

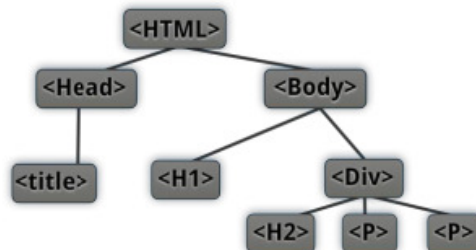
jQuery

jQuery es una librería de Javascript que nos permite simplificar la manera de interactuar con el árbol DOM del documento HTML con el que estemos trabajando en ese momento. Nos permite manejar eventos, desarrollar animaciones e interactuar con la tecnología AJAX con la página web. jQuery es software libre y nos ofrece una serie de funcionalidades desarrolladas en JavaScript, simplificando al programador su trabajo.

Concepto DOM

El **Modelo en Objetos para la Representación de Documentos** ó **DOM** (*Document Object Model*), es un estándar de objetos para representar documentos **HTML** y **XML**. A través de él, los programas pueden acceder y modificar el contenido, estructura y estilo.

La siguiente imagen es una **representación** del **DOM** del documento visto anteriormente:



La imagen anterior muestra el documento **Html** en forma de un árbol. Esta representación es conocida como un árbol **DOM**.

El responsable del DOM es el World Wide Web Consortium ([W3C](http://www.w3.org/)).

ENLACE AL CÓDIGO jQuery

Lo podemos hacer de varias formas, como puede ser enlazarlo en nuestra web como un fichero .js normal, o bien, enlazarlo de forma remota, que es lo que están haciendo últimamente muchos desarrolladores ya que esto asegura el mejor tiempo de carga posible, ya que muchos de los que visiten la web, ya pueden tener la librería cargada en caché, lo que ahorraría tiempo de carga. Esta forma de hacerlo sería la siguiente:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.4.0/jquery.min.js"></script>
```

EJECUTAR EL CÓDIGO CUANDO EL DOM ESTÁ LISTO

Lo primero que nos debemos de asegurar es que el documento está listo para trabajar con él. Con esto nos aseguramos dos cosas. Primero, que el código no se ejecute hasta que el DOM no esté completamente listo, asegurándonos de que todos los elementos existen y están cargados. Lo segundo que nos aseguramos de que el código está separado de HTML y CSS.

Para conseguir esto, el código es el siguiente:

```
$(document).ready(function () {  
  // Todo el código jQuery va aquí  
});
```

SELECCIONANDO ELEMENTOS CON JQUERY

jQuery nos permite seleccionar elementos de nuestro código HTML utilizando el símbolo \$. Veamos algunos ejemplos:

- `$("div");` // Selecciona todos los elementos html
- `$("#myElement");` // Seleccionamos un elemento por medio de su ID
- `$(".myClass");` // Seleccionamos un elemento por medio de su clase.
- `$("p#myElement");` // Seleccionamos elementos P con el id que se le indique
- `$("ul li a.navigation");` // Seleccionamos elementos A con la clase navigation que estén dentro de elementos ul o li.

jQuery también acepta todos los selectores CSS incluso de CSS3. Algunos ejemplos son:

- `$("p > a");` // Selecciona elementos A que son hijos directos de etiquetas P
- `$("input[type=text]");` // Selecciona inputs que tienen un tipo determinado
- `$("a:first");` // Selecciona los primeros A de la página
- `$("li:first-child");` // Cada elemento de la lista que es el primer hijo de una lista

OBTENER ACCESO A NOMBRES DE CLASES CSS

jQuery permite de forma sencilla añadir, quitar, cambiar la clase CSS que utilice un elemento. Veamos algunos ejemplos:

- `$("div").addClass("content");` // Añade la clase content a todos los div
- `$("div").removeClass("content");` // Elimina la clase content de todos los div

MANIPULANDO ESTILOS CSS CON JQUERY

jQuery también nos permite añadir nuevas propiedades CSS a los elementos de una forma sencilla. Veamos algunos ejemplos:

- `$("p").css("width", "400px");` // Añade un ancho a todos las etiquetas P
- `$("#myElement").css("color", "blue");` // Asigna el color azul a los textos del id indicado
- `$("ul").css("border", "solid 1px #ccc");` // Añade border a todas las listas

AÑADIR, ELIMINAR Y ANEXAR LOS ELEMENTOS Y CONTENIDOS

Vamos a ver las distintas formas que nos permite jQuery manipular los elementos incluyendo el contenido de estos elementos.

Obtener el código HTML de cualquier elemento, similar a innerHTML en JavaScript.

- `var myElementHTML = $("#myElement").html();`

Si solo queremos los textos de un elemento

- `var myElementHTML = $("#myElement").text();`

Usando un código similar, podemos cambiar el contenido del HTML o del Texto de un elemento.

- `$("#myElement").html("<p>This is the new content.</p>");`

Añadir contenido a un elemento

- `$("#myElement").append("<p>This is the new content.</p>");`

MANEJO DE EVENTOS CON JQUERY

Para especificar un manejador de eventos podemos utilizar el siguiente código

```
$("#a").click(function() {  
  // hacer algo aquí  
  // cuando se hace clic en cualquier anclaje});
```

El código que vaya dentro de la función, solo se ejecutará cuando se pulse un enlace. jQuery incluye blur, focus, hover, keydown, load, mousemove, resize, scroll, submit, select.

MOSTRAR Y OCULTAR ELEMENTOS CON JQUERY

Las instrucciones para mostrar u ocultar elementos es la siguiente:

```
$("#myElement").hide("slow", function() {  
  //acciones que hace cuando se oculta  
})  
$("#myElement").show("fast", function() {  
  //acciones que hace cuando se muestra  
})  
$("#myElement").toggle(1000, function() {  
  //acciones que hace cuando cambia de estado  
})
```

Toggle lo que permite es cambiar entre los estados dependiendo del valor que se le indique en el primer parámetro

ANIMACIÓN Y EFECTOS CON JQUERY

jQuery nos permite poner elementos animados, parar secuencias a mitad de carga.... Algunos ejemplos son:

```
$("#myElement").slideDown(fast, function() {  
  // hacer algo cuando se deslice hacia abajo  
})  
$("#myElement").slideUp(slow, function() {  
  // hacer algo cuando se deslice hacia arriba  
})  
$("#myElement").slideToggle(1000, function() {  
  // hacer algo cuando se deslice hacia arriba / abajo  
})
```

Para animar un elemento, jQuery lo hace utilizando los estilos CSS pero en vez de ponerlo de golpe, lo que hace es ir poniendo lo forma gradual, dándole así el aspecto de animación.

```
$("#myElement").animate(  
{  
  opacity: .3,  
  width: "500px",  
  height: "700px"  
}, 2000, function() {  
  // de llamada opcional después de la última animación  
});
```