

09

Introdução às Tecnologias Web

HTML Forms + JavaScript



Perguntas

```
var u = location.href  
open (u);
```

**A partir de um nó do DOM posso saber quem é
o seu pai? Como?**

O que é um *Event Handler*?

Qual o significado?

```
3 var container;
4
5 function setOrange () {
6   var pars = container.getElementsByTagName("p");
7   var i;
8   for (i = 0; i < pars.length; i++) {
9     if (pars[i].innerHTML == "Laranja") {
10      pars[i].style.color = "orange";
11      break;
12    }
13  }
14 }
15
16 function addMaca () {
17   var newP = document.createElement("p");
18   var newText = document.createTextNode ("Maçã");
19   var pClass = document.createAttribute ("class");
20   pClass.value = "fruta";
21   newP.appendChild (newText);
22   newP.setAttributeNode (pClass);
23   newP.addEventListener ("click", myFunc);
24
25   var pos = document.getElementById("pera");
26   container.insertBefore (newP, pos);
27 }
28
29 function myFunc () {
30   alert("Clicou na Maçã!");
31 }
32
33 function mainFunction () {
34   container = document.getElementById("cc");
35   document.getElementById ("pera").addEventListener ("click", addMaca);
36
37   setOrange ();
38 }
39
40 window.onload = mainFunction;
```

```
10 <body>
11 <div id="cc">
12   <p class='fruta'>Banana</p>
13   <p class='fruta'>Laranja</p>
14   <p class='fruta' id='pera'>Pera</p>
15   <p class='fruta'>Melão</p>
16 </div>
17 </body>
```

Qual o significado?

```
3 var container;
4
5 function setOrange () {
6   var pars = container.getElementsByTagName("p");
7   var i;
8   for (i = 0; i < pars.length; i++) {
9     if (pars[i].innerHTML == "Laranja") {
10      pars[i].style.color = "orange";
11      break;
12    }
13  }
14 }
15
16 function addMaca () {
17   var newP = document.createElement("p");
18   var newText = document.createTextNode ("Maçã");
19   var pClass = document.createAttribute ("class");
20   pClass.value = "fruta";
21   newP.appendChild (newText);
22   newP.setAttributeNode (pClass);
23   newP.addEventListener ("click", myFunc);
24
25   var pos = document.getElementById("pera");
26   container.insertBefore (newP, pos);
27 }
28
29 function myFunc () {
30   alert("Clicou na Maçã!");
31 }
32
33 function mainFunction () {
34   container = document.getElementById("cc");
35   document.getElementById ("pera").addEventListener ("click", addMaca);
36
37   setOrange ();
38 }
39
40 window.onload = mainFunction;
```

```
10 <body>
11 <div id="cc">
12   <p class='fruta'>Banana</p>
13   <p class='fruta'>Laranja</p>
14   <p class='fruta' id='pera'>Pera</p>
15   <p class='fruta'>Melão</p>
16 </div>
17 </body>
```

Resumo Aula Anterior

Objeto Window

Janela do browser = objeto global

Oferece vários métodos e propriedades

DOM

Estrutura lógica da página

Encontrar + Alterar + Criar + Remover elementos HTML

Eventos

Gerados quando algo acontece

Event Handler – Código executado quando acontece evento

Um elemento pode ter vários *Event Handlers*

Sumário

Formulários em HTML

Formulários e JavaScript

Web Storage

Mais eventos

01

FORMS EM HTML

Forms

Servem para recolher dados dos utilizadores

Usadas para pesquisas, registos, etc.

Contêm elementos de vários tipos

Campos de texto

Checkboxes

Radio Buttons

etc.

http://www.w3schools.com/html/html_forms.asp

Estrutura da Form (simples)

<form>

Elementos da form

</form>

Elementos da form devem ter nome

Só assim são enviados para o servidor

Campos de Texto

Linha de Texto

```
<input type="text" name="username"
size="15" maxlength="30" value="Manuel">
```

Password

```
<input type="password" name="password"
maxlength="30" >
```

Área de Texto

```
<textarea name="comments" cols="20"
rows="4">Comentários aqui ...</textarea>
```

Campos de Escolha

Qual o seu género favorito?

☒ Rock ☐ Pop ☐ Jazz ☐ Classica

Radio Buttons

```
<input type="radio" name="genero" value="rock" checked> Rock
```

```
<input type="radio" name="genero" value="pop"> Pop
```

```
<input type="radio" name="genero" value="jazz"> Jazz
```

Campos de Escolha

Quais os seus géneros favoritos?

☒ Rock ☐ Pop ☐ Jazz ☐ Classica

Checkbox

```
<input type="checkbox" name="genero" value="rock" checked> Rock
```

```
<input type="checkbox" name="genero" value="pop"> Pop
```

```
<input type="checkbox" name="genero" value="jazz"> Jazz
```

Campos de Escolha

Drop down list

```
<select name="generos">
  <option value="rock" selected> Rock
</option>
  <option value="pop">Pop</option>
  <option value="jazz">Jazz</option>
</select>
```

Qual o seu género favorito?

Rock ▾

Campos de Escolha

Multiple Select box

```
<select name="generos" multiple>
  <option value="rock" selected> Rock
</option>
  <option value="pop">Pop</option>
  <option value="jazz">Jazz</option>
</select>
```

No CSS definir o width e height

Neste caso width:150px; height:80px

Qual o seu género favorito?

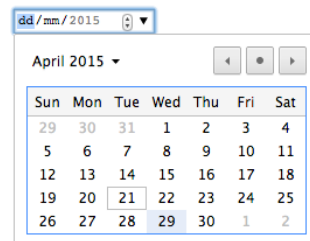
Rock
Pop
Jazz
Classica

Campos de Datas

Date

```
<input type="date" name="dnasc">
```

```
<input type="date" name="dnasc"  
  min="2015-01-01" max="2015-12-31">
```



Outros Campos

Cores

Email

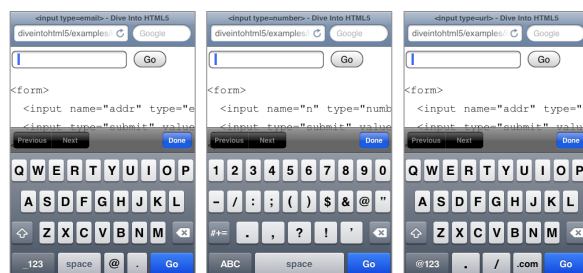
Números

Intervalos de valores

Telefones

URLs

etc.



O Botão Submit

Envia a form para o servidor

Necessário indicar qual o *form-handler*

Script no servidor que vai processar os dados

```
<form action="http://www.exemplo.com/
exec_form.php">
  <input type="text" name="firstname">
  <input type="submit" value="Enviar">
</form>
```

Como funcionam as Forms?

Browser

Please fill out this form.

Title*
Mr

First name*

Last name*

Email*

Contact us

Pares nome-valor
de cada elemento



Nova página criada
pelo servidor



Servidor



Servidor processa
informação recebida

Método para enviar os dados

```
<form action="http://www.exemplo.com/
exec_form.php" method="GET">
```

ou

```
<form action="http://www.exemplo.com/
exec_form.php" method="POST">
```

GET

Método por omissão

Usar quando não há informação sensível

Dados da form **aparecem** na URL

POST

Usar quando há informação sensível (ex. passwords)

Dados da form **não aparecem** na URL

Fazer upload de um ficheiro

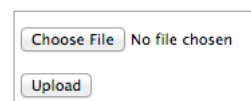
File input box

```
<form action="http://www.exemplo.com/
upload.php" method="POST">
```

```
<input type="file" name="musica"> <br>
```

```
<input type="submit" value="Upload">
```

```
</form>
```



Choose File No file chosen

Upload

Campos Acessíveis (Labels)

Usados para tornar os formulários acessíveis

Podem ser colocados

À volta do texto

Usando o id do campo

```
<label>Nome: <input type="text" name="n">
</label>
```

ou

```
<label for="nome">Nome: </label>
<input id="nome" type="text" name="n">
```

Regras para posicionar o Label

À esquerda ou em cima

Campos de texto, Áreas de texto

Select boxes, File uploads

Datas, Cores

etc.

À direita

Checkboxes

Radio buttons

Agrupamento de campos da Form

Permite agrupar elementos relacionados

```
<fieldset>
  <legend>Contactos</legend>
  <label>Email:<br>
  <input type="text" name="email">
</label> <br>
  ...
  <label>Telefone:<br>
  <input type="text" name="phone">
</label>
</fieldset>
```



Validação da Form

HTML5 introduziu validação de Forms

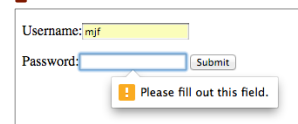
Antigamente era feita com JavaScript

Validação ajuda a reduzir a carga no servidor

Validação feita pelo browser

```
<label>Username:<input type="text"
  name="username" required>
</label> <br>

<label>Password:<input type="password"
  name="password" required>
</label>
```



Mais novidades do HTML5

Autocomplete

Autofocus

Placeholder

Required

etc.

<http://diveintohtml5.info/forms.html>

02

ACEDER AOS CAMPOS DA FORM COM JS

Exemplo de Form

```
<form name="morada">

  <label>Nome: <input type="text" name="nome">
</label><br><br>

  <label>Rua: <input type="text" name="rua">
</label><br><br>

  <label>Código Postal: <input type="text"
name="cp"></label><br><br>

  <button onclick="myFunc2()">Aceder Form</button>

</form>
```



Nome:

Rua:

Código Postal:

Aceder à Form e seus Campos

Podem ser acedidos usando métodos standard

ex. `getElementById()`, `getElementsByTagName()`

Usando outros métodos

```
document.morada    // form com name="morada"
document.forms.morada
```

document.forms

Devolve array de forms da página

Aceder à Form e seus Campos

document.forms.morada.elements

Lista de elementos da form “morada”

document.forms.morada.elements[0]

Primeiro elemento da form “morada”

document.forms.morada.elements.nome

document.forms.morada.elements[“nome”]

Elemento da form “morada” com name “nome”

Aceder valor elemento da Form

```
<form name="morada">
  <label>Nome: <input type="text" name="nome">
</label> <br><br>
  <label>Rua: <input type="text" name="rua">
</label> <br><br>
  <label>Código Postal: <input type="text" name="cp">
</label> <br><br>
  <button onclick="myFunc2()">Aceder Form</button>
</form>
```

```
function myFunc2 () {
  var myText;
  var ff = document.forms.morada;
  myText = ff.elements.nome.value;
  alert(myText);
}
```

Aceder valor elemento da Form

```
<form name="estilos">
  <p>Qual o seu género favorito?</p>
  <input type="radio" name="genero" value="rock"> Rock
  <input type="radio" name="genero" value="pop"> Pop
  <input type="radio" name="genero" value="jazz"> Jazz
  <input type="radio" name="genero" value="classica">
    Classica
  <button onclick="myFunc3()">Aceder Form</button>
</form>
```

```
function myFunc3 () {
  var selEstilo=document.forms.estilos.elements.genero;
  // Array com os vários géneros
  var estilo;
  for(var i = 0; i < selEstilo.length; i++) {
    if (selEstilo[i].checked) {
      estilo = selEstilo[i].value; break;}
  }
  alert(estilo);
}
```

Métodos da Form

submit()

Envia a Form para o servidor

reset()

Limpa todos os campos da Form

Exemplo

```
document.forms.estilos.reset();
```


Eventos da Form e seus Elementos

onchange

onfocus

oninput

oninvalid

onsearch

onselect

onreset

onsubmit

onkeydown

onkeypress

onkeyup

onclick

onmouseover

etc.

03

WEB STORAGE

(ARMAZENAMENTO LOCAL)

Armazenamento local no HTML

Cookies

Incluem pedidos ao servidor (lentidão)

Informação é enviada para o servidor

Têm um limite de 4KB

Armazenamento Local

A informação nunca é enviada para o servidor

Têm um limite de 5MB

O armazenamento é por domínio

Armazenamento local melhor que cookies

Armazenamento local no HTML

Dois objectos para Web Storage

localStorage – Válido entre sessões

sessionStorage – Válido apenas numa sessão (janela)

Guardam pares chave-valor

Atenção que valores são guardados como strings

Usar Number(), parseInt() e parseFloat() ao recuperar um número

Comportam-se como objetos normais

Têm propriedades e métodos

Métodos para guardar e usar dados

setItem ("var1", valor)

Guarda um valor com a chave "var1"

getItem ("var1")

Devolve o valor com chave "var1"

key (i)

Devolve a chave do par i

removeItem ("var1")

Remove o item "var1"

clear ()

Remove todos os itens

Exemplo: Guardar e usar dados

```
function myFunc4 () {  
    localStorage.setItem("x", 1);  
    localStorage.setItem("y", 3);  
    for(var i = 0; i < localStorage.length; i++) {  
        var chave = localStorage.key(i);  
        var value = localStorage.getItem(chave);  
        console.log(chave + " : " + value);  
    }  
    localStorage.removeItem("x");  
    localStorage.clear();  
}
```

04

MAIS SOBRE EVENTOS

Objecto event

Podemos obter informação do evento

Elemento onde o evento foi gerado

Tecla carregada

Botão do rato

etc.

Usar propriedades do objeto event

target – elemento HTML que gerou o evento

type – nome do evento

timeStamp – tempo em que o evento foi gerado

Exemplo: Apanhar uma tecla

```
mBox.addEventListener("keyPress",function ()
{ myFunc(event);});

function myFunc (event) {
    var carCode = event.which || event.charCode ||
event.keyCode;
    var car = String.fromCharCode(carCode);
    alert ("Letra : " + car);
}
```

which, charCode, keyCode

Devolvem o código unicode do caracter produzido

http://www.w3schools.com/jsref/event_key_keycode.asp

Exemplo: Apanhar um botão do rato

```
mBox.addEventListener("mousedown", function ()
{ myFunc(event);});
function myFunc (event) {
    var mClick = event.which || event.button;
    alert ("Botão : " + mClick);
    if (event.altKey) alert ("Botão + tecla Alt");
}
```

W3C

Esquerdo – 0

Meio -1

Direito - 2

Chrome, Safari

Esquerdo – 1

Meio - 2

Direito - 3

Exemplo: Passar o elemento HTML

```
myH1.addEventListener("click",  
    function () { myFunc(myH1,"red");});  
  
mBox.addEventListener("keypress",  
    function () { myFunc(mBox,"blue");});  
  
function myFunc (elem, color) {  
    elem.style.color = color;  
}
```

Permite chamar a mesma função de vários elementos HTML

05

RESUMINDO

Resumo

Formulários em HTML

Vários tipos de elementos
Texto, escolhas, datas, etc.

Formulários e JavaScript

Aceder aos elementos e valores
Métodos e eventos das Forms

Web Storage

SessionStorage e LocalStorage

Mais eventos

Usar o evento gerado

Próxima Aula

