

jQuery on off son dos métodos de JQuery que resuelven una casuística que aparece muchas veces cuando trabajamos con el framework. Normalmente cuando gestionamos eventos en jQuery implementamos un código similar al que se muestra a continuación.

```
<html>
<script type="text/javascript" src="jquery-2.1.3.min.js">
</script>
<script>

$(document).ready(function() {

$("#boton").click(function() {

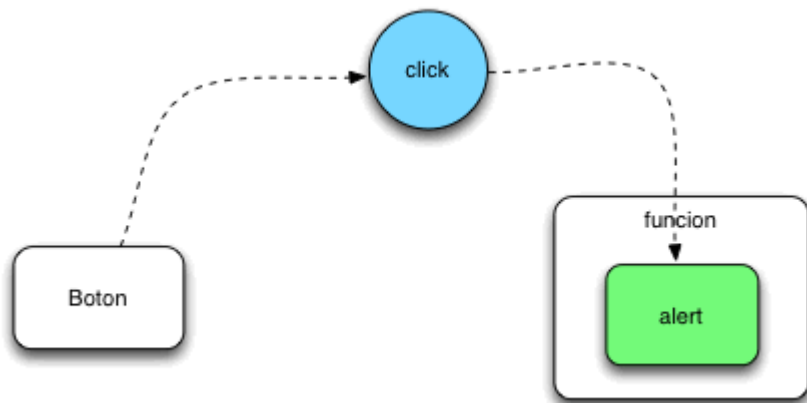
    alert("has pulsado al boton");

});

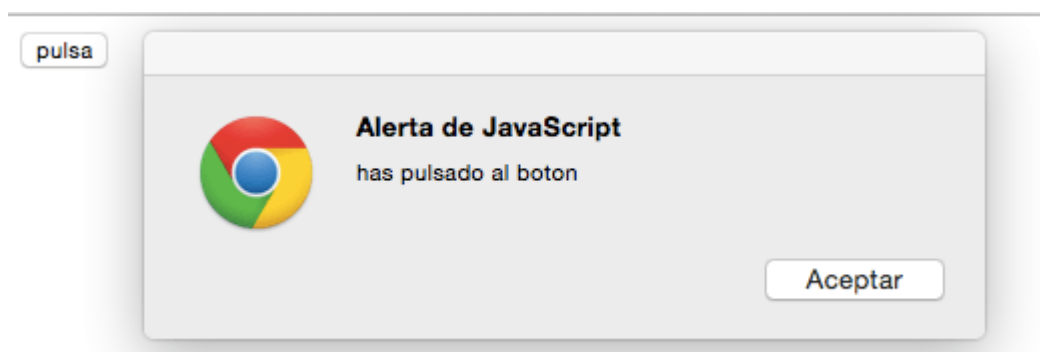
});

</script>
<body>
<input type="button" id="boton" value="pulsa"/>
</body>
</html>
```

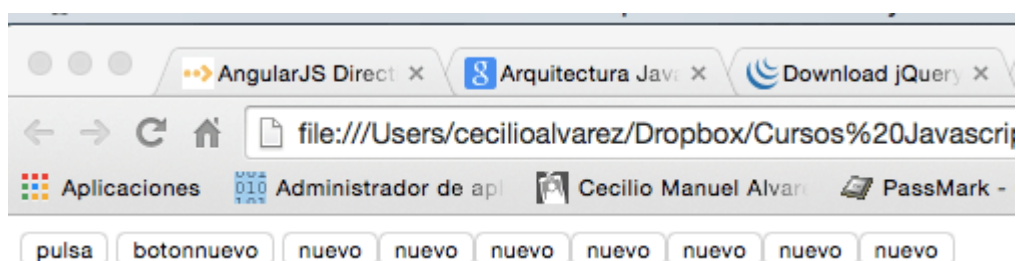
En este caso disponemos de un botón para el cual a través de jQuery registramos un evento click y una función handler.



A este evento le asignamos una función anónima que nos muestra un mensaje por pantalla y todo funciona correctamente.



¿Ahora bien que sucede si tenemos otro botón que nos genera nuevos botones en la página?.



Aquí es donde empiezan los problemas ya que aunque registremos un handler para el

evento click este no se ejecutará. El siguiente código no funciona correctamente ya que los botones no responden al evento de pulsar.

```
<html>
<script type="text/javascript" src="jquery-2.1.3.min.js">
</script>
<script>

$(document).ready(function() {

$("#boton").click(function() {

    alert("has pulsado al boton");

});

$("#botonNuevo").click(function() {

    $("<input type='button' value='nuevo'
class='nuevo' />").appendTo("body");

});

$(".nuevo").click(function() {

    alert("has pulsado el boton nuevo");
});

});
```

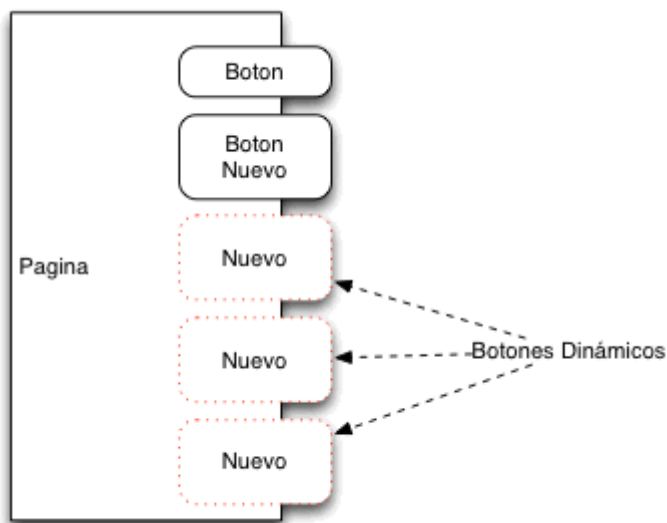
```

</script>
<body>
<input type="button" id="boton" value="pulsa"/>
<input type="button" id="botonNuevo" value="botonnuevo"/>

</body>
</html>

```

Esto se debe a que los botones son creados después de cargar la página a través de la pulsación de un botón. Por lo tanto jQuery no puede registrar los eventos para elementos que en ese momento todavía no existen.



Para solventar esto debemos usar los métodos On/Off que nos permiten delegar la gestión de eventos a un elemento superior de la página que exista previamente. En este caso podemos definir una zona en la que los controles se van a dibujar.

```

<html>

```

```
<script type="text/javascript" src="jquery-2.1.3.min.js">
</script>
<script>

$(document).ready(function() {

$("#boton").click(function() {

    alert("has pulsado al boton");

});

$("#botonNuevo").click(function() {

    $("<input type='button' value='nuevo'
class='nuevo' />").appendTo("#zona");

});

$("#zona").on("click", ".nuevo", function() {

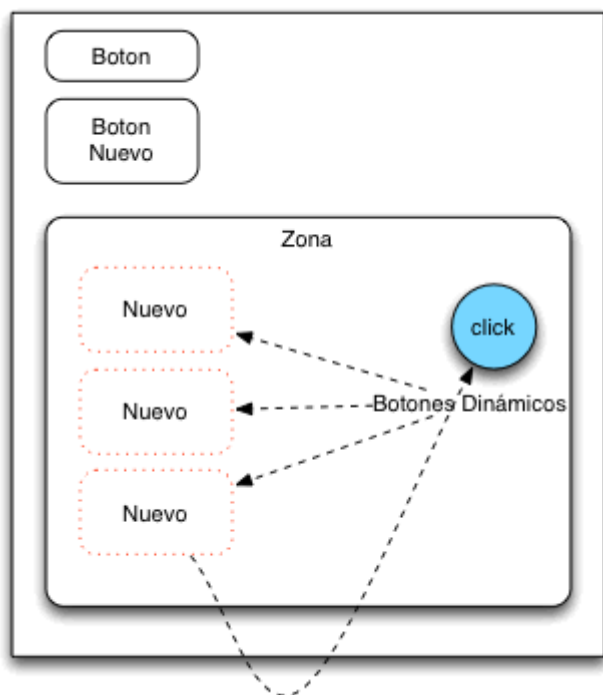
    alert("has pulsado el boton nuevo");
});

});

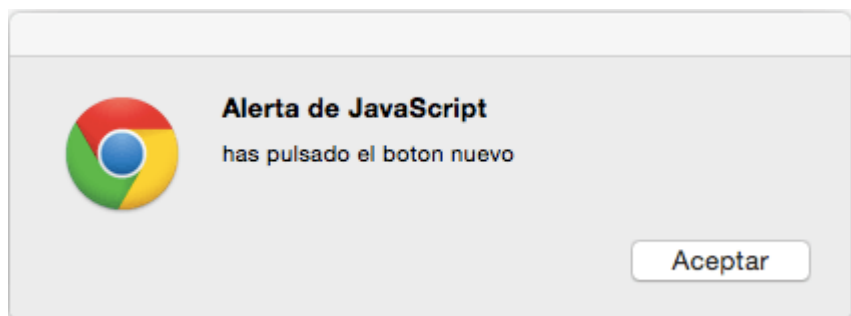
</script>
<body>
<input type="button" id="boton" value="pulsa"/>
<input type="button" id="botonNuevo" value="botonnuevo"/>
```

```
<div id="zona">  
  
</div>  
</body>  
</html>
```

En este caso hemos diseñado una zona en el documento que captura todos los eventos click que se producen en sus nodos hijos.



De esta manera el mensaje saldrá sin problemas:



Otros artículos relacionados: [jQuery Event Bubbling](#) , [jQuery WrapperSet](#) , [jQuery find vs filter](#)