DISEÑO DE INTERFACES

U.D.4

Librería jQuery

Objetivos

- Librerías de JavaScript.
- Librerías de actualización dinámica.
- jQuery.

Librerías útiles de JavaScript

- Las librerías o bibliotecas de JavaScript contienen funciones, métodos u objetos para realizar tareas específicas en una página o aplicación Web con el fin de ahorrar tiempo y esfuerzo.

- Para utilizar una librería en una aplicación, generalmente basta con añadir <script> al elemento <head> utilizando el atributo "src" que hace referencia a la ruta de origen o URL de la librería.

- Ejemplos: documento "Librerías de JavaScript"
  https://kinsta.com/es/blog/bibliotecas-javascript/#qu-son-las-bibliotecas-javascript

Librerías de actualización dinámica

- Junto con la tecnología AJAX, están las librerías que implementan una gran cantidad de funciones y controles:
  - Independiente de la tecnología del servidor (ej. PHP, etc.).
  - Manejar de manera transparente las incompatibilidades de los diferentes navegadores.
  - Manejar la comunicación asíncrona, sin necesidad de realizar la gestión de las operaciones de bajo nivel, como por ejemplo el manejo de estados y de tipos de errores.
  - Acceso sencillo al árbol DOM.
  - Información de errores para facilitar su utilización al desarrollador.
  - Proporcionar controles y objetos gráficos configurables, como por ejemplo botones, calendarios, campos de texto, etc.

Librería jQuery

- Es una librería de funciones clásica de JavaScript (creada en 2006) con licencia GPL que permite:
  - Selección y manipulación de elementos HTML y CSS.
  - Funciones de eventos en HTML.
  - Efectos y animaciones. Soporta efectos complejos como arrastrar y soltar.
  - Soporte de temas y de manejo directo con el teclado.
  - Funciones de búsqueda.
  - Edición en el mismo control.
  - Soporta la tecnología AJAX.
  - Compatibilidad con todos los navegadores.
  - "Write less, do more" (escribe menos, haz más).

Selector jQuery

- La forma básica de interactuar es mediante la función $() que recibe como parámetro el identificador de un elemento HTML o el nombre de una etiqueta HTML.

  - Para utilizar la librería:

    ```
    <script src="jquery-3.6.1.min.js"></script>
    ```

  - Sintaxis: $(selector).action()
    - $: Símbolo para definir jQuery.
    - selector: Consulta sobre elementos HTML.
    - action: Acción que ejecuta sobre los elementos.

  - Ejemplo: Momento de ejecución de código JS cuando se carga una página.

    ```
    window.onload = function() { /*Código JS*/ };
    $(document).ready(function() { /*Código jQuery*/ });
    ```

jQuery

- Ejemplos de uso:

$("p") // Se seleccionan todos los elementos de tipo párrafo.

$("p.intro") // Todos los párrafos con class=intro.

$("p#demo") // Todos los párrafos con id=demo.

$("[href]") // Todos los elementos con atributo href.

$("[href='#']") // Todos los elementos con atributo href="#".

$("[href!='#']") // Todos los elementos con atributo href diferente de #.

$("[href$='.jpg']") // Todos los elementos con atributo href que acabe en .jpg.

$("p").css("background-color","yellow"); // Se modifica el color de fondo de todos los párrafos a amarillo.

$("p#intro:first") // El primer párrafo con id="intro".

$("ul li:first") // El primer elemento <li> de cada <ul>.

$("div#intro.head") // Todos los elementos con class="head" dentro de un con id="intro".

jQuery

- Ejemplos de uso:

$(selector).html(contenido) // Cambia el contenido de un elemento HTML.

$(#boton).fadeOut() // Se agrega un efecto visual al elemento con identificador "boton".

$(this).hide() // Oculta el elemento actual.

$(this).toggle() // Conmuta entre ocultar y mostrar el elemento.

$("input[name='nombre']").val() // Se obtiene el valor del elemento <input> de name nombre.

$(document).ready(function(){ }) // Se ejecuta la función cuando se cargue la página Web.

$(input:button).click(function(){ }) // Se ejecuta la función al pulsar en el elemento <input> de tipo button.

$.ajax({ }) // Método para trabajar con AJAX.

jQuery

- Ejemplo: Crea una página Web que muestre un párrafo y se oculte al pulsar sobre un botón haciendo uso del selector jQuery.

- Ejemplo: Continuando con el ejemplo anterior, al pulsar en el botón, el párrafo debe desaparecer. Pero si se vuelve a pulsar, debe aparecer de nuevo.

Ejercicios

- Realiza todos los ejercicios del bloque 'Ejercicios 5.1'.

- Realiza todos los ejercicios del bloque 'Ejercicios 5.6'.

INTEGRACIÓN DE CONTENIDO INTERACTIVO

CAPÍTULO 5

Módulo de Diseño de Interfaces Web

C.F.G.S. Técnico Superior en Desarrollo de Aplicaciones Web

4.1 ELEMENTOS INTERACTIVOS BÁSICOS Y AVANZADOS

Back to Google Logos

4.1.1 INTRODUCCIÓN A JQUERY

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

Una de las primeras instrucciones que todo código jQuery debe tener es

$(document).ready(function() { ... });

La función más usada en jQuery es $(). Esta función se utiliza para seleccionar un elemento del árbol DOM a través del identificador, a través del nombre de la clase o de la etiqueta (CSS).

- *$("#miIdentificador");*
- *$("a");*
- *$("#miParrafo, a");*
- *$(".miClase");*
- *$("a[rel]"); //Con un determinado atributo*
- *$("input[@type=radio]"); //Con un valor para un atributo*
- *$("/html/body//p"); //Usando XPath*

4.1.2 MANEJO DE EVENTOS

- Para cada evento disponible hay dos posibilidades de manejo:
  - Se le puede pasar una función que determine su comportamiento.
    - *$("p").click(function() { alert($(this).text());});*
  - No se le pasa función, entonces se ejecuta el evento JavaScript del elemento. jQuery tiene tantas funciones como eventos estándar de JavaScript. El nombre de cada función es el mismo que el del evento, pero sin el habitual prefijo “on” de los eventos:
    - *$("p").click();*

4.1.2 MANEJO DE EVENTOS (II)

- **bind() y unbind():** Este método permite asociar a un elemento un número ilimitado de eventos, todo de una vez. Esto método facilita la asignación de eventos a los elementos, haciendo un código más legible y una ejecución más óptima.

```
$("p").bind("click mouseenter mouseleave", function(e){
        if ($(this).css("color")!="rgb(255, 0, 0)")
                $(this).css("color", "rgb(255, 0, 0)");
        else
                $(this).css("color", "rgb(0, 0, 255)");
})
```

- **toggle():** Se le pasan dos funciones. La primera vez que se hace clic sobre el elemento (y todas las veces impares), se ejecuta la primera función y la segunda vez que se hace clic el elemento (y todas las veces pares) se ejecuta la segunda función:

```
var poner_Rojo= function() {$(this).css("color", "rgb(255, 0, 0)");     }
var poner_Azul= function() {$(this).css("color", "rgb(0, 0, 255)");     }
$(document).ready(function() {$("p").toggle(poner_Rojo,poner_Azul);     });
```

4.2 CAMBIO DE LAS PROPIEDADES DE UN ELEMENTO

4.2.1 CAMBIO DE LAS PROPIEDADES CSS

- *Tres ejemplos:*

  - *$("p").css("color");*
  - *$("p").css("color","red");*
  - *$("p").css({ color: "red", background: "blue", font-weight: "bold" });*

4.2.2 AÑADIR NUEVOS ELEMENTOS

- *append()* permite insertar un contenido como el último hijo (árbol DOM) de cada elemento de la colección seleccionada. Si lo que se desea es insertarlo como primer hijo, entonces se usa *.prepend()* de la misma manera.

```
$(document).ready(function() {
        $(".misdivs").bind("click mouseenter mouseleave", function(e){
        $(".misdivs").append("<p><b>Texto</b>insertado en los div</p>");
        $(".misdivs").append($("h1"));
        })
});
```

4.2.3 AÑADIR TEXTO

- *.text().* Este método permite obtener o añadir un texto a un elemento determinado (que permita texto). Puede ser utilizado tanto en HTML como en XML:
  - En la forma *.text()* obtiene el texto de un elemento (no es válido para elementos de formularios tipo <input> ya que para ello se utiliza .val()).
  - En la forma *.text(cadena_de_texto)* escribe ese texto en el elemento que lo invoca.

```
$("p").text("<b>Some</b> new text.");
```

4.2.4 OTROS MÉTODOS

- *.innerWidth()* e *.innerHeight()*: para un elemento devuelve sus dimensiones internas (anchura y altura, respectivamente) contando el *padding* del elemento pero no el borde.
- *.outerWidth()* e *outerHeight()*: devuelve las dimensiones externas (anchura y altura, respectivamente) del elemento contando el *padding* del elemento y su borde.
- .offset(): indica la posición del elemento real, teniendo en cuenta los márgenes del elemento. Lo devuelve en un objeto con dos atributos *top* y *left*.

4.3 EJECUCIÓN DE SECUENCIA DE FUNCIONES

- los *callbacks:*

```
$("#misdivs").fadeOut(2000);
$("#misdivs").css({background: "blue"});
$("#misdivs").fadeIn(2000);
```

Esto mismo se hubiese conseguido haciendo una secuencia de este tipo:

```
$("#misdivs").fadeOut(2000).css({background: " blue "}).fadeIn(2000);
```

- *Pila de ejecución de funciones:*

```
$("#micapa").fadeOut(1000, function(){
        $("#micapa").css({'top': 300, 'left':200});
        $("#micapa").fadeIn(1000);
});
```

```
var ocultar=function ()
{$("#misdivs3").css({background: "green"});
$("#misdivs3").fadeIn(2000); };
…
$("#misdivs3").fadeOut(2000, ocultar);
```

4.4 COMPORTAMIENTO DE LOS ELEMENTOS. EFECTOS VISUALES

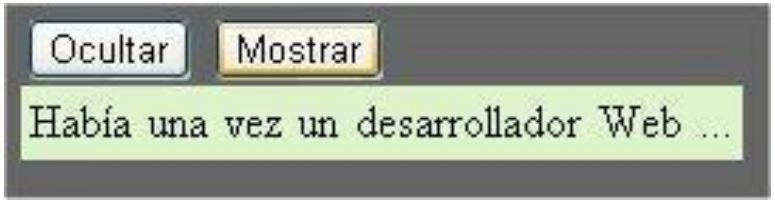
4.4.1 EFECTOS BÁSICOS

- *.show ([duración] [, easing] [, callback])*

  *$("span").**show(500);***

- *.hide ([duración] [, easing] [, callback])*

  *$("span:last-child").**hide("fast", ocultar);***



4.4.2 EFECTO *FADDING (OPACIDAD Y TRANSPARENCIAS)*

- *.fadeIn ([duración] [, easing] [, callback])*
- *.fadeOut ([duración] [, easing] [, callback])*
- *.fadeToggle ([duración] [, easing] [, callback])*

```
$("span").click(function () {
        $(this).fadeOut(1000, function () {
        $("div").text("'" + $(this).text() + "' ¡Se ha desvanecido!");
        $(this).remove();
        }); //function del Fade
}); //Function del Click
```

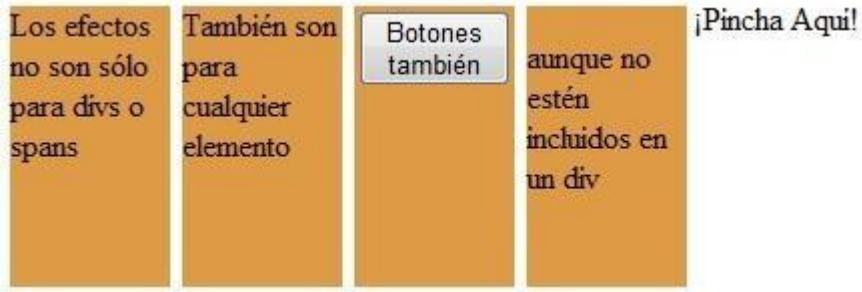
Encuentra los adjetivos 'cálido' ¡Se ha desvanecido!

La chaqueta roja el lo mejor cuando hace un tiempo ¿verdad?

4.4.3 EFECTO *SLIDING (DESPLAZAMIENTO)*

- *.slideUp ([duración] [, easing] [, callback])*
- *.slideDown ([duración] [, easing] [, callback])*
- *.slideToggle ([duración] [, easing] [, callback])*

```
$(document.body).click(function () {
                    if ($("div:first").is(":hidden")) {
                    $("div").slideDown("slow");
                    } else {
                    $("div").hide();
                    }
            });
```



4.4.4 OTROS EFECTOS

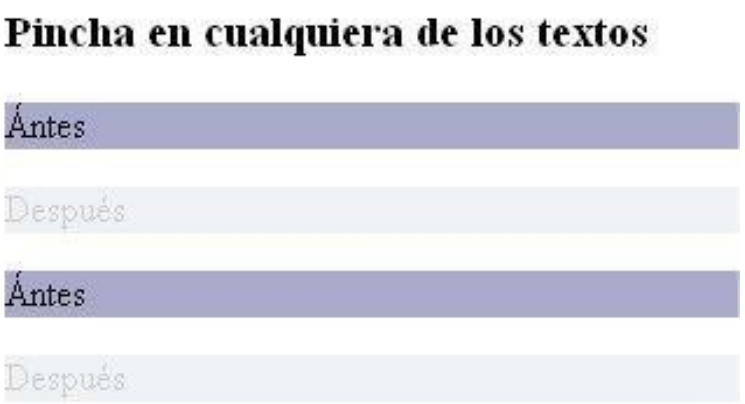
- *.animate (propiedades, [duración] [, easing] [, callback])*

```
$("#left").click(function(){//Animación del botón izquierdo.
        $(".block").animate({"left": "-=50px"}, "slow");
      });
      $("div").click(function(){ //al hacer click en div, cambia de tamaño
       $("div").animate({
         width: "70%",
         opacity: 0.5,
         marginLeft: "0.6in",
         fontSize: "3em",
         borderWidth: "10px"
       }, 1500 );
      });
```

4.4.4 OTROS EFECTOS (II)

- *.delay(duración [, NombreCola])*

```
$(document).ready(function(){
        $("div").click(function() {
          $("div.antes").slideUp(300);
          $("div.antes").fadeIn(800);
          $("div.despues").slideUp(300);
          $("div.despues").delay(800);
          $("div.despues").fadeIn(800);
          });
    });
```



Hacer Actividad 4.3 del libro

4.5 COMPORTAMIENTO INTERACTIVO

4.5.1 EVENTOS DE RATÓN

- *Elemento.Evento (función (handler));*

  - ***.click()***. Se lanza un evento clic cuando el usuario, colocado sobre un elemento (objeto) presiona el botón izquierdo del ratón y lo levanta, todo dentro del mismo objeto. Cualquier elemento HTML puede recibir este evento.
  - ***.dblclick(), .focusin(), .focusout(), .mousedown(), .mouseup(), .mouseenter(), .mouseleave(), .mousemove(), .mouseover(), .mouseout()***.

- *Elemento.hover (función_entrada (handler), función_salida (handler));*

```
$("li#localizacion").hover(function () {$("span#opcion_loc").show();
                                }, function () {
                                $("span#opcion_loc").hide();       });
```

4.5.2 EVENTOS DE TECLADO

- *Elemento.Evento (función (handler));*

  - ***.****focusin(), .focusout(),.keydown(), .keyup(), .keypress().*

```
$(document).ready(function() {
    $('#objetivo').keypress(function(e) {
      e.preventDefault();
      $("#salidapress").html(e.which + ": " + String.fromCharCode(e.which))
                        });
          });
```

4.5.3 EVENTOS DE VENTANA

- *.scroll().* Este evento se atiende cuando sobre un elemento que tiene barras de desplazamiento se mueve una de ellas.
- *.resize()*. El evento se lanza cuando a un elemento tipo ventana se le cambia el tamaño.

4.6 REPRODUCCIÓN DE SONIDO, VÍDEO Y ANIMACIÓN

- *<audio> y <video> HTML5*
- *Otras alternativas basadas en jQuery*

  - *Acorn Media Player*:
  - *Video JS*:
  - *jPlayer*:
  - *Flare Video*:
  - *Media Element*:
  - *Simple Player*:

INTEGRACIÓN DE CONTENIDO INTERACTIVO

CAPÍTULO 4

Módulo de Diseño de Interfaces Web

C.F.G.S. Técnico Superior en Desarrollo de Aplicaciones Web.