



# Aplicaciones WEB

## jQuery Básico

Oscar Fernando Aristizábal Cardona  
Ingeniero de Sistemas y Telecomunicaciones  
ofaczero@gmail.com

# Lo que usted ya debe saber

Antes de comenzar el estudio de jQuery, usted debe tener un conocimiento básico de:

- HTML
- CSS
- JavaScript

# Qué es jQuery

- jQuery es una librería de funciones JavaScript.
- jQuery simplifica enormemente la programación de JavaScript (Escribir menos - Hacer más).
- jQuery es fácil de aprender.
- La librería jQuery se puede añadir a una página web con una sola línea de marcado.

# Características de la librería jQuery

- ⦿ Selecciones de elementos HTML
- ⦿ Manipulación de elementos HTML
- ⦿ Manipulación CSS
- ⦿ Funciones de eventos HTML
- ⦿ Efectos y Animaciones Javascript
- ⦿ Recorrido y Modificación del DOM HTML
- ⦿ AJAX
- ⦿ Utilidades

# 1. Instalar jQuery

Hay varias formas de usar jQuery en su sitio web:

- ▶ Descargue la librería jQuery de [www.jquery.com](http://www.jquery.com)

Hay dos versiones de jQuery disponibles para la descarga:

**-Versión de producción** - Una versión ligera y comprimida.

**-Versión de desarrollo** - Para pruebas y desarrollo (código sin comprimir y legible)

## 2. Instalar jQuery

- ▶ Incluya jQuery desde un CDN (Content Delivery Network), como Google
- Si no desea descargar y alojar jQuery, puede incluirlo desde un CDN.

```
<head>
<script type="text/javascript" src="http://
ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></
script>
</head>
```

# Adicionar la librería jQuery en una página web

La librería jQuery se almacena en un único archivo JavaScript que contiene todos los métodos de jQuery.

Se puede añadir a una página web con el siguiente marcado:

```
<head>
  <script type="text/javascript" src="jquery.js"></script>
</head>
```

Tenga en cuenta que la etiqueta `<script>` debe estar dentro de la sección `<head>` de la página.

# La función Document Ready

Esta función evita que cualquier código jQuery se ejecute antes de que el documento se termine de cargar.

```
$(document).ready(function(){  
    // Las funciones jQuery van aquí...  
});  
  
// Método corto  
$(function(){  
    // Las funciones jQuery van aquí...  
});
```

# Ejemplo Básico de jQuery

```
<html>
  <head>
    <script type="text/javascript" src="jquery.js"></script>
    <script type="text/javascript">
      $(document).ready(function(){
        $("button").click(function(){
          $("p").hide();
        });
      });
    </script>
  </head>
  <body>
    <h2>Ejemplo Básico jQuery</h2>
    <p>Esto es un Párrafo.</p>
    <p>Esto es otro Párrafo.</p>
    <button>Click me</button>
  </body>
</html>
```

# Sintaxis jQuery

La sintaxis de jQuery está hecha a la medida para la selección de elementos HTML y realizar algunas acciones sobre ellos.

La sintaxis básica es: `$(selector).acción()`

- Un signo de dólar \$ para definir jQuery
- Un (selector) para consultar o encontrar elementos HTML
- Una .acción() para llevar a cabo en el elemento

# Ejemplos Sintaxis jQuery

- **\$ (this).hide ()**
  - Oculta el elemento HTML actual.
  
- **\$ ("#prueba").hide ()**
  - Oculta el elemento con id = "prueba".
  
- **\$ ("p").hide ()**
  - Oculta todos los elementos <p>.
  
- **\$ (.prueba").hide ()**
  - Oculta todos los elementos con class = "prueba".

# 1. Selectores jQuery

Los selectores de jQuery le permiten seleccionar y manipular elementos HTML.

Con los selectores de jQuery se pueden encontrar elementos en función de su id, clase, tipo, atributo, valores de los atributos y muchos más.

Se basa en los actuales selectores CSS, y además, tiene algunos selectores personalizados.

Todo tipo de selectores de jQuery, comienza con el signo de dólar y entre paréntesis: \$( ).

# 2. Selectores jQuery

## Ejemplos:

`$(*)`

`$(".[href]")`

`$(this)`

`$("a[target='_blank'])")`

`$("#div")`

`$("a[target!='_blank'])")`

`$("#prueba")`

`$(".button")`

`$(".color")`

`$("tr:even")`

`$("#p:first")`

`$("tr:odd")`

`$("ul li:last")`

# 1. Eventos jQuery

jQuery está hecho a la medida para controlar eventos.

Los controladores de eventos son métodos que se llaman cuando "algo pasa" en HTML.

Ejemplos:

- Al mover el cursor del ratón sobre un elemento.
- Al seleccionar un botón de opción.
- Al hacer clic en un elemento.

# 2. Eventos jQuery

**Eventos del Ratón: click, dblclick, mouseenter, mouseleave.**

**Eventos del Teclado: keypress, keydown, keyup.**

**Eventos de Formulario: submit, change, focus, blur.**

**Eventos del Documento/Ventana: load, resize, scroll, unload.**

# 3. Eventos jQuery

## Ejemplos:

- `$(document).ready(function)` // Evento ready, cuando el documento está cargado
- `$(selector).click(function)` // Evento click, cuando se presiona un elemento HTML seleccionado.
- `$(selector).dblclick(function)` // Evento dblclick, cuando se presiona rápidamente 2 veces un elemento HTML seleccionado.
- `$(selector).focus(function)` // Evento focus, cuando se enfoca un elemento HTML seleccionado.
- `$(selector).mouseover(function)` // Evento mouseover, cuando se pasa el cursor por encima del elemento HTML seleccionado.

# Efectos jQuery

La librería jQuery ofrece varias técnicas para añadir animación a una página web. Estos incluyen animaciones simples que se utilizan con frecuencia, y la capacidad para elaborar sofisticados efectos personalizados.

- **`$(selector).hide() // Oculta los elementos HTML seleccionados`**
- **`$(selector).show() // Muestra los elementos HTML seleccionados`**
- **`$(selector).toggle() // Oculta y Muestra los elementos HTML seleccionados`**
- **`$(selector).slideDown() // Desplaza hacia abajo los elementos HTML seleccionados`**
- **`$(selector).slideUp() // Desplaza hacia arriba los elementos HTML seleccionados`**
- **`$(selector).slideToggle()// Oculta y Muestra desplazando de arriba hacia abajo los elementos HTML seleccionados`**

# Efectos jQuery

- `$(selector).fadeIn()` // Va mostrando de forma desvanecida los elementos HTML seleccionados
- `$(selector).fadeOut()` // Va ocultando de forma desvanecida los elementos HTML seleccionados
- `$(selector).fadeTo()` // Desvanece hasta cierto valor de opacidad los elementos HTML seleccionados
- `$(selector).fadeToggle()` // Desvanece y muestra los elementos HTML seleccionados
- `$(selector).animate()` // Animación personalizada de los elementos HTML seleccionados
- `$(selector).stop()` // Se utiliza para detener una animación o efecto antes de que se termine.

# Función Callback jQuery

Las sentencias de JavaScript se ejecuta línea a línea.

Sin embargo, cuando se utilizan animaciones, la siguiente línea de código se puede ejecutar a pesar de que la animación no ha finalizado, esto puede provocar errores.

Para evitar estos errores, se puede utilizar una función **callback**.

Una función **callback** se ejecuta después que la animación actual (efecto) ha finalizado.

```
$( "p" ).hide(2000, function(){
    alert("EL párrafo ha desaparecido");
});
```

# Método de Encadenamiento jQuery

Existe una técnica llamada encadenamiento, que nos permite ejecutar varios comandos con jQuery, uno tras otro, en el mismo elemento(s).

Para encadenar una acción, sólo tiene que añadir la acción a la acción anterior.

```
$("#p1").css("color","blue").slideUp(2000).slideDown(2000);
```

# 1. Manipulación DOM - jQuery

## Métodos GET/SET:

Manipulación DOM (**Document Object Model**)

Una parte muy importante de jQuery es la posibilidad de manipular el DOM.

jQuery contiene unos poderosos métodos (funciones) para cambiar y manipular los elementos y atributos HTML.

**text( )** - Establece o devuelve el contenido de texto de los elementos seleccionados.

**html( )** - Establece o devuelve el contenido de los elementos seleccionados (incluyendo el formato HTML).

**val( )** - Establece o devuelve el valor de los campos del formulario.

**attr()** - Se utiliza para obtener valores de atributos.

# 2. Manipulación DOM - jQuery

## Ejemplo de métodos GET:

```
$("#btn1").click(function(){
    alert("Texto: " + $("#caja").text());
});
```

```
$("#btn2").click(function(){
    alert("HTML: " + $("#caja").html());
});
```

```
$("#btn3").click(function(){
    alert("Valor: " + $("#campo").val());
});
```

```
$("#btn4").click(function(){
    alert($(".enlace").attr("href"));
});
```

# 3. Manipulación DOM - jQuery

## Ejemplo de métodos SET:

```
$("#btn1").click(function(){
    $("#caja1").text("Hola Mundo!");
});

$("#btn2").click(function(){
    $("#caja2").html("<b><em>Hola</em> mundo!</b>");
});

$("#btn3").click(function(){
    $("#campo").val("Valor nuevo");
});

$("#btn4").click(function(){
    $(".enlace").attr("href", "http://jquery.com");
});
```

# 4. Manipulación DOM - jQuery

## Añadir nuevos elementos/contenidos HTML:

**append( )** - Inserta contenido en el extremo de los elementos seleccionados.

**prepend( )** - Inserta contenido al principio de los elementos seleccionados.

**after( )** - Inserta el contenido después de los elementos seleccionados.

**before( )** - Inserta el contenido antes de que los elementos seleccionados.

```
$( "ol" ).append( "<li>Lista</li>" );
```

```
$( ".texto" ).prepend( "Hola." );
```

```
$( "#caja" ).after( "Texto Despues" );
```

```
$( "#caja" ).before( "Texto Antes" );
```

# 5. Manipulación DOM - jQuery

**Remover elementos/contenidos HTML:**

**remove( )** - Elimina el elemento seleccionado (y sus elementos secundarios)

**empty( )** - Elimina los elementos secundarios del elemento seleccionado

```
$("#div1").remove();
```

```
$("#div1").empty();
```

# Manipulación CSS - jQuery

## Métodos GET/SET:

**addClass( )** - Añade una o más clases a los elementos seleccionados.

**removeClass( )** - Elimina una o más clases de los elementos seleccionados.

**toggleClass( )** - Cambia entre la adición / eliminación de las clases de los elementos seleccionados.

**css ( )** - Establece o devuelve el atributo de estilo.

- **css (nombre)** - Retorna el valor de la propiedad CSS
- **css (nombre, valor)** - conjunto de propiedades CSS y el valor
- **css ({propiedades})** - Establecer múltiples propiedades CSS y valores

# 1. Dimensiones - jQuery

jQuery tiene varios métodos para trabajar dimensiones.

- **width()** // Retorna el ancho sin bordes, margen y relleno.
- **height()** // Retorna el alto sin bordes, margen y relleno.
- **innerWidth()** // Retorna el ancho con relleno.
- **innerHeight()** // Retorna el alto con relleno.
- **outerWidth()** // Retorna el ancho con relleno y bordes.
- **outerHeight()** // Retorna el alto con relleno y bordes.

# 2. Dimensiones - jQuery

Ejemplos:

```
$("button.btn1").click(function(){
    var txt = "";
    txt += "Ancho: " + $("#caja").width() + "</br>";
    txt += "Alto: " + $("#caja").height();
    $("#caja").html(txt);
```

```
$("button.btn2").click(function(){
    var txt = "";
    txt += "Documento ancho/alto: " + $(document).width();
    txt += "/" + $(document).height() + "\n";
    txt += "Ventana ancho/alto: " + $(window).width();
    txt += "/" + $(window).height();
    alert(txt);
});
```

# 1. AJAX - jQuery

## ¿Qué es Ajax?

Asincrono JavaScript y XML.

En definitiva, se trata de cargar datos en segundo plano y mostrarlos en la página web, Sin recargar la página web.

Ejemplos de aplicaciones que usan Ajax: Gmail, Google Maps, Youtube, Facebook.

jQuery ofrece varios métodos para la funcionalidad de Ajax.

Con los métodos de jQuery ajax, puede solicitar texto, HTML, XML y JSON desde un servidor remoto a través de HTTP GET y HTTP POST.

Usted podrá cargar los datos externos directamente en el elemento seleccionado HTML de su página web!

# 2. AJAX - jQuery

## Método load()

**load()** es un método de AJAX simple, pero poderoso.

**load()** carga los datos de un servidor y pone los datos devueltos en el elemento seleccionado.

### Sintaxis:

```
$(selector).load(URL,data,callback);
```

El parámetro **URL** Especifica la dirección URL que desee cargar.

El parámetro opcional **data** especifica un conjunto de querystring clave/valor a enviar junto con la solicitud.

El parámetro opcional **callback** es el nombre de una función a ejecutar después de que el método **load()** se ha completado.

# 3. AJAX - jQuery

Ejemplos:

```
$("#contenedor").load("documento.txt");
```

```
$("#contenedor").load("documento.txt #p1");
```

```
$("#contenedor").load("documento.txt",function(respuesta,estado,xhr){  
    if(estado == "success")  
        alert("El contenido se ha cargado!");  
    if(estado == "error")  
        alert("Error: "+xhr.status+": "+xhr.statusText);  
});
```

# 4. AJAX - jQuery

## Métodos GET/POST:

Los métodos **get()** y **post()** se utilizan para solicitar los datos del servidor con una solicitud HTTP GET o POST.

Los métodos que son comúnmente utilizados para la petición-respuesta entre un cliente y un servidor son: **GET** y **POST**.

### **GET:**

Se utiliza básicamente para conseguir (recuperar) algunos datos del servidor.  
Nota: El método GET puede devolver datos almacenados en caché.

### **POST:**

También se puede usar para obtener datos desde el servidor. Sin embargo, nunca el método POST almacena en memoria caché los datos, y con frecuencia se utiliza para enviar datos junto con la solicitud.

# 5. AJAX - jQuery

## Método `$.get()`:

El método `$.get()` solicita datos del servidor con una solicitud HTTP GET.

### Sintaxis:

```
$.get(URL, callback);
```

El parámetro **URL** Especifica la dirección URL que desea solicitar.

El parámetro opcional **callback** es el nombre de una función que se ejecutará si la solicitud se realiza correctamente.

### Ejemplo:

```
$(".btn").click(function(){
    $.get("prueba.php", function(data, status){
        alert("Datos: " + data + "\nEstado: " + status);
    });
});
```

# 6. AJAX - jQuery

## Método `$.post()`:

El método `$.post()` solicita datos del servidor con una solicitud HTTP POST.

### Sintaxis:

```
$.post(URL ,data ,callback);
```

El parámetro **URL** Especifica la dirección URL que desea solicitar.

El parámetro opcional **data** especifica algunos datos para enviar junto con la solicitud.

El parámetro opcional **callback** es el nombre de una función que se ejecutará si la solicitud se realiza correctamente.

# 7. AJAX - jQuery

Ejemplo Método `$.post()`:

```
$(".btn").click(function(){
    $.post("prueba.php",
    {
        nombre:"Jose Eustasio Rivera",
        ciudad:"Medellin"
    },
    function(data,status){
        alert("Datos: " + data + "\nEstado: " + status);
    });
});

<?php
// prueba.php
$nom = $_POST['nombre'];
$ciu = $_POST['ciudad'];
echo
echo "Bienvenido: ".$nom;
echo "A la ciudad de:". $ciu;
?>
```

completar: ajax, delay