

Programación Front End (nivel inicial)

Ing. Leonardo Torres, Lic. Gustavo Lafuente



FACULTAD DE INGENIERÍA
Universidad Nacional de la Pampa



Argentina
programa
4.0

Librerías en JavaScript

Programación Orientada a la Web

Librerías en Javascript



La aparición de JavaScript ha provocado la existencia de una gran variedad de Frameworks JavaScript para manipular el DOM desde el cliente o bien trabajar del lado del servidor. A los Frameworks en JavaScript también se los puede encontrar como Librerías JavaScript (**JavaScript Libraries**).



Algunos Framework:

- Del lado del cliente: Prototype.js, **jQuery**, **AngularJS**, lightbox (lightbox 2), W3.js
- Del lado del servidor: Node.js

Prototype JS



Se orienta al desarrollo sencillo y dinámico de aplicaciones web. Es una **herramienta** que implementa las técnicas **AJAX**.

Prototype tiene su función **\$()** (también llamada dólar).

Se convirtió en poco tiempo en una referencia básica de AJAX y es la base de muchos otros frameworks e inspiró a librerías relacionadas (ej. jQuery)

Prototype JS



La **Función `$()`** (también se suele llamar función dólar) es bastante útil, al principio puede parecer un poco abstracta, esta función es un atajo a la función del DOM **`document.getElementById()`**

// Con JavaScript

```
var elemento = document.getElementById('primero');
```

// Con Prototype

```
var elemento = $('primero');
```

Prototype JS



La función **F()** permite obtener el valor de cualquier elemento.

// Con JavaScript

```
document.getElementById("municipio").value
```

// Con Prototype

```
$F("municipio")
```

Otras funciones:

\$\$() permite seleccionar elementos de la página utilizando selectores de CSS

\$A() *convierten a array cualquier cosa que parezca un array*

\$H() crea array asociados. **\$R()** crea rango de valores

Mas funciones en <http://prototypejs.org>

jQuery es una biblioteca multiplataforma de JavaScript, de fácil de aprendizaje. El propósito de jQuery es facilitar el uso de JavaScript en su sitio web.

jQuery simplifica muchas de las cosas complicadas de JavaScript, como las llamadas AJAX y la manipulación DOM.

Características:

- Manipulación HTML / DOM
- Manipulación CSS
- Métodos de eventos HTML
- Efectos y animaciones
- AJAX
- Utilidades

*¿Por qué **jQuery**?*

Hay muchas otras bibliotecas de JavaScript (como prototype.js), pero **jQuery** es probablemente la más popular y también la más ampliable.

Muchas de las empresas más grandes de la Web utilizan jQuery, como: Google, Microsoft, IBM, Netflix.

También lo es usado por los gestores de contenidos como Wordpress y Joomla, como así también los framework de Bootstrap y Material Design (Google)

Formas de uso:

- *Descargando la librería e invocándola*

`<head>`

`<script src="jquery-3.5.1.min.js"></script>`

`</head>`

- *Haciendo referencia a un CDN (Content Delivery Network). Por ejemplo de google*

`<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>`

jQuery Sintaxis

`$(selector).action()`

- *Un signo \$ para definir / acceder a jQuery*
- *Un (selector) para "consultar (o buscar)" elementos HTML*
- *Una acción jQuery () que se realizará en los elementos*

jQuery Sintaxis ejemplo

`$(this).hide()` – *Ocultar el elemento actual.*

`$("p").hide()` – *Ocultar todos los elementos `<p>`.*

`$(".test").hide()` – *Ocultar todos los elementos con `class="test"`.*

`$("#test").hide()` – *ocultar el elemento con `id="test"`.*

Todos estos métodos deben ser cargados dentro de una función (Ready) para evitar que los eventos se ejecuten antes que termine la carga de la página web. Es decir que los eventos que creamos deben estar contenidos en dicha función.

```
$(document).ready(function(){  
    //colocar aquí los metodos jQuery  
});
```

También podemos encontrar un método mas abreviado

```
$(function(){  
    //colocar aquí los metodos jQuery  
});
```

SELECTORES

Los selectores son la parte mas importante de la librería jQuery

`$(selector)`

Podemos seleccionar el párrafo <P> colocando `$("p")`

Ejemplo:

```
$(document).ready(function(){  
    $("button").click(function(){  
        $("p").hide();  
    });  
});
```

(1_Ocultar_parrafos.html)*

Ejemplos de selectores

`$("")` todos los elementos*

`$(this)` el elemento actual

`$("#p.intro")` los elementos `<p>` con `class="intro"`

`$("#p:first")` selecciona el primer `<p>`

`$("#tr:odd")` los tr impares de una tabla

(` 2_cambia_color_tr_pares.html`)*

Eventos

Todas las acciones que el usuario realiza sobre la pagina puede ser capturada por eventos.

- *eventos del mouse*
- *Eventos del teclado*
- *Eventos de ante la acción de los formularios*
- *Eventos de la ventana del navegador*

Ejemplo de Evento

capturar el clic de un párrafo

```
$("p").click();
```

El siguiente paso es definir qué debería suceder cuando se desencadena el evento. Debes pasar una función al evento:

```
$("p").click(function(){  
    // acciones aqui  
    $(this).hide();  
});
```

*(*3_Ocultar_parrafos_al_cliquearse.html)*

jQuery - Efectos



➤ *hide() and show()*

(como vimos ocultar podemos mostrar)

```
$("#hide").click(function(){  
    $("p").hide();  
});
```

```
$("#show").click(function(){  
    $("p").show();  
});
```

(*4_Oculto_muestra.html)

➤ *Fade*

- *fadeIn()*
- *fadeOut()*
- *fadeToggle()*
- *fadeTo()*

(*5_fadein.html)

(*5_fadeToggle.html)

➤ Slide

- *slideDown()*
- *slideUp()*
- *slideToggle()* //combina los dos efectos anteriores

(*6_slideToggle.html)

➤ Animaciones - **animate()**

- `$(selector).animate({params},speed,callback);`

➤ **Stop()** – (frena las animaciones o efectos)

- `$(selector).animate({params},speed,callback);`

➤ Devolución de llamadas (**Callback Functions**)

- `$(selector).hide(speed,callback);`

▪ *Esto permite que el script se ejecute paso a paso*

➤ Con jQuery, puede encadenar acciones / métodos.

([*7_animate.html](#)) [7_stop.html](#); [7_pasoapaso.html](#))

jQuery contiene métodos poderosos para cambiar y manipular elementos y atributos HTML.

➤ **Obtener elementos (get)**

- *text()*: establece o devuelve el contenido de texto de los elementos seleccionados
- *html()*: establece o devuelve el contenido de los elementos seleccionados (incluido el marcado HTML)
- *val()*: establece o devuelve el valor de los campos del formulario
- *attr()*: obtiene un atributo de un elemento

```
$("button").click(function(){  
    alert($("#w3s").attr("href"));  
});
```

(* 8-get.html)

➤ **Setear elementos (set)**

Se utilizan los mismos métodos que el de obtener: `text()`, `html()`, `val()`, `attr()`

```
$(document).ready(function(){  
    $("#btn1").click(function(){  
        $("#test1").text("Hola mundo!");  
    });  
    $("#btn2").click(function(){  
        $("#test2").html("<b>Hello world!</b>");  
    });  
    $("#btn3").click(function(){  
        $("#test3").val("Dolly Duck");  
    });  
});
```

(* 9-setear.html)

➤ **adicionar elementos o contenido (add)**

- *append ()*: inserta contenido al final de los elementos seleccionados
`$("p").append("algún texto.");`
- *prepend ()*: inserta contenido al comienzo de los elementos seleccionados
`$("p").prepend("algún texto.");`
- *after ()*: inserta contenido después de los elementos seleccionados
`$("img").after("Some text after");`
- *before ()*: inserta contenido antes de los elementos seleccionados
`$("img").before("Some text after");`

(* 10-append.html)

➤ **Remove elementos o contenido**

- *remove ()*: elimina el elemento seleccionado (y sus elementos secundarios)
- *Empty()* - Elimina los elementos secundarios del elemento seleccionado

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

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

```
$("p").remove(".test"); //remueve la clase test de todos los <p>
```

➤ **Manipular hojas de estilo**

jQuery tiene varios métodos para la manipulación de CSS. Veremos los siguientes métodos:

- *`addClass ()`: agrega una o más clases a los elementos seleccionados*
- *`removeClass ()`: elimina una o más clases de los elementos seleccionados*
- *`toggleClass ()`: alterna entre agregar / eliminar clases de los elementos seleccionados*
- *`css ()`: establece o devuelve el atributo de estilo (se ve en la pagina siguiente)*

*(*11- estilos.html)*

➤ **Manipular hojas de estilo (continuación)**

El método `css ()` establece o devuelve una o más propiedades de estilo para los elementos seleccionados.

- Sintaxis para retornar un valor:

```
css("propertyname");
```

Ejemplo:

```
$("p").css("background-color");
```

- Sintaxis para establecer un valor:

```
css("propertyname", "value");
```

Ejemplo:

```
$("p").css("background-color": "yellow");
```

```
$("p").css({"background-color": "yellow", "font-size": "200%"}); //Multiple asignación
```

(*12- css.html)

Métodos de Ajax con jQuery

- *load()*
- *get()*
- *post()*
- *ajax()*

➤ **Metodo Load()**

- *Es el método mas simple de Ajax con jQuery*
- *El resultado devuelto es colocado en un contendor*
- *Solo muestra el texto, si el texto devuelto tiene etiquetas html no las muestra.*

`$(selector).load(URL,data,callback);`

URL: *Valor requerido, Dirección de la url que se desea llamar*

data: *(opcional) es la dupla clave/valor de los datos que se enviaran al servidor*

callback: *es opcional, es el nombre de una función que se ejecutará después de que se complete el método load ().*

jQuery – AJAX – Load()



```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  
<script>
```

```
$(document).ready(function(){  
    $("button").click(function(){  
        $("#div1").load("Holamundo.txt");  
    });  
});  
</script>
```

(*1_load.html)

jQuery – AJAX – Load()



➤ Validar Nickname

```
<script>
```

```
$(document).ready(function(){
```

```
    $("#nick").keyup(function(){
```

```
        pnickname=$("#nick").val();
```

```
        $("#comprobar_mensaje").load("ajax_comprobar_nick.php",{nickname:pnickname}, function(response, status, xhr) {
```

```
            if (status == "error") {
```

```
                var msg = "Error!, algo ha sucedido: ";
```

```
            }
```

```
        });
```

```
    });
```

```
});
```

```
</script>
```

(*2_load_comprobar_nick.html

3_load_comprobar:Nick(función).html)

➤ **Metodos `get()` y `post()`**

- *Se utilizan para solicitar datos del servidor con una solicitud HTTP GET o POST.*
- *GET: solicita datos de un recurso específico. Se usa básicamente para obtener (recuperar) algunos datos del servidor. Este método puede devolver datos almacenados en cache.*
- *POST: envía datos para su procesamiento a un recurso específico. POST también se puede utilizar para obtener algunos datos del servidor. Sin embargo, el método POST NUNCA almacena datos en caché*
- *Respetar el contenido devuelto, si la respuesta tiene etiquetas html las interpretará el navegador.*

jQuery – AJAX – get()



➤ **Metodo get()**

`$.get(URL,callback);`

- *URL: requerido,url donde se apunta la funcion.*
- *callback: es el nombre de la función que se ejecuta cuando esta ok la consulta*

jQuery – AJAX – get()



```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $.get("holamundo.txt", function(data, status){
      alert("Data: " + data + "\nStatus: " + status);
    });
  });
});</script>
```

(*4_get.html)

jQuery – AJAX – get()



➤ **Validar Nickname**

`<script>`

```
$(document).ready(function(){  
    $("#nick").keyup(function(){  
        pnickname=$("#nick").val();  
        $.get("ajax_comprobar_nick.php?nickname="+pnickname,  
            function(data, status){  
                $("#comprobar_mensaje").html(data);  
            });  
    });  
});
```

`</script>`

`(*5_get_comprobar_nick.html)`

jQuery – AJAX – post()



➤ **Metodo post()**

`$.post(URL,data,callback);`

URL: requerido, url donde se apunta la función.

Data: El parámetro de datos opcional, especifica los datos para enviar junto con la solicitud.

callback: es el nombre de la función que se ejecuta cuando esta ok la consulta

jQuery – AJAX – post()



```
<script>
$(document).ready(function(){
  $("button").click(function(){
    $.post("post.php",
    {
      nombre: "Jose Lopez",
      ciudad: "General pico"
    },
    function(data,status){
      alert("Data: " + data + "\nStatus: " + status);
    });
  });
});
</script>
```

jQuery – AJAX – post()



➤ **Validar Nickname**

`<script>`

```
$(document).ready(function(){
    $("#nick").keyup(function(){
        pnickname=$("#nick").val();
        $.post("ajax_comprobar_nick.php?nickname=",{nickname:pnickname},
        function(data, status){
            $("#comprobar_mensaje").html(data);
        });
    });
});
```

`</script>`

(*7_post_comprobar_nick.html)

jQuery – AJAX – post()



Metodo ajax()

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        $.ajax({url: "holamundo.txt", success: function(result){
            $("#div1").html(result);
        }});
    });
});
</script>
```

(*8_Ajax_jquery.html)

jQuery – AJAX – ajax()



➤ **Validar Nickname**

// Función que envía y recibe respuesta con AJAX

```
$.ajax({  
    type: 'POST', // Envío con método POST  
    url: './ajax_comprobar_nick.php', // Fichero destino (el PHP que trata los datos)  
    data: { nickname: pnickname} // Datos que se envían  
}).done(function( msg ) {  
    $("#comprobar_mensaje").html(msg);  
    $(document).scrollTo('#comprobar_mensaje'); $("#comprobar_mensaje").focus();  
    }).fail(function (jqXHR, textStatus, errorThrown){ // Función que se ejecuta si algo ha ido mal  
        // Mostramos en consola el mensaje con el error que se ha producido  
        $("#comprobar_mensaje").html("ocurrio un error: "+ textStatus + " "+ errorThrown);  
    });  
(*9_ajax_comprobar_nick.html)
```


jQuery – noConflict() Method



- *¿Qué sucede si desea usar otros marcos en sus páginas, mientras sigue usando jQuery?*
- *jQuery usa el signo \$ como atajo para jQuery.*
- *Hay muchos otros marcos de JavaScript populares como: Angular, Backbone, Ember, Knockout, prototype y más.*
- *Si dos frameworks diferentes están usando el mismo atajo, uno de ellos podría dejar de funcionar.*
- *El equipo de jQuery ya ha pensado en esto e implementó el método noConflict ().*

jQuery – noConflict() Method



- ***El método noConflict () libera la retención del identificador de acceso directo \$, para que otros scripts puedan usarlo.***
- ***Para usar JQuery se puede usar el nombre en jQuery en lugar de \$***

<script>

```
$.noConflict();  
jQuery(document).ready(function(){  
    jQuery("button").click(function(){  
        jQuery("p").text("jQuery is still working!");  
    });  
});
```

</script>

jQuery – noConflict() Method



- ***El método noConflict () libera la retención del identificador de acceso directo \$, para que otros scripts puedan usarlo.***
- ***Para usar JQuery se puede usar el nombre en jQuery en lugar de \$***

<script>

```
$.noConflict();  
jQuery(document).ready(function(){  
    jQuery("button").click(function(){  
        jQuery("p").text("jQuery is still working!");  
    });  
});
```

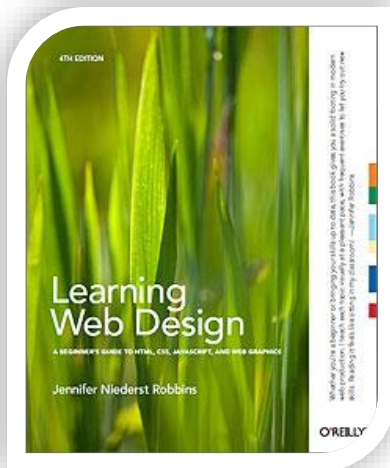
</script>

jQuery – noConflict() Method



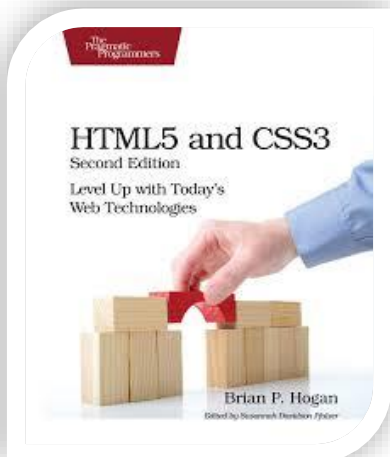
➤ ***También puede crear su propio atajo con mucha facilidad. El método noConflict() devuelve una referencia a jQuery***

```
var jq = $.noConflict();  
  
jq(document).ready(function(){  
    jq("button").click(function(){  
        jq("p").text("jQuery is still working!");  
    });  
});
```



Learning Web Design A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics (Fourth Edition)

Autor Jennifer Niederst Robbins ISBN: 978-1-449-31927-4



HTML5 and CSS3: Level Up with Today's Web Technologies (Second Edition)

Autor Brian P. Hogan

ISBN-13: 978-1937785598



Api jQuery

<https://api.jquery.com/>

Desarrolladores Web

<https://desarrolloweb.com/articulos/demo-simple-jquery.html>

w3scool

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

Herramientas



Sublime Text

<http://www.sublimetext.com/>



NetBeans

<https://netbeans.org/>



Dreamweaver – Adobe ©

<https://www.adobe.com/la/products/dreamweaver.html>



Webstorm – JetBrains

<http://www.jetbrains.com/webstorm/>



Microsoft Visual Studio for Web ®

<https://www.visualstudio.com/es/vs/visual-studio-express/>

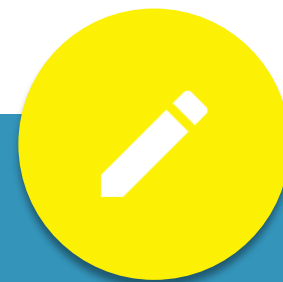
Recursos educativos



Campus Virtual
FACULTAD DE INGENIERÍA

Campus Virtual Facultad de Ingeniería

<http://campusvirtual.eng.unlpam.edu.ar/>



Programación Front End 2023



FACULTAD DE INGENIERÍA
Universidad Nacional de la Pampa



**Argentina
programa
4.0**