

JQUERY

- `$(document).ready(function { --codigo aquí-- });` Esta función se utiliza cuando se carga la página.
- `$("#nombreElemento")` -> Esta sintaxis se usa para seleccionar un elemento por id de la página html.
- `$(".nombreClase")` -> Esta sintaxis se usa para seleccionar un elemento por clase de la página html.
- `$("#nombreElemento.nombreClase")` -> Esta sintaxis se usa para seleccionar un elemento por id y por clase para algo más específico de la página html.
- `$("#nombreElemento").css("propiedad","valor")` -> Se utiliza para darle propiedades al objeto por css.
- `$("#nombreElemento").remove();` -> Elimina el elemento de la página.
- `$("p")` -> Esta sintaxis se usa para seleccionar un elemento por etiqueta de la página html.
- `$("input[type='text']")` -> Esta sintaxis se usa para seleccionar un elemento por atributo de la página html.

Ejemplos: `$("input[type='text']")`

`$("[type='text']")`

`$("[type]")`

`$("[type^=abc]")` -> Lo caracteres que inicia el valor del atributo

`$("[type$=abc]")` -> Lo caracteres que terminan el valor del atributo

- `$(".men .tit")` -> Seleccionar por jerarquía. Estamos seleccionando el "Hola" o lo que esta dentro del p, o para seleccionar los hijos inmediatos `$(".men > .tit")`

Ejemplo: `<div class="men">`

`<p class="tit">Hola</p>`

`</div>`

- `$("div+div")` -> Para seleccionar un div precedido de otro div.
- `$("*")` -> Selector universal.
- `$("*", "body")` -> Selector universal dentro de una etiqueta.
- Ejemplo construcción de una tabla

```
<table>
  <thead>
    <tr>
      <th></th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td></td>
    </tr>
  </tbody>
</table>
```

- `$("tbody tr")` -> Selecciona el cuerpo de la tabla.
- `$("tbody tr:first")` -> Selecciona el primer elemento del cuerpo de la tabla.
- `$("tbody tr:last")` -> Selecciona el ultimo elemento del cuerpo de la tabla.
- `$("tbody tr:even")` -> Selecciona los elemento de índice par del cuerpo de la tabla.
- `$("tbody tr:odd")` -> Selecciona los elemento de índice impar del cuerpo de la tabla.
- `$("tbody tr:eq(2)")` -> Selecciona el tercer elemento del cuerpo de la tabla.
- `$("tbody tr:gt(1)")` -> Selecciona el tercer elemento y hacia adelante del cuerpo de la tabla.
- `$("tbody tr:lt(2)")` -> Selecciona el segundo elemento y hacia atrás del cuerpo de la tabla.

- `$("td:first-child")` -> Selección primera columna del cuerpo de la tabla.
- `$("td:last-child")` -> Selección última columna del cuerpo de la tabla.
- `$("td:nth-child(2)")` -> Selección la segunda columna del cuerpo de la tabla.
- `$("td:nth-child(even)")` -> Selección las columnas pares del cuerpo de la tabla.
- `$("td:nth-child(odd)")` -> Selección las columnas impares del cuerpo de la tabla.
- `$(":enabled")` -> Selecciona los que tienen esa propiedad.

Ejemplos: : text

 : checkbox

 : checked

 : button

 : hidden

- `$("input:not(:password)")` -> Selecciona los elementos input que no tenga la propiedad password.
- `$("div,p")` -> Selección de varios elementos separados por comas. Se pueden mezclar clases, ids y elementos.
- Se pueden anidar varias propiedades en una sola línea
Ejemplo: `$("#div1").css("color","red").css("background","blue")`
- `$("span").eq(1)` -> Retorna el 2 elemento de varios span
- Todos los ejemplos anteriores no envían un elemento jquery ahora si queremos que nos retorne un elemento html usamos la palabra `get()`.
Ejemplo: `$("span").get(0).innerHTML` -> Retorna la cadena de texto dentro del span
- `$("span").last().css({color:"red", border:"1px solid black"})` -> Se pueden asignar varias propiedad al mismo tiempo separados por comas.
- `$("#span3").index("span")` -> Se obtiene el índice del span3 dentro de todos los elementos span
- `$("#lista > li").index($("#elemento2"))` -> De una lista llamada #lista se obtiene el index del hijo #elemento2.
- `$(this)` -> Cuando doy click en un elemento esta línea de código obtiene este elemento seleccionado.
- `$(this).parent()` -> Selecciona el elemento padre.
Ejemplo: Si selección una posición de una tabla, va a obtener toda la fila donde se encuentra el elemento.
- `.click(function(){})` -> Cuando se da click sobre un objeto.
- `.hover(function(){})` -> Cuando se pasa el mouse por encima de un objeto.
- En el siguiente ejemplo trabajamos con el hover para ejecutar una función para colorear una fila de una tabla, y vemos que al final tenemos otra función que funciona como un else, para quitarle el color.

```
$(".valor_columna_1").hover(function(){
    $(this).parent().css("background-color", "red");
}, function(){
    $(this).parent().css("background-color", "");
});
```

- `$("tr").children("td")` -> Selecciona todos los hijos de la fila, el td es opcional por si tenemos otros elementos diferentes de td. Children solo funciona con los primeros hijos inmediatos de un objeto.
- `$("#divExterno").find("p")` -> Selecciona todos los elementos p dentro del divExterno. Este selección todos los descendientes.
- `$("#divExterno").siblings("p")` -> Retorna todos los hermanos del divExterno que tenga la propiedad p
- `$("#divExterno").prev()` -> Selecciona el elemento anterior al seleccionado.
- `$("#divExterno").next()` -> Selecciona el elemento posterior al seleccionado.

- `$("#divExterno").prevall("p")` -> Selecciona los elementos anteriores al seleccionado. En este caso se seleccionan todos los tipo p. Si quito las comillas p, selecciona todo.
- `$("#divExterno").nextall("div")` -> Selecciona los elementos posteriores al seleccionado. En este caso se seleccionan todos los tipo div
- `$("p").not("")` -> Selecciona todos excepto el que tenga las propiedades dentro de los paréntesis. Para el siguiente ejemplo se crea una función y se escoge todos los p que no tengan el texto "p2"

```
<body>
  <h1>Programando en jQuery!</h1>
  <h2>Módulo 2 - Colecciones en jQuery</h2>

  <p name="p-primera">p1</p>
  <p name="p-segunda">p2</p>
  <p name="p-tercera">p3</p>
  <p name="p-cuarta">p4</p>

  <script type="text/javascript">
    // Esta función se ejecuta cuando el DOM termina de cargar
    $(function(){

      // $("p").not("[name$=tercera]").css("color", "red");

      $("p").not(function(index){
        return $(this).text() === "p2";
      }).css("color", "blue");

    });
  </script>
```

- `$("p").filter("")` -> Selecciona todos los elementos que tenga la coincidencia. Es el opuesto de not del ejemplo anterior.

```
<body>
  <h1>Programando en jQuery!</h1>
  <h2>Módulo 2 - Colecciones en jQuery</h2>

  <p name="p-primera">p1</p>
  <p name="p-segunda">p2</p>
  <p name="p-tercera">p3</p>
  <p name="p-cuarta">p4</p>

  <script type="text/javascript">
    // Esta función se ejecuta cuando el DOM termina de cargar
    $(function(){

      // $("p").filter("[name$=tercera]").css("color", "red");

      $("p").filter(function(index){
        return $(this).text() === "p2" || index === 3;
      }).css("color", "blue");

    });
  </script>
```

- `.has` -> Selecciona el elemento que cumple con la condición.

```
<div>
  <input type="button" value="boton 1" />
</div>
<br />
<div>
  <input type="text"/>
</div>

<script type="text/javascript">
  // Esta función se ejecuta cuando el DOM termina de cargar
  $(function(){
    $("div").has("input[type=button]").css("border", "2px dashed black");
  });
</script>

</body>
```

- .map -> Tomar elementos y crear la función, .join unir elementos.

```
<body>
  <h1>Programando en jQuery!</h1>
  <h2>Módulo 2 - Colecciones en jQuery</h2>

  <p>felipe</p>
  <p>gavilan</p>
  <p>whatever</p>

  <input id="resultado" type="text">

  <script type="text/javascript">
    $(function(){
      var texto = $("p").map(function(index){
        return $(this).text().substring(0,1);
      }).get().join(",");

      $("#resultado").val(texto);

    });
  </script>
</body>
```

felipe
gavilan
whatever
f,g,w

```
<body>
  <h1>Programando en jQuery!</h1>
  <h2>Módulo 2 - Colecciones en jQuery</h2>

  <input id="resultado" type="text">

  <script type="text/javascript">
    $(function(){
      var arreglo = [2,3,4,5,6];

      var texto = $.map(arreglo, function(item, indice){
        return item.toString() + " " + indice;
      }).join(",");

      $("#resultado").val(texto);

    });
  </script>
```

2,0,3 1,4,2,5 3,6 4

- .grep -> Selecciona los elementos del arreglo que cumplan con la condición del return.

```
<body>
  <h1>Programando en jQuery</h1>
  <h2>Módulo 2 - Colecciones en jQuery</h2>

  <input type="text" id="resultado" />

  <script type="text/javascript">
    $(function(){
      var arr = [1,2,3,4,5,6,7,8,9,10];

      var arrFiltrado = $.grep(arr, function(el, index){
        return el < 5;
      });

      $("#resultado").val(arrFiltrado.join(","));

    });
  </script>
</body>
```

1,2,3,4

- .each -> Sirve para realizar iteraciones

```
$(function(){
    $("#enviar").click(function(){
        var items = [];

        // Iteramos cada fila de la tabla
        $("tbody tr").each(function(el){

            var itemOrden = {};

            var tds = $(this).find("td");
            itemOrden.Id = tds.filter(":eq(0)").text();
            itemOrden.Descripcion = tds.filter(":eq(1)").text();
            itemOrden.Precio = parseFloat(tds.filter(":eq(2)").text());
            itemOrden.Envolver = tds.find(":checkbox").prop("checked");

            // metemos el objeto itemOrden en el arreglo
            items.push(itemOrden);

        });

        var Orden = {};

        Orden.items = items;

        var total = 0;

        // Usamos $.each para iterar un arreglo común y corriente
        $.each(items, function(index, value){
            total += value.Precio;
        });

        // toFixed(2) sirve para redondear a dos cifras decimales
        Orden.total = total.toFixed(2);

        //console.log(Orden);

        // Cuando enviamos mensajes a un servidor, lo hacemos en formato JSON
        var data = JSON.stringify(Orden);

        console.log(data);
    });
});
```

- .append -> Sirve para agregar después de un elemento otro elemento, y adentro del elemento. O mover elementos existentes
- .prepend -> Sirve para agregar antes de un elemento otro elemento, y adentro del elemento. O mover elementos existentes
- .appendTo -> Sirve igual que el append pero se invierte el orden de los elementos. Solo funciona con elementos html.

```
$("#destino").append($("#mover"));
$("#mover").appendTo("#destino");
```

- .prependTo -> Sirve igual que el prepend pero se invierte el orden de los elementos. Solo funciona con elementos html.
- Ejemplo agregar div por jquery

```
<script type="text/javascript">
$(function(){

    $("<div>", {class: "mensaje", 'data-id': 45}).append("hola").appendTo("body");

    $(".mensaje").css("color", "Red");

});
</script>
```

- Ejemplo agregar elementos en una tabla

```

<body>
  <h1>Programando en jQuery</h1>
  <h3>Módulo 3: jQuery y HTML</h3>

  <table id="miTabla">
    <thead>
      <tr>
        <td>header 1</td>
        <td>header 2</td>
      </tr>
    </thead>
    <tbody>
    </tbody>
  </table>

  <script type="text/javascript">
    $(function(){

      var miObjeto1 = {valor1: "A", valor2: "B"};
      var miObjeto2 = {valor1: "C", valor2: "D"};
      var miObjeto3 = {valor1: "E", valor2: "F"};
      var miObjeto4 = {valor1: "G", valor2: "H"};
      var miArreglo = [];
      miArreglo.push(miObjeto1);
      miArreglo.push(miObjeto2);
      miArreglo.push(miObjeto3);
      miArreglo.push(miObjeto4);

      for(var i =0; i < miArreglo.length; i++){
        var $td1 = $("<td>").append(miArreglo[i].valor1);
        var $td2 = $("<td>").append(miArreglo[i].valor2);

        $("<tr>").append($td1).append($td2).appendTo("tbody");
      }

    });
  </script>

```

Módulo 3: jQuery y HTML

header 1	header 2
A	B
C	D
E	F
G	H

- .after -> Adiciona el elemento después y por fuera del elemento seleccionado.
- .before -> Adiciona el elemento antes y por fuera del elemento seleccionado.
- .clone() -> Clona un elemento y lo colocamos donde deseamos.
Ejemplo: `var btn = $("[name=miboton]").clone();`
`$(“table”).after(btn);`
- .wrap -> Meter un elemento dentro de otro elemento. Lo envuelve.
Ejemplo: `$(“span”).wrap("<div>");`
- .wrapall -> Meter todos los elemento deseados dentro de otro elemento. Lo envuelve.
Ejemplo: `$(“span,input”).wrapall("<div>");`
- .unwrap -> Saca todos los elemento deseados de otro elemento. Lo envuelve.
- .remove() -> Elimina un elemento de la página.
- .empty() -> Vacía el contenido de un elemento de la página.
- .show() -> Muestra un elemento oculto.
- .hide() -> Oculta un elemento.
- .toggle -> funciona como el show y el hide. Pero cambia el estado de la visibilidad de un objeto en una sola línea.
- .toggleClass -> Cambia el estado de una clase de un objeto. Le agrega la clase o se la quita.
- Agregar, editar o eliminar atributos de un elemento

```

$(“input”).attr(“miAtributo2”, “456”);
$(“input”).attr(“miAtributo”, “nuevo valor”);
$(“input”).removeAttr(“miAtributo”);

```

- .prop -> Sirve para trabajar con atributos de elementos.
- Selección un checkbox.
Ejemplo: `$(“input[type=checkbox]”).prop(“checked”,true);`

- Habilitar o deshabilitar un botón.
Ejemplo: `$("input[type=button"]').prop("disabled",true);`
- `.addClass` -> Para adicionar clases a un elemento.
- `.hasClass` -> Para saber si tiene una clase.
- `.removeClass` -> Para remover una clase.
- El primer ejemplo crea divs a partir de solo el primero, el segundo ejemplo crea los divs sin importar que no hayan sido creados al cargar la página, cosa que no pasa con el primero. Esto se hace con la función **.on**

```
<script type="text/javascript">
    $(function(){
        /*
        $("div").click(function(){
            var cantidadDeDivs = $("div").length;
            var numeroDelDivACrear = cantidadDeDivs + 1;
            $("div").append("div " + numeroDelDivACrear).appendTo("body");
        });
        */

        $(document).on("click", "div", function(){
            var cantidadDeDivs = $("div").length;
            var numeroDelDivACrear = cantidadDeDivs + 1;
            $("div").append("div " + numeroDelDivACrear).appendTo("body");
        });
    });
</script>
```

- `.off` -> Sirve para que finalice la lógica del ejemplo anterior.

```
if (numeroDelDivACrear === 5){
    $(document).off("click", "div");
}
```

- `.one` -> Sirve para que el evento se ejecute solo una vez.

```
<script type="text/javascript">
    $(function(){
        $(document).one("click", "div", function(){
            var cantidadDeDivs = $("div").length;
            var numeroDelDivACrear = cantidadDeDivs + 1;

            $("div").append("div " + numeroDelDivACrear).appendTo("body");
        });
    });
</script>
```

- `.trigger` -> Sirve para ejecutar un evento de otro elemento.

```
$("#boton").click(function(){
    $("div").trigger("click");
});
```

NOTA: Para este caso, como se crean nuevos div este método se aplica para todos, por lo tanto tendríamos que usar el código de la siguiente manera `$("div").first().trigger("click");` ó `$("div").first().click();`

- `.submit()` -> Para ejecutar un botón que esta fuera del form
Ejemplo: `$(form).submit();`
- `.click()` -> Dispara un evento cuando se da click sobre un elemento html.

```
$("#enviar").click(function(){
    $("#formPersona").submit();
});

$("#limpiar").click(function(){
    $("#formPersona input[type=text]").val("");
});
```

- .blur() -> Cuando tengo el focus en un textbox y doy click afuera del elemento se dispara.
- .change() -> Cuando cambia el valor de un elemento.

```

$("#textbox").change(function(){
    //alert("texto cambiado");
});

$("#checkbox").change(function(){
    if ($("#checkbox").is(":checked")){
        $("#seleccionado").text("checkbox seleccionado");
    }
    else{
        $("#seleccionado").text("checkbox no seleccionado");
    }
});

$("#tipoDocumento").change(function(){
    var valorSeleccionado = $(this).children(":selected").text();
    $("#textbox").val(valorSeleccionado);
    $("#textbox").change();
});

```

- .dblclick -> Dispara un evento cuando se da doble click sobre un elemento html.
- .focusin -> Dispara un evento cuando se hace focus sobre un elemento.
- .focusout -> Dispara un evento cuando se hace focus sobre otro elemento diferente al actual.
- .hover() -> Dispara un evento cuando se pasa el mouse por encima de un elemento.
- .keydown -> Dispara un evento cuando se presiona una tecla.
- .keyup -> Dispara un evento cuando se suelta una tecla.