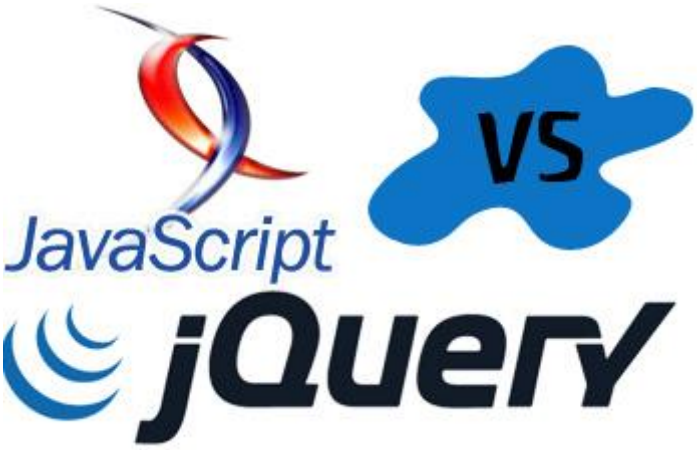
# *Ejemplos prácticos de JavaScript sin utilizar jQuery*



Dar forma a un portal web requiere del uso de varias tecnologías. Entre ellas podemos destacar PHP, HTML, CSS o JavaScript. Cada una de ellas tiene un papel fundamental en el diseño y desarrollo de la página web. A su vez, dentro de estas tecnologías, nos podemos encontrar con ciertos framework que buscan facilitarnos nuestro trabajo, aunque en muchos casos los utilizamos para realizar cosas sencillas para las que en realidad no sería necesario hacer uso de ellos. Hoy nos centraremos en el uso de jQuery.

## ¿Qué es jQuery?


jQuery es simplemente una librería JavaScript de código abierto. Está optimizado para funcionar de forma correcta en los principales navegadores del mercado, siendo compatible con CSS3. El objetivo principal de este framework es hacer la programación "scripting" mucho más ágil y rápida del lado del cliente. Existen otras muchas librerías JavaScript como MooTools, pero jQuery se ha convertido en la más popular debido a su facilidad de uso y su gran potencia.

Una cosa hay que tener clara, JavaScript y jQuery no son dos lenguajes de programación diferentes ya que en ambos casos se utilizan sentencias típicas de JavaScript. La diferencia radica en que jQuery ha sido optimizado para realizar determinadas tareas habituales pero utilizando un menor número líneas de código.

## Ventajas de jQuery

La fama adquirida por este framework respecto a sus rivales es debido a las muchas ventajas que presenta:

- Posibilidad de añadir plugins fácilmente. Cada nuevo plugin ofrecerá nuevas funcionalidades
- Licencia open source. Esto se traduce en que se dispone de un soporte constante y rápido, publicándose nuevas actualizaciones de forma constante
- Ofrece una excelente integración con AJAX
- Su curva de aprendizaje es pequeña. Con muy poco esfuerzo, cualquier desarrollador podrá sacarle el máximo partido

# Ejemplos prácticos de JavaScript sin jQuery

Ya hemos hablado de cómo jQuery puede simplificarnos el desarrollo de una página web, a la vez que abre un amplio abanico de posibilidades, pero en ocasiones es utilizado de forma incontrolada, sobre todo en proyectos pequeños donde su uso no es del todo necesario. Muchas veces, jQuery es mal utilizado sólo para asignar un evento a unos botones o colocar una animación a algún elemento, cosas que podemos hacer con JavaScript sin necesidad de tener que importar la librería. A continuación realizaremos un repaso por ciertas acciones que podemos realizar sin tener que utilizar jQuery.

## Detectar que el DOM está listo para ser manipulado

Una página no puede ser modificada hasta que todo el documento esté listo, es decir, todos los elementos estén cargados. Debido a esto, es necesario que nuestro código JavaScript esté metido dentro de una función que detecte este estado.

Esta función que se encarga de detectar este estado, sería la siguiente.

```
document.addEventListener('DOMContentLoaded', function () {
   console.log( " Estoy listo para modificar!" );
});
```

## Seleccionar elementos

Uno de los principales motivos por el que se suele utilizar jQuery, es por su sencillez a la hora de seleccionar los elementos del DOM con los que se quiere trabajar. Normalmente, basta con hacer uso de la función "$()" para acceder a ellos mediante el nombre de una clase, un id o el nombre de los tags.

Lo que no todo el mundo sabe es que JavaScript también ofrece un API que permite seleccionar elementos de la misma forma que jQuery.

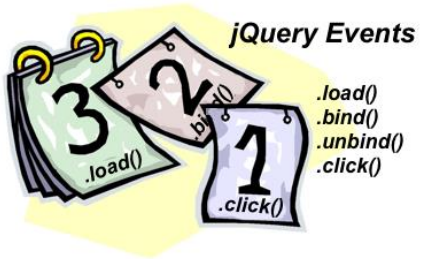
Por ejemplo, si queremos seleccionar el primer elemento que coincida con el selector indicado, se haría de la siguiente forma.

```
var variable = document.querySelector(".miSelector");
```

Si queremos que nos devuelva todos los elementos que coincidan con ese selector, entonces lo podríamos hacer de la siguiente manera.

```
var variable = document.querySelectorALL(".miSelector");
```

## Añadir y eliminar eventos




Trabajar con eventos en JavaScript puede resultar igual de fácil que si lo hacemos con jQuery, con la ventaja de que no será necesario importar ninguna librería. En cualquier momento, podemos añadir o eliminar eventos que estén asociados a un determinado elemento de la página. Para ello, lo primero que debemos hacer es seleccionar el elemento, de la misma forma que lo hemos hecho anteriormente.

```
var boton = document.querySelector("button");
```

Definimos la función que se ejecutará cuando se produzca tal evento.

```
var botonPulsado = function () {
   console.log("Se ha pulsado el botón");
};
```

Finalizamos añadiendo el evento al elemento seleccionado, para que se ejecute la función definida anteriormente.

```
boton.addEventListener("click", botonPulsado);
```

Al igual que añadimos el evento, también lo podemos eliminar. Para ello, utilizaríamos la siguiente instrucción.

```
boton.removeEventListener("click", botonPulsado);
```

## Obtener y modificar el contenido de los elementos

Otra de las acciones habituales, es recuperar el valor de un determinado elemento o bien asignarle un nuevo valor. La forma que tenemos de hacerlo sin hacer uso de jQuery y solo con JavaScript puro sería la siguiente.

Lo primero, como siempre, es seleccionar el elemento.

```
var miElemento = document.querySelector("#selector");
```

A continuación, podemos recuperar su valor utilizando el atributo "textContent".

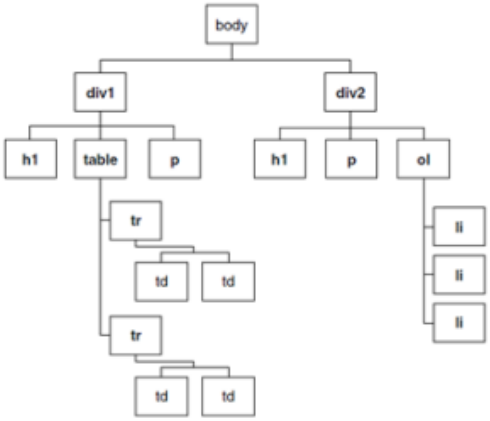
```
var miContenido  = miElemento.textContent;
```

Si queremos asignarle un nuevo valor, lo haríamos de la siguiente manera.

**miElemento.textContent = "Nuevo contenido";**

De esta forma, el elemento que hemos seleccionado al principio, tomaría el valor de "Nuevo contenido".

## Navegando por el DOM de la página




Moverse entre los elementos que forman parte del DOM de una web puede ser muy útil en determinados momentos. Para esto, el API de JavaScript ofrece funciones que pueden resultar muy útiles partiendo del elemento seleccionado mediante la función "document.querySelector()".

**var ejemplo = document.querySelector('#selector);**

Algunas funciones que tenemos para movernos por el DOM son las siguientes.

**var padre = ejemplo.parentNode;**
**var hijo = ejemplo.children;**
**var anterior = ejemplo.previousElementSibling;**
**var siguiente = ejemplo.nextElementSibling;**

## Llamadas AJAX

AJAX es un conjunto de tecnologías que permiten realizar llamadas asíncronas al servidor. Aunque este tipo de llamadas son mucho más sencillas con jQuery, no queríamos dejar la oportunidad de mostraros cómo se puede hacer utilizando JavaScript puro.

```
var request = new XMLHttpRequest();
//Dirección a la que accedemos de forma asíncrona a por los datos
request.open('GET', 'ajax/test.php', true);

request.onload = function (e) {
  if (request.readyState === 4) {

    // Revisamos si la obtención de los datos fue satisfactoria

    if (request.status === 200) {
      console.log(request.responseText);
    } else {
      console.error(request.statusText);
    }
  }
};
```

Puede parecer complicado, pero no lo es.

A lo largo de este libro blanco, hemos querido mostraros cómo no siempre es necesario utilizar jQuery en nuestros proyectos, ya que todo lo que ofrece se puede hacer mediante código puro de JavaScript. Ya sabéis, si vuestro proyecto es pequeño, quizás no os interese utilizar este famoso framework.