para button, checkbox, radio, link, reset e submit
- onmouseover, qd o rato passa por um objecto, para link
- onselect, qd um objecto é seleccionado, para textarea e text
- onsubmit, qd um botão submit recebe o click

Variáveis

- Criada com a atribuição de valores
- Variáveis locais, dentro de uma função
- Variáveis globais, criadas fora de uma função

Mensagens

- alert(mensagem)
- confirm (mensagem)
- receptor= prompt("Mensagem","Texto")

Exemplo

```
if (confirm ("Continuar??")){  
    alert("Continuar")  
}  
else {  
    alert("Abortar")  
}
```

Funções

- Uma função é um conjunto de instruções, que só são executadas quando a função for activada. Ex:
 - function NomeFunção (Parâmetros)
 - function Idade (Anos) {
 - if (Anos > 17)
 - { alert ("Maior de Idade") }
 - else
 - { alert ("menor de Idade") }
 - }
 - Para activar a função, uma FORM:
 - <form>
 - <input type=text size=2 maxlength=2 name="Tempo"
 - onchange="Idade(Tempo.value)">
 - </form>

Funções

- Return, especifica os argumentos retornados pela função

Funções Pré-definidas

- Ver a [especificação](#)
- Categorias:
 - eval
 - isFinite
 - isNaN
 - parseInt and parseFloat
 - Number and String
 - escape and unescape

Exemplos

- Calcular um valor de uma string
 - `eval : Result = eval (" (10 * 20) + 2 - 8"`)
- `isNaN` (is not a number)
 - `floatValue=parseFloat(toFloat)`
 - `if(isNaN(floatValue)) {`
 - `notFloat()`
 - `}`
 - `else {`
 - `isFloat()`
 - `}`

Objectos

- Aceder à propriedade de um objecto com *objectName.propertyName*
- Não esquecer, javascript é case sensitive
- Atribuir valores às propriedades (o objecto carro já existe):
 - *Carro.fabricante="Renault"*
Carro.modelo="Clio"
Carro.ano = 1995;
- Criar o objecto tipo:

```
function carro(fabricante, modelo, ano) {  
    this.fabricante = fabricante  
    this.modelo = modelo  
    this.ano = ano  
}
```
- Criar objectos
 - *meucarro = new carro("Citroen", "Saxo", 2000)*

Arrays

- Array é um conjunto de valores que podem ser acedidos por um nome ou valor.
- Para criar um array
 - `arrayObjectName = new Array(element0, element1, ..., elementN)`
 - `arrayObjectName = new Array(arrayLength)`
- Adicionar valores
 - `pessoa[1]="João"`
`pessoa[2]="Maria"`
`pessoa[3]="Xico"`
 - ou
 - `meuArray = new Array("olá", umaVar, 3.14159)`

Array's

- Aceder aos seus elementos através da sua posição ordinal
- ex. `array[0]` → 1º elemento
- Ver a spec para alguns métodos suportados no array:
 - `sort`, `push`, `pop`, ...

Outros objectos

- Boolean
- Date
- Function
- Math
- Number
- String

Object Model

