

JQUERY

JQUERY

- Es una librería de Javascript muy extendida que facilita:
 - Manejo del DOM
 - Gestión de eventos
 - Animaciones
 - Uso de AJAX
 - Compatibilidad con diversos navegadores
 - “Write less, do more”

JQUERY

- Para incorporarla a nuestro proyecto, tenemos dos opciones:
 - Descargarla de su web <https://jquery.com>, añadir el script a nuestro sitio web e incluir el script cuando vayamos a utilizarlo
 - Incluir jQuery desde un CDN como por ejemplo <https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js>
 - Esta opción tiene la ventaja de que facilita que nuestro navegador tire de caché y no descargue la librería, pero dependemos de que el CDN esté online
- En cualquiera de los casos, es importante cargar el script de jQuery antes que los scripts que vayan a utilizarlo

JQUERY

- Es importante que nos adaptemos a jQuery (Don't fight the framework!)
- La sintaxis tipo de jQuery es: `$(selector).acción()`:
 - El símbolo `$` es la función principal de jQuery
 - Admite múltiples tipos de valores que veremos más adelante
 - El selector obtiene elemento/s html
 - La acción a realizar sobre dichos elementos
- Por ejemplo:
 - `$(document)`: devuelve el documento
 - `$("p").text('Hola')`: pone el texto "Hola" en todos los elementos `<p>`
 - `$(".table-big").hide()`: oculta todos los elementos con `class="table-big"`
 - `$("#txt").html("hola")`: cambia el contenido del `id="txt"`
 - `$(this).hide()`: oculta el elemento actual

JQUERY - SELECTORES

- Nos permiten acceder y manipular elementos del html
 - Encuentran elementos por su id, nombre, clase, tipo, atributos, valores...
 - Similar al `document.getElementsByTagName()`, pero más potente
- Algunos ejemplos:
 - Para localizar elementos por su tag: `$("tag")` `$("p")`
 - Obtener un elemento por su id: `$("#id")` `$("#texto4")`
 - Encontrar los de una clase: `$(".clase")` `$(".table-big")`
 - Devuelve todos los elementos: `$("*")`
- Y muchos más...

JQUERY - SELECTORES

\$(this)	Elemento actual
\$("p.intro")	Todos los <p> con class="intro"
\$("p:first")	Primer <p>
\$("ul li:first")	Primer de la primera
\$("ul li:first-child")	Primer de cada
\$("[href]")	Todos los elementos que tengan un atributo href
\$("a[target='_blank']")	Todos los <a> cuyo atributo target valga "_blank"
\$("a[target!='_blank']")	Todos los <a> cuyo target sea distinto a "_blank"
\$(":button")	Todos los <button> y los <input> con type="button"
\$("tr:even")	Todos <tr> pares
\$("tr:odd")	Todos los <tr> impares

JQUERY - EVENTOS

- jQuery facilita enormemente la gestión de eventos
- Algunos de los eventos que podemos gestionar son:
 - De ratón: click, dbclick, mouseenter, mouseleave
 - De teclado: keypress, keydown, keyup
 - De forms: submit, change, focus, blur
 - De document/ventana: load, resize, scroll, unload
- Para asignar un manejador cuando hacemos en cualquier `<p>` que haga que se oculte ese `<p>` en concreto:
 - `$("#p").click(function() {$(this).hide()});`

JQUERY - EVENTOS

- Podemos añadir manejadores para diferentes eventos con **on**:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <p>Hola!</p>

  <script>
    $("p").on({
      mouseenter: function(){
        $(this).css("background-color", "lightgray");
      },
      mouseleave: function(){
        $(this).css("background-color", "lightblue");
      },
      click: function(){
        $(this).css("background-color", "yellow");
      }
    });
  </script>
</body>
</html>
```


JQUERY - EVENTO DOCUMENT.READY

- Habitualmente, querremos esperar a que la página esté cargada antes de empezar a ejecutar nuestros scripts
 - Esto se consigue con `$(document).ready(function(){});`
- Funcionamiento es similar a utilizar el event listener; `document.addEventListener('DOMContentLoaded', ...)`
 - Utilizando JS puro, debemos tener en cuenta la compatibilidad de los diferentes navegadores. El event listener de `DOMContentLoaded` no funciona con algunos navegadores antiguos
 - JQuery se encarga por nosotros de que todo funcione, independientemente del navegador
- Esto evita que intentemos acceder a un elemento del documento que todavía no esté creado

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script>
    $(document).ready(function() {
      console.log( "Ya ha cargado la página!" );
    });
  </script>
</body>
</html>
```


JQUERY - ACCIONES

- Una vez hemos seleccionado el/los elementos que queremos manipular, podemos realizar acciones como, por ejemplo:
 - `show()`: muestra el elemento
 - `hide()`: oculta el elemento
 - `text()`: accede al texto del elemento
 - `html()`: accede al contenido html del elemento
 - `attr(a)`: accede al atributo a
 - `each(f)`: ejecuta la función f para cada elemento
 - Y muchas más...

JQUERY - MÁS INFORMACIÓN

- En la web oficial tenemos documentación y ejemplos para seguir aprendiendo jQuery
 - <https://jquery.com>
 - Otra web útil es: <https://www.w3schools.com/jquery/>

EJEMPLO JQUERY

- Podéis ver una reimplementación del ejercicio de lista de empleados utilizando jQuery en la carpeta 01_ejercicio_fundamentos_con_jquery del [repositorio](#)

Nombre	Apellidos	Fecha de nacimiento	Sexo	Tiempo parcial	Borrar
Alfonso	Izquierdo	Invalid Date	Hombre	✓	<button>Borrar</button>
Lara	Márquez	20/12/1991	Otro	✗	<button>Borrar</button>
John	Muller	Invalid Date	No Especificado	✗	<button>Borrar</button>

Nombre


Apellidos

☐ Hombre

☐ Mujer

☐ Otro

☐ No especificado

Fecha de nacimiento 

☐ Tiempo parcial