

JS-JQuery

- **jQuery:**

- Librería de js
- Gestión de eventos, efectos y animaciones.
- Se añade al final del body o en el header con el atributo defer: `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>`
- Accedemos a jQuery mediante `$(selector).action()`
 - Selector: elemento HTML sobre el que queremos actuar
 - Acción: acción a realizar sobre el elemento
 - jQuery también tiene acciones que no requieren selector. `$.action`

- **Selectores:**

- Window: `$(window)`
- Document: `$(document)`
- This: `$(this)`
- SelectoresCSS (revisar sección CSS)

`$("p")`

`$(".rojo")`

`$("#p1")`

- `:contains(texto)`: elementos que contengan texto
 - Cuidado con las comillas, hay que alternar las comillas entre simples, dobles y template literals.

`$(":contains('ipsum')");`

`$(':contains("ipsum")');`

`$(`:contains("ipsum")`);`

`$(":contains(`ipsum`)");` //No devuelve el resultado esperado

- Se puede usar con otros selectores antes de los dos puntos

`$("#p:contains('ipsum')");`

- Mas: <https://api.jquery.com/category/selectors/>

- **Eventos:**

- `.click(handler)`: Se ejecuta la función handler cuando se hace click sobre el elemento

```
$("#p").click(function(){  
  console.log("Has pulsado ", $(this));  
});
```

```
function info(){  
  console.log("Has pulsado ", $(this));  
}  
$("##boton1").click(info);
```

- Cuidado con las funciones con argumentos

```
function info(mensaje){  
  console.log(mensaje);  
}  
$("#p").click(info("Hola"));
```

- Hay que usar una función anónima como wrapper

```
$("#p").click(function(){  
  info("Hola");  
});
```

- `$(handler)`: Se ejecuta cuando el Document Object Model ha terminado de cargar
 - Punto de comienzo para ejecutar JS
 - Pueden no haberse cargado los assets (imágenes, vídeos...)

```
$(function(){  
  console.log("Bienvenido");  
});
```

- `.on("load", handler)`: Se ejecuta handler cuando se ha terminado de cargar el elemento.
 - En vez de load, otros nombres de evento: click, keyDown...

```
$(img).on("load", function(){
```

```
console.log("Se ha cargado la imagen");
});
```

- `$(window).on("load", handler)`: handler se ejecuta cuando se han terminado de cargar los assets de la página.

```
$(window).on("load", function(){
console.log("Bienvenido. Ya está todo listo.");
});
```

- `.on(handler)`: Permite asociar múltiples acciones

```
$("p").on({
mouseenter: function(){
$(this).css("background-color", "green");
},
mouseleave: function(){
$(this).css("background-color", "lightblue");
},
click: function(){
$(this).css("background-color", "yellow");
}
});
```

- Múltiples eventos de ratón, teclado, formularios...: **`.click()`, `.keydown()`, `.blur()`, `.dblclick()`, `.keypress()`, `.focus()`, `.hover()`, `.keyup()`, `.select()`, `.mousedown()`**
- Todos los handlers disponen de información del evento que lo activó en un objeto evento: <https://api.jquery.com/category/events/event-object/>

```
$("p").click(function(e){
console.log(e);
});
```

- Propiedades del objeto evento:

`e.pageX`, `e.pageY` //Posición del ratón

`e.type` //Tipo de evento

[`e.target`](#) //El elemento HTML que inició el evento

`e.timestamp` //Tiempo desde que se cargó la página

- Funciones del objeto evento

`e.preventDefault()` //Previene la acción por defecto

e.stopPropagation() //Evita que el evento se propague a los padres

- **CSS:**

- .css(propertyname): Devolver el valor de una propiedadCSS
 - Encaso de que haya multiples elementos solo devuelve el del primero

```
$("#p").css("background-color");
```

- .css(propertyname,value): Cambiar el valor de una propiedad

```
$("#p").css("background-color", "yellow");
```

- .css({"propertyname":"value","propertyname":"value",...}): Actualizar múltiples propiedades CSS a la vez

```
$("#p1").css({"background-color": "yellow",  
"color": "green",  
"font-size": "24px"  
});
```

- .addClass(clase) / .removeClass(clase): añade / elimina la clase de los elementos HTML

```
$("#p1").addClass("rojo");
```

- .toggleClass(clase): añade la clase a los elementos si no la tienen o la elimina si ya la tienen

```
$("#p").toggleClass("rojo");
```

- MAS: <https://api.jquery.com/css/> .hasClass(), .height(), .width(), .position()...

- **Efectos:**

- .hide(): Oculta el elemento
- .show(): Muestra el elemento

```
$("#boton1").click(function(){  
$("#p1").hide();  
});  
$("#boton2").click(function(){  
$("#p1").show();  
});
```

- `.toggle()`: Oculta el elemento si está visible y lo muestra si está oculto

```
$("#boton1").click(function(){
$("#p").toggle();
});
```

- `fadeIn()`: Muestra el elemento con efecto fade
- `fadeOut()`: Oculta el elemento con efecto fade
- `fadeToggle()`: Muestra/oculta el elemento con efecto fade

```
$("#p1").hide();
$("#boton1").click(function(){
$("#p1").fadeIn();
$("#p2").fadeOut();
$("#p3").fadeToggle(5000);
});
```

- `slideDown()`: Desliza el elemento hacia abajo para mostrarlo
- `slideUp()`: Desliza el elemento hacia arriba para ocultarlo
- `slideToggle()`: Desliza el elemento hacia arriba/abajo

```
$("#boton1").click(function(){
$("#p1").slideToggle();
});
```

- Parámetros de los efectos

```
$(selector).hide(speed,callback);
```

- `speed`: "slow", "fast", Un número que representa los ms.
- `callback`: Función que se ejecuta cuando termina el efecto

```
$("#p1").hide(1000);
$("#p").toggle("slow", function(){
console.log("Terminó el efecto");
});
```

- Se pueden encadenar múltiples efectos (chaining)
 - Cuando termina uno se ejecuta el siguiente, excepto los que son instantáneos

```
$("#p1").css("color", "red")
.slideUp(3000)
```

`.slideDown(3000);`

- MAS: <https://api.jquery.com/category/effects/>

- **Manipulación del DOM:**

- `.html()` : devuelve el contenido del primer elemento HTML

`console.log($(".body").html());`

`console.log($(".div").html());`

- `.html(contenido)`: modifica el contenido de todos los elementos HTML

`$(".div").html("Sin contenido");`

- `.text()` : devuelve el texto combinado de todos los elementos y sus descendientes
 - Ignora las etiquetas html

`console.log($(".body").text());`

- `.text(texto)`: modifica el texto de todos los elementos HTML
 - Trata las etiquetas html como texto

`$(".p").text("Sin contenido");`

- `.val()` : devuelve el valor del primer elemento
 - Se usa con elementos de formulario (input, select, textarea...)

`console.log($(".input").val());`

- `.val(valor)`: modifica el valor de todos los elementos HTML

`$(".input").val("Prueba el texto");`

- `.attr(nombreAtributo)` : devuelve el valor de un atributo HTML

`console.log($(".a").attr("href"));`

- `.attr(nombreAtributo, valor)`: modifica el valor del atributo

`$(".a").attr("href", "https://jquery.com/");`

- `.prepend(contenido)` / `.append(contenido)`: añade contenido al principio / final dentro cada elemento HTML

`$("#p1").prepend("Al principio");`

```
$("#p1").append("<span>Al final</span>");
```

- Permiten añadir múltiple contenido

```
let txt1 = "<h1>Un título</h1>";
```

```
let txt2 = "<h1>Otro título</h1>";
```

```
$("#body").append(txt1, txt2);
```

- .before(contenido) / .after(contenido): añade contenido al antes / detrás de cada elemento HTML

```
$("#p1").before("<h2>Título</h2>");
```

```
$("#p1").after("<span>Comentario final</span>");
```

- Permiten añadir múltiple contenido

```
let txt1 = "<h2>Un título</h2>";
```

```
let txt2 = "<p>Párrafo de introducción</p>";
```

```
$("#p1").before(txt1, txt2);
```

- .remove() : elimina los elementos HTML y todo su contenido

```
$("#p").remove();
```

- .remove(selector): elimina los elementos HTML que cumplan el selector

```
$("#p").remove(":contains('Lorem')");
```

- .empty() : vacía los elementos HTML

```
$("#p").empty();
```

- .children() : Devuelve los hijos de los elementos HTML

```
$("#body").children();
```

- .parent() : devuelve el elemento padre de los elementos HTML

```
$("#a").parent();
```

- .siblings() : devuelve los hermanos de los elementos HTML

```
$("#p1").siblings();
```

- MAS: <https://api.jquery.com/category/manipulation/>

- **Plantilla:**

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Demo</title>
<style type="text/css">
p{padding: 10px;}
#p1{padding: 150px;
background-color: #e5eccc;}
.rojo{color: red;}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js">
</script>
</head>
<body>
<h1>Prueba jQuery</h1>
<p id="p1">This is paragraph 1</p>
<p id="p2" class="rojo">This is paragraph 2</p>
<p id="p3">This is paragraph 3</p>
<div>
<a href="https://www.google.es/">Enlace</a>
</div>
<div>
<input type="text">
</div>
<div>
<button id="boton1">Botón 1</button>
<button id="boton2">Botón 2</button>
</div>
<form>
<button id="boton_form">Submit</button>
</form>
</body>
</html>
```