



Asociando y creando eventos

Juan Diego Pérez Jiménez
@pekechis
<https://github.com/pekechis>

ÍNDICE

- Introducción.
- Control de la propagación de eventos.
- Conectar / Desconectar Handlers.
- Disparando eventos.
- Creando mis propios eventos.

1. Introducción



Además de las operaciones que hemos visto anteriormente **(CAPTURA-RESPUESTA)** jQuery nos permite **operaciones avanzadas** con **EVENTOS**.

2. Control de Propagación de eventos.



Una vez **capturado un evento**, y desde el objeto **EVENT** podemos **CONTROLAR** con algunos de sus métodos la **PROPAGACIÓN** de dicho evento.

2. Control de Propagación de eventos.

```
$(function() {  
    $("some_selector").on("...",function(event) {  
  
        //Evita el comportamiento por defecto  
        event.preventDefault();  
  
        //Para los propagación hacia arriba  
        event.stopPropagation();  
  
        //Para el resto de los handlers y la propagación hacia arriba  
        event.stopImmediatePropagation();  
    });  
});
```


2. Control de Propagación de eventos.

```
$(function() {  
  
    $("a").click(function (event) {  
        event.preventDefault();  
        alert("No me voy a ningún sitio");  
        //Si algún padre tuviera algún handler no se ejecutaría  
        event.stopImmediatePropagation();  
    });  
});
```

3. Conectar / Desconectar Handlers

(desconectando)

```
$(function() {
```

```
//.off() hace el EFECTO CONTRARIO a .on()
```

```
//Borra todos los handlers asociados a un selector
```

```
$("some_selector").off();
```

```
//Borra una serie de handlers asociados a un selector
```

```
$("some_selector").off("event1 event2...eventN");
```

```
});
```

3. Conectar / Desconectar Handlers

(desconectando)

```
$(function() {
```

```
//Borra una serie de handlers asociados a un selector
```

```
//por asociación delegada
```

```
$(“some_selector”).off(“event1 event2...eventN”, “selector2”);
```

```
//Tiene que tener la misma definición que cuando habíamos hecho
```

```
//un .on() delegado.
```

```
//Tercer parámetro “**” borra sólo los delegados
```

```
});
```


3. Conectar / Desconectar Handlers

(elementos nuevos)



Los **handlers** añadidos con **.on()** como hasta ahora presentan un **PROBLEMA: No** se asocian a elementos **NUEVOS**.

3. Conectar / Desconectar Handlers

(elementos nuevos)



En **versiones antiguas** de jQuery se usaban para ello las funciones **.live()** y **.die()**. Para todos los **elementos seleccionados ahora o en el futuro**.

3. Conectar / Desconectar Handlers

(elementos nuevos)



En las últimas versiones se usa la **ASIGNACIÓN DELEGADA** al padre.

3. Conectar / Desconectar Handlers

(elementos nuevos)

```
$(function() {
```

```
//Handler únicamente para los elementos que ya EXISTEN (ASIGNACIÓN //DIRECTA)
```

```
$("#p").on("click", function (event) {  
    alert("Click en párrafo");  
    event.stopImmediatePropagation();  
});
```

```
//Handler para los nuevos y los que voy a seguir creando
```

```
$("#div.output").on("click", "p", function (event) {  
    alert("Click en párrafo- DELEGADA");  
    event.stopImmediatePropagation();  
});
```

```
});
```


4. Disparando eventos



En ocasiones puedo **SIMULAR** que se ha producido un evento **LANZÁNDOLO** cuando me interesa.



4. Disparando eventos

```
$(function() {
```

```
//Simulo que se ha producido un eventos en los elementos del selector
```

```
$(“some_selector”).trigger(“some_event”);
```

```
//El argumento puede ser un objeto Evento
```

```
$(“some_selector”).trigger(event);
```

```
//BUBBLE PROPAGATION
```



```
});
```

4. Disparando eventos

```
$(function() {
```

```
//Ejecuto todos los handlers asociados evento
```

```
$(“some_selector”).triggerHandler(“some_event”);
```

```
//Ejecuto todos los handlers asociados a un objeto del tipo evento
```

```
$(“some_selector”).triggerHandler(event);
```

```
//Previamente el evento debe tener handler asociado
```

```
// NO BUBBLE PROPAGATION
```

```
// NO DEFAULT BEHAVIOR
```

```
});
```



5. Creando mis propios eventos

```
$(function() {  
  
    //Uso el nombre de "mi evento"  
    $( "some_selector").on("mi_evento".....);  
  
    //Lo utilizo con trigger  
    $( ".....").trigger("mievento");  
    $( ".....").triggerHandler("mievento");  
  
});
```


THANKS!

Any questions?

You can find me at @pekechis &
<https://github.com/pekechis>