



Índice

| | |
|--|---|
| Semana 1..... | 3 |
| Introducción..... | 3 |
| ¿Qué es jQuery?..... | 3 |
| ¿Por qué usar jQuery?..... | 3 |
| Cómo agregar jQuery a sus páginas web..... | 3 |
| 1.- Descargando jQuery..... | 3 |
| 2.- CDN de jQuery..... | 4 |
| Sintaxis de jQuery..... | 4 |
| Desarrollo de una tienda con JS y JQ..... | 6 |
| Arranque y carga de datos..... | 6 |

Semana 1

Introducción

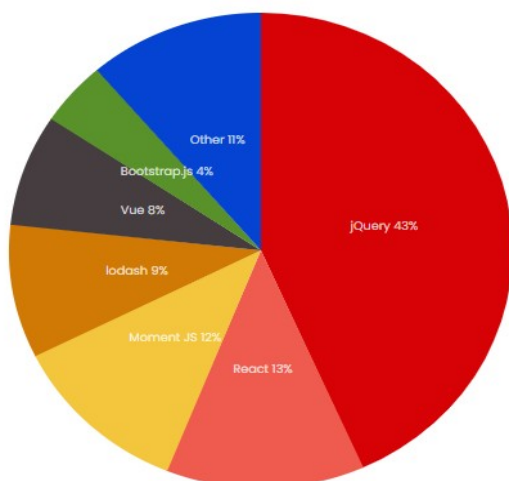
¿Qué es jQuery?

Es una biblioteca de JavaScript popular diseñada para facilitar la manipulación del DOM (Document Object Model), el manejo de eventos, las animaciones y la interacción con servidores a través de AJAX.

Su principal ventaja es hacer que el código JavaScript sea más simple y fácil de entender, lo que permite a los desarrolladores escribir menos código para tareas típicas, como elegir y cambiar elementos de una página web. Además, jQuery proporciona una API unificada que funciona en todos los principales navegadores, lo que ayuda a resolver problemas de compatibilidad entre navegadores.

¿Por qué usar jQuery?

Hay muchas otras bibliotecas de JavaScript, pero jQuery es probablemente la más popular y también la más extensible.



Muchas de las empresas más grandes de la Web utilizan jQuery, como:

- Google
- Microsoft
- IBM
- Netflix

Cómo agregar jQuery a sus páginas web

Existen varias formas de comenzar a usar jQuery en su sitio web. Puede:

- Descargar la biblioteca jQuery desde <https://jquery.com>
- Incluir jQuery desde un CDN, como Google

1.- Descargando jQuery

Hay dos versiones de jQuery disponibles para descargar:

- Versión de producción: es para su sitio web en vivo porque se ha minimizado y comprimido.
- Versión de desarrollo: es para pruebas y desarrollo (código sin comprimir y legible)

Ambas versiones se pueden descargar desde [jQuery.com](https://jquery.com) .

La biblioteca jQuery es un único archivo JavaScript, y se hace referencia a él con la `<script>`etiqueta HTML (tenga en cuenta que la `<script>`etiqueta debe estar dentro de la `<head>`sección):

```
<head>
<script src="jquery-3.7.1.min.js"></script>
</head>
```

Consejo:

Coloque el archivo descargado en el mismo directorio de las páginas donde desea utilizarlo.

2.- CDN de jQuery

Si no desea descargar y alojar jQuery usted mismo, puede incluirlo desde una CDN (red de distribución de contenido).

Google es un ejemplo de alguien que aloja jQuery:

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
</head>
```

Sintaxis de jQuery

La sintaxis jQuery está hecha a medida para seleccionar elementos HTML y realizar alguna acción en dichos elementos.

La sintaxis básica es: **`$(selector) .action ()`**

- Un signo \$ para definir/acceder a jQuery
- Un (selector) para "consultar (o encontrar)" elementos HTML
- Una acción jQuery () que se realizará en el elemento(s)
-

Ejemplos:

- ◆ `$(this).hide()` --> oculta el elemento actual.
- ◆ `$("p").hide()` --> oculta todos los elementos `<p>`.
- ◆ `$(".test").hide()` --> oculta todos los elementos con `clase="test"`.
- ◆ `$("#test").hide()` --> oculta el elemento con `id="test"`.

Advertencia

Para evitar que cualquier código jQuery se ejecute antes de que el documento termine de cargarse (esté listo), se ponen todos los métodos jQuery dentro del evento “documento listo”:

```
$(document).ready(function(){  
    // jQuery methods go here...  
});
```

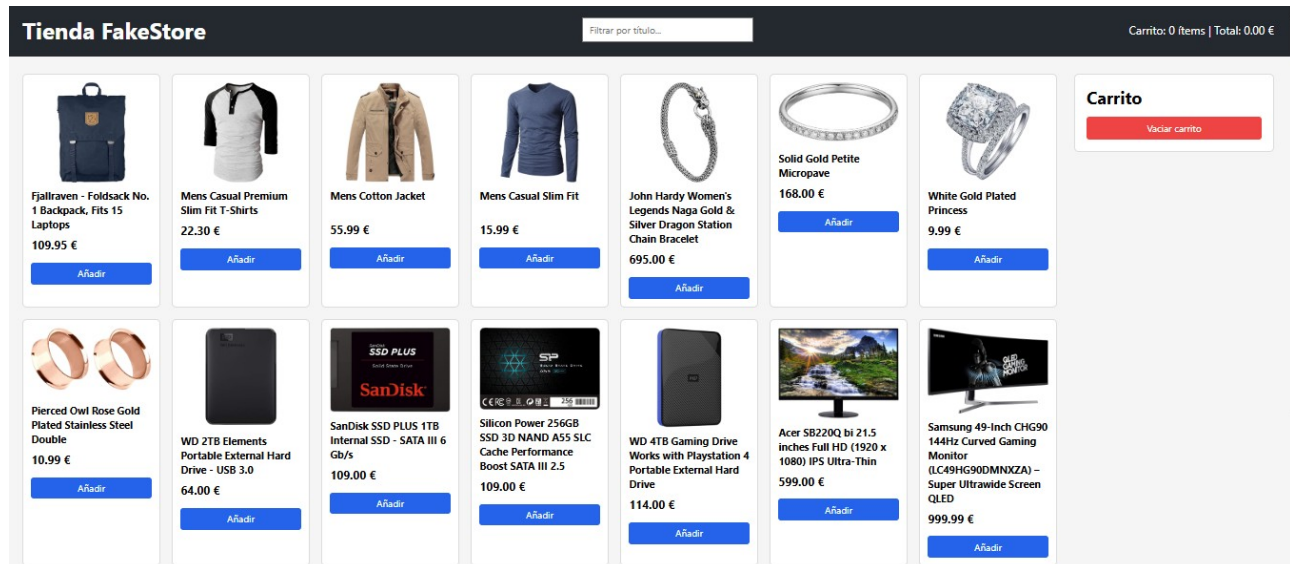
Hay un método aún más corto para el evento “documento listo”:

```
$(function(){  
    // jQuery methods go here...  
});
```

Es una buena práctica esperar a que el documento esté completamente cargado y listo antes de trabajar con él. Esto también le permite tener el código JavaScript antes del cuerpo del documento, en la sección de encabezado.

Desarrollo de una tienda con JS y JQ

Aprenderemos JQ a través de la creación de una tienda usando la API de desarrollo <https://fakestoreapi.com> siguiendo un patrón “datos ↔ lógica ↔ vista”.



Arranque y carga de datos

Teoría que aprenderás

- El elemento Selector: básicos y de atributos.
- Uso de fetch.
- Modificación de atributos: `.attr()`, `.html()`, `.text()`.
- Manejo de clases: `.addClass()`, `.removeClass()`, `.toggleClass()`
- Inserción de elementos en el DOM (`.append()`, `.prepend()`, `.after()`, `.before()`).`outline`
- Eliminación de elementos: `.remove()`, `.empty()`.

Desarrolla

Carga de librería jquery en el proyecto

- Descarga la librerías jquery de: <https://code.jquery.com/jquery-3.7.1.js>
- Crea el la carpeta /libs y dentro el fichero /jquery_3_7_1.js y pega el contenido anterior. Fíjate que `_3_7_1` hace referencia a la versión.
- En el index, dentro del body, crea el `<script>` que apunte a `jquery_3_7_1.js`.

Creación de la entrada a la aplicación

- Crea la carpeta “js” y dentro el fichero main.js
- En el index, dentro del body, crea el <script> que apunte a main.js.

Obtener los datos de la API

La idea es que main.js traiga los datos a través de un Manager y quede independiente de la fuente. Para ello creamos las carpetas

- Crea la carpeta “js” y dentro el fichero ApiStorageService.js
- Crea la carpeta “js” y dentro el fichero ProductManager.js
- En main.js
 - Importar e instanciar ProductManager.js
 - Crear método asíncrono cargarProductos() con try-catch para cargar los datos desde ProductManager.js
- En ProductManager.js
 - Importar e instanciar ApiStoreService.js
 - Crear método asíncrono fetchAll() con try-catch para cargar los datos desde ApiStoreService.js
- En ApiStorageService.js
 - Crear método asíncrono getAllProducts() con try-catch para cargar los datos desde "<https://fakestoreapi.com>" usando fetch.