

UNIDAD4: INTERACTIVIDAD: PARTE1

COMPONENTES INTERACTIVOS JQUERY

1. COMPONENTES

- Es un elemento software que ofrece una funcionalidad concreta
- Facilita la reutilización, distribución y mantenimiento
- Herramientas para generación de componentes:
 - **JQUERY**: imprescindible en el frontend
 - **BOOTSTRAP**: para componentes más dinámicos

2. JQUERY

- Facilita el desarrollo de aplicaciones en entorno cliente
- Es una librería enriquecida de Javascript
- Tiene compatibilidad con cualquier navegador
- Es de código abierto, reduce el número de líneas de código y simplifica la realización de múltiples tareas.
- Es un APLI de funciones o framework que permite:
 - 2.1.1. Multitud de funciones
 - 2.1.2. Simplificación para interactuar con documento HTML
 - 2.1.3. Manipulación del árbol DOM (Document Object Model) (árbol con nodos del documento)
 - 2.1.4. Sencilla utilización de efectos vistosos
 - 2.1.5. Manejo de eventos
 - 2.1.6. Interacción de AJAX (*Asynchronous JavaScript And XML*)
 - 2.1.7. Creación de plugins personalizados
- Pag oficial: jquery.com
- Pag API: api.jquery.com

3. CARACTERÍSTICAS DE JQUERY

- Total compatibilidad con navegadores.
- Completa manipulación de árbol DOM.
- Se puede utilizar con otras librerías como Prototype pues evita conflictos con ellas.
- Manipulación de estilos CSS.
- Gestión de eventos.
- Uso de efectos enriquecidos, animaciones y desarrollo de plugins personalizados.
- Permite ser enriquecida con multitud de extensiones.

4. VENTAJAS FRENTE A JAVASCRIPT

- Cantidad de código es menor.
- Proporciona gran variedad de efectos y utilidades.
- Solventa problemas en la compatibilidad de navegadores.
- Manipulación relativamente sencilla del DOM.
- Permite excelente integración con AJAX: técnica de programación web para crear aplicaciones interactivas.

5. EMPEZANDO CON JQUERY

INCLUSIÓN DE LA LIBRERÍA Existen 2 formas:

1.- Descargando la última versión del framework desde <http://jquery.com/download>

Existen 2 versiones:

- Comprimida: para aplicaciones puestas en producción que permite que la carga del sitio sea más rápida
- Sin comprimir: para desarrollo que permite ver el código fuente de las funciones

Se descarga un único **archivo js** que se debe almacenar en disco

Downloading jQuery

Compressed and uncompressed copies of jQuery files are available. The compressed file saves bandwidth and improves performance in production with a compressed file. The map file is *not* required for users to run jQuery 1.11.0/2.1.0 the `//# sourceMappingURL` comment is not included in the c

To locally download these files, right-click the link and select "Save as..." f

jQuery

For help when upgrading jQuery, please see the [upgrade guide](#) most relev [plugin](#).

[Download the compressed, production jQuery 3.4.1](#)

[Download the uncompressed, development jQuery 3.4.1](#)

[https://code.jquery.com/jquery-3.4.1.js](#)

[code.jquery.com/jquery-3.4.1.js](#)

Aplicaciones Iniciar sesión Google GestionarFP IES AL-ANDALUS IES AGUADULCE IES Cura Valera. Hu...

```

/*!
 * jQuery JavaScript Library v3.4.1
 * https://jquery.com/
 *
 * Includes Sizzle.js
 * https://sizzlejs.com/
 *
 * Copyright JS Foundation and other contributors
 * Released under the MIT license
 * https://jquery.org/license
 *
 * Date: 2019-05-01T21:04Z
 */
( function( global, factory ) {
    "use strict";

    if ( typeof module === "object" && typeof module.exports === "object" ) {
        // For CommonJS and CommonJS-like environments where
        // is present, execute the factory and get jQuery.
        // For environments that do not have a `window` wi
        // (such as Node.js), expose a factory as module.e
        // This accentuates the need for the creation of a
        // e.g. var jQuery = require("jquery")(window);
        // See ticket #14549 for more info.
        module.exports = global.document ?
            factory( global, true ) :
            function( w ) {

```

Atrás	Alt + Flecha izquierda
Reenviar	Alt + Flecha derecha
Volver a cargar	Ctrl + R
Guardar como...	Ctrl + S
Imprimir...	Ctrl + P
Enviar...	
Enviar a ANE-LX1	
Traducir a español	
Ver código fuente de la página	Ctrl + U
Inspeccionar	Ctrl + Mayús + I

jquery-3.4.1.js

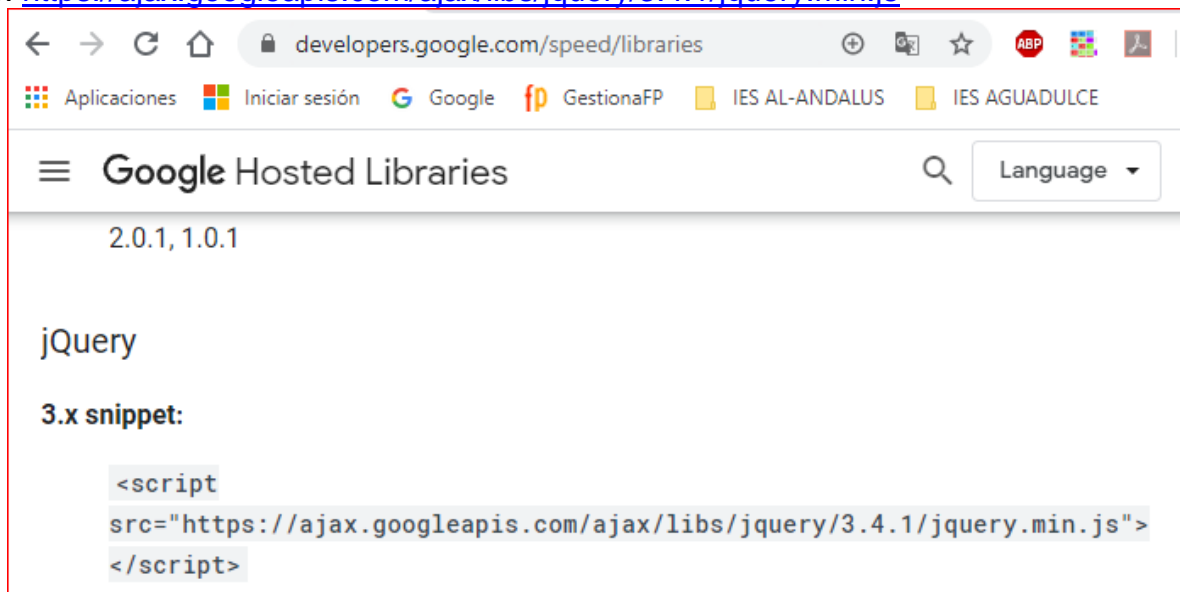
Incluirlo en el body: ejemplo1

```

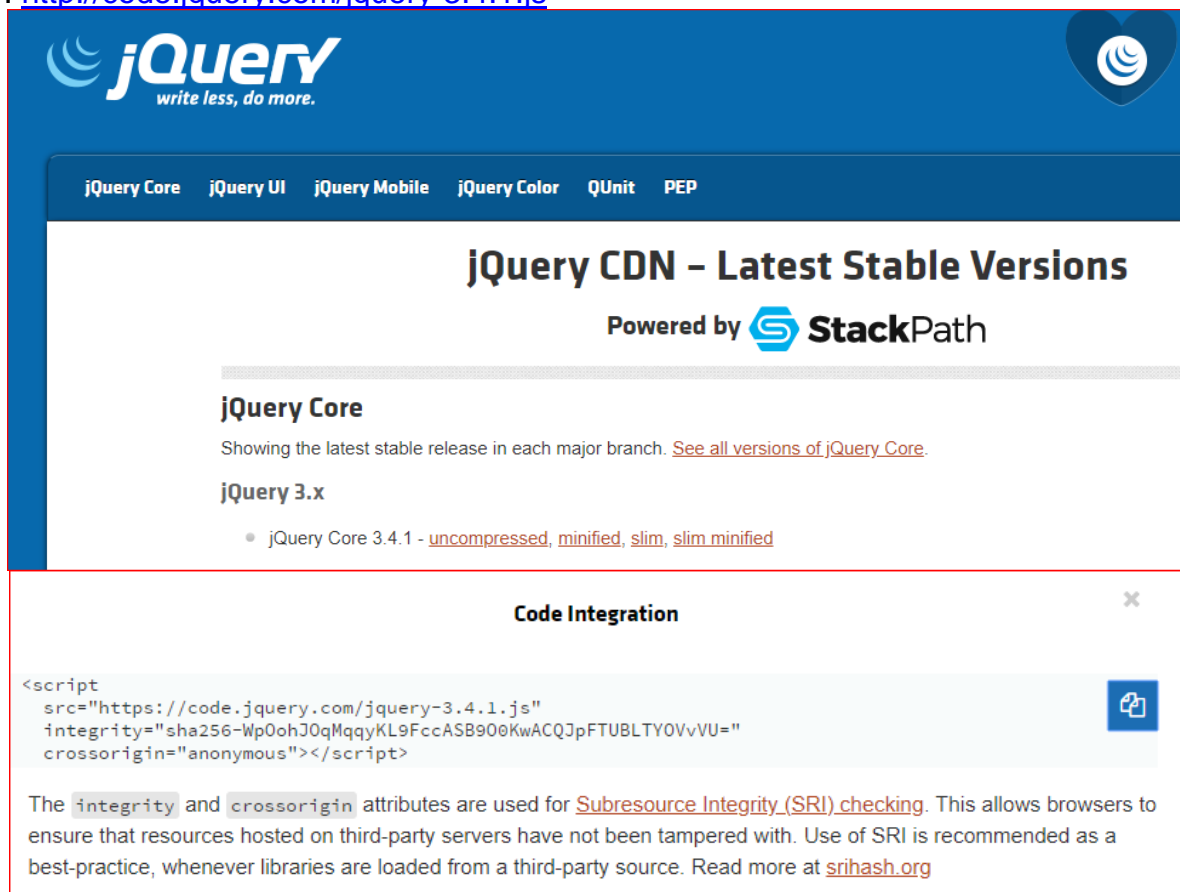
<body>
  <!--elementos html-->
  <a href="http://jquery.com/">jQuery</a>
  <p id="pfirst">First content</p>
  <!--inclusión de archivo js descargado-->
  <script src="jquery-3.4.1.js"></script>
  <!--inclusión de archivo js realizado-->
  <script src="js/example1.js"> </script>
</body>

```

2.- A través de una versión online de **CDN**(Red de distribución de contenidos) de: **Google**: <https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js>



jQuery: <http://code.jquery.com/jquery-3.4.1.js>



Esta manera reporta beneficios porque:

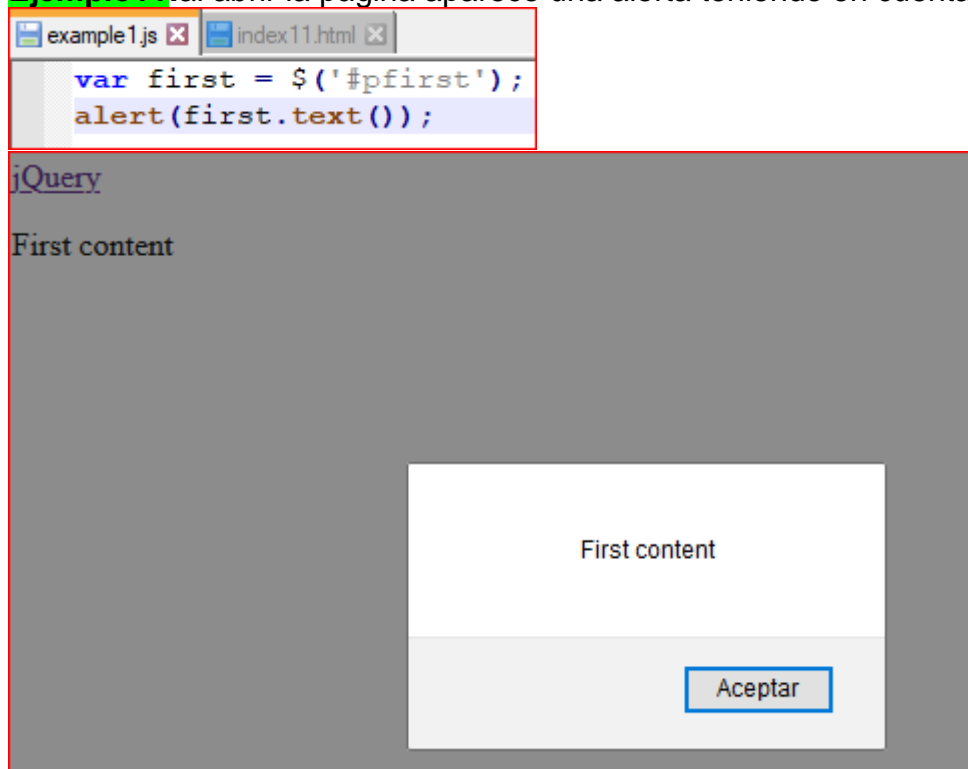
- Los CDNs tienen servidores distribuidos por todo el mundo, permitiendo servir el contenido más rápidamente pues se proporcionan los archivos desde el servidor más cercano a la ubicación del usuario.
- En los CDNs muy utilizados, es posible que el usuario haya entrado en algún sitio desde el mismo CDN que nuestra aplicación, por lo que algunos archivos ya estarán cachados en el navegador del usuario.

INCLUSIÓN DEL CÓDIGO SCRIPT : copiando el enlace **ejemplo11**

```
<body>
  <!--elementos html-->
  <a href="http://jquery.com/">jQuery</a>
  <p id="pfirst">First content</p>
  <!--inclusión de archivo js descargado-->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">
  </script>
  <!--inclusión de archivo js realizado-->
  <script src="js/example1.js"> </script>
</body>
```

- **Al final del body** en ambos casos, puesto que las buenas prácticas sobre rendimiento de aplicaciones indican que mientras se descargan los scripts pueden bloquear la descarga de otros contenidos importantes de la página como las imágenes.
- **En el head** cuando se utiliza la técnica de Google Analytics, mediante atributo **async** (crea el enlace al código JavaScript dinámicamente mientras que la carga se realiza asíncrona).
- Utilizando el **atributo defer** que indica al navegador que el código JavaScript se debe procesar después de que la página haya sido cargada completamente. Pero se producen también bloqueo y los scripts se ejecutan en el orden en el que han sido definidos.

Ejemplo11: al abrir la página aparece una alerta teniendo en cuenta el identificador



6. CONCEPTOS BÁSICOS

EJECUCIÓN DE CÓDIGO CUANDO LA PÁGINA HA SIDO CARGADA

- Para realizar acciones con Javascript la página debe haber sido cargada completamente.
- **jQuery** permite realizar acciones aunque queden elementos sin cargar mediante **`$(document).ready(function(){`**

```
console.log('document is ready');
});
```

O la versión abreviada

```
$(function(){
    console.log('document is ready');
});
```

Ejemplo12: al pulsar en el enlace se muestra un **alert**

```
<!--inclusión de código js -->
<script>
$(document).ready( //espera a que el documento esté listo
function() {
    $("a").click( //selecciona todos los enlaces
function(event){ //implementa el evento para el click
    alert("Hello everybody!"); //alerta
    });
}
);
</script>
```

jQuery

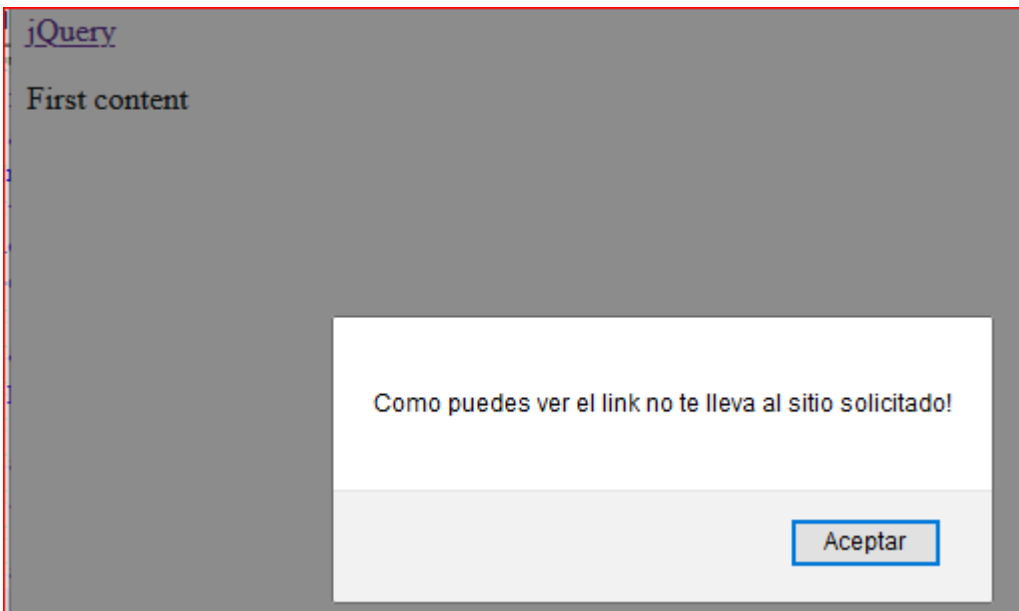
First content

Hello everybody!

Aceptar

Ejemplo13: igual al anterior pero cancelando el comportamiento por defecto **utilizando preventDefault**

```
</script>
<!--inclusión de código js -->
<script>
$(document).ready( //espera a que el documento esté listo
function() {
    $("a").click( //selecciona todos los enlaces
function(event){ //implementa el evento para el click
    alert("Como puedes ver el link no te lleva al sitio solicitado!");
    event.preventDefault(); //cancela el comportamiento por defecto
    });
}
);
</script>
```



7. FUNCIÓN `$()`

- Es la forma de interactuar con HTML
- Es equivalente a **`document.getElementById()`**
- Recibe por parámetro: **expresión**, selector CSS, **nombre de etiqueta HTML**
- Devuelve como resultado todos los **nodos** que coincidan con la expresión.

7.1. SELECCIÓN DE ELEMENTOS

- En <http://api.jquery.com/category/selectors/> encontramos todos los selectores

Category: Selectors

Borrowing from CSS 1–3, and then adding its own, jQuery offers a powerful set of tools for matching a set of document.

To use any of the meta-characters (such as `! "$ % & ' () * + , . / : ; < = > ? @ [\] ^ ` { | } ~`) as a literal part of a name, escaped with two backslashes: `\\`. For example, an element with `id="foo.bar"`, can use the selector `$("#foo\\.bar")`. The W3C CSS specification contains the [complete set of rules regarding valid CSS selectors](#). Also useful is Mathias Bynens on [CSS character escape sequences for identifiers](#).

All Selector ("*")

Selects all elements.

Also in:

:animated Selector

Select all elements that are in the progress of an animation at the time the selector is run.

Also in: [Selectors > Basic Filter](#) | [Selectors > Pseudo-Classes](#)

Attribute Contains Prefix Selector `[name|="value"]`

Also in: [Selectors > Basic Filter](#) | [Selectors > Pseudo-Classes](#)

Los más habituales son:

- `$('#pfirst')` devuelve el elemento con **id** pfirst
- `$('.enable')` devuelve los elementos de la **clase** enable
- `$('li.enable')` devuelve los li de la clase enable

- `$('#div.enable')` devuelve los elementos los elementos div de la clase enable
- `$('#li')` devuelve los items de la lista
- `$('#input[name=username]');` devuelve el atributo de nombre ...
- `$('#center ul.clients li');` : mediante selectores CSS
- `$('#p a')` : todos los enlaces descendientes de un párrafo
- `$('#div, #pfirst, h1.title')` Selecciones múltiples
- `$('#*');` : todos los elementos de la página
- Pseudoselectores
 - `$('#p[img]')` : todos los párrafos que contienen una imagen
 - `$('#login :input')` : todos los input del formulario login
 - `$('#p:lt(3)')` : los tres primeros párrafos del documento
 - `$('#div:hidden')` : todos los div ocultos

7.2. EJEMPLO DE USO DE SELECTORES

- Quitar una clase y añadir otra clase
`$(".enable").removeClass("enable").addClass("block");`
- Anidara componente con efectos gráficos
`$(".enable").slideToggle("slow");`
- Mostrar el contenido del enlace

```
<a id="link" class="linksClass" href="http://jquery.com">
alert($('#link').text());
```
- Mostrar el contenido de la página
`$("#body").append($("#linksClass").text());`
- Trabajar con elemento DOM a través de su selección

```
var lItems = $('#li');
var rawLItem = lItems [0]; // lItems.get(0)
var content = rawLItem.innerHTML;
```

Ejemplo 2: Utilización de id y selector de etiqueta

Al posicionar sobre el enlace el texto del párrafo con id cambia a rojo

Añade una clase en el head y posteriormente en el body incluye el script

```

<html>
  <head>
    <meta charset="utf-8">
    <title>Adding and removing classes</title>
    <style type="text/css">
      .disallowed{
        color: red;
        font-weight: bold;
      }
    </style>
  </head>
  <body>
    <a href="http://jquery.com/">jQuery</a>
    <p id="pfirst">First content</p>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">
    </script>
    $(document).ready(function(){//cuando el documento esté listo
      $("a").mouseover(function(event){//evento de posicionar en enlace
        $("#pfirst").addClass("disallowed");//añadir clase al identificador
      });
      $("a").mouseout(function(event){//al salir de posicionar en enlace
        $("#pfirst").removeClass("disallowed");//quita clase del identificador
      });
    });
  </script>
</body>
</html>

```

jQuery

First content

jQuery

First content

7.3. SELECTORES AVANZADOS

SELECTORES DE JERARQUÍAS

- `parent > child` hijos del elemento padre
- `el1 el2` elemento 2 que desciendan del 1
- `el1 + el2` elemento 2 inmediatamente precedidos por el elemento1
- `el1 ~ el2` elemento 2 precedidos por el elemento1, aunque existan elementos intermedios
- `:first-child` elementos que son el primer hijo de otro elemento
- `:last-child` elementos que son el último hijo de otro elemento
- `:nth-child(n)` elementos que son el nésimo hijo de su padre
- `:only-child` elementos que son el único hijo de su padre

7.3.1. EJEMPLO SELECTORES

Identificador y lista

Ejemplo3: index3: añade clase a los elementos de lista con identificador media


```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>EJEMPLO DE SELECTORES</title>
    <style type="text/css">
      .horizontal{ <!--clase a utilizar-->
        float: left;
        margin:20px;
        list-style:none;
      }
    </style>
  </head>
  <body>
    <ul id="media">
      <li>Series
        <ul>
          <li><a href="/sherlock/">Sherlock</a></li>
          <li>The Walking Dead</li>
          <li>Silicon Valley</li>
          <li>Braking Bad</li>
          <li>Game of Thrones</li>
        </ul>
      </li>
      <li>Films
        <ul>
          <li><a href="transformers.pdf">Transformers</a></li>
          <li>Interstellar</li>
          <li>The Hobbit</li>
          <li>The Imitation Game</li>
        </ul>
      </li>
      <li>Books
        <ul>
          <li>The Lord of the Ring
            (<a href="mailto:anaiesfp2@gmail.com">email</a>)
            <ul>
              <li> Part I</li>
              <li> Part II</li>
            </ul>
          </li>
          <li><a href="https://www.warnerbros.co.uk/brands/wizarding-world">Harry
            Potter</a></li>
          <li>The Hunger Games</li>
        </ul>
      </li>
    </ul>
  </body>
</html>

```

```

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"
></script>
<script>
    //aplicación de clase a los li del identificador media y posterior eliminación
    $(document).ready(
    function() { //cuando el documento esté listo
        $("#media > li").addClass("horizontal"); //añade clase al identificador para
        hijos del primer nivel
    });
</script>
</body>
</html>

```

Series

- [Sherlock](#)
- The Walking Dead
- Silicon Valley
- Braking Bad
- Game of Thrones

Films

- [Transformers](#)
- Interstellar
- The Hobbit
- The Imitation Game

Books

- The Lord of the Ring ([email](#))
 - Part I
 - Part II
- [Harry Potter](#)
- The Hunger Games

SELECTORES POSICIONALES

- **:first** primer elemento
- **:last** último elemento
- **:not(selector)** elementos que no coinciden con selector
- **:even** elementos impares, comenzando por 0
- **:odd** elementos pares, comenzando por 0
- **:eq(n)** elemento igual a puesto n
- **:gt(n)** elemento con índice mayor a n
- **:lt(n)** elemento con índice menor a n

EJEMPLO: EN EL ARCHIVO **index3.html** añade los siguientes selectores y cambia las propiedades a la clase como tamaño de fuente y color de fuente

- Al primer elemento de la lista
`$("#media li:first ").addClass("horizontal");`
- Al último elemento de la lista: cuidado quién es el último elemento
`$("#media li:last ")` `$("#media > li:last ")`
- No a las anclas:
`$("#media li:not(li a)")`
- A las listas pares

```
$("#media ul li:odd") $("#media ul:odd") $("#media >li:odd")
```

- A las listas impares

```
$("#media ul li:even") $("#media ul:even") $("#media >li:even")
```

- Lista de lugar 2

```
$("#media >li:eq(1)")
```

- La lista mayor que 1

```
$("#media >li:gt(0)")
```

FILTRADO DE CONTENIDO

- **:contains(text)** elementos que contienen texto

```
$("#media ul li ul:contains('Part')")
```

- **:empty** elementos sin hijos

- **:has(selector)** elementos que contienen al menos un elemento que coincide con selector

```
$("#media ul li:has(a)")
```

- **:parent** último elemento

```
$("#media ul li:parent")
```

- **:header** selecciona encabezados

- **:animated** elementos que forman parte de una animación

SELECTORES DE ATRIBUTO

- **:E[name]**: elementos que contienen el atributo con independencia de su valor.
- **:E[name=value]**: elementos que contienen el atributo con el valor dado.
- **:E[name^=value]**: elementos que contienen el atributo con el valor que empiece por la cadena especificada.
- **:E[name\$=value]**: elementos que contienen el atributo con el valor que termine por la cadena especificada.
- **:E[name*=value]**: elementos que contienen el atributo con el valor que contenga la subcadena especificada.
- **:E[name|=value]**: elementos que contienen el atributo con el valor dado o que empiezan por el valor dado seguido de un guion.
- **:E[name!=value]**: elementos que no contienen el atributo o que bien lo contienen, pero su valor no coincide con el especificado.
- **:E[name=value] [name2=value2]**: elementos coincidentes con dichos filtros.

EJEMPLOS DE SELECTORES DE ATRIBUTO

- **:img[alt]**: todas las imágenes que tengan atributos alt.
- **\$('a[href\$=".txt"]')**: enlaces con un atributo href que termine por txt.
- **\$('a[title^=Descarga]')**: enlaces cuyo título comience por Descarga.

```
$("a[href$='.pdf']")
```

- `$('a[href^="mailto:"]')`: los enlaces que apunten a direcciones de correo.
- `$('a[href^="http://:"]')`: los enlaces que apunten a páginas web.

`$("a[href^='mailto:']")`

`$("a[href^='https://']")`

FILTRADO POR VISIBILIDAD

- `:hidden`: elementos que están ocultos.
- `:visible`: elementos visibles.

ELEMENTOS DE FORMULARIO

- `:input`: elementos input, textarea, select y button.
- `:text`: inputs de tipo text.
- `:password`: inputs de tipo password.
- `:radio`: inputs de tipo radio.
- `:checkbox`: inputs de tipo checkbox.
- `:submit`: inputs de tipo submit.
- `:image`: inputs de tipo image.
- `:reset`: inputs de tipo reset.
- `:file`: inputs de tipo file.
- `:button`: inputs de tipo button o bien elementos button.
- `:hidden`: elementos de formulario ocultos.
- `:enabled`: elementos de formulario activados.
- `:disabled`: elementos de formulario desactivados.
- `:checked`: checkboxes y radio buttons que están pulsados.
- `:selected`: elementos de una lista de opciones que estén seleccionados.

Es posible combinar los selectores:

```
$( ':radio:checked' )
$( 'input[name=nombre]:radio:checked' )
```

SELECTORES ESPECIALES

<http://docs.jquery.com/Selectors>

Ejercicios para practicar: <https://elh.mx/jquery/iniciando-en-jquery-buenas-practicas-ejercicios-para-practicar-selectores-y-mas/>

7.4. COMPROBAR SELECCIÓN

```
if ( $( 'div.central' ).length ) { ... }
```

7.5. GUARDAR SELECCIONES MEDIANTE VARIABLES

```
var $listItems = $(li);
```

EL \$ no es obligatorio, pero se considera una buena práctica

7.6. FILTROS SOBRE ELEMENTOS SELECCIONADOS PREVIAMENTE

- `.not()` no cumplen una condición
 - `$('div').not('.active')`: div que no poseen la clase activo
 - `$("div").not(".verde, #azul").css("border-color", "red");` div que no sean del identificador azul ni de clase verde y se le aplica el estilo borde rojo

- **.filter()** cumplen una condición
 - `$('#ul li').filter('.active')`
 - `$('#li').filter(":even").css("background-color", "red")` para los item de la lista que sean impares establece color de fondo rojo
- **.slice(init,fin)** los elementos contenidos entre inicio y fin, sin incluir el último
 - `$('#div').slice(2,4)` div 2 y 3, excluyendo el 4(se empieza en 0)
- **.eq(index)** igual a posición índice (se empieza en 0)
 - `$('#ul li').eq(4)` quinto item de una lista desordenada
- **.first()** el primero
 - `$('#div').first()` devuelve el primer div
- **.has()** elementos que tienen
 - `$('#li').has('ul')` los elementos de la lista que contengan un ul interior

7.7. BUSCAR ELEMENTOS

- **.next()** hermano siguiente
- **.nextAll()** todos los hermanos siguientes
- **.prev()** hermano anterior
- **.prevAll()** todos los hermanos anteriores
- **.parent()** padre directo
- **.parents()** grupo de padres
- **.children()** hijos directos
- **.siblings()** hermanos únicos
- **.contents()** todos los hijos incluido el texto con `$(“p”).contents()`
- **.add()** añade elementos al selector
- **.closest()** comprueba si el elemento actual coincide con la condición especificada para retornarlo. En caso contrario, seguirá realizando la búsqueda hacia arriba, padrea padre hasta el primero que coincida con el patrón
- **.find()** elementos descendientes del selector

7.8. ENCADENAMIENTO DE MÉTODOS

- `$('#central').find('p').eq(3).html("text for the fourth paragraph");`
- El método **end()** permite reestablecer la selección a todos los elementos seleccionados antes de haber realizado las modificaciones
- El método **andBack()** permite seleccionar el elemento DOM anterior

7.9. OBTENER Y ESTABLECER CONTENIDO

Utilizando los métodos:

- **text()**: devuelve o establece el texto contenido en los elementos
- **html()**: devuelve o establece el texto y los marcadores de HTML
- **val()**: devuelve o establece el valor de los campos del formulario
- **attr()**: devuelve o establece el valor de los atributos de un elemento

- **removeAttr():** borra un atributo de un elemento
- **removeProp():** borra propiedad de un elemento

Ejemplo:

```
<body>
  <h1>PRUEBA SELECTORES</h1>
  <a href="https://www.cuatrovientos.org">cuatro vientos</a>

  <script src=
    "https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"
  ></script>
  <script>

    $(document).ready(
      function() { //cuando el documento esté listo
        alert($("#h1").html()); //muestra el contenido de h1
        //establece atributo en h1
        $("#h1").html("Hola mundo"); //modifica h1
        alert($("#a").attr("href")); //muestra el atributo de a
        $("#a").html("google"); //modifica a
        $("#a").attr("href", "http://www.google.es"); //cambia
        atributo de a

      });
    }

  </script>
</body>
</html>
```

Ejemplo html(): index8: Establece manejador para el **evento click** de todos los párrafos de la página. Luego con la **sentencia \$(this)** se captura el elemento que ha lanzado ese evento (en este caso el párrafo sobre el que hacemos click). Se guarda en una variable el html del párrafo y se imprime a continuación dentro del mismo elemento.

```

<html>
  <head>
    <meta charset="utf-8">
    <title>uso del método .html()</title>
    <style>
      p{
        margin:8px;
        font-size:20px;
        color:blue;
        cursor:pointer;//el cursor es un puntero
      }
      b{text-decoration:underline;}
    </style>
  </head>
  <body>
    <p><b>click</b> para cambiar el <span id="etiqueta">html</span> </p>
    <p>a modo <span id="texto">texto</span> </p>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"
    ></script>
    <script>
      $("p").click(
        function(){
          var entrada =$(this).html();//se introduce el código del html en entrada
          $this.text(entrada); //se pasa entrada dentro del mismo elemento.
        }
      );
    </script>
  </body>
</html>

```

Ejemplo removeProp() index10

Utiliza la propiedad código para añadir un valor, que posteriormente elimina.

Se usa un párrafo vacío al que se añaden las propiedades

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>ejemplo prop y removeProp </title>
  </head>
  <body>
    <p></p>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"
    ></script>
    <script>
      var $para=$("p");// el párrafo se guarda en variable para
      $para.prop("codigo", 1234); //inicializa la propiedad
      $para.append("El código del lugar secreto es:", String($para.prop("codigo")),
        " . "); // muestra la propiedad
      $para.removeProp("codigo"); //elimina propiedad
      $para.append(" Ahora el código del lugar secreto es: ", String($para.prop(
        "codigo")), " ."); // muestra la propiedad
    </script>
  </body>
</html>

```

El código del lugar secreto es:1234 . Ahora el código del lugar secreto es: undefined .

Ejemplo val(): index20

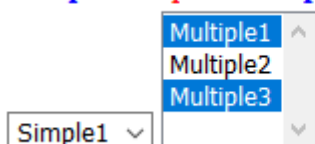

```

<title>ejemplo val </title>
<style>
    p{color:red; margin:4px;}
    b{color:blue;}
</style>
</head>
<body>
<p></p>
<select id="simple">
    <option> Simple1</option>
    <option> Simple2</option>
</select>
<select id="multiple" multiple="multiple">
    <option selected="selected"> Multiple1</option>
    <option> Multiple2</option>
    <option selected="selected"> Multiple3</option>
</select>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"
></script>
<script>
    function mostrarval() {
        var valorsimple = $("#simple").val(); //se guarda valor simple
        var valormultiple = $("#multiple").val(); //se guarda valores múltiples
        //se llevan a párrafos las variables
        $("p").html("<b>Simple:</b>" + valorsimple + "<b>Multiple:</b>" +
            valormultiple);
    }
    $("select").change(mostrarval); //mostrar valor se pasa a change para los select
    mostrarval(); //se ejecuta la función
</script>
</body>

```

Simple:Simple1 Multiple:Multiple1,Multiple3



7.10. AÑADIR O ELIMINAR CONTENIDO O ELEMENTOS

Utilizando los métodos:

- **append()**: inserta al final de los elementos seleccionados
- **prepend()**:inserta contenido antes de los elementos seleccionados
- **after()**:inserta después
- **before()**:inserta antes
- **remove()**: elimina elementos seleccionados y sus hijos
- **empty()**:elimina hijos de los elementos seleccionados

7.11. MANIPULACIÓN DE CSS

- **addClass()**: añade una o más clases a los elementos seleccionados
- **removeClass()**:elimina una o más clases a los elementos seleccionados
- **hasClass()**:tiene clase determinada
- **toggleClass()**:cambia entre adición/eliminación de clases dependiendo si la clase ya estaba presente o no
- **css()**:establece o devuelve atributo de estilo

EJEMPLO: **css()** index5.html


```

<script>
    $(document).ready(
        function() { //cuando el documento esté listo
            alert($("#h1").css("font-size")); //devuelve el valor css
            $("#a").css("font-size", "30px"); //modifica el valor css
            $("#h1").css({"color" : "red", "background-color" : "yellow" });
            //modifica varios valores css
        });
    </script>

```

Ejemplo addclass(): index6 añade la clase “selected” al último párrafo de la página y la clase resaltado al primer párrafo

```

index6.html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>OBTENER Y ESTABLECER CONTENIDO</title>
    <style>
      .selected{color:blue;}
      .resaltado{background-color:yellow;}
    </style>
  </head>
  <body>
    <p>PÁRRAFO1</p>
    <p>PÁRRAFO2</p>
    <p>PÁRRAFO3</p>
    <script src=
      "https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"
    ></script>
    <script>
      $(document).ready(
        function() { //cuando el documento esté listo
          $("#p:last").addClass("selected"); //añade clase al último
          párrafo
          $("#p:first").addClass("resaltado"); //añade clase al primerr
          párrafo
        });
      </script>
    </body>
  </html>

```

ACTIVIDAD 4.1: Selecciona el primer elemento de la primera lista y oculta dicho elemento utilizando la propiedad adecuada de CSS.

- Lista 1:
 - o Tortilla
 - o Jamón
 - o Queso
- Lista 2:
 - o Coca Cola
 - o Leche
 - o Té

ACTIVIDAD 4.2: Convertir el siguiente código JavaScript a JQuery

```

<script type="text/javascript">
    var setList = function () {
        var x = document.getElementById("number").value;
        var lista = "<ul>";
        for (var i = 1; i <= x; i++) {
            lista += "<li>" + i + "</li>";
        }
        lista += "</ul>";
        document.getElementById("contenedorLista").innerHTML = lista;
    };
</script>

```

Ejemplo hasClass y append():index7

```

<style>
    .selected{color:blue;}
    .resaltado{background-color:yellow;}
</style>
</head>
<body>
    <p class="selected">PÁRRAFO1</p>
    <p class="resaltado">PÁRRAFO2</p>
    <p>PÁRRAFO3</p>
    <div id="resultado1"> El primer párrafo tiene la clase selected:</div>
    <div id="resultado2"> El segundo párrafo tiene la clase resaltado:</div>
    <div id="resultado3"> El tercer párrafo tiene la clase selected o resaltado:
    </div>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"
    ></script>
    <script>
    $(document).ready(
    function(){//cuando el documento esté listo
        //añade al div si el primer párrafo tiene la clase selected
        $("div#resultado1").append($("p:first").hasClass("selected").toString());
        //añade al div si el segundo párrafo tiene la clase
        $("div#resultado2").append($("p:eq(1)").hasClass("resaltado").toString());
        //añade al div si el tercer párrafo tiene la clase
        $("div#resultado3").append($("p:last").hasClass("selected").toString());
    });
    </script>
</body>

```

PÁRRAFO1

PÁRRAFO2

PÁRRAFO3

El primer párrafo tiene la clase selected:true

El segundo párrafo tiene la clase resaltado:true

El tercer párrafo tiene la clase selected o resaltado:false

Ejemplo:removeClass():index8: quita clase de los párrafos impares

```

<head>
  <meta charset="utf-8">
  <title>uso del método .html()</title>
  <style>
    p{
      margin:4px;
      font-size:16px;
      font-weight:bold;
    }
    .azul{color:blue;}
    .debajo{text-decoration:underline;}
    .resaltado{background-color:yellow;}
  </style>
</head>
<body>
  <p class="azul debajo resaltado">Hola</p>
  <p class="azul debajo resaltado">Hola1</p>
  <p class="azul debajo resaltado">Hola2</p>
  <p class="azul debajo resaltado">Hola3</p>

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4
  ></script>
  <script>
    //quita la clase azul y resaltado en los párrafos impares
    $("p:even").removeClass("azul resaltado");
  </script>
</body>
</html>

```

Hola

Hola1

Hola2

Hola3

Ejemplo toggleClass() index9

Cambia de clase al pulsar sobre el párrafo

```

<title>ejemplo toggleClas</title>
<style>
  p{
    margin:4px;
    font-size:16px;
    font-weight:bolder;
  }
  .azul{color:blue;}
  .debajo{text-decoration:underline;}
  .resaltado{background-color:yellow;}
</style>

</head>
<body>
  <p class="azul ">Hola</p>
  <p class="azul ">Hola1</p>
  <p class="azul ">Hola2</p>
  <p class="azul ">Hola3</p>

  <script src="https://ajax.googleapis.com/ajax/libs
  ></script>
  <script>
    //al hacer clic en párrafo cambia de clase
    $("p").click(
      function() {
        $(this).toggleClass("resaltado");
      }
    );
  </script>
</body>

```

Hola
 Hola1
 Hola2
 Hola3

7.12. TRABAJANDO CON DIMENSIONES

- width():ancho sin incluir relleno ni borde ni margen
- height():alto sin incluir relleno ni borde ni margen
- innerwidth():ancho con relleno
- innerheight()altura con relleno
- outerwidth()ancho con relleno y borde
- outerheight()ancho con relleno y borde

7.13. RECORRIDO DE COLECCIONES CON EACH

- Equivalente al clásico for
- **Sintaxis**

```

$.each( array,
  function(i, value){
    ...acciones...
  });

```

array: lista de elemntos **i:**índice que se actualiza solo **value:**cada valor del array

Ejemplo:Index22

```

<html>
  <head>
    <title>index22</title>
    <meta charset="utf-8">
  </head>
  <body>
    <p>ejemplos de bucle para recorrer array</p>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"
    ></script>
    <script>
      //ejemplo con vector de 2 números enteros
      $.each([50,90],
        function(i,value){ //array
          document.write( "Posición " + i + " = " + value + "<br>");
        });
      //ejemplo con vector de nombres que llevan su propio índice
      var estudiantes = {"nombre-0":"Juan","nombre-1":"Enrique"};
      $.each(estudiantes,
        function(j,value){ //array
          document.write( j + " = " + value + "<br>");
        });
    </script>
  </body>
</html>

```

Ejemplos de bucle para recorrer array

Posición 0 = 50

Posición 1 = 90

nombre-0 = Juan

nombre-1 = Enrique

7.14. EJEMPLOS DE MANIPULACIÓN DEL DOM

.after() .append() .before() .prepend() .css() .empty() .height() .remove() .replaceAll

.after():inserta después del elemento seleccionado

Ejemplo :index23

```

<style>
  p{background:yellow;}
</style>
</head>
<body>
  <p>Ejemplo de after</p>
  <script src="https://ajax.googleapis.com
  ></script>
  <script>
    $("p").after("<b>Hola</b>")
  </script>

```

Ejemplo de after

Hola

.append():añade al final del elemento seleccionado

Ejemplo :index24

```

<style>
    p{background:yellow;}
</style>
</head>
<body>
    <p>Ejemplo de append </p>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">
    </script>
    <script>
        $ ("p").append("<b>Hola</b>")
    </script>

```

Ejemplo de append **Hola**

.before():contrario a after

Ejemplo :index25

```

<style>
    p{background:yellow;}
</style>
</head>
<body>
    <p>Ejemplo de before </p>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">
    </script>
    <script>
        $ ("p").before("<b>Hola</b>")
    </script>

```

Hola

Ejemplo de before

.prepend():contrario a append

Ejemplo :index25

```

<script>
    $ ("p").prepend("<b>Hola</b>")
</script>

```

Hola Ejemplo de before

.css():extrae o establece propiedad css

Ejemplo:index26

```

<title>index23</title> <meta charset="utf-8">
<style>
    div{
        width:100px;
        height:100px;
        margin:5px;
        float:left;
    }
</style>
</head>
<body>
    <div style="background-color:blue;" ></div>
    <div style="background-color:rgb(15,99,30);" ></div>
    <div style="background-color:#123456;" ></div>
    <div style="background-color:#f11;" ></div>
    <div id="resultado" style="clear:both;border:2px solid orange" ></div>
    <script src=
    "https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script>
    //al hacer click en el div obtiene su color de fondo
    $("div").click(
        function(){
            //se guarda color en variable
            var color=$(this).css("background-color");
            //se lleva el color al div con id resultado
            $("#resultado").html("El color es: " + "<span style='color=color'"
            + color + "</span>");
        }
    );
    </script>
</body>

```



El color es:
rgb(0, 0, 255)

.empty():borra nodos hijos del elemento seleccionado

Ejemplo:index27

```

<html>
  <head>
    <title>index27</title>  <meta charset="utf-8">
  <body>
    <p>Eso es un párrafo a eliminar cuando se pulsa el botón</p>
    <button style="cursor:pointer">Llama la la función para eliminar el párrafo
    </button>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"
    ></script>
    <script>
      //al hacer click en botón borra el párrafo
      $("button").click(function () {
        $("p").empty();
      });
    </script>
  </body>
</html>

```

.height():obtiene la altura del elemento seleccionado

Ejemplo:index28

```

<html>
  <head>
    <title>index27</title>  <meta charset="utf-8">
    <style>
      body{ background-color:yellow;}
      button{font-size:12px;margin:2px;}
      p{width:150px;border:1px red solid;text-align:center;}
      div{color:red;font-wight:bold;}
    </style>
  <body>
    <p>Eso es un párrafo</p>
    <button id="hp" style="cursor:pointer">Obten Altura del párrafo</button>
    <button id="hd" style="cursor:pointer">Obten Altura del documento</button>
    <button id="hw" style="cursor:pointer">Obten Altura de la ventana</button>
    <div></div>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"
    ></script>
    <script>
      //al hacer clic en cada botón se obtiene la altura de de los elementos
      $("#hp").click(function () {
        $("div").text("La Altura del Párrafo es:" + $("p").height());
      });
      $("#hd").click(function () {
        $("div").text("La Altura del Documento es:" + $(document).height());
      });
      $("#hw").click(function () {
        $("div").text("La Altura de la Ventana es:" + $(window).height());
      });
    </script>
  </body>
</html>

```

Eso es un párrafo

Obten Altura del párrafo

Obten Altura del documento

Obten Altura de la ventana

La Altura del Documento es:236

.replaceAll():reemplaza todos los elemento seleccionados por un elemento dado

Ejemplo: index29


```

<html>
  <head>
    <title>index29</title> <meta charset="utf-8">
  <body>
    <p>Eso es un párrafo1 </p>
    <p>Eso es un párrafo2 </p>
    <p>Eso es un párrafo3 </p>

    <div></div>
    <script src="https://ajax.googleapis.com/ajax/libs/jc
    ></script>
    <script>
      //se reemplazan los párrafos por Parrafo en negrita
      $("<b> Parrafos </b> <br>").replaceAll("p");
    </script>
  </body>
</html>

```

Parrafos
Parrafos
Parrafos

.remove():borra elemento seleccionado

Ejemplo: index30

```

<html>
  <head>
    <title>index30</title> <meta charset="utf-8">
    <style>
      body{ background-color:yellow;}
      button{font-size:12px;margin:2px;}
      p{width:150px;border:1px red solid;text-align:center;}
      div{color:red;font-wight:bold;}
    </style>
  <body>
    <p>Eso es un párrafo</p>
    <button id="hp" style="cursor:pointer">Borra el párrafo</button>

    <div></div>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/j
    ></script>
    <script>
      //al hacer clic en cada botón se borra el párrafo
      $("button").click(function () {
        $("p").remove();
      });
    </script>
  </body>
</html>

```

Eso es un párrafo

Borra el párrafo

Borra el párrafo

ACTIVIDAD 4.3

7.15. EVENTOS

Ya hemos utilizada el evento **click**

```
$("#link").click(function() {
    alert("Hello");
});
```

Cuyo equivalente en Javascript (más código) es **onclick** :

```
<a id="link" href="#" onclick="functionName('value')">Clic aquí</a>
```

JQuery permite **vincular** un evento mediante el método **on** y **desvincular** con **off**

```
$('#link').on('click', function() {
    console.log('Hello');
});
```

Para consultar la **lista de eventos** de JQuery:

<http://api.jquery.com/category/Events/>

Los **eventos más habituales** son:

.bind(): asigna un evento a un manejador de eventos y permite asignar múltiples eventos a la vez. Realmente click() es la manera corta de llamar a bind("click")

ejemplo: index40

```
<style>
    .test{
        height:100px; width:100px;
        background-color:red;
    }
    .inside{
        height:100px; width:100px;
        background-color:blue;
    }
    .outside{
        height:100px; width:100px;
        background-color:yellow;
    }
</style>
</head>
<body>
    <div class="test"> lugar para aplicar clase </div>
    <script src=
    "https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.
    <script>
        $("div.test").bind({
            mouseenter:function(){ // al entrar con el mouse
                $(this).addClass("inside");
            },
            mouseleave:function(){ // al salir con el mouse
                $(this).addClass("outside");
            }
        });
    </script>
```

La documentación de JQuery aconseja utilizar on() en lugar de bind()

.delegate(): es similar a bind() ya que no sirve para capturar eventos, sino para asignarlos a manejadores, con la diferencia que puede asignar eventos a elementos que aún no se encuentran en la página.

```

<p>Pulsa sobre mí</p>
<script src="https://ajax.googleapis.com/ajax/libs
"></script>
<script>
    //al hacer clic en el párrafo delega a que des
    $("body").delegate("p","click", function() {
        $(this).after("<p> Otro párrafo</p>");
    });
</script>

```

Pulsa sobre mí

Otro párrafo

.click(): se dispara al hacer clic en el elemento seleccionado

Ejemplo

```

<body>
    <div id="target">
        Click here
    </div>
    <div id="other">
        Trigger the handler
    </div>

    <script>
        $("#target").click(function () {
            alert("Handler for .click() called.");
        });
    </script>

```

.blur(): pierde el foco de selección

Ejemplo

```

<form>
    <input type="text" id="campo1" value="campo1"></input>
    <input type="text" value="campo2"></input>
</form>
<div> Disparador del evento handler está ligado al pimer input </div>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.
"></script>
<script>
    //al salir del campo1 se activa el blur
    $("#campo1").blur(function() {
        document.write("Se ha llamado a handler para blur");
    });
</script>

```

campo1

campo2

Disparador del evento handler está ligado al pimer input

Se ha llamado a handler para blur

.focus(): se dispara cuando el elemento recibe el foco(contrario a blur)

Ejemplo

index45

```

<p>
  <input type="text"></input>
  <span style="display:none">Encendido el foco</span>
</p>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery"></script>
<script>
  //al poner el foco en input
  $("input").focus(function() {
    //hace visible el párrafo con css
    //y luego lo oculta con fadeOut durante 1000 segundos
    $(this).next("span").css("display", "inline").fadeOut(1000);
  });
</script>

```

Encendido el foco

.hover(): se dispara cuando el puntero del ratón pasa por encima

Ejemplo: index46

```

ul{margin-left:20px; color:blue;}
li{cursor:default;}
span{color:red;}
</style>
</head>
<body>
  <ul>
    <li>Leche</li>
    <li>Pan</li>
    <li class="fade">Galletas</li>
    <li class="fade">Calcetines</li>
  </ul>
  <script src=
  "https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  //al pasar el ratón por los elementos de la lista
  $("li").hover(
    function() {
      //añade ****
      $(this).append("<span>****</span>");
    },
    function() {
      //encuentra la última tira de **** y la borra
      $(this).find("span:last").remove();
    }
  );
  //al pasar por los elementos de fade los desaparecer y aparece
  $("li.fade").hover(function() {
    $(this).fadeOut(100);
    $(this).fadeIn(100);
  });
</script>

```

- Leche****
- Pan
- Galletas
- Calcetines

.change(): cambia el valor de un elemento y tiene 3 controles: input textarea select

Ejemplo: index42

```

<form>
  <select class="opciones">
    <option value="opcion1" >Opción1</option>
    <option value="opcion2" >Opción2</option>
  </select>
</form>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">
</script>
<script>
  //avisa cuando cambia de opción seleccionada
  $(".opciones").change(function() {
    document.write("Se ha cambiado de opción");
  });
</script>

```

Opción1 ▾

Se ha cambiado de opción

Event.preventDefault(): cancela el comportamiento por defecto de una determinada acción. Por ejemplo cuando pinchamos en un enlace en lugar de redirigirnos a la dirección indicada podremos cancelar esa redirección

```

<a href="http://code.jquery.com">Al hacer clic va a cancelar la redirección a
la página de jquery</a>
<div></div>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">
</script>
<script>
  //al hacer clic en ancla previene de ir al enlace
  $("a").click(function(event) {
    event.preventDefault(); //previene
    //avisa en div
    $("div").append("Se ha detenido el evento por defecto: "+event.type);
  });
</script>

```

Al hacer clic va a cancelar la redirección a la página de jquery

Se ha detenido el evento por defecto: click

.on(): vincula manejadores al elemento seleccionado. Es el método recomendado a partir de la versión 1.7 y se puede revertir con el método off()

Ejemplo: index47

```

<form>
  <input type="text" id="tarea" value="campo1"></input>
  <input type="text" value="campo2"></input>
</form>
<div id="otro">Disparador del handler</div>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">
</script>
<script>
  //el evento se liga al campo1 y se desencadena blur
  $("#tarea").on("blur",
    function() {
      document.write("El disparador blur ha sido llamado");
    });
</script>

```

El disparador blur ha sido llamado

.ready(): todo lo que incluya se ejecuta cuando la página esté disponible

Ejemplo: index48

```
<style>
  p{color:red;}
</style>
</head>
<body>
  <p>No cargado todavía</p>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4
  ></script>
  <script>
    $(document).ready(function() {
      $("p").text("Párrafo cargado para su manipulación");
    });
  </script>
```

Párrafo cargado para su manipulación

.submit(): se dispara cuando se intenta enviar un formulario mediante los elementos de

<input type="submit">

<input type="image">

< button="submit">

Ejemplo: index49

```
<style>
  p{margin:0;color:blue;}
  div,p{margin-left:10px;}
  span{color:red;}
</style>
</head>
<body>
<p>Introduce correcto para validar</p>
  <form action="javascript:alert("Éxito!!!");">
    <div>
      <input type="text" ></input>
      <input type="submit" ></input>
    </div>
  </form>
  <span></span>
  <script src=
  "https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.
  <script type="text/javascript">
    $("form").submit(function() {
      //si en el primer input se introduce el valor correcto
      if($("input:first").val()=="correcto"){
        //en el span pone el texto Válido
        $("span").text("Válidado...");
        return true;
      }
      //sino en span pone Inválido y lo hace desaparecer
      $("span").text("INVálidado....").fadeOut(1000);
      return false;
    });
  </script>
```

Introduce correcto para validar

Válidado....

ACTIVIDAD 4.4

7.16. EFECTOS

jQuery ofrece la posibilidad de crear muchos efectos. La lista completa la puedes consultar en:

<http://docs.jquery.com/Effects>

Los más utilizados son: **show()** **hide()** **toggle()** **slideDown()** **slideUp()** **fadeIn()**

.show(): muestra los elementos si en ese momento están ocultos

.hide(): oculta los elementos que en este momento estén visibles

.toggle(): combinación de los 2 anteriores: oculta/visualiza

Ejemplo

```
<p id="uno" style="display:none">Hola1</p>
<a href="#">Pulsa en este enlace</a>
<p id="dos" style="display:none">Hola2</p>
<button>Pulsa en este botón </button>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4
></script>
<script>
    $(document).ready(function() {
        $("p").show();//muestra párrafo oculto
        $("a").click(function() { //al hacer clic en el enlace
            $("#uno").hide();//oculta párrafo visible
            $(this).hide();//oculta enlace
        });
        $("button").click(function() { //al hacer clic en botón
            $("#dos").toggle();//aparece o desaparece
        });
    });
</script>
```

.slideDown(): hace aparecer un elemento mediante un efecto de cortina descendente

Ejemplo


```

<style>
    div{
        background-color:red; margin:3px;
        width:80px;height:40px; display:none; float:left;
    }
</style>
</head>
<body>
    Clickea sobre mí
    <div></div>
    <div></div>
    <div></div>
    <div></div>
    <script src=
    "https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery
    <script>
        $(document).ready(function() {
            $(document.body).click(function() {
                if($("#div:first").is(":hidden")){
                    $("#div").slideDown("slow");
                }
                else{
                    $("#div").hide();
                }
            });
        });
    </script>

```

.slideUp() :igual al anterior pero ascendente

Ejemplo

```

    div{
        background-color:red; margin:3px;
        width:80px;height:40px; display:none; float:left;
    }
</style>
</head>
<body>
    Clickea sobre mí
    <div></div>
    <div></div>
    <div></div>
    <div></div>
    <script src=
    "https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.m
    <script>
        $(document).ready(function() {
            $(document.body).click(function() {
                if($("#div:first").is(":hidden")){
                    $("#div:first").show("slow");
                }
                else{
                    $("#div").slideUp();
                }
            });
        });
    </script>

```


.fadeIn():hace aparecer poco a poco un elemento ajustando su opacidad

.fadeOut():hace desaparecer poco a poco un elemento ajustando su opacidad

Ejemplo

```
<style>
    div{
        margin:3px;
        width:80px;height:40px; display:none; float:left;
    }
    span{color:red;cursor:pointer;}
    div#uno{background-color:red;}
    div#dos{background-color:green;}
    div#tres{background-color:blue;}
</style>
</head>
<body>
    <p>Si clickeas sobre este párrafo lo verás desaparecer</p>
    <span>Clickea sobre mí</span>
    <div id="uno"></div>
    <div id="dos"></div>
    <div id="tres"></div>
    <script src=
    "https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.js">
    <script>
        $(document).ready(function() {
            $(document.body).click(function() {
                $("div:hidden:first").fadeIn("slow");
            });
            $("p").click(function() {
                $("p").fadeOut("slow");
            });
        });
    </script>
```

7.17. JQUERY UI

<http://jqueryui.com/>

EFFECTOS

- Para utilizar los efectos es necesario referenciar la librería <https://code.jquery.com/ui/>

The screenshot shows the 'jQuery UI - All Versions' page, powered by StackPath. A 'Code Integration' modal is open, displaying a script tag for jQuery UI 1.12.1 with SRI attributes. Below the script tag, a note explains that the 'integrity' and 'crossorigin' attributes are used for Subresource Integrity (SRI) checking to ensure resources have not been tampered with, recommending SRI as a best-practice for third-party libraries.

```
<script
src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"
integrity="sha256-T0Vest3yCU7pafRw9r+settMBX6JkKN06dqBnpQ8d30="
crossorigin="anonymous"></script>
```

The `integrity` and `crossorigin` attributes are used for [Subresource Integrity \(SRI\) checking](#). This allows browsers to ensure that resources hosted on third-party servers have not been tampered with. Use of SRI is recommended as a best-practice, whenever libraries are loaded from a third-party source. Read more at srihash.org

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.js">
</script>
<!--NUEVA LIBERÍA PARA EL UI-->
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
</script>
```

- Adicionalmente es necesario incluir el **theme** u hoja de estilos asociada a la librería:

jQuery UI 1.12

jQuery UI 1.12.1 - [uncompressed](#) [minified](#)

Themes

[base](#) [black-tie](#) [blitzer](#) [cupertino](#) [dark-hive](#) [dot-luv](#) [eggplant](#) [excite-bike](#) [flick](#) [hot-sneaks](#) [humanity](#) [le-frog](#) [mint-choc](#) [overcast](#) [pepper-grinder](#) [redmond](#) [smoothness](#) [south-street](#) [start](#) [sunny](#) [swanky-purse](#) [trontastic](#) [ui-darkness](#) [ui-lightness](#) [vader](#)

```
<!--NUEVO ENLACE PARA LAS CLASES DE LOS EFECTOS DISPONIBLES EN themes-->
<link rel="stylesheet" href=
"https://code.jquery.com/ui/1.12.1/themes/smoothness/jquery-ui.css"/>
<style type="text/css">
```

- Todos los efectos están en: <https://jqueryui.com/effect/>

EJEMPLO: index60:formulario con efecto en botones

Nombre Usuario

Contraseña

Login

Bounce Highlight Pulsate Shake

```
<html>
  <head>
    <title>index60</title>
    <meta charset="utf-8">
    <!--NUEVO ENLACE PARA LAS CLASES DE LOS EFECTOS DISPONIBLES EN themes-->
    <link rel="stylesheet" href=
      "https://code.jquery.com/ui/1.12.1/themes/smoothness/jquery-ui.css"/>
    <style type="text/css">
      *{font:normal 12px arial,verdana;}
    </style>
  </head>
  <body>
    <form id="loginsimple" style="border:3px solid red;width:300px;paddin:6px;">
      <p>
        <label for="username"> Nombreusuario</label>
        <input id="username" type="text"></input>
      </p>
      <p>
        <label for="password"> Contraseña</label>
        <input id="password" type="text"></input>
      </p>
      <p>
        <input type="submit" value="Login"></input>
      </p>
    </form>
    <p>
      <input type="button" id="bounce" value="Bounce"></input>
      <input type="button" id="highlight" value="Highlight"></input>
      <input type="button" id="pulsate" value="Pulsate"></input>
      <input type="button" id="shake" value="Shake"></input>
    </p>
```

```

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.js"
></script>
<!--NUEVA LIBERÍA PARA EL UI-->
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
<script>
    //al hacer clic en los botones con id se produce el efecto
    $("#bounce").click(function() {
        $("#loginsimple").effect("bounce");
    });
    $("#highlight").click(function() {
        $("#loginsimple").effect("highlight");
    });
    $("#pulsate").click(function() {
        $("#loginsimple").effect("pulsate");
    });
    $("#shake").click(function() {
        $("#loginsimple").effect("shake");
    });
    $.webshims.polyfill();//para que sea visible en todos los navegadores
</script>
</body>
</html>

```

COMPATIBILIDAD HACIA ATRÁS HTML5

Utilizando librerías como **modernizr** y **polyfiller**

```

<script type="text/javascript" src="js/modernizr-custom.js"></script>
<script type="text/javascript" src="js/polyfiller.js"></script>
<script type="text/javascript">

```

Para descargar las librerías: <http://afarkas.github.io/webshim/demos/>

ACTIVIDAD 4.5

PARTE2: JQUERY AJAX

- **AJAX**(Asynchronous JavaScript and XML)
 - permite establecer una comunicación asíncrona con el servidor para intercambiar datos.
 - La comunicación usa protocolo HTTP
 - Permite establecer una comunicación del entorno cliente con el servidor sin que se produzca refresco en el navegador(sin necesidad de recargar la página del navegador)
 - Ofrece una web dinámica con diseño más atractivo
 - Ej:autocompletado en un campo de búsqueda, un chat web
 - Su funcionamiento es similar a Javascript pero de una manera más comprensible y elegante
- **jQuery** dispone de funciones relacionadas con **AJAX**
 - la función principal es jQuery.ajax()
 - los parámetros a utilizar se pueden consultar en <https://api.jquery.com/category/ajax/>

Ejemplo

```
$.ajax({
  type: "POST",
  url: "request.php",
  data: { name: "César", location: "Oviedo" }
}).done(function( response ) {
  alert( "Data Received: " + response);
});
```

type: especifica el tipo de petición; GET O POST

url: dirección del recurso al cual vamos a enviar la petición AJAX

data: parámetros que se envían en la llamada expresados en un array

done: especifica la función que será ejecutada tras la finalización de la llamada

Otros parámetros de uso común son:

- **dataType:** permite especificar el formato de los datos esperado para la respuesta (xml, json, jsonp, html, text, script)
- **async:** establece si la petición será asíncrona o no. De forma predeterminada el valor es **true**. Si se establece a **false**, la petición bloqueará la ejecución de otros códigos hasta que dicha petición haya finalizado.
- **cache:** establece si la petición será guardada en la caché del navegador. De forma predeterminada es **true** para todos los dataType excepto para script y jsonp.

Funciones de callback

Una vez se termina la petición AJAX existen tres métodos para recoger los datos recibidos, mediante las funciones:

- **done():** cuando la llamada es satisfactoria
- **fail():** cuando se produce algún error
- **always():** cuando la llamada se completa, independiente si hay error o no

Métodos útiles relacionados con las llamadas:

- **beforeSend():** para antes de realizar la llamada
- **dataFilter():** invocada inmediatamente después de recibir una respuesta satisfactoria. Recibe el dataType y el valor de la respuesta.

Ejemplo

```
$.ajax({
  type: "POST",
  url: "request.php",
  data: {name: "César", location: "Oviedo"}
}).done(function (msg) {
  $("body").append("Data Received: " + msg);
}).fail(function () {
  $("body").append("Error!");
}).always(function () {
  $("body").append("<br />This is always written");
});
```

1. MÉTODOS CONVENIENTES O SHORTCUTS

\$.get: realiza petición GET a una URL provista

\$.post: realiza petición POST a una URL provista

\$getScript: añade un script a la página

\$getJSON: realiza una petición GET a una URL provista y espera que un dato JSON sea devuelto.

Los métodos anteriores deben tener en orden los siguientes argumentos:

- **url:** en donde se realizará la petición
- **data:** con información a enviar al servidor. Es opcional y puede ser tanto una seire de **parámetros** como un objeto. No es valida para \$getScript
- **success callback:** función que se ejecuta en cas que la petición haya sido satisfactoria
- **dataType:** opcionalmente se puede especificar el tipo de dato que se espera recibir.

Ejemplo:

```
$.post('request.php', {name:"César", location:"Oviedo"}, function(msg){  
    alert("Data Received: "+msg);  
});
```

EJEMPLO GET: index70:probar con xampp

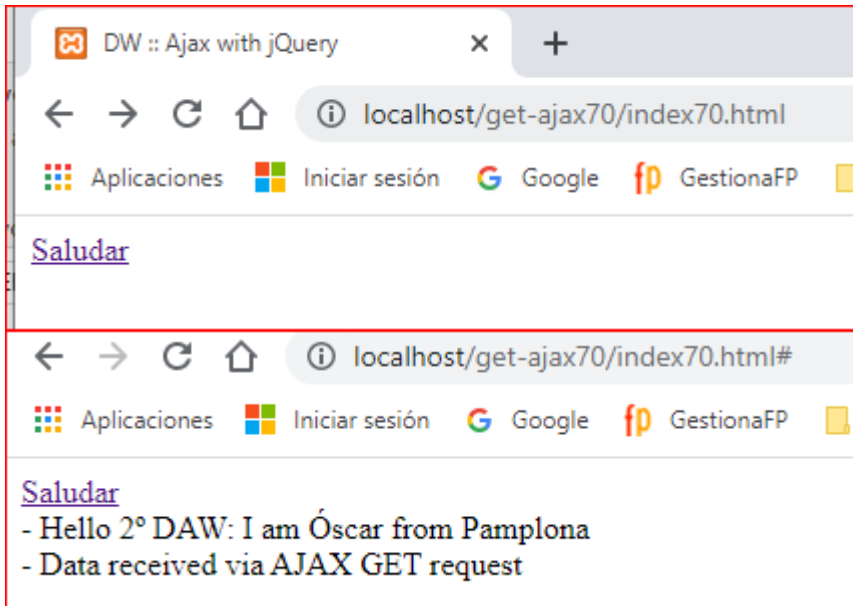
```
index70.html x get-ajax70.js x get-ajax2.php x  
<!DOCTYPE html>  
- <html>  
-   <head>  
       <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
       <meta charset="utf-8">  
       <title>DW :: Ajax with jQuery</title>  
   </head>  
-   <body>  
       <a href="#">Saludar</a>  
       <div id="resultsId"></div>  
       <script src="http://code.jquery.com/jquery-3.4.1.min.js" type=  
       "text/javascript"></script>  
       <script src="get-ajax70.js" type="text/javascript"></script>  
   </body>  
</html>
```

```
index70.html x get-ajax70.js x get-ajax2.php x  
- function helloDAW2 () {  
    $.get("get-ajax2.php",  
        {name: "Óscar", location: "Pamplona"},  
-    function (data) {  
        $("#resultsId").html(data);  
    });  
};  
  
//Esto es lo primero que se ejecuta  
//Es lo mismo que: $( document ).ready(function() {  
- $(function () {  
-     $('body a').on('click', function () {  
         helloDAW2 ();  
     });  
});
```

```

index70.html x get-ajax70.js x get-ajax2.php x
- <?php
  //isset, devuelve true si está definida
- if (isset($_GET['name']) && isset($_GET['location'])) {
    echo "- Hello 2º DAW: I am " . $_GET['name'] . " from " . $_GET['location'];
    echo "<br />- Data received via AJAX GET request";
  }
?>

```



EJEMPLO POST: Index 71

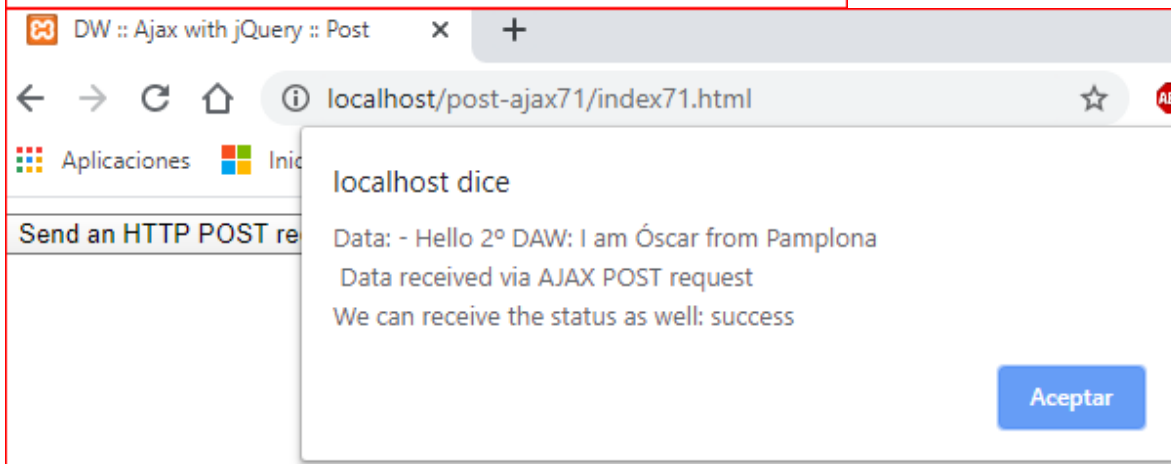
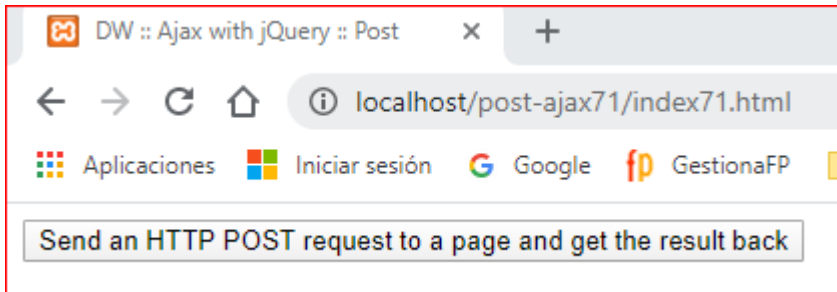
```

<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="author" content="Eugenia Pérez, eugenia_perez@cuatrovientos.org">
    <title>DW :: Ajax with jQuery :: Post</title>
  </head>
  <body>
    <button>Send an HTTP POST request to a page and get the result back</button>
    <script src="http://code.jquery.com/jquery-3.4.1.min.js" type=
    "text/javascript"></script>
    <script type="text/javascript">
      $(document).ready(function () {
        $("button").click(
          function () {
            $.post("post-ajax.php",
              {
                name: "Óscar",
                location: "Pamplona"
              },
            function (data, status) {
              alert("Data: " + data + "\nWe can receive the status as well: " + status);
            });
          });
      });
    </script>
  </body>
</html>

```



```
index71.html x post-ajax.php x
- <?php
- if (isset($_POST['name']) && isset($_POST['location'])) {
    echo "- Hello 2º DAW: I am " . $_POST['name'] . " from " . $_POST['location'];
    echo "\n Data received via AJAX POST request";
}
?>
```



PARTE3: JQUERY VALIDATION

- Las herramientas de validación de formularios de JQuery están <https://jqueryvalidation.org/>
- La validación cliente permite ganar eficiencia, menos carga y mejor experiencia de usuario

- Descarga el plugin de validación *jquery.validate.min.js* disponible en el CDN e incluirlo en /js de nuestro directorio
- Hacer uso del método **validate** sobre el formulario a validar

```
$(document).ready(function() {
    // validate signup form on keyup and submit
    var validator = $("#signupform").validate({
```

- Los parámetros que recibe el método validate es:
 - **Rules:** para cada campo que será referenciado por id: **required**, **lettersonly**, **digits**, **minlength**, **maxlength**, **email**, **remote**, **equalTo**, **iban**, etc

```
rules: {
    firstname: "required",
    lastname: "required",
    username: {
        required: true,
        minlength: 2
    },
    password: {
        required: true,
        minlength: 5
    },
    // ...
}
```

- **Messages:** permite personalizar mensajes de error que se proporcionan a los usuarios cuando no se cumplen la regla de validación

```
messages: {
  firstname: "Enter your firstname",
  lastname: "Enter your lastname",
  username: {
    required: "Enter a username",
    minlength: jQuery.format("Enter at least {0} characters")
  },
  password: {
    required: "Provide a password",
    rangelength: jQuery.format("Enter at least {0} characters")
  },
}
```

- **errorPlacement, errorContainer, errorLabelContainer, wrapper, errorElement:** especifica dónde y cómo queremos formatear los mensajes de error. Son opcionales y en caso de no especificarse, los errores se muestran en rojo al lado del campo con errores de validación.

```
errorPlacement: function(error, element) {
  if ( element.is(":radio") )
    error.appendTo( element.parent().next().next() );
  else if ( element.is(":checkbox") )
    error.appendTo ( element.next() );
  else //Another element like an input
    error.appendTo( element.parent().next() );
},
// specifying a submitHandler prevents the default submit, good for the demo
submitHandler: function() {
  alert("submitted!");
}
```

- **submitHandler:** especifica cualquier función o instrucción a realizar cuando se envía el formulario

```
submitHandler: function() {
  alert("submitted!");
}
```

EJEMPLO: basado en sitio **rememberthemilk**: donde en el formulario de alta de usuarios es validado en el cliente a través de las herramientas de validación de jQuery.



Signup

First Name

Last Name ✖ Enter your lastname

Username ✖ Enter a username

Password ✖ Provide a password

Confirm Password ✖ Repeat your password

Email Address ✖ Please enter a valid email address

Which Looks Right ☐ 14/02/07 ☐ 02/14/07 ✖ Choose your preferred dateformat

☐ I have read and accept the Terms of Use. ✖ Please, accept the terms

[Show script used on this page](#)


```
<script id="demo" type="text/javascript">
    $(document).ready(function() {
        // validate signup form on keyup and submit
        var validator = $("#signupform").validate({
            //Define the rules of the form
            rules: {
                firstname: "required",
                lastname: "required",
                username: {
                    required: true,
                    minlength: 2
                },
                password: {
                    required: true,
                    minlength: 5
                },
                password_confirm: {
                    required: true,
                    minlength: 5,
                    equalTo: "#password"
                },
                email: {
                    required: true,
                    email: true
                },
                dateformat: "required",
                terms: "required"
            },
```

```
//Define the messages for the rules
messages: {
  firstname: "Enter your firstname",
  lastname: "Enter your lastname",
  username: {
    required: "Enter a username",
    minlength: jQuery.format("Enter at least {0} characters")
  },
  password: {
    required: "Provide a password",
    rangelength: jQuery.format("Enter at least {0} characters")
  },
  password_confirm: {
    required: "Repeat your password",
    minlength: jQuery.format("Enter at least {0} characters"),
    equalTo: "Enter the same password as above"
  },
  email: {
    required: "Please enter a valid email address",
    minlength: "Please enter a valid email address"
  },
  dateformat: "Choose your preferred dateformat",
  terms: "Please, accept the terms"
},
```

ACTIVIDAD 4.7

PARTE5: JQUERY MOBILE:

ACTIVIAD 4.8 PRUEBA LOS CÓDIGOS QUE SE SUMINISTRAN EN EL DOCUMENTO

 JQUERYMOBILE.pdf