Diseño y Programación Web

Tema 5

JavaScript (jQuery)

Enrique Moguel
enrique@unex.es

jQuery

- **Librería JavaScript** multiplataforma.
- Muy extendida.
- *Open Source* (Licencia MIT).
- Simplifica muchas de las operaciones más comunes de JavaScript y AJAX.
- Soporta la creación de *plugins*.

jQuery

- Contiene gran número de funciones JavaScript.
  - *“Write less, do more”*
- El elemento principal de la librería es el objeto JavaScript **jQuery**.

  jQuery()
  $()
  - Manipulación de elemento DOM.

jQuery

- Para empezar a trabajar con jQuery vamos a utilizar una página de pruebas básicas.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title of the document</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/
    3.5.0/jquery.min.js"></script>
  </head>
  <body>
    <a href="www.example.com">Link</a>
    <script>
      $("a").attr("href", "https://www.unex.es");
    </script>
  </body>
</html>
```

jQuery

- El atributo **src** del elemento *script* debe apuntar a una copia de jQuery.
  - Puede estar online.
  - Podemos descargarla e incluirla en nuestros proyectos.
  - http://jquery.com/download/
  - Descargamos la versión no comprimida de jQuery 3.6.0.
  - Un único archivo de 289 KB con todo el código de la librería. El comprimido son 90 KB.

jQuery – Primeros pasos

- Es bastante habitual en JavaScript utilizar la función ***onload()*** para ejecutar código antes de que el navegador termine de cargar una página.
  - No se ejecuta hasta que todas las imágenes se han cargado (incluyendo la publicidad).
  - jQuery proporciona un método para ejecutar código en cuanto el DOM puede ser manipulado.
    - Evento ***ready***.

https://stackoverflow.com/questions/1140402/how-to-add-jquery-in-js-file

jQuery – Primeros pasos

```
<script>
    $( document ).ready(function() {
        $( "a" ).click(function( event ) {
            alert( "¡Mensaje de alerta!" );
        });
    });
</script>
```

jQuery – Primeros pasos

- También podemos eliminar el comportamiento por defecto de los eventos:

event.preventDefault();

- Y manipular el CSS de los elementos:

$("a").addClass("test");

$("a").removeClass("test");

```
<style>
    a.test {
        font-weight: bold;
    }
</style>
```

jQuery – Primeros pasos

- jQuery proporciona efectos para crear webs más atractivas visualmente.

```
<script>
    $( document ).ready(function() {
        $( "a" ).click(function( event ) {
            event.preventDefault();
            $( this ).hide( "slow" );
        });
    });
</script>
```

jQuery – Primeros pasos

- Como ya hemos visto anteriormente, JavaScript permite pasar funciones como parámetros para ejecutarse posteriormente.
  - Estas funciones se denominan ***callback***.
  - Muy utilizadas en jQuery.

$.get("myhtmlpage.html", myCallBack);

- Cuando termine de ejecutarse ***$.get()*** se ejecutará ***myCallBack()***.

jQuery – Primeros pasos

- Es posible pasar parámetros a las funciones *callbacks*.
  - Así **NO**

```
$.get( "myhtmlpage.html",
    myCallBack( param1, param2 ) );
```

  - Así **SÍ**

```
$.get( "myhtmlpage.html", function( {
    myCallBack( param1, param2 );
});
```

jQuery - Core

- Núcleo de la librería.
  - Contiene las funciones más utilizadas.
  - El elemento principal es la función **$()**.
    - ***$(selector).action()***
    - ***Selector*** permite seleccionar los elementos DOM que serán encapsulados por jQuery.
    - ***Action*** es la función de jQuery que se aplicará sobre dichos elementos.

jQuery - Core

- Para seleccionar elemento jQuery soporta la mayoría de los selectores de CSS3:
  - También incluye algunos no estándares.
  - Documentación en: http://api.jquery.com/category/selectors/

jQuery - Core

- Podemos seleccionar por id:

```
$( "#myId" ); // IDs must be unique per page
```

- Por nombre de clase:

```
$( ".myClass" );
```

- Por atributo:

```
$( "input[name='first_name']" );
```

jQuery - Core

- Y de otras muchas formas:

```
$( "a.external:first" );
$( "tr:odd" );

// Select all input-like elements in a form (more on this below).
$( "#myForm :input" );
$( "div:visible" );

// All except the first three divs.
$( "div:gt(2)" );

// All currently animated divs.
$( "div:animated" );
```

jQuery - Core

- Y de otras muchas formas:

Syntax	Description
$(this)	Current HTML element
$("p")	All <p> elements
$("p.intro")	All <p> elements with class="intro"
$("p#intro")	All <p> elements with id="intro"
$("p.intro:first")	The first <p> element with class="intro"
$(".intro")	All elements with class="intro"
$("#intro")	The first element with id="intro"
$("ul li:first")	The first element of the first
$("ul li:first-child")	The first element of every
$("[href$='.jpg']")	All elements with an href attribute that ends with ".jpg"
$("div#intro .head")	All elements with class="head" inside a element with id="intro"

jQuery - Core

- La utilización de unos selectores u otros afecta significativamente al rendimiento del código JavaScript.

- Selectores lo suficientemente específicos consiguen una ejecución mucho más rápida.

jQuery - Core

- Podemos comprobar si una selección devuelve algún objeto.

```
// Testing whether a selection contains elements
if ( $( "div.foo" ).length ) {
    ...
}
```

- Y guardar los resultados para utilizarlos más tarde.

```
var divs = $( "div" );
```

jQuery - Core

- Las selecciones se pueden refinar y filtrar.

```
// Refining selections
$( "div.foo" ).has( "p" );           // div.foo elements that contains <p> tags
$( "h1" ).not( ".bar" );             // h1 elements that don't have a class of bar
$( "ul li" ).filter ( ".current" ); // unordered list items with class of current
$( "ul li" ).first();                // just the first unordered list item
$( "ul li" ).eq (5);                 // the sixth
```

jQuery - Core

- A la hora de trabajar con los elementos seleccionados jQuery habitualmente sobrecarga los métodos.
  - *Getters* y *setters* se denominan igual.
  - Los *setters* afectan a todos los elementos de una selección.

```
// The .html() method used as a setter:
$( "h1" ).html( "hello world" );
```

  - Los *getters* devuelven el valor del primer elemento en la selección.

```
// The .html() method used as a getter:
$( "h1" ).html();
```

jQuery - Core

- Los setters devuelven un objeto *jQuery*.
- Permite encadenar las llamadas a métodos (muy habitual en código *jQuery*).

```
$( "#content" )
    .find( "h3" )
    .eq( 2 )
    .html( "new text for the third h3!" );
```

```
$( "#content" )
    .find( "h3" )
    .eq( 2 )
        .html( "new text for the third h3!" )
        .end() // Restores the selection to all h3s in # content
    .eq( 0 )
        .html( "new text for the third h3!" );
```

jQuery - Core

- *jQuery* proporciona múltiples métodos para manipular los elementos seleccionados.

`.html()` – Get or set the HTML contents.

`.text()` – Get or set the text contents; HTML will be stripped.

`.attr()` – Get or set the value of the provided attribute.

`.width()` – Get or set the width in pixels of the first element in the selection as an integer.

`.height()` – Get or set the height in pixels of the first element in the selection as an integer.

`.position()` – Get an object with position information for the first element in the selection, relative to its first positioned ancestor. *This is a getter only*.

`.val()` – Get or set the value of form elements.

jQuery - Core

- Por ejemplo, la función **.attr()** permite añadir o consultar información de los atributos de un elemento.

```
$( "a" ).attr( "href", "allMyHrefsAreTheSameNow.html" );

$( "a" ).attr({
    title: "all titles are the same too!",
    href: "somethingNew.html"
});
```

```
// Returns the href for the first a element in the document
$( "a" ).attr( "href" );
```

jQuery - Core

- También podemos mover los elementos en el árbol DOM.
  - Mover los elementos seleccionados en relación a otro elemento.

    .insertAfter()
  - Mover un elemento en relación a los elementos seleccionados.

    .after()

    .insertBefore(); .before(); .appendTo()
    .append(); .prependTo(); .prepend()

jQuery - Core

- El método *.clone()* permite crear copias de elementos.
- Con el parámetro *true* se copian también los datos y eventos asociados a los elementos.
- *remove()* and *detach()* permite eliminar elementos.
- *empty()* elimina el contenido.

jQuery - Core

- jQuery proporciona un mecanismo muy sencillo para crear nuevos elementos.

```
// Creating new elements from an HTML string
$( "<p>This is a new paragraph</p>" );
$( "<li class=\"new\">new list item</li>" );
```

```
// Creating a new element with an attribute object
$( "<a/>", {
    html: "This is a <strong>new</strong> link",
    "class": "new",
    href: "foo.html"
});
```

jQuery - Core

- jQuery proporciona gran cantidad de métodos para desplazarnos por los elementos seleccionados.
  - **Padres**

    .parent(); .parents(); .parentsUntil(); .closest();

  - **Hijos**

    .children(); .find();

  - **Hermanos**

    .prev(); .next(); .siblings(); .nextAll(); .nextUntil(); .prevAll(); .prevUntil();

http://api.jquery.com/category/traversing/

jQuery - Core

- Podemos manipular el CSS:

```
// Getting CSS propierties
$( "h1" ).css( "fontSize" );  // Returns a string such as "19px"
$( "h1" ).css( "font-size" ); // Also works
```

```
// Setting CSS propierties
$( "h1" ).css( "fontSize", "100px" ); // Setting an individual property

// Setting multiple properties
$( "h1" ).css({
    fontSize: "100px",
    color: "red"
});
```

jQuery - Core

- Podemos manipular el CSS:

```
// Working with classes
var h1 = $( "h1" );

h1.addClass( "big" );
h1.removeClass( "big" );
h1.toggleClass( "big" );

if ( h1.hasClass( "big" ) ){
    ...
}
```

jQuery - Core

- Podemos manipular el CSS:

```
// Basic dimensions methods

// Sets the width of all <h1> elements
$( "h1" ).width( "50px" );

// Gets the width of the first <h1> element
$( "h1" ).width();

// Sets the height of all <h1> elements
$( "h1" ).height( "50px" );

// Sets the height of the first <h1> element
$( "h1" ).height();

// Returns an object containing position information for
// the first <h1> relative to its "offset (positioned) parent"
$( "h1" ).position();
```

jQuery - Eventos

- jQuery proporciona herramientas para trabajar con eventos:
  http://api.jquery.com/category/events/

- La mayoría de eventos son disparados por el usuario.
  - *Clicks, movimientos de ratón, entrada de texto…*
  - Excepto *load* y *unload* que son disparados por el navegador.

https://api.jquery.com/load/#entry-examples

jQuery - Eventos

```
// Event setup using a convenience method
$( "p" ).click(function() {
    console.log( "You clicked a paragraph!" );
})


// Equivalent event setup using the '.on()' method
$( "p" ).on( "click", function() {
    console.log( "click" );
})
```

https://api.jquery.com/category/events/

https://www.w3schools.com/jquERy/jquery_events.asp

jQuery - Eventos

- Toda función asociada a un evento recibe un objeto evento:
  - *.preventDefault()*: Evita el comportamiento por defecto del evento.
  - *pageX*, *pageY*: Posición del ratón cuando se disparó el evento.
  - *type*: Tipo del evento.
  - *which*: Botón o tecla pulsados.
  - *data*: Información adicional enviada con el evento.
  - *target*, *namespace*, *timeStamp*, *stopPropagation()*.

jQuery - Efectos

- jQuery simplifica en gran medida la inclusión de efectos sencillos en las páginas web.

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

- Por ejemplo, podemos mostrar u ocultar elementos.

```
// Instantaneously hide all paragraphs
$( "p" ).hide();

// Instantaneously show all divs that
// have the hidden style class
$( "div.hidden" ).show();
```

jQuery - Efectos

- Podemos controlar la velocidad del efecto.
  - *slow*, *normal* o *fast*.

```
// Slowly hide all paragraphs
$( "p" ).hide( "slow" );

// Quickly show all divs that
// have the hidden style class
$( "div.hidden" ).show( "fast" );
```

  - O de forma *manual*.

```
// Hide all paragraphs over half a second
$( "p" ).hide( 500 );

// Show all divs that have the hidden
// style class over 1.25 secods
$( "div.hidden" ).show( 1250 );
```

jQuery - Efectos

- Se puede tener un control más manual de estos efectos.

```
$("#panel").slideDown("slow");
```

https://www.w3schools.com/jquERy/tryit.asp?filename=tryjquery_slide_down

```
$("#panel").slideUp("slow");
```

https://www.w3schools.com/jquERy/tryit.asp?filename=tryjquery_slide_up

```
$("#panel").slideToggle("slow");
```

https://www.w3schools.com/jquERy/tryit.asp?filename=tryjquery_slide_toggle

jQuery - Efectos

- Se puede tener un control más manual de estos efectos.

```
$("#div2").fadeIn("slow");
```

https://www.w3schools.com/jquERy/tryit.asp?filename=tryjquery_fadein

```
$("#div3").fadeOut(3000);
```

https://www.w3schools.com/jquERy/tryit.asp?filename=tryjquery_fadeout

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

https://www.w3schools.com/jquERy/tryit.asp?filename=tryjquery_fadetoggle

jQuery - Efectos

- Se puede ejecutar código cuando finaliza una animación.

```
// Fade in all hidden paragraphs;
// then add a style class to them
$( "p.hidden" ).fadeIn( 750, function () {
    // this = DOM element which has just finished being animated
    $ ( this ).addClass( "lookAtMe" );
})
```

jQuery - Efectos

- Se pueden crear efectos personalizados con el método *animate()*.

```
$("button").click(function(){
  $("div").animate({
    left: '250px',
    opacity: '0.5',
    height: '150px',
    width: '150px'
  });
});
```

https://www.w3schools.com/jquERy/tryit.asp?filename=tryjquery_animation1_multicss

jQuery - AJAX

- jQuery proporciona múltiples métodos de soporte AJAX.
- El principal es ***$.ajax()***.

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

```
// Using the core $.ajax() method
$.ajax({
    // the URL for the request
    url: "post.php",

    // the data to send (will be converted to a query string)
    data: {
        id: 123
    },

    // whether this is a POST or GET request
    type: "GET",

    // the type of data we expect back
    dataType : "json",

    // code to run if the request succeeds;
    // the response is passed to the function
    success: function( json ) {
        $( "<h1/>" ).text( json.title ).appendTo( "body" );
        $( "<div class=\"content\"/>").html( json.html ).appendTo( "body" );
    },

    // code to run if the request fails; the raw request and
    // status codes are passed to the function
    error: function( xhr, status, errorThrown ) {
        alert( "Sorry, there was a problem!" );
        console.log( "Error: " + errorThrown );
        console.log( "Status: " + status );
        console.dir( xhr );
    },

    // code to run regardless of success or failure
    complete: function( xhr, status ) {
        alert( "The request is complete!" );
    }
});
```

jQuery - AJAX

- Tenemos métodos adicionales para serializar formularios.

```
// Turning form data into a query string
$( "#myForm" ).serialize();

// creates a query string like this:
// field_1=something&field_2=somethingElse
```

- Validación.

```
// Using validation to check for the presence of an input
$( "#form" ).submit(function ( event ) {
  // if .required's value's length is zero
  if ( $( ".required" ).val().length === 0) {
      return false;
  } else {
      // run $.ajax here
  }
});
```

jQuery - AJAX

- Uso de AJAX con jQuery:

https://sutilweb.com/lenguajes/jquery/uso-de-ajax-con-jquery/

https://sutilweb.com/lenguajes/jquery/ajax-jquery-con-mensaje-de-carga/

jQuery

https://www.w3schools.com/jquery/jquery_examples.asp

jQuery - Plugins

- jQuery tienen un sistema de *plugins* que permiten ampliar su funcionalidad.
  - Podemos crear nuestros propios *plugins*.
  - Podemos utilizar los *plugins* existentes.

    http://plugins.jquery.com

jQuery - Plugins

- Slick carousel:
  https://plugins.jquery.com/slick/

- jCarousel:
  https://plugins.jquery.com/jcarousel/

- Pickdate:
  https://plugins.jquery.com/pickadate/

- ScrollMagic:
  https://plugins.jquery.com/ScrollMagic/

AnimateScroll:
https://plugins.jquery.com/animatescroll/

jQuery - Plugins

- ScrollMe:
  https://plugins.jquery.com/scrollme/
- ImagesLoaded:
  https://plugins.jquery.com/imagesloaded/
- Fotorama:
  https://plugins.jquery.com/fotorama/
- **jQuery UI**:
  https://jqueryui.com/demos/
- Y un largo etc.

jQuery

- Podemos descargar una versión de la librería que contenga solo los elementos que vamos a utilizar.

  http://jqueryui.com/download/

- Seleccionamos los componentes, descargamos e incluimos en nuestro código.