

Javascript + JQuery

The logo for JavaScript, featuring the letters "JS" in a bold, black, sans-serif font on a yellow square background, followed by the word "JavaScript" in a bold, black, sans-serif font on a white background.

JS JavaScript



Qué es JQuery

- jQuery es una biblioteca de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML (manipular el árbol DOM, manejar eventos, desarrollar animaciones ...)
- Es software libre y de código abierto
- Con las funciones de esta biblioteca se logran buenos resultados en menos tiempo y espacio que usando sólo JavaScript
- Descarga: <https://jquery.com/>

Función básica \$()

- **\$()** es un atajo de la función JavaScript `document.getElementById()`
 - Recibe una cadena de texto que usa para seleccionar los elementos
 - Usando el carácter "," puede seleccionar un número ilimitado de elementos
 - Ejemplos
 - // Selecciona todos los enlaces de la página
`$("a")`
 - // Selecciona el elemento cuyo id sea "primero"
`$("#primero")`
 - // Selecciona todos los h1 con class "titular"
`$(h1.titular")`
 - // Selecciona todo lo anterior en un solo paso
`$("a, #primero, h1.titular")`

Qué más permite seleccionar \$()?

- Soporta casi todos los selectores definidos por CSS 3 y también permite utilizar XPath
 - // Selecciona todos los párrafos de la página que tengan al menos un enlace
`$('p[a]')`
 - // Selecciona todos los radiobutton de los formularios de la página
`$('input:radio')`
 - // Selecciona todos los enlaces que contengan la palabra "Imprimir"
`$('a:contains("Imprimir")')`
 - // Selecciona los div que no están ocultos
`$('div:visible')`
 - // Selecciona todos los elementos pares de una lista
`$("ul#menuPrincipal li:even")`
 - Y más selectores en <http://api.jquery.com/category/selectors/>

Comienza a ejecutar antes

- En lugar de `window.onload()`, jQuery propone el siguiente código para ejecutar las instrucciones una vez que se ha cargado la página:
 - `$(document).ready(function() {`
... acá se cargan las funciones asociadas a eventos ...
`});`
- **Ventaja:** no espera que se carguen todos los elementos de la página. Sólo que se haya descargado el contenido HTML y el árbol DOM ya está disponible para ser manipulado
 - Las aplicaciones JavaScript desarrolladas con jQuery pueden iniciarse más rápidamente que las aplicaciones JavaScript tradicionales.

Agrupar funcionalidad de eventos

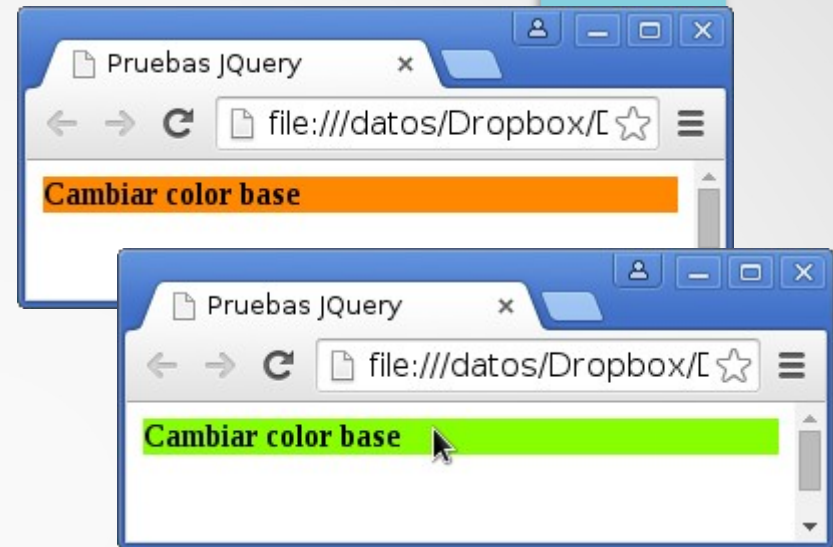
- Permite establecer la función para manejar evento a un grupo de elementos
 - // Establece función del evento 'onclick'
// para todos los párrafos de la página
`$('p').click(function() {
 alert($(this).text());
});`
 - // Establece función “valida” al evento 'onblur' para todos
// los elementos input de tipo text del formulario form1
`$("#form1 :input:text").blur(function() {
 valida($(this));
});`

JQuery: cambiar valor de un atributo

- Método `css` recibe como parámetros el atributo y el valor
- Ejemplo: cambiar el color cuando el mouse está encima y cuando se retira dejarlo como antes
 - En archivo de JavaScript

```
$(document).ready(function){  
    $("#elegido").mouseover(function(event){  
        $("#elegido").css("background-color", "#88ff00");  
    });  
  
    $("#elegido").mouseout(function(event){  
        $("#elegido").css("background-color", "#ff8800");  
    });  
});
```
 - En Html

```
<div id="elegido"> Cambiar color base </div>
```



Jquery: averiguar el valor de un atributo

- Sea el siguiente HTML
`<input id="texto1" type="text" value="Jose"/>
`
`<input id="texto2" type="text" value="Garcia"/>
`
`<input id="texto3" type="text" value="Neuquén"/>
`
- Para saber el valor de un atributo se utiliza
`$(selector).attr(nombre-atributo)`
Ejemplo: `$("#texto3").attr("value")` devuelve “Neuquén”
- Atención: Si se invoca el método `attr` con un selector que hay varios elementos de ese tipo, `attr()` devuelve el valor del atributo del primero de los elementos que haya
Ejemplo: `$("input:text").attr("id")` devuelve “texto1”

Jquery: cambiar atributo de un elemento

- También es posible modificar un atributo, usando el mismo método con dos parámetros (el primero el nombre del atributo, y el segundo el nuevo valor)
- Sea el siguiente HTML

```
<input id="texto1" type="text" value="Jose"/><br>
<input id="texto2" type="text" value="Garcia"/> <br>
<input id="texto3" type="text" value="Neuquén"/> <br>
```
- Se puede cambiar el identificador del primer input con `$("#texto1").attr("id","texto1-1");`
- Se puede cambiar el tipo del tercer input para que pase de texto a radio button `$("#texto3").attr("type","radio");`

Jquery: Agregar o remover una clase

- También se puede cambiar el aspecto o comportamiento asignándole o quitándole la pertenencia a una clase

- En Javascript

```
$("#sinClase").mouseover(function(event){  
    $("#sinClase").addClass("nueva");  
});
```

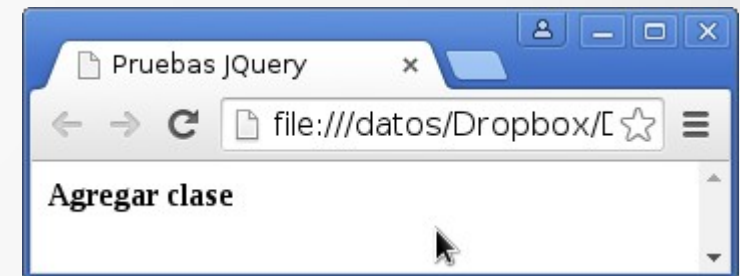
```
$("#sinClase").mouseout(function(event){  
    $("#sinClase").removeClass("nueva");  
});
```

- En CSS

```
.nueva{ background-color: #0000ff;}
```

- En HTML

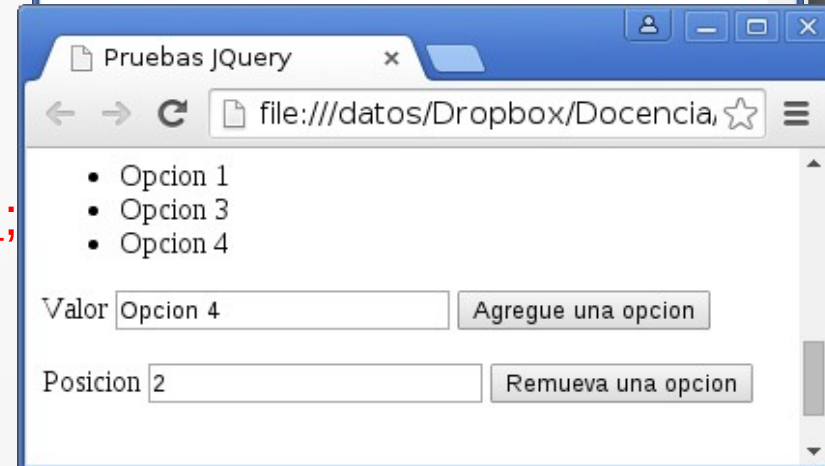
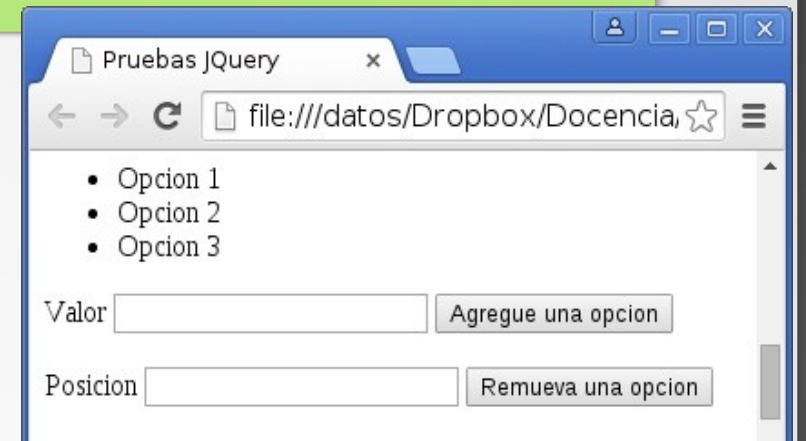
```
<div id="sinClase"> Agregar clase </div>
```



Agregar y eliminar elementos

- Ejemplo: agregar nuevas opciones o eliminar una de las existentes al usar los botones con id agregaOp y remueveOp
- ```
$(document).ready(function(){
 $("#agregaOp").click(function(event){
 var texto=$("#nom").val();
 $("#miLista").append("" + texto + "");
 });

 $("#remueveOp").click(function(event){
 var ini=parseInt($("#pos").val());
 $("#miLista").children().slice(ini-1,ini).remove();
 });
});
```



# Creando animaciones

- JQuery provee varias funciones para mostrar, ocultar, desvanecer y aparecer en barridos horizontales, verticales, etc
  - **hide()**: oculta el elemento (contrario: **show**)
  - **fadeOut()**: el elemento que lo recibe desaparece de la página haciendo una transición suavizada que acaba con el valor de opacidad cero (efecto contrario **fadeIn**)
  - **slideDown()**: muestra un elemento con un efecto de deslizamiento (efecto contrario **slideUp**)

*A todas estas funciones se les puede agregar un parámetro indicando la duración en milisegundos*

# Animate

- **Animate** es una función que permite actualizar en tiempo real la plantilla css
- Parámetros  
**.animate( propiedades, [duración], [función de animación], [callback])**
  - Propiedades: parámetro obligatorio. Indica qué atributos CSS se quieren actualizar, con sus nuevos valores. Sólo permite el cambio de propiedades CSS que tengan valores numéricos.
  - Duración: duración de la animación, en milisegundos, o una palabra como "fast" o "slow".
  - Función de animación: indica cómo se realizará la animación (suave al principio y rápida al final, o igual de rápida todo el tiempo). Dos posibles valores: "swing" (por defecto) y "linear".
  - Callback: Ofrece la posibilidad de indicar una función a ejecutarse cuando se ha terminado totalmente de producir el efecto.

# Animate

- **Animate** es una función que permite actualizar en tiempo real la plantilla css
- Parámetros  
**.animate( propiedades, [duración], [función de animación], [callback])**
  - Propiedades: parámetro obligatorio. Indica qué atributos CSS se quieren actualizar, con sus nuevos valores. Sólo permite el cambio de propiedades CSS que tengan valores numéricos.
  - Duración: duración de la animación, en milisegundos, o una palabra como "fast" o "slow".
  - Función de animación: indica cómo se realizará la animación (suave al principio y rápida al final, o igual de rápida todo el tiempo). Dos posibles valores: "swing" (por defecto) y "linear".
  - Callback: Ofrece la posibilidad de indicar una función a ejecutarse cuando se ha terminado totalmente de producir el efecto.

# Un script para animar una div con texto

- ```
function miAnimacion(){
    capa = $("#micapa");
    capa.fadeOut(2000);
    capa.fadeIn(2000);
    capa.slideUp(5000);
    capa.slideDown(5000);
    capa.animate({ "font-size": "1.5em" }, 2000);
    capa.hide(1000);
    capa.show(1000);
    capa.animate({ "left": "350px", "top": "50px" }, 1500);
    capa.animate({ "font-size": "0.75em" }, 2000);
    capa.animate({ "left": "100px", "top": "20px" },
        1500, miAnimacion);
}
```

Referencias

- **Fundamentos de jQuery** (2013)
Rebecca Murphey (Trad. Leandro D'Onofrio)
https://librosweb.es/libro/fundamentos_jquery/
- **Manual de jQuery**
Miguel Angel Alvarez
<http://www.desarrolloweb.com/manuales/manual-jquery.html>