**INTRODUCCIÓN**

jQuery es una biblioteca JavaScript la cual nos va a simplificar algunas funciones con lo que su propósito es facilitar el uso de JavaScript en su sitio web.

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

jQuery también simplifica muchas de las cosas complicadas de JavaScript, como la manipulación DOM.

La biblioteca jQuery contiene las siguientes características:

* Manipulación HTML / DOM
* Manipulación CSS
* Métodos de eventos HTML
* Efectos y animaciones

**COMO INSTALAR**

Podemos hacerlo de dos maneras:

* Descargando desde la página : <https://jquery.com/download/>
* Usando de manera online: script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

Al estar en el servidor de google debería ir más rápido que el nuestro.

**Sintaxis de jQuery**

La sintaxis de jQuery está hecha a medida para **seleccionar** elementos HTML y realizar alguna **acción** en los elementos.

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

* Un signo $ para definir / acceder a jQuery
* Un ( *selector* ) para "consultar (o buscar)" elementos HTML
* Una *acción* jQuery () que se realizará en los elementos

**El evento Document Ready**

Para llamar a los métodos jQuery utilizamos un evento de preparación en nuestro codigo:

$(document).ready(function(){  
  
  *// metodos de jQuery*   
});

Esto es para evitar que se ejecute el código jQuery antes de que el documento termine de cargarse (esté listo).

Es una buena práctica esperar a que el documento esté completamente cargado y listo antes de trabajar con él. Esto también le permite tener su código JavaScript antes del cuerpo de su documento, en la sección de encabezado.

A continuación, se muestran algunos ejemplos de acciones que pueden fallar si se ejecutan métodos antes de que el documento esté completamente cargado:

* Intentando ocultar un elemento que aún no se ha creado
* Intentando obtener el tamaño de una imagen que aún no está cargada

## Selectores de jQuery

## Podemos encontrar cuatro tipos de selectores:

## Etiquetas

El selector de etiquetas de jQuery selecciona elementos en función del nombre de la etiqueta.

Ejemplo:

## $(document).ready(function(){   $("button").click(function(){     $("p").hide();   }); });

## Identificador

jQuery usa el atributo id de una etiqueta HTML para encontrar el elemento específico. #*id*

Una identificación debe ser única dentro de una página, por lo que debe usar el selector #id cuando desee encontrar un elemento único y único.

Para encontrar un elemento con una identificación específica, escriba un carácter hash, seguido por la identificación del elemento HTML:

$(document).ready(function(){  
  $("button").click(function(){  
    $("#test").hide();  
  });  
});

## Clase

JQuery busca elementos con la clase indicada.

Para buscar elementos con una clase específica, escriba un carácter de punto, seguido del nombre de la clase:

$(document).ready(function(){  
  $("button").click(function(){  
    $(".test").hide();  
  });  
});

**Métodos de eventos de jQuery**

Todas las acciones de los diferentes visitantes a las que puede responder una página web se denominan eventos.

Un evento representa el momento preciso en que sucede algo.

Ejemplos:

* moviendo el mouse sobre un elemento
* seleccionando un botón de radio
* haciendo clic en un elemento

El término **"activación"** se usa a menudo con eventos. Ejemplo: "El evento de pulsación de tecla se activa en el momento en que pulsa una tecla".

A continuación, se muestran algunos eventos DOM habituales:

**$ (documento) .ready ()**

El $(document).ready()método nos permite ejecutar una función cuando el documento está completamente cargado.

**click()**

La función se ejecuta cuando el usuario hace clic en el elemento HTML.

El siguiente ejemplo dice: Cuando un evento de clic se dispara en un <p>elemento; ocultar el <p>elemento actual:

$("p").click(function(){  
  $(this).hide();  
});

**dblclick ()**

La función se ejecuta cuando el usuario hace doble clic en el elemento HTML:

$("p").dblclick(function(){  
  $(this).hide();  
});

**mouseenter ()**

La función se ejecuta cuando el puntero del mouse pasa por encima del elemento HTML:

$("#p1").mouseenter(function(){  
  alert("You entered p1!");  
});

**mouseleave ()**

La función se ejecuta cuando el puntero del mouse sale del elemento HTML:

$("#p1").mouseleave(function(){  
  alert("Bye! You now leave p1!");  
});

**mousedown ()**

La función se ejecuta cuando se presiona el botón izquierdo, medio o derecho del mouse, mientras el mouse está sobre el elemento HTML:

$("#p1").mousedown(function(){  
  alert("Mouse down over p1!");  
});

**mouseup ()**

La función se ejecuta, cuando se suelta el botón izquierdo, medio o derecho del mouse, mientras el mouse está sobre el elemento HTML.

**hover ()**

El hover()método son dos funciones y es una combinación de los métodos mouseenter()y mouseleave().

La primera función se ejecuta cuando el mouse pasa por encima al elemento HTML, y la segunda función se ejecuta cuando el mouse sale del elemento HTML:

**focus()**

La función se ejecuta cuando el campo del formulario se selecciona:

$("input").focus(function(){  
  $(this).css("background-color", "#cccccc");  
});

**hide() y show();**

Sintaxis:

$(*selector*).hide(*speed,callback*);  
  
$(*selector*).show(*speed,callback*);

$("#hide").click(function(){  
  $("p").hide();  
});  
  
$("#show").click(function(){  
  $("p").show();  
});

**fadein()**

$(*selector*).fadeIn(*speed,callback*);

El parámetro de velocidad opcional especifica la duración del efecto. Puede tomar los siguientes valores: "lento", "rápido" o milisegundos.

El parámetro de entrada de llamada opcional es una función que se ejecutará después de que se complete el desvanecimiento.

El siguiente ejemplo demuestra el fadeIn()método con diferentes parámetros:

$("button").click(function(){  
  $("#div1").fadeIn();  
  $("#div2").fadeIn("slow");  
  $("#div3").fadeIn(3000);  
});

**fadeout()**

$(*selector*).fadeOut(*speed,callback*);

El parámetro de velocidad opcional especifica la duración del efecto. Puede tomar los siguientes valores: "lento", "rápido" o milisegundos.

El parámetro de entrada de llamada opcional es una función que se ejecutará después de que se complete el desvanecimiento.

El siguiente ejemplo demuestra el fadeOut()método con diferentes parámetros:

$("button").click(function(){  
  $("#div1").fadeOut();  
  $("#div2").fadeOut("slow");  
  $("#div3").fadeOut(3000);  
});

**slidedown()**

$(*selector*).slideDown(*speed,callback*);

El parámetro de velocidad opcional especifica la duración del efecto. Puede tomar los siguientes valores: "lento", "rápido" o milisegundos.

El parámetro de devolución de llamada opcional es una función que se ejecutará después de que se complete el deslizamiento.

El siguiente ejemplo demuestra el slideDown()método:

$("#flip").click(function(){  
  $("#panel").slideDown();  
});

**slideup()**

$(*selector*).slideUp(*speed,callback*);

El parámetro de velocidad opcional especifica la duración del efecto. Puede tomar los siguientes valores: "lento", "rápido" o milisegundos.

El parámetro de entrada de llamada opcional es una función que se ejecutará después de que se complete el deslizamiento.

El siguiente ejemplo demuestra el slideUp()método:

$("#flip").click(function(){  
  $("#panel").slideUp();  
});

<style>

#panel, #flip {

padding: 5px;

text-align: center;

background-color: #e5eecc;

border: solid 1px #c3c3c3;

}

#panel {

padding: 50px;

display: none;

}

</style>

</head>

<body>

<div id="flip">Haz clic para ver el panel</div>

<div id="panel">Se muestra panel</div>

</body>

**animate()**

$(*selector*).animate({*params*}*,speed,callback*);

El parámetro params define las propiedades CSS que se animarán.

El parámetro de velocidad opcional especifica la duración del efecto. Puede tomar los siguientes valores: "lento", "rápido" o milisegundos.

El parámetro de entrada de llamada opcional es una función que se ejecutará después de que se complete la animación.

El siguiente ejemplo demuestra un uso simple del animate()método; mueve un elemento <div> hacia la derecha, hasta que alcanza una propiedad izquierda de 250px:

Podemos introducir más propiedades como por ejemplo

 left: '250px',  
 opacity: '0.5',  
 height: '150px',  
 width: '150px'

**stop()**

Detiene animaciones o efectos antes de que finalice.

$(*selector*).stop(*stopAll,goToEnd*);

El parámetro opcional stopAll especifica si también se debe borrar la cola de animación o no. El valor predeterminado es falso, lo que significa que solo se detendrá la animación activa, lo que permitirá que las animaciones en cola se realicen posteriormente.

El parámetro opcional goToEnd especifica si se completa o no la animación actual inmediatamente. El valor predeterminado es falso.

Entonces, de forma predeterminada, el stop()método elimina la animación actual que se realiza en el elemento seleccionado.

El siguiente ejemplo demuestra el stop()método, sin parámetros:

$("#stop").click(function(){  
  $("#panel").stop();  
});

**Encadenar Jquery**

Permite ejecutar múltiples comandos de jQuery, uno tras otro, en el mismo elemento (s) con los navegadores no tienen que encontrar el mismo elemento (s) más de una vez.

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

## jQuery DOM HTML

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

jQuery viene con un montón de métodos relacionados con DOM que facilitan el acceso y la manipulación de elementos y atributos.

Tres métodos jQuery simples pero útiles para la manipulación de DOM son:

* text() - Establece o devuelve el contenido de texto de los elementos seleccionados
* html() - Establece o devuelve el contenido de los elementos seleccionados (incluido el marcado HTML)
* val() - Establece o devuelve el valor de los campos del formulario.

## $("#btn1").click(function(){   alert("Text: " + $("#test").text()); }); $("#btn2").click(function(){   alert("HTML: " + $("#test").html()); });

Si introducimos un valor a texto podemos cambiarlo de valor

$("#test").text("texto nuevo")

Para los atributos si queremos obtener el valor lo haremos con attr ()

$("button").click(function(){  
  alert($("#objeto").attr("href"));  
});

Si introducimos un valor al atributo podemos cambiarlo de valor

## Agregar elementos

Usaremos los siguientes métodos

* append() - Inserta contenido al final de los elementos seleccionados

$("p").append("Texto de ayuda");

Podemos agregar varios elementos de una vez:

$("p").append("Texto de ayuda",("Texto de ayuda2",("Texto de ayuda3");

* prepend() - Inserta contenido al comienzo de los elementos seleccionados
* $("p").append("Texto de ayuda");

## Eliminar elementos

El remove()método jQuery elimina los elementos seleccionados y sus elementos secundarios.

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

Mientras el empty() elimina los elementos secundarios de los elementos seleccionados.

## Metodo CSS

El css()establece o devuelve una o más propiedades de estilo para los elementos seleccionados.

Para devolver el valor de una propiedad CSS especificada, utilizamos la siguiente sintaxis:

css("*propertyname*");

El siguiente ejemplo devolverá el valor de color de fondo del PRIMER elemento coincidente:

$("p").css("background-color");

Para establecer propiedades en CSS, utilizamos la siguiente sintaxis:

css({"*propertyname*":"*value*","*propertyname*":"*value*",...});

El siguiente ejemplo establecerá un color de fondo y un tamaño de fuente para TODOS los elementos coincidentes:

$("p").css({"background-color": "yellow", "font-size": "200%"});