



Desarrollo de Aplicaciones Web

DOCENTE: Daniel López Lozano





Tema 5.

Páginas web dinámicas.

Índice de contenidos

- ❑ Creación y borrado de elementos.
- ❑ Generación de contenido en páginas web dinámicas.

- ❑ Para borrar elementos necesitamos la función remove.
- ❑ Se aplica sobre una selección de elementos con \$ directamente o sobre una variable.
- ❑ Tenemos que tener en cuenta que no solo desaparece de la página como hace hide.
- ❑ Elimina completamente el objeto JavaScript con sus datos, estilos asignados y eventos asignados.

```
$("p").remove();  
//equivale tambien a  
let parrafos=$("p");  
parrafos.remove();
```

```
$("input[type=radio]").remove();  
$("#nombre").remove();  
$("table.especial").remove();
```

```
$("p").last().remove();  
//equivalente a  
let ultimo_p=$("p").last();  
ultimo_p.remove();
```

- ❑ Hasta ahora para introducir contenido en nuestra web hemos utilizado `document.write`
- ❑ Si la página no esta vacía esta función borra el código HTML que hubiera.
- ❑ Además no permite añadir código de forma selectiva, o sea no podemos colocar el código en un lugar específico.
- ❑ Por ejemplo añadir opciones a un select, crear varios div para distintos ítems que obtengamos de una base de datos etc.
- ❑ También necesitamos añadir eventos en la página sobre elementos que todavía no existen.

- ❑ Para crear elementos nuevos en JQuery es muy sencillo con el operador \$.

```
let nuevo_parrafo=$("<p>Nuevo parrafo con contenido</p>");  
let nuevo_enlace=$("<a>Enlace creado con jquery</a>");  
let nueva_imagen=$("<img></img>");  
nueva_imagen.attr("src","encendida.gif");  
let nuevo_div=$("<div>Nuevo div articulo</div>");  
nuevo_div.addClass("articulo");
```

- ❑ Los elementos nuevo no aparecerán hasta que se añadan a la página, por ejemplo directamente al body con el método append.

```
$("body").append(nuevo_parrafo);  
...  
$("body").append(nuevo_div);
```

- ❑ Podemos añadir no solo al body, también a otros elementos.

```
let opcion=$( "<option>Nueva opcion</option>" );  
opcion.attr("value", "Nuevo");  
$( "select" ).append(opcion);
```

- ❑ También a elementos que acabamos de crear

```
let lista_nueva=$( "<ul><li>Punto 1</li></ul>" );  
let nuevo1=$( "<li>Punto al principio</li>" );  
let nuevo2=$( "<li>Punto al final</li>" );  
lista_nueva.prepend(nuevo1);  
lista_nueva.append(nuevo2);
```

- ❑ Observamos que prepend añade al principio del elemento y append añade también pero al final

- ❑ Podemos crear elementos que dependan de otras fuentes introducidos a través de variables.

```
let url=$("#url").val();  
let enlace=$("#<a></a>");  
enlace.text(nombre);  
enlace.attr("href",url);  
$("#div#contenido").append(enlace);
```

- ❑ Si queremos que los elementos que se creen dinámicamente reaccionen a distintos eventos tenemos que añadir el manejador en el mismo momento de la creación.
- ❑ Por ejemplo un manejador que carguemos en la función ready no afectara a los elementos de listas creados dinámicamente.

- ❑ Podemos añadir no solo al body, también a otros elementos.

```
let nuevo=$("<li>Elemento con evento</li>");
nuevo.click(Accion1);
nuevo.dblclick(Accion2);

function Accion1()
{
    alert("Has pulsado el elemento "+$(this).text());
    $(this).remove();
}

function Accion2()
{
    $(this).css("background-color","red");
}
```

- ❑ Podemos generar códigos HTML con gran cantidad de características usando las funciones de JQuery y bucles

```
let select_nuevo=$("<select></select>");
let opcion;
for(let i=1;i<=100;i++){
    opcion=$("<option>Numero "+i+"</option>");
    opcion.attr("value",i);
    opcion.css("font-size",25);
    opcion.css("background-color","green");
    select_nuevo.append(opcion);
}
$("body").append(select_nuevo);
```

- ❑ Podemos insertar un hermano delante o detrás de un elemento simplemente con las funciones after y before.

```
$("img").after("Texto despues de la imagen");
$("img").before("Texto antes de la imagen");
```

- ❑ Podemos construir un elemento todo lo complejo que queramos que dependa de eventos.
- ❑ Combinando con números aleatorios se puede crear tableros de juego dinámicos con relativamente poco código.

```
let fila_nueva;  
let casilla;  
let tabla=$( "<table></table>" );  
for( let i=1; i<=20; i++ )  
{  
    fila_nueva=$( "<tr></tr>" );  
    for( let j=1; j<=10; j++ )  
    {  
        casilla=$( "<td>Casilla "+i+", "+j+"</td>" );  
        casilla.click( Accion );  
        casilla.attr( "id", "casilla"+i+j );  
        fila_nueva.append( casilla );  
        //podemos añadir mas eventos, estilos etc  
    }  
    tabla.append( fila_nueva );  
}  
$( "body" ).append( tabla );
```

Bibliografía

- ❑ Gauchat, Juan Diego: “El gran libro de HTML5, CSS3 y Javascript”. Editorial Marcombo. 2012
- ❑ W3SCHOOL “Manual de referencia y Tutoriales”
<http://www.w3schools.com/> Última visita Septiembre 2017
- ❑ Vara, J.M. y otros: “Desarrollo Web en Entorno Cliente. CFGS”. Editorial Ra-Ma. 2012
- ❑ Comesaña, J.L. : Técnico en Desarrollo de Aplicaciones Web.
<http://www.sitiolibre.com/daw.php>
Última visita Septiembre 2017
- ❑ “Programación en Javascript”. Colección de artículos disponibles en la url.
<http://www.desarrolloweb.com/manuales/> Última visita: Septiembre 2017.
- ❑ Píldoras Informáticas. Curso de JavaScript.
<https://www.pildorasinformaticas.es/course/javascript-desde-0/>
Última visita Septiembre 2018.