

## TEXT CLASS REVIEW

### TEMAS QUE TRATAREMOS EN LA CUE

- Qué es JQuery
- Manipulación del DOM con JQuery
- Plugins de JQuery

Dentro de la programación con **JavaScript**, existen múltiples librerías y *frameworks* que se pueden utilizar, uno de ellos es la famosa librería **JQuery**. Esta biblioteca simplifica enormemente la programación con **JavaScript**, considerándose de fácil aprendizaje y proporcionando un acceso simple a una serie de funciones y métodos de **JavaScript** ya prediseñados.

### QUE ES JQUERY

En el sitio oficial de **JQuery** encontramos una respuesta a esta interrogante. Nos dice que “**JQuery** es una biblioteca de **JavaScript** rápida, pequeña y con muchas funciones. Hace que cosas como el recorrido y la manipulación del documento **HTML**, el manejo de eventos, la animación y **Ajax** sean mucho más simples con una **API** fácil de usar que funciona en una multitud de navegadores. Con una combinación de versatilidad y extensibilidad, **JQuery** ha cambiado la forma en que millones de personas escriben **JavaScript**”.



Ilustración 1 Logo JQuery

Entonces, como bien recalcamos previamente, **JQuery** es una biblioteca de **JavaScript** liviana y que facilita en gran medida el uso de este lenguaje de programación en nuestras páginas web.

Es necesario dejar en claro que JQuery no es un lenguaje de programación si no, una biblioteca de JavaScript.

**jQuery** toma muchas tareas comunes que requieren muchas líneas de código **JavaScript** para llevarlas a cabo y las envuelve en métodos que se pueden llamar con una sola línea de código.

La biblioteca **jQuery** contiene las siguientes características:

- Manipulación HTML / DOM
- Manipulación de CSS
- Métodos de eventos HTML
- Efectos y animaciones
- AJAX
- Utilidades

Una de las grandes ventajas del uso de esta librería se basa en un problema muy recurrente al desarrollar en el “lado del cliente”. Antes de **JQuery**, al desarrollar, nos veíamos en la obligación de revisar cada uno de nuestros elementos en cada uno de los diversos navegadores, para evaluar cómo funcionaba en ellos y realizar los “arreglos” para tener el diseño adecuado. Con la llegada de **JQuery** la principal ventaja es que ya no necesitamos preocuparnos sobre si el navegador del usuario es **Edge, Chrome, Firefox** u otro, ya que la propia librería se encargará de ejecutar el código que sea compatible con el software del cliente que está accediendo a nuestra web.

Es importante considerar que es necesario conocer **JavaScript** para trabajar con **JQuery** ya que, al escribir código **JQuery** realmente estamos programando en **JavaScript**.

**JQuery** es un software libre y de código abierto. Fue creado por [John Resig](#) y presentado el 14 de enero de 2006. Se cree que **JQuery** sigue siendo la biblioteca de **JavaScript** más utilizada, estando presente en un 77,2% de los sitios web, según un informe de [W3Techs](#). Por eso y por todas sus funcionalidades, es importante que aprendamos a usar esta poderosa herramienta.

## SINTAXIS DE JQUERY

**JQuery** tiene una sintaxis un poco diferente a **JavaScript** puro, esta consta de un signo de dólar, que indica que se está trabajando con **JQuery** y, entre paréntesis, encontraremos indicado el selector al cual nos dirigimos. Esto puede ir seguido de una acción a realizar con este selector.

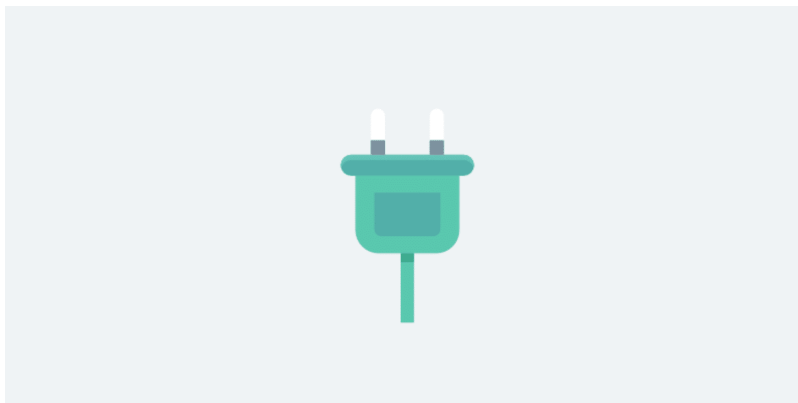
**`$(selector).acción();`**

**JQuery** al ser una librería grande y conocida tiene múltiples funciones que podemos utilizar.

## QUÉ ES UN PLUGIN JQUERY

*Plugin* (puede ser plug-in) es un concepto de la lengua inglesa que puede entenderse como “inserción” o como “complemento”.

Un *plugin*, en palabras simples, no es más que un método propio de la función **JQuery** creado para extender la funcionalidad de un objeto **JQuery**, es decir, añade una funcionalidad adicional o una nueva característica al software.



*Ilustración 2 Concepto de enchufar o insertar una funcionalidad*

Una de las principales ventajas del uso de *plugin* es que facilitan la colaboración de desarrolladores externos con el *software*. Como desventaja, en algunas ocasiones puede ocurrir algún conflicto entre el plugin y la aplicación principal, lo que puede provocar fallos en la ejecución. En estos casos, por lo general, el *software* brinda la opción de desactivar el *plugin* o incluso, eliminarlo.

Los *plugins* se utilizaron por primera vez en la década de los 70, en editores de texto como por ejemplo **Hypercard** o software de publicación como **QuarkXPress** para el **Macintosh de Apple**.

## TIPOS DE PLUGINS

Podemos decir que existen *plugins* para prácticamente cualquier ámbito de *software* o tipo de programa. Los más conocidos son los *plugins* para navegadores de internet como **Firefox** o **Chrome**. Entre los más populares podríamos nombrar a **Flash Player** para vídeos de tipo Flash o **Acrobat Reader** para visualizar archivos PDF.

Pero no solo existen los *plugins* de navegadores. Algunos otros ejemplos son:

- **Plugins de audio:** Para estudios digitales de grabación, como [Kontakt](#) o [Nexus](#).
- **Plugins sociales:** permiten incorporar funciones en populares redes sociales. Algunos *plugins* podrían ser [Social Slider Feed](#) y [Highlight and Share](#).
- **Plugins para IDE:** permite habilitar nuevos lenguajes de programación en entornos de desarrollo integrados. Aquí podemos considerar [File-icons](#) o [CheckStyle](#).
- **Plugins CMS:** permiten incorporar funciones que dan la posibilidad a cualquier persona, incluso principiante de utilizar los sistemas de gestión de contenidos para crear páginas web rápidas y sencillas. Dentro de los ejemplos que podemos encontrar existe [Sidebar Login](#) y [Simple CMS](#).

## PLUGINS COMUNES DE BOOTSTRAP JQUERY

**Bootstrap** trae consigo *plugins* de **JQuery** que amplían su funcionalidad, otorgando un sitio más interactivo. Hablar en detalle de cada uno es complejo, ya que la cantidad de *plugins* disponibles para uso son muchos, pero podemos nombrar algunos:

- [Fuel UX](#)
- [Bootstrap Lightbox](#)
- [Bootbox.js](#)
- [Bootstrap Switch](#)
- [JQuery File Upload](#)
- [Pick a Color](#)

## QUÉ MÁS PODEMOS REALIZAR CON JQUERY

Con la biblioteca **JQuery** podemos realizar la captura de datos y manipulación de datos del **DOM**, de la misma forma que lo hacemos con **Vanilla JavaScript** (**JavaScript** puro, sin librería ni *framework*), pero utilizando los beneficios del uso de **JQuery**. Podemos crear animaciones, eventos, etc.

Para conocer más sobre estos interesantes temas podemos ingresar a los enlaces que se comparten a continuación:

- [Tutorial de JQuery](#)
- [JQuery](#)
- [JQuery write les, do more](#)
- [Manual de JQuery](#)