Javascript

JavaScript

- *Client-side scripting language*
  - Interpretado directamente pelo *browser (on the fly),* não necessita de ser compilado

JavaScript

- Sintaxe
  - *É case sensitive*.
  - // símbolo do comentário
    - /* comentário para múltiplas linhas */
  - Um *script* é composto por um conjunto de *statements* (declarações) que indicam ao browser a ação a realizar

```
<h1> Data Gerada por um Script</h1>
<script>
       var data = new Date();

       document.write(data.getDate());
       document.write(" / ");
       document.write(data.getMonth()+1);
       document.write(" / ");
       document.write(data.getFullYear());
 </script>
```


JavaScript

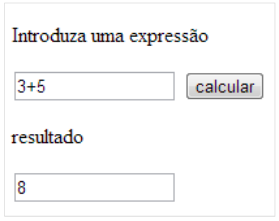
- Declaração de variáveis
  - *loosely typed*, não é necessário definir o tipo de variável uma vez que este é automaticamente assumido de acordo com a declaração efectuada
  - *Keyword:* ***var***
    - Nomes devem começar por uma letra ou por underscore "_"
    - Não devem conter espaços nem caracteres especiais (! . , / \ + * = …)

```
var x=10;  // numeric
var x="ten"; // string
var x;
```

JavaScript

- Inserção Directa *(embedded script)*
  - ***<script> … </script>***
    - o *script* executado como ***resposta a um evento***, e.g. *onclick*

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
    <script type="text/javascript">
        function calculo(formulario)
        {
        if (confirm ("Confirma operação?"))
        formulario.result.value=eval(formulario.expr.value)
        else
        alert ("introduza novos dados")}
    </script>
<title> Javascript - Resposta a um Evento </title>
</head>
<body>
<form>
    <p>Introduza uma expressão</p>
    <input type="text"  id="expr" size="15" >
    <input type="button" value="calcular"  onclick="calculo(this.form)" >
    <p>resultado</p>
    <input type="text"  id="result" size="15">
</form>
</body>
</html>
```



JavaScript

- Evento
  - Acções que podem ser detectadas pelo *JavaScript* e que provocam uma execução específica:
    - Chamada de uma função.
    - A função só é executada após a ocorrência do respectivo evento.
  - Exemplos:
    - Click do rato
    - Carregamento de uma página ou imagem
    - *Mouse over*
    - Submissão de um formulário
    - Selecção de um elemento de um formulário…

JavaScript

- Eventos
  - É necessário um identificador (*event handler*) que inicia a execução do script quando esse evento ocorre.
    - Utilizados neste trabalho:

Mouse Events

Property	Description
onclick	The event occurs when the user clicks on an element
ondblclick	The event occurs when the user double-clicks on an element
onmousedown	The event occurs when a user presses a mouse button over an element
onmousemove	The event occurs when the pointer is moving while it is over an element
onmouseover	The event occurs when the pointer is moved onto an element
onmouseout	The event occurs when a user moves the mouse pointer out of an element
onmouseup	The event occurs when a user releases a mouse button over an element

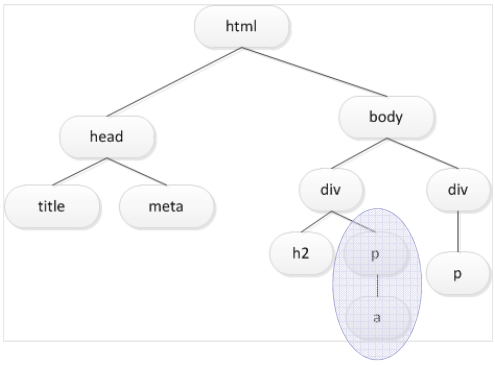
DOM

- Document Object Model (HTML DOM)
  - W3C standard
  - HTML DOM disponibiliza um conjunto de métodos/propriedades (API) para permitir o acesso de uma linguagem de programação (**Javascript**, …) a elementos HTML
    - Aceder a elementos HTML pelo seu nome ou atributos e efetuar várias operações:
      - adicionar,
      - modificar,
      - apagar elementos e respetivo conteúdo

DOM

- DOM nodes
  - Cada elemento é um nó
  - O conteúdo **também é considerado** um nó

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title> Titulo </title>
</head>
<body >
    <div>
        <h2> Subtítulo h2 </h2>
        <p> Primeiro parágrafo que contém
        um <a href="http://www.isec.pt"> link </a>externo</p>
    </div>
    <div>
        <p> Segundo parágrafo </p>
    </div>
</body>
</html>
```



DOM

- *DOM Methods*

Método (*DOM Queries*)	Descrição
document.getElementById()	retorna o elemento que tem o id com o valor especificado
document.getElementsByClassName()	retorna a lista de nós que tem o atributo class com o nome especificado
document.getElementsByName()	retorna a lista de nós cujo atributo name corresponde ao nome especificado
document.getElementsByTagName()	retorna a lista de nós cuja designação da tag corresponde ao nome especificado
…	…

DOM

- DOM Properties

Propriedades	Descrição
innerHTML	acesso/alteração do conteúdo HTML de um elemento
nodeValue	Valor de um nó (se for um nó de texto, retorna o seu valor)
textContent	Acesso/alteração de texto
className	Retorna/altera o valor do atributo *class* de um elemento
…	…

DOM

- *Acesso aos DOM nodes*
  - *getElementById()*

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
</head>
<body >
<p id="Id1"> Exemplo da Aplicação _do getElementById</p>
<script  type="text/javascript">
    var txt=document.getElementById("Id1").innerHTML;
    document.write(txt);
</script>
</body>
</html>
```

Exemplo da Aplicação do getElementById

Exemplo da Aplicação do getElementById

DOM

- *Acesso aos DOM nodes*
  - *getElementsByTagName()*
    - retorna todos os elementos com o nome especificado numa *nodeList*

```
<body >
<p id="Id1"> Exemplo da Aplicação do getElementsByTagName</p>
<script  type="text/javascript">
    var txt=document.getElementsByTagName("p")[0].innerHTML;
    document.write(txt);
</script>
</body>
```

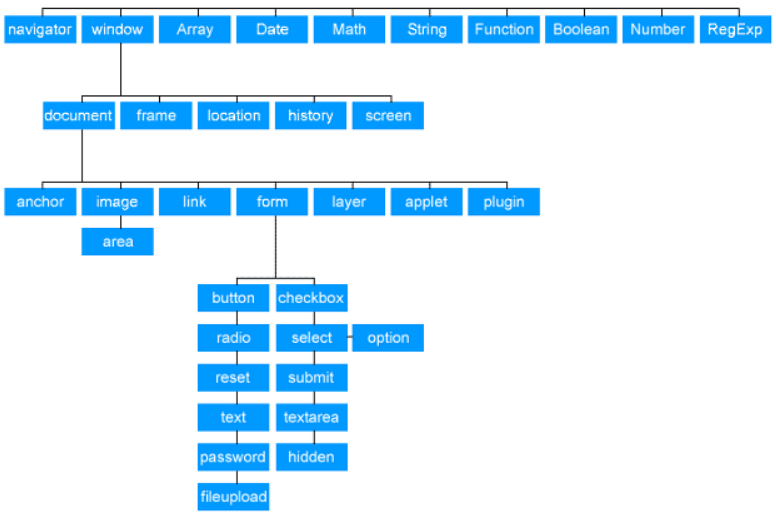
Exemplo da Aplicação do getElementsByTagName

Exemplo da Aplicação do getElementsByTagName

Importante: *Os elementos tem que ser indexados, mesmo que apenas exista uma ocorrência do TagName (posição [0])*

DOM

- Acesso aos DOM nodes
  - Hierarquia de Objetos



- Aceder a um formulário:

document.valueAttributeNameIdForm

- Aceder a um campo do formulário:

document.valueAttributeNameIdForm.valueAttributeNameIdElementHTML

DOM

- Alteração de conteúdo dos *DOM nodes*
  - *innerHTML*
    - permite alterar/definir o conteúdo (texto/markup) de um dado elemento

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <style>
        p {color:red;}
    </style>
</head>
<body >
<p id="Id1" class="class1" onClick="altera()"> Exemplo da Aplicação</p>

<script  type="text/javascript">
function altera()
{
    var txt=document.getElementById("Id1");
    txt.innerHTML="<h1>Tecnologias Web</h1>";
}
</script>

</body>
</html>
```

DOM

- Alteração de atributos.
  - *setAttribute()*
    - método que permite múltiplas alterações: alterar uma imagem (src); um destino (href)

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <style>
        p {color:orange;}
    </style>
</head>
<body >
<p id="Id1" class="class1" onClick="altera()"> Exemplo da Aplicação </p>

<script  type="text/javascript">
function altera()
{
    var txt=document.getElementById("Id1");
    txt.setAttribute('style','color:red;');
}
</script>

</body>
</html>
```