



CFGS DAW

Mòdul 6: Desenvolupament web en entorn client

INS Joan d'Àustria

HTML y Javascript



- Por el momento, solo sabemos escribir código javascript que se ejecuta en el momento que el navegador procesa el código
- Funciones y eventos forman parte de otra UF, pero vamos a introducir un evento simple. Este evento nos va a permitir ejecutar código javascript a petición del usuario

HTML y Javascript



- La mayoría de los elementos html pueden ser clicados. El elemento por excelencia para este comportamiento es el botón:
`<input type="button" value="Hola">`

Podemos añadir a los elementos html funcionalidad, de forma que, cuando cliquemos sobre ellos se ejecute javascript

HTML y Javascript



- Para esto utilizaremos un evento llamado **onclick**

```
<body>  
<input type="button" value="Hola" onclick="javascript:alert('Hola  
mundo');">  
</body>
```

- Aunque no es la forma más recomendada, podemos escribir más de una instrucción de código javascript

```
<body>  
<input type="button" value="Hola" onclick="javascript:alert('Hola  
mundo');var x=3; alert (x);">  
</body>
```

HTML y Javascript



- Mezclado no agitado

```
<body>
<script>
  var x;
  document.write("<form name='form1'>");
  document.write("<b>Selecciona un color para el fondo de página:</b><br>");
  document.write("<select name='combo1'>");
  document.write("<option value='o1' onclick='x=\"red\";'>Rojo</option>");
  document.write("<option value='o2' onclick='x=\"yellow\";'>Amarillo</option>");
  document.write("<option value='o3' onclick='x=\"green\";'>Verde</option>");
  document.write("<option value='o4' onclick='x=\"white\";'>Blanco</option>");
  document.write("</select>");
  document.write("<input type='button' value='Modifica'
onclick='document.bgColor=x'>");
  document.write("</form>");
</script>
</body>
```



Trabajo con ventanas



- El objeto window presenta el método open:
 - `window.open(url,target,opciones)`
- Hay que tener presente que estas ventanas se tratan como popups y que algunos navegadores las pueden tener deshabilitadas o bloqueadas

Trabajo con ventanas



- Abrir ventanas: atributo target

Valores
_blank (valor por defecto)
_parent
_self
_top
Nombre

Trabajo con ventanas



- Abrir ventanas: atributo opciones
 - Listado separado por comas con todas las opciones que queramos de apariencia de la nueva ventana

Valores		
height=pixels (≥ 100)	resizable=yes no 1 0	toolbar=yes no 1 0
left=pixels (≥ 0)	scrollbars=yes no 1 0	top=pixels (≥ 0)
location=yes no 1 0	status=yes no 1 0	width=pixels (≥ 100)
menubar=yes no 1 0	titlebar=yes no 1 0	

*En negrita aparecen los valores por defecto si no se indica lo contrario. Con letra de color negro algunas condiciones que tienen que cumplir los valores

Trabajo con ventanas



```
window.open("http://www.insjoandaustria.org/");
```

```
window.open("http://www.insjoandaustria.org/","jdaWindow");
```

```
window.open("http://www.insjoandaustria.org/","jdaWindow",  
"height=400,width=400,top=10,left=10,resizable=yes");
```

Trabajo con ventanas



- El método `open` crea un nuevo objeto `Window` que podemos recoger en una variable para trabajar con el:

```
var miventana=window.open("about:blank","jdaWindow",  
"height=400,width=400,top=10,left=10,resizable=yes");
```

```
miventana.resizeTo(500, 500);  
miventana.moveTo(100, 100);
```

- Una vez cerrada, la referencia del objeto todavía existe, pero sólo podremos consultar la propiedad `closed`

```
miventana.close();  
alert(miventana.closed); //true
```

Comunicación entre ventanas



```
<input type="button" value="prueba"
onclick='var myWindow1=window.open("about:blank", "ventana",
"width=300, height=200");
myWindow1.document.write("<html>");
myWindow1.document.write("<head>");
myWindow1.document.write("<title>Ventana nueva</title>");
myWindow1.document.write("</head>");
myWindow1.document.write("<body>");
myWindow1.document.writeln("Se usan las propiedades: ");
myWindow1.document.write("<li>height=200</li> <li>width=300</li>");
myWindow1.document.write("</body>");
myWindow1.document.write("</html>");'>
```

```
var miventana = window.open("http://insjoandaustria.org", "_blank");
if (miventana == null){
    alert("Debes habilitar las ventanas emergentes");
}
```





Frames

- HTML permite dividir una ventana en varias partes “independientes” mediante el uso de Frames
- Los frames se representan en HTML mediante las etiquetas frameset y frame

Frames



- Web de la API de Java:
(<http://docs.oracle.com/javase/7/docs/api>)

The screenshot shows the Java Platform Standard Edition 7 API Specification website. The browser address bar shows the URL <http://docs.oracle.com/javase/7/docs/api/>. The page has a navigation bar with tabs: Overview, Package, Class, Use, Tree, Deprecated, Index, and Help. Below the navigation bar, there are links for 'Prev', 'Next', 'Frames', and 'No Frames'. The main content area is titled 'Java™ Platform, Standard Edition 7 API Specification'. Below the title, there is a paragraph stating 'This document is the API specification for the Java™ Platform, Standard Edition.' and a link 'See: Description'. On the left side, there is a sidebar with a list of 'All Classes' and a list of 'Packages'. The 'Packages' list is expanded, showing a table with two columns: 'Package' and 'Description'. The table lists various Java packages and their descriptions, including java.applet, java.awt, java.awt.color, java.awt.datatransfer, java.awt.dnd, java.awt.event, java.awt.font, java.awt.geom, java.awt.im, java.awt.im.spi, java.awt.image, java.awt.image.renderable, java.awt.print, java.beans, java.beans.beancontext, and java.io.

Frames



- Elementos Frameset y Frame

Frameset
rows
cols

Frame
frameborder
marginheight
marginwidth
Name
noresize
scrolling
src



Frames

- Elementos Frameset y Frame

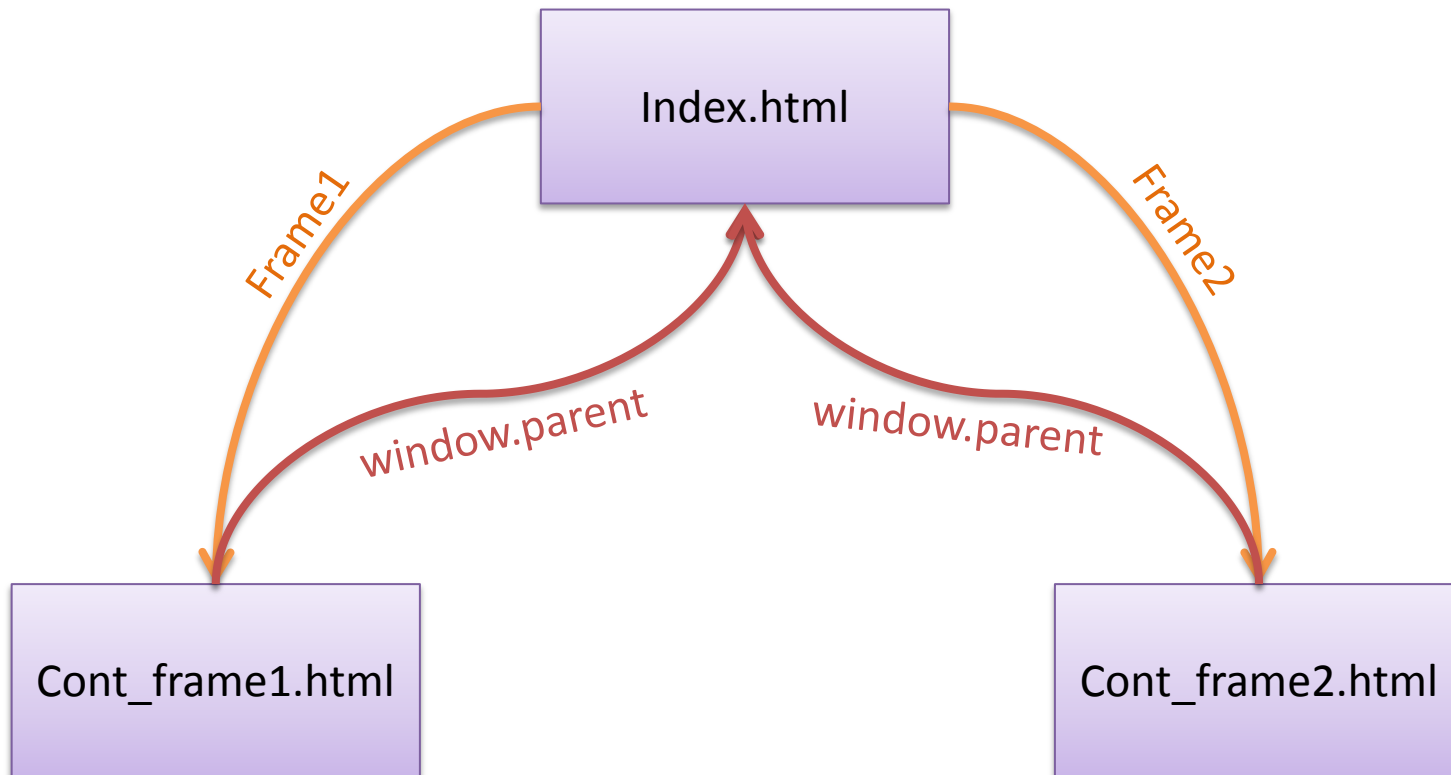
```
<HTML>
<HEAD>
<TITLE>A simple frameset document</TITLE>
</HEAD>
<FRAMESET cols="20%, 80%">
    <FRAME src="cont_frame1.html" name="Frame1">
    <FRAME src="cont_frame2.html" name="Frame2">
</FRAMESET>
</HTML>
```

Frame1	Frame2
--------	--------



Frames

- Elementos Frameset y Frame





Frames

- Elementos Frameset y Frame
 - Mediante esta estructura jerárquica podemos establecer una comunicación entre los frames

cont_frame1.html

```
<html>
<body>
<input type="button" value="Prueba"
onclick="javascript:window.parent.Frame2.document.bgColor='red';>
</body>
</html>
```

iFrames



- La comunicación con iframes sólo es posible si el contenido del iframe pertenece al mismo dominio

Cookies



- Cuando un servidor sirve una página, se olvida de que la ha servido y no recuerda nada al respecto (validación, preferencias...)
- Una cookie es un pequeño fichero de texto que se almacena en el cliente con la siguiente información:
 - Una pareja de nombre-valor con la información almacenada
 - Una fecha de caducidad de la información a partir de la cual ya no es válida (en formato UTC) precedida de la palabra expires=
 - El dominio y ruta del servidor que ha servido la cookie

Cookies



- Las cookies son accesibles a través de la propiedad `document.cookie`, que inicialmente podemos tratar como un string separando las tres partes por un punto y coma “;” y un espacio

```
document.cookie = 'user=jordi; expires=Thu, 18 Oct 2012 15:47:11 UTC; path=/'
```

- Leer una cookie de esta manera puede resultar un poco engorroso

Cookies



- Podemos emplear funciones que nos lo faciliten la tarea

```
function createCookie(name, value, days) {  
    if (days) {  
        var date = new Date();  
        date.setTime(date.getTime()+  
            (days*24*60*60*1000));  
        var expires = "";  
        expires="+date.toGMTString();  
    }  
    else  
        var expires = "";  
    document.cookie = name+"="+  
        value+expires+"; path=/";  
}
```

<http://www.quirksmode.org/js/cookies.html>

```
function readCookie(name) {  
    var nameEQ = name + "=";  
    var ca = document.cookie.split(';');  
    for(var i=0;i < ca.length;i++) {  
        var c = ca[i];  
        while (c.charAt(0)==' ')  
            c = c.substring(1,c.length);  
        if (c.indexOf(nameEQ) == 0)  
            return c.substring(nameEQ.length,c.length);  
    }  
    return null;  
}
```