

# JavaScript - JQuery

# Qué es un *framework*?

- Un *framework* es un **esquema** base sobre el que construir una aplicació. Lo podemos entender como una **capa adicional** por encima del lenguaje de programación.
- Su objetivo es **normalizar** o **estandarizar** el proceso de desarrollo, para que el programador no tenga que “perder tiempo” en pensar la estructura de su aplicación y pueda así centrarse en los aspectos realmente relevantes.
- Además de facilitar el proceso, un *framework* también ayuda a que el código resultante esté más ordenado y, por tanto, pueda ser más fácilmente comprensible tanto por él como por otros desarrolladores.
- Existen *frameworks* para todo tipo de lenguajes de programación: **PHP** (Symfony, Laravel), **Java** (Hibernate, Struts), **JavaScript** (jQuery, Angular JS)...

# Javascript

Qué framework **Javascript** utilizaremos?

Actualmente existen multitud de bibliotecas para Javascript, pero aquí utilizaremos una de las más extendidas, **jQuery**.

Además de ser muy sencilla de utilizar, se estima que tres de cada cuatro sitios web que utilizan una biblioteca Javascript la usan, incluyendo empresas como Amazon, Microsoft, BBC o Twitter.

# Javascript

## Obtener jQuery

Podemos descargar jQuery desde su página web:

[www.jquery.com](http://www.jquery.com)

Obtendremos un archivo .js, que deberemos incluir en nuestro documento HTML:

```
<script src="jquery.js"></script>
```

## Fundamentos de jQuery

<http://librojquery.com/>

# Javascript

## Cómo funciona jQuery?

jQuery utiliza el símbolo **\$()** como función que nos permitirá interactuar con la librería.

La función **\$()** nos permite "seleccionar" elementos del **DOM** (**D**ocument **O**bject **M**odel) y, a partir de ahí, empezar a interactuar con el lenguaje de programación.

# Javascript

## Ejemplo:

```
<html>
<head>
  <title>Exemple</title>
  <script src="jquery.js"></script>
  <script>
    $(document).ready(function(){
      var resultat = $('#p1').text();
      alert(resultat)
    });
  </script>
</head>
<body>
  <p id="p1">¡Hola Mundo!</p>
</body>
</html>
```

# Javascript

## `$(document).ready`

**Todo** el código que creemos (Javacript o jQuery, son compatibles), deberá ir **dentro** de esta función, para que así se cargue una vez se haya cargado todo el contenido html y css de la página.

```
$(document).ready(function(){
```

```
    ... código ...
```

```
}
```

# Javascript

jQuery admite tres tipus de selectores:

- Por etiqueta: `$('h1')`
- Por id: `$('#cabecera')`
- **Por clase:** `$('.titulo')`



# Javascript

Como nueva funcionalidad, hay que decir que la función **\$()** admite un segundo parámetro que nos permite definir el contexto en el que queremos que se realice la búsqueda del selector. Por defecto, si el parámetro no existe, la función realiza la búsqueda en todo el documento. Si aplicamos un contexto, sólo la realizará en el contexto:

```
<div>
```

```
    <p>Content</p>
```

```
< /div>
```

```
<div id="p1">
```

```
    <p>Content 2</p>
```

```
</div>
```

```
$('p', $('#p1'));
```

Nomès retornarà el segon "p"

# Javascript

Seleccionados un conjunto de elementos, podemos manipular sus propiedades CSS con el método **.css**, así:

```
$('#p').css('color', 'red');
```

También pueden cambiar muchas propiedades a la vez. Para ello, la función recibirá un objeto Javascript y quedará en el siguiente formato:

```
$('#p').css({ 'color': 'grey',  
'padding': '5px',  
'background-color': 'yellow' });
```

# Javascript

El método **.css** también nos permite obtener el valor de una propiedad css de un elemento, esté ésta definida en el atributo 'style' o en una clase (a diferencia de lo que pasaba en javascript!):

```
var col = $('p').css('color');  
$('div').css('color',col);
```

# Javascript

También podemos añadirles o quitarles clases CSS:

```
$('#p').addClass('estilo1');  
$('#p').removeClass('estilo1');
```

También puede utilizarse el método “toggleClass” para activar o desactivar una clase (si la tenía, se la quita, y sino, se la añade):

```
$('#p').toggleClass('estilo1');
```

O comprobar si un elemento tiene o no una clase con hasClass:

```
if($('#elemento').hasClass('estilo1')){ ...
```

# Javascript

Hay que decir que existen algunos métodos en forma de acceso directo, como por ejemplo:

- **.text ()**: devolverá el texto de un elemento determinado.
- **.val ()**: devolverá el valor de un elemento determinado.
- **.show ()**: nos mostrará el elemento si estaba escondido.
- **.hide ()**: lo esconderá.
- **.append('<p>Ejemplo</p>')**: añade el código indicado al final del elemento

# Javascript

## Eventos

### .click()

```
$( "#target" ).click(function() {  
    alert( "Handler for .click() called." );  
});
```

### .dblclick()

```
$( "#target" ).dblclick(function() {  
    alert( "Handler for .dblclick() called." );  
});
```

```
<div id="target">  
    Click here  
</div>
```

# Javascript

## Eventos

### .focus()

```
$( "#target" ).focus(function() {  
    alert( "Handler for .focus() called." );  
});
```

### .keypress()

```
$( "#target" ).keypress(function(event) {  
    //  
});
```

<label for="target">Type Something:</label>

<input id="target" type="text">

# Javascript

## Eventos .submit()

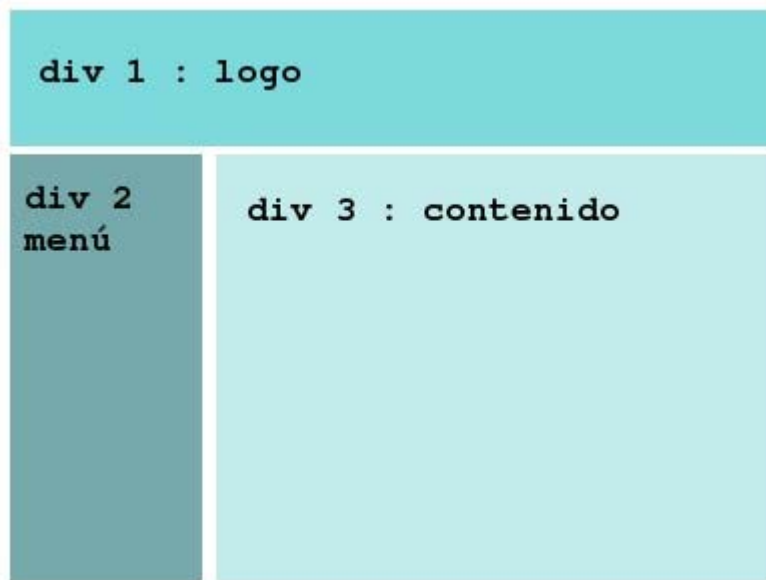
```
$( "form" ).submit(function( event ) {  
    if ( $( "input:first" ).val() === "correct" ) {  
        $( "span" ).text( "Validated..." ).show();  
        return;  
    }  
  
    $( "span" ).text( "Not valid!" ).show().fadeOut( 1000 );  
    event.preventDefault(); // impide que se envía el formulario  
});
```

```
<p>Type 'correct' to validate.</p>  
<form action="javascript:alert( 'success!' );">  
  <input type="text">  
    <input type="submit">  
</form>  
  
<span></span>
```



# Javascript

Navegación sin recargar toda la web



En este caso tendríamos unas opciones en el menú (div 2), y según sobre la que hagamos click, cambiará el contenido de 'div 3', un típico funcionamiento de web.

# Javascript

Navegación sin recargar toda la web

```
<script language="javascript">  
function mostrarContenido(){  
    $.post("pagina.html", function(data){  
        $("#div3").html(data);  
    });  
}  
  
$("#div3").load("pagina.html");
```

# Javascript

## JQuery UI (User Interface)

### Interacciones

Añade comportamientos complejos a los elementos:

**Draggable:** Hace al elemento arrastrable.

**Droppable:** Permite que el elemento responda a elementos arrastrables.

**Resizable:** Permite redimensionar el elemento.

**Selectable:** Permite seleccionar entre una lista de elementos.

**Sortable:** Ordena una lista de elementos.

# Javascript

## Widgets

Es un conjunto completo de controles UI. Cada control tiene un conjunto de opciones configurables y se les pueden aplicar estilos CSS.

**Accordion:** Menú con efecto acordeón.

**Autocomplete:** Caja con autocompletado.

**Button:** Botón.

**Dialog:** Ventanas con contenido.

**Slider:** Elemento para elegir en un rango de valores.

**Tabs:** Pestañas.

**Datepicker:** Calendario gráfico.

**Progressbar:** Barra de progreso.

# Javascript

## Obtener jQueryUI

Podemos descargar jQuery desde su página web:

[www.jqueryui.com](http://www.jqueryui.com)

Seguidamente deberemos enlazar la librería en nuestro documento HTML / PHP:

```
<script src="//code.jquery.com/jquery-1.9.1.js"></script>  
<script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>  
<link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">  
<link rel="stylesheet" href="/resources/demos/style.css">
```