

# U6. Ajax

## **Parte II. Ajax con JQuery**

Desarrollo Web en Entorno Cliente

# Contenidos

- Conceptos básicos JQuery
  - **Recordatorio** JQuery: DOM, eventos, etc.
  - Consultar/modificar value con **.val()**.
  - Recorrer array con **\$.each**
- Funciones específicas peticiones Ajax
- Función genérica peticiones Ajax

# ¿Cómo incluimos JQuery?

- Incluimos JQuery localmente:

```
<head><script  
src="lib/jquery2.2.3.min.js"></script></head>
```

- Incluimos JQuery a través de un CDN:

```
<head>  
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js">  
</script>  
</head>
```

# Selectores

A partir de la función **\$()** es posible seleccionar un elemento del árbol DOM mediante JQuery empleando la sintaxis CSS:

**\$(selectorCSS).acción**

- `$("#midentificador");`
- `$("a");`
- `$("#miParrafo, a");`
- `$(".miClase");`
- `$("a[rel]");` *//Con un determinado atributo*

Para asegurarnos de que la página está cargada empleamos la función **ready()**: `$(document).ready( function() { ... } );`

# Modificar el DOM

- Incorporamos texto en un elemento con **html()** o **text()**
- Incluimos un elemento hijo de otro con **append(elemento)**
- Incluimos un atributo con **attr( "nomAtributo","valAtributo" )**
- Incluimos clases con **addClass( "clase1 clase2 ..." )**

- **JQ** permite **concatenar** varias funciones:

//creo el div, asigno la clase, el id y el contenido en 1 linea

```
var alumno = $(document.createElement("div"))  
  .addClass("alumno").attr("id", "al12345")  
  .html("Alumno 12345");
```

//incluimos el div en body

```
$("body").append(alumno);
```

# Contenidos interactivos

- Podemos incluir contenidos interactivos empleando JQuery mediante
  - Eventos
  - Efectos
  - Animaciones
- Por ejemplo:
  - `$("#p").hide(); //oculta el párrafo`
  - `$("#p").show(); //muestra el párrafo`
  - `$("#p").css('color','red'); //cambia la propiedad color de p`

# Eventos

- Podemos emplear funciones de eventos específicos:

```
$(selector).click(function(){ /* cuerpo de función*/ } )
```

- Podemos vincular eventos con on

```
$(selector).on("evento", function(){ /*cuerpo función*/ } )
```

Ej: `$("p").on("click", function(){  
 alert("Hola!");  
});`

- Podemos desvincular eventos con off

```
$(selector).off("evento")
```

Ej: `$("button").click(function(){  
 $("p").off("click");  
});`

# Consultar/modificar value

- Con [attr](#) podemos consultar/modificar cualquier atributo:
  - **attr( “nomAtributo” ),** devuelve el valor
  - **attr( “nomAtributo”, “valAtributo” ),** cambia el valor
- Con [val\(\)](#) hacemos lo mismo pero sólo para value: **val()** devuelve el valor y **val(“nuevoValor”)** lo cambia.

```
$("#button").click(function(){  
    var nombre = $("#input:text").val(); //almaceno el valor  
    $("#input:text").val("Pepe"); //cambio el valor  
});
```



# Recorrer array

- Podemos recorrer un array en JQuery empleando un for pero en JQuery [\\$.each](#) permite recorrerlo de una manera sencilla:
  - Mostramos los elementos de un array:

```
var arr = ["Ana","Pepe"];  
$.each(arr, function(i, elemento){  
    console.log("El índice " + i + "es " + elemento);  
});
```
  - Mostramos el valor de los input de un formulario:

```
$( "input:text" ).each( function( i, elemento ) {  
    console.log("Valor input nº " + i + " " + elemento.val());  
});
```

# Función específica Ajax

- **\$(selector).load()**("urlAcargar") carga el contenido solicitado en el elemento seleccionado en selector. Ej:  
`$('#resultado').load("datos.txt");`
- Load admite más parámetros:  
**\$(selector).load(url, function(datos, status, xhr){ //uso datos...}**  
`$("#button").click(function(){  
 $("#div1").load("datos.txt", function(respuesta, status, xhr){  
 if(status == "success")  
 alert("Me han enviado: " + respuesta);  
 if(status == "error")  
 alert("Error: " + xhr.status + ": " + xhr.statusText);  
 });  
});`

# Función específica Ajax

- Podemos emplear funciones más específicas para un método y para el tipo de datos JSON.
- Como mínimo debemos indicar: URL y function para definir una función que manipule los datos.
- [\\$.get](#)(URL,function(datos,status){...});
- [\\$.post](#)(URL,[datosEnvio],function(datos,status,xhr){...}, [tipoDatosEnvio]);
  - *Opcionalmente. Puedo enviar datos e indicar su tipo.*
- [\\$.getJSON](#)(URL,[datosEnvio],function(datos,status,xhr){...});
  - *Opcionalmente, puedo enviar datos JSON*

# Función genérica Ajax

- \$.Ajax() permite especificar cualquier cosa respecto a la petición Ajax añadiendo parámetros.

- Ejemplo:

```
$.ajax({  
  url: 'http://localhost/alumnos.json',  
  data: {usuario: 'Pepe'},  
  error: function() {  
    $('#info').html('<p>Hay un error!!</p>');  
  },  
  dataType: 'jsonp',  
  success: function(data) {  
    $('#info').text("La nota del alumno es " + data.nota);  
  },  
  type: 'GET'  
});
```