12004217@centres.cult.gva.es
c/ Pintor Soler Blasco, 3, 12003 Castelló
http://moodle.ieselcaminas.org
tlf:964232553 Fax:964224702
CIF: Q-6255120-E

DESARROLLO WEB EN ENTORNO CLIENTE

BLOQUE 3
PRACTICA 2

JQUERY-AJAX EFECTOS PLUGINS- WIDGETS

INTRODUCCIÓN

Con esta práctica finalizamos este bloque. Conocemos ya la sintaxis básica de jQuery, en esta segunda práctica utilizaremos Ajax a través de jQuery, trabajaremos con algunos efectos de animación, crearemos plugins y entenderemos el concepto de widget con algunos ejemplos.

PRACTICA 1

AJAX en jQuery

Ya vimos que era Ajax, para que servía y como se utilizaba implementado el objeto XMLHttpRequest, jQuery trae incorporado un módulo para trabajar con AJAX, que hace muy sencillo su uso. Ejemplo básicos de utilización de **$.ajax** podrían ser los siguientes:

Ejemplo 1

```
$.ajax({
    type: "GET",
    url: "ejer.js",
    dataType: "script"
    async:false;
});
```

Ejemplo 2

```
$.ajax({
      type: "POST",
      url: "programa.php",
      data: "nombre=Maria&apellido=Garcia",
      success: respuestacorrecta,
      error: respuestaerronea
});
```

En el Ejemplo 1 utilizamos el método GET como método de envío, en el segundo ejemplo utilizamos el POST y le mandamos con data los parámetros que el programa php necesitará para ejecutarse correctamente.

La siguiente tabla muestra todas las opciones que se pueden definir para el método $.ajax:

Opción	Descripción
`Async`	Indica si la petición es asíncrona. Su valor por defecto es `true`, el habitual para las peticiones AJAX
`beforeSend`	Permite indicar una función que modifique el objeto `XMLHttpRequest` antes de realizar la petición. El propio objeto `XMLHttpRequest` se pasa como único argumento de la función
`complete`	Permite establecer la función que se ejecuta cuando una petición se ha completado (y después de ejecutar, si se han establecido, las funciones de `success` o `error`). La función recibe el objeto `XMLHttpRequest` como primer parámetro y el resultado de la petición como segundo argumento
`contentType`	Indica el valor de la cabecera `Content-Type` utilizada para realizar la petición. Su valor por defecto es `application/x-www-form-urlencoded`
`data`	Información que se incluye en la petición. Se utiliza para enviar parámetros al servidor. Si es una cadena de texto, se envía tal cual, por lo que su formato debería ser `parametro1=valor1&parametro2=valor2`. También se puede indicar un array asociativo de pares clave/valor que se convierten automáticamente en una cadena tipo *query string*
`dataType`	El tipo de dato que se espera como respuesta. Si no se indica ningún valor, jQuery lo deduce a partir de las cabeceras de la respuesta. Los posibles valores son: `xml` (se devuelve un documento XML correspondiente al valor `responseXML`), `html` (devuelve directamente la respuesta del servidor mediante el valor `responseText`), `script` (se evalúa la respuesta como si fuera JavaScript y se devuelve el resultado) y `json` (se evalúa la respuesta como si fuera JSON y se devuelve el objeto JavaScript generado)
`error`	Indica la función que se ejecuta cuando se produce un error durante la petición. Esta función recibe el objeto `XMLHttpRequest` como primer parámetro, una cadena de texto indicando el error como segundo parámetro y un objeto con la excepción producida como tercer parámetro
`ifModified`	Permite considerar como correcta la petición solamente si la respuesta recibida es diferente de la anterior respuesta. Por defecto su valor es `false`
`processData`	Indica si se transforman los datos de la opción `data` para convertirlos en una cadena de texto. Si se indica un valor de `false`, no se realiza esta transformación automática
`success`	Permite establecer la función que se ejecuta cuando una petición se ha completado de forma correcta. La función recibe como primer parámetro los datos recibidos del servidor, previamente formateados según se especifique en la opción `dataType`
`timeout`	Indica el tiempo máximo, en milisegundos, que la petición espera la respuesta del servidor antes de anular la petición
`type`	El tipo de petición que se realiza. Su valor por defecto es `GET`, aunque también se puede utilizar el método `POST`
`url`	La URL del servidor a la que se realiza la petición

(Fuente: www.librosweb.es – Manual Introducción a Ajax)

Puedes ver información complementaria sobre el método $.ajax en el siguiente enlace:

http://www.w3schools.com/jquery/ajax_ajax.asp

Además de esta función genérica existen otras en jQuery para realizar funciones especificas como pueden ser: $.load, $.get(), $.post, etc.

Su utilización es muy sencilla, veamos la sintaxis de estas funciones y algunos ejemplos:

$.load

La función más simple para cargar contenido en Ajax es load(). Su sintaxis es la siguiente:

load(url,parametros,callback)

- **url:** La URL a la que se envía la petición.
- **parámetros:** Parámetros que pasamos junto con la petición.

- **callback:** Llamada a la función que se encargará de gestionar la respuesta del servidor.

El primer parámetro es obligatorio, los otros dos son opcionales.

Por ejemplo

$("#contenedor").load("pagina.html");

Cargará la página y la dejará dentro del div “contenedor”.

Consultar este enlace para ver más información: http://www.desarrolloweb.com/articulos/uso-ajax-jquery.html

$.get

La función $.get se utiliza para realizar peticiones de forma sencilla con este método. Su sintaxis es la siguiente:

$.get(url, datos, funcion, tipo_dato_respuesta)

El primer parámetro es obligatorio, los otros dos son opcionales.

Ejemplo:

$.get(“pagina.php", { nombre:"Maria", ciudad:"Castellon" });

Consultar este enlace para ver más información:
http://www.desarrolloweb.com/articulos/funcion-get-jquery-ajax.html

$.post

Funciona de la misma manera que la anterior y con la misma sintaxis.

```
$.post('pagina.html', function(data) {
  $('.result').html(data);
});
```

Consultar este enlace para mayor información:

http://api.jquery.com/jQuery.post/

EFECTOS EN JQUERY

Ya os habréis dado cuenta que uno de los atractivos más importantes de jQuery son los efectos que podéis asignar a los elementos de la página. Vimos eventos básicos como hide, slow, fadeOut y fadeIn ,aunque los vamos a repasar, también vamos a ver métodos de animación nuevos como el animate y algunos efectos de deslizamiento.

animate

Uno de los efectos más interesantes que podemos utilizar es el método animate a partir de la modificación de propiedades CSS. Para invocarlo tenemos que indicar una serie de parámetros, aunque sólo el primero es obligatorio:

.animate(Propiedades, [Duración], [Función de animación], [Callback])

Vemos un ejemplo:

```
$(document).ready(function() {
    $("#boton1").click(function() {
        $("#elemento1").animate({
            height: "20px",
            width: "10px",
        }, 1000);
        $("#elemento2").animate({
            height: "50px",
            width: "400px"
        }, 1000);
    });
});
```

Algunos enlaces interesantes:

http://www.desarrolloweb.com/articulos/animate-jquery-animacion.html

http://christian.sietemedia.com.mx/animate/animate.html

Efectos de fundido

Son efectos interesantes para provocar efecto de fundido (desvanecer y aparecer un contenido) en jQuery. Serian los métodos : fadeOut(), fadeIn(), fadeTo().

Ejemplo:

```
<script type="text/javascript">
$(document).ready(function(){
  $(document).ready(function () {
    $('#contenedor').fadeIn(1200);
  });
});
</script>
```

http://www.desarrolloweb.com/articulos/fading-jquery.html

Efectos de deslizamiento

Estos efectos se utilizan para deslizar hacia abajo (slideDown) o hacia arriba (slideUp) el contenido indicado.

Ejemplo:

```
$("button").click(function(){
$("p").slideDown();
});
```

Enlace:
http://api.jquery.com/category/effects/sliding/

Cola de efectos

Cada vez que invocamos un efecto en jQuery este se va introduciendo en una cola de efectos para ejecutarse automáticamente. Si llamamos al método queue() podemos trabajar con los método que se encuentren en la cola en esos momentos. A veces hay que trabajar con métodos de esta cola de efectos para que se gestione correctamente como son stop(), delay(), etc.

Nota: Los temas 46-51 del manual de desarrolloweb hablan sobre este tema.

https://desarrolloweb.com/articulos/colas-efectos-jquery.html

PLUGINS

Un plugin es una extensión que se añade al objeto jQuery para poder ampliar los métodos de trabajo que el elemento tiene en sí.
¿Cuándo creamos un plugin? Cuando tengamos que hacer algo sobre algún elemento(o elementos) que no tenga implementado ningún método para hacer lo que queramos hacer.
Un ejemplo: Queremos que todos los text de un formulario al tomar el foco se pongan en azul.
Para ello crearíamos un plugin que accediese a todos los text y con la propiedad css correspondiente les aplicase el color.
Los plugins normalmente se almacenan en librerías para que se puedan utilizar cada vez que sea necesario.
Para crear plugins hay que seguir unas normas que utilizan todos los desarrolladores. Estos plugins creados pueden ser de uso particular o se pueden compartir con los usuarios, como hacen algunos autores, lo que nos permite ampliar de forma extraordinaria las capacidades de jQuery y los efectos que podemos obtener.

Os adjunto dos enlaces sencillos que explican cómo crear un plugin:

http://www.desarrolloweb.com/articulos/plugins-jquery.html

http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=jQueryPlugin

Os adjunto dos enlaces para ver diferentes plugins:

http://webgenio.com/2012/06/30-plugins-jquery-para-slideshows-y-pases-de-diapositivas/

http://www.desarrolloweb.com/manuales/taller-jquery.html

Nota: En el manual de desarrolloweb los temas 34-42 tratan sobre plugins.

Widgets

En informática, un ***widget*** es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de *widgets* o *Widget Engine*. Entre sus objetivos están dar fácil acceso a funciones frecuentemente usadas y proveer de información visual. Aunque no es condición indispensable, los widgets suelen ser utilizados para ser "empotrados" en otra página web, copiando el código que el mismo widget pone a disposición del usuario. Dado que son pequeñas aplicaciones, los *widgets* pueden hacer todo lo que la imaginación desee e interactuar con servicios e información distribuida en Internet; pueden ser vistosos relojes en pantalla, notas, calculadoras, calendarios, agendas, juegos, ventanas con información del tiempo en su ciudad, incluso sistemas de tiendas de comercio, etcétera. (definición de la wikipedia).
La forma de utilizar un widget es similar a utilizar un plugin, muchos confunden ambos términos. La diferencia principal es visual, los widgets tienen una interfaz gráfica de usuario mientras que los plugins no siempre la tienen, como hemos visto en el apartado anterior.

Otra diferencia importante es que los widgets forma parte de jQuery UI, una biblioteca de componentes para jQuery, y los plugins no.
Con la utilización de Widgets la programación web desde el lado cliente llega a su máximo nivel ya que los efectos que se consiguen son realmente espectaculares.

Enlaces:

http://es.wikipedia.org/wiki/JQuery_UI
https://deideaaapp.org/primeros-pasos-con-jquery-ui/
https://jqueryui.com/accordion/

Los ejercicios que debéis entregarme son los siguientes:

1º) Crea una página que nos muestre el contenido de un fichero de texto utilizando $.ajax al pinchar sobre el botón “Mostrar fichero de texto”.

2º) Crea una página que nos cargue otra página html utilizando el método $.load al pinchar sobre el botón “Carga una página html”.

3º) Crea una página que nos muestre el contenido de un fichero XML utilizando el método $.get al pinchar sobre el botón “Mostrar fichero XML”.

Ver como referencia el siguiente enlace:
http://www.returngis.net/2011/01/leer-archivos-xml-con-jquery/

4º) Crea una página que utilizando el método $.post pase uno o dos parámetros a un fichero php y nos devuelva un resultado.

5º) Crea un div con el contenido que consideres y haz que se desplace por la pantalla aplicándole los efectos que hemos visto. Puedes utilizar también el método delay para retrasar la ejecución de los efectos. También tienes que hacer que el efecto se ejecute permanentemente.

6º) Tenemos una página con varios párrafos y un botón desaparecer, crea un plugin sencillo para que al pinchar sobre el botón desaparezcan todos los párrafos de la pantalla.

7º) Crea una página web con el pluging carrusel (imágenes que tú quieras) y con el widget acordeon.

MATERIALES

- Editor de textos sin formato(Notepad++,Gedit…
- Navegador de Internet(Explorer,Firefox…)
- La librería jQuery accesible para ejecutar los scripts.
- La librería jQuery Ui para trabajar con Widgets

.

BIBLIOGRAFIA Y OTROS RECURSOS

- El manual “Introducción a Ajax” de la página www.librosweb.es
- El manual de jQuery de desarrolloweb y otros recursos de Internet.
- “Mantenimiento de portales en Internet” de Xavier Munté y Jordi Sabaté de la editorial InforBooks.
- “Diseño de páginas Web” de J.Mariano González Romano y J.Manuel Cordero Valle.
- “Superutilidades para JavaScript” de Jeff Frentzen y Henry Sobotka.

CRITERIOS DE EVALUACIÓN

La práctica se evaluará como: “satisfactoria” o ”no satisfactoria”.

FORMATO DE ENTREGA

En formato electrónico en fichero comprimido, con el nombre del alumno, según el ejemplo:

Apellido1_nombre_bloq3prac2.zip