C.F.G.S.: DESARROLLO DE APLICACIONES WEB
Módulo: DESARROLLO WEB EN ENTORNO CLIENTE

# 01 INTRODUCCIÓN A JQUERY


jQuery

Software

jQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Wikipedia

**Fecha del lanzamiento inicial:** 26 de agosto de 2006

**Autor original:** John Resig

**Desarrollador:** Equipo de desarrollo

**Estado actual:** En desarrollo

**Licencia:** MIT

**Plataformas admitidas:** véase soporte en navegadores web

**Lenguaje de programación:** JavaScript

http://jquery.com

https://www.w3schools.com/jquery/

## 1. ¿Qué es jQuery?

**jQuery** es una librería o biblioteca de Javascript(creado en 2006 por John Resig) que contienen funciones y métodos javascript compatibles con todos los navegadores.

Se utiliza para no tener que desarrollar tareas básicas que en esta librería ya están implementadas y probadas. Nos facilitará la creación de aplicaciones complejas del lado del cliente: efectos dinámicos, recorrido y manejo del DOM, tratamiento de eventos, aplicaciones que hacen uso de Ajax, galerías, etc…

Su éxito se debe a:

- Es gratuito.
- Es una biblioteca relativamente poco pesada (ocupa poca memoria).
- Crea una capa de abstracción entre las funcionalidades y los navegadores, el programador no tiene que controlar la compatibilidad.
- Aprovecha los conocimientos que pudiera tener el usuario sobre CSS para ofrecerle un método de selección de nodos del DOM muy refinado.
- Puede ampliarse a través de plugins.
- Su código es compacto, eficiente y muy depurado.

Para utilizarlo tendremos que incluir en nuestras páginas un script que contenga el código de jQuery.

El sitio web oficial de jQuery es:

http://jquery.com

El origen del código, podemos ubicarlo en nuestro propio servidor o requerirlo de alguno de los varios proveedores que lo tienen a disposición pública

* **http://jquery.com/download/**

  (última versión jquery-3.7.1)

* **jQuery** a través de Media Template: El url de la biblioteca es
  http://code.jquery.com/jquery-3.7.1.min.js

* **Google**. Éste es probablemente el más popular de todos debido a la magnitud del CDN *(Content Delivery Network (CDN) es un conjunto de servidores que contiene copias de una misma serie de contenidos de una web para agilizar su* entrega) de esta empresa. Su url es

**http://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js**

El archivo ocupa muy poco espacio, lo que es bastante razonable y no retrasará mucho la carga de nuestra página (si nuestro servidor envía los datos comprimidos, lo que es bastante normal, el peso de jQuery será aún menor). Además, nuestro servidor lo enviará al cliente la primera vez que visite una página del sitio. En siguientes páginas el cliente ya tendrá el archivo, *por lo que no necesitará transferirlo y lo tomará de la caché*. Las ventajas a la hora de desarrollo de las aplicaciones, así como las puertas que nos abre jQuery compensan extraordinariamente el peso del paquete.

No obstante, si nuestra aplicación se utiliza en un servidor sin conexión a Internet, por ejemplo, en una intranet, tendremos que utilizar una versión local.

jQuery es un producto con una aceptación por parte de los programadores muy buena. Es un producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de su mejora y actualización. Otra cosa muy interesante es la comunidad de creadores de plugins, lo que hace fácil encontrar soluciones ya creadas en jQuery para implementar asuntos como interfaces de usuario, galerías, votaciones, efectos diversos, etc.

## 2. Utilizar jQuery

Para utilizar jQuery tenemos que tener en cuenta lo siguiente:

1. Necesitamos un script que contenga el código de jQuery (archivo js) para incluirlo en nuestras páginas.
    a. Podemos obtenerlo de internet

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

    b. O trabajar en local con la última versión de la librería descargada. Hay dos posibilidades:

- Versión PRODUCTION. Está minimizada, ocupa menos espacio. Es adecuada para aplicaciones en producción

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

- Versión DEVELOPMENT. El código no está comprimido, ocupa más pero se puede ver la implementación de las funciones. Adecuada para el desarrollo de las aplicaciones.

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

2. Detectar el momento en el que la página está lista para ejecutar código Javascript que hace uso del DOM

Cuando hacemos ciertas acciones complejas (que modifican cualquier cosa de la página) con Javascript tenemos que estar seguros que la página haya terminado de cargar y esté lista para recibir comandos Javascript que utilicen la estructura del documento, el DOM. Hay que hacerlo cuando el navegador ha recibido el código HTML completo y lo ha procesado. Para ello, jQuery incluye una manera de hacer acciones justo cuando ya está lista la página, aunque haya elementos que no hayan sido cargados del todo. Esto se hace con la siguiente sentencia:

```
$(document).ready(function(){
//código a ejecutar cuando el DOM está listo para recibir instrucciones.
});
```

Con $(document) se obtiene una referencia al documento (la página web) que se está cargando.

Con el método ready() se define un evento que se desata al quedar listo el documento para realizar acciones sobre el DOM de la página.

3. Insertar código

Podemos utilizar la función jQuery(*selector*), que suele expresarse con su alias $(*selector*). Sirve para seleccionar todos los nodos del DOM que coinciden con el selector indicado.

```
$("a").click(function(evento){
//aquí el código que se debe ejecutar al hacer clic
} );
```

Para crear un evento click sobre un elemento tenemos que invocar al método click sobre ese elemento y pasarle como parámetro una función con el código que queremos que se ejecute cuando se hace clic.

***Ejemplo 01, Ejemplo 02, Ejemplo 03, Ejemplo 04***

## 3. FUNCIÓN $

El corazón de jQuery es la función sobrecargada del mismo nombre, jQuery, que ofrece distinta funcionalidad dependiendo de los parámetros utilizados. Se utiliza normalmente con su alias $.

Retorna un **objeto de tipo jQuery** que contiene los nodos que coinciden con el parámetro pasado.

Dicho objeto es una mezcla entre un Array y un NodeList

Array:

- Tiene la propiedad length. Se utiliza para comprobar si una selección tiene elementos.
- Se puede acceder a sus elementos mediante el operador [*índice*], que devuelve la referencia al objeto Element correspondiente del DOM

NodeList:

- Las modificaciones realizadas sobre los elementos se aplican automáticamente en el navegador

Por ejemplo:

$("p") ----> Retorna un objeto de tipo jQuery con todos los párrafos del documento.

$("p").length ---> indica cuantos párrafos hay en el documento

$("p").eq(0).html() ---> es el contenido html del primer párrafo, lo podemos obtener de esta manera y también modificarlo.

$("p").eq(0).html("Cambio su innerHTML")

***Ejemplo 05***

Aporta tres mejoras importantes:

- Iteración **automática**. Podemos gestionar de una sola vez todos los nodos de un determinado tipo indicado en el selector.

  *$('p').hide():* ocultaría todos los párrafos del documento.

  *$("p").css("background-color", "#eee"):* cambia el color de fondo de todos los párrafos( usando estilos)

  En el DOM teníamos que iterar manualmente a través de los nodos de los objetos de tipo NodeList para poder acceder a sus propiedades y métodos.

- **Concatenación de métodos**. Sobre un objeto jQuery podemos concatenar varios métodos simplemente escribiéndolos separados por puntos.

  Por ejemplo: $('p').addClass('desplegando').slideDown(); asignaría a todos los párrafos del documento la clase 'desplegano' y empezaría a desplegarlos hacia abajo.

  ***Ejemplo 06, Ejemplo 07***

- **Iteración individual**. Sobre cualquier objeto jQuery podemos aplicar el método each(*función(índice,referencia_elemento_DOM*){…}) que ejecuta la función pasada como argumento una vez por cada elemento que contenga el objeto jQuery, enviando a su vez como argumentos a dicha función el índice del elemento sobre el que se está iterando y una referencia al mismo. Dentro de la función, el operador this es una referencia a dicho elemento.

```
$("p").each(function(i){
      if(i%2==0){
            $(this).css("background-color", "#eee");
              }
     else {
              $(this).css("background-color", "#ccc");
      }
     });
```

  this hace referencia al elemento del DOM .

  ***Ejemplo 08***