Documento de Apoyo

FASE DESARROLLO

jQuery (HTML/CSS)

Métodos para manipular el DOM

- jQuery nos permite acceder al DOM de manera sencilla con ayuda de los siguientes métodos.

text() – Asigna o regresa el contenido de texto de los elementos seleccionados.

```
$(“selector").text(“nuevoContenido”);
```

html() - Asigna o regresa el contenido de los elementos seleccionados (incluyendo etiquetas HTML)

```
$(“selector").html(“nuevoContenido”);
```

val() - Asigna o regresa el valor de los campos de formulario.

```
$(“selector").val(“nuevoContenido”);
```

Asignar atributos – attr()

- Attr() también es usado para asignar o cambiar valores de atributos. Ejemplo:

- ```
  $(“selector").attr(“atributo", “nuevoValor");
  ```

- ```
  $(“selector").attr({
          “atributo1" : “nuevoValor1",
          "atributo2" : "nuevoValor2"
      });
  ```

Añadir nuevo contenido HTML

- Los siguientes métodos son usados para crear nuevo contenido.

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

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

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

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

Crear nuevos elementos con append() y prepend()

```
function appendText() {
    var txt1 = "<p>Text.</p>";                // Create element with HTML
    var txt2 = $("<p></p>").text("Text.");    // Create with jQuery
    var txt3 = document.createElement("p");   // Create with DOM
    txt3.innerHTML = "Text.";
    $("body").append(txt1, txt2, txt3);       // Append the new elements
}
```

Borrar Elementos y Contenido

- Para borrar elementos y contenido existen dos métodos principales:

remove() – Borra los elementos seleccionados (Incluyendo elementos hijos).

empty() - Borra los elementos hijos de los elementos seleccionados.

```
$("#div1").remove(“Elemento”);
```

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

Remove

- El método **remove()** recibe un parámetro que permite filtrar el elemento que se quiere borrar.

  El siguiente ejemplo demuestra como se pueden borrar todos los elementos que tengan la clase "prueba":

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

- Este ejemplo borra los elementos **<p>** con las clases **.prueba**, **.texto** y el identificador **#demo**.

```
$("p").remove(".prueba, .texto, #demo");
```

Asignar clases CSS

- jQuery tiene varios métodos para manipular CSS. Referenciaremos algunos de ellos:

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

removeClass() - Borra una o mas clases de los elementos seleccionados.

toggleClass() - Alterna entre las funciones añadir/borrar de los elementos seleccionados.

css() - Asigna el atributo style.

Documento de Apoyo

jQuery (HTML)

- El siguiente ejemplo muestra como añadir clases a diferentes elementos.

```
.importante {font-weight: bold; font-size: 3em; }
.azul {color: blue; }


$("button").click(function(){
    $("h1, h2, p").addClass(“azul");
    $("div").addClass("importante");
});
```

- También se pueden asignar varias clases dentro de un mismo método.

removeClass()

- **removeClass()** permite eliminar una clase específica de los elementos seleccionados:

```
$("button").click(function(){
    $("h1, h2, p").removeClass(“azul");
});
```

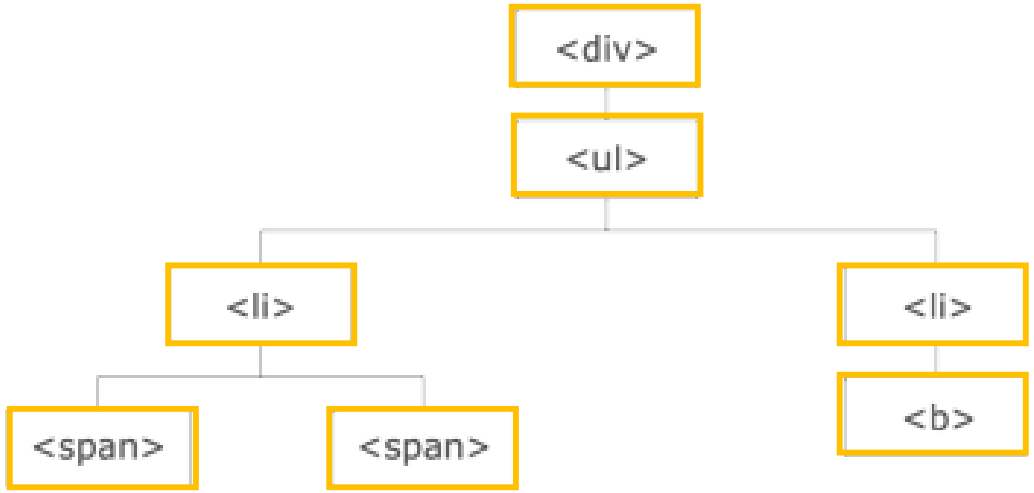
toggleClass()

- Este método alterna entre las funciones añadir/borrar clases de los elementos seleccionados:

```
$("button").click(function(){
    $("h1, h2, p").toggleClass(“azul");
});
```

Atravesando documento HTML

Con jQuery podemos desplazarnos fácilmente hacia arriba **(antepasados)**, hacia abajo **(descendientes)** y lateralmente **(hermanos)** en el árbol de la familia, empezando por el elemento seleccionado. A este movimiento se le llama atravesar - o moverse a través - del DOM.



Antepasados

Un antepasado es un **padre (parent)**, **abuelo (grandparent)**, **bisabuelo (great-grandparent)**, y así sucesivamente.

Tres métodos jQuery útiles para recorrer el árbol DOM son:

parent()

parents();

parentsUntil ();

parent();

El método parent () devuelve el elemento padre directo del elemento seleccionado.

Este método solo atraviesa un nivel hacia arriba en el árbol DOM

```
$(document).ready(function(){
    $("span").parent();
});
```

parents();

El método parents(); regresa todos los ancestros del elemento seleccionado hasta la raíz del elemento.

El siguiente ejemplo regresa todos los antepasados del elemento ****

```
$(document).ready(function(){
    $("span").parents();
});
```

parentsUntil();

El método **parentsUntil();** devuelve todos los elementos antecesores entre dos argumentos dados.

En este ejemplo se devuelven todos los elementos antecesores entre un **** y ** **:

```
$(document).ready(function(){
    $("span").parentsUntil("div");
});
```

Descendientes

Los descendientes pueden ser: **“child”**, **“grandchild”**, **“great-grandchild”**.

Los métodos para recorrer el DOM hacia abajo son los siguientes:

children();

find();

children();

El método children() devuelve el elemento hijo directo del elemento seleccionado.

Este método solo atraviesa un nivel hacia abajo en el árbol DOM

```
$(document).ready(function(){
    $("div").children();
});
```

jQuery (HTML)

children(); puede utilizar un parámetro opcional para filtrar la búsqueda de los elementos **child**.

El siguiente ejemplo devuelve todos los elementos **<p>** con el nombre de clase **“uno"**, que son hijos directos de ** **:

```
$(document).ready(function(){
    $("div").children("p.uno");
});
```

find();

Find(); devuelve todos los descendientes del elemento seleccionado, realiza el recorrido hasta el último elemento descendiente.

```
$(document).ready(function(){
    $("div").find("span");
});
```

```
$(document).ready(function(){
    $("div").find("*");
});
```

Reducir búsqueda

Los tres métodos básicos de filtro son **first()**. **last()**. **eq()** los cuales permiten seleccionar un elemento especifico basado su posición dentro de un grupo de elementos.

Otros métodos de filtrado, como **filter ()** y **not ()** permiten seleccionar elementos que coincidan o no con ciertos criterios.

first();

First() regresa el primer elemento de los elementos seleccionados.

El siguiente ejemplo selecciona el primer elemento **<p>** dentro del elemento ** **

```
$(document).ready(function(){
    $("div p").first();
});
```

last();

Last() regresa el último elemento de los elementos seleccionados.

El siguiente ejemplo selecciona el último elemento **<p>** dentro del elemento ** **

```
$(document).ready(function(){
    $("div p").last();
});
```

eq();

eq() regresa un elemento con un número de índice especifico dentro del elemento seleccionado.

Los números índice o index empiezan en 0, es decir que el primer elemento tendrá asignada la posición 0 y no 1. El siguiente ejemplo selecciona el segundo elemento **<p>:**

```
$(document).ready(function(){
    $("p").eq(1);
});
```

Filter();

filter() permite especificar criterios.

Los elementos que no coincidan con los criterios se eliminarán y se devolverán los que coincidan.

El siguiente ejemplo devuelve todos los elementos **<p>** que tienen la clase **"intro"**:

```
$(document).ready(function(){
    $("p").filter(".intro");
});
```

Not();

Realizará la acción opuesta a el método **filter().**

not() devuelve los elementos que no coinciden con los criterios.

El siguiente ejemplo devuelve todos los elementos **<p>** que no tienen la clase **"intro"**:

```
$(document).ready(function(){
    $("p").not(".intro");
});
```