

# Eventos en jQuery

```
$(".mienlace").click(function(mievento){  
    mievento.preventDefault();  
    alert("Has hecho clic");  
});
```

1. El evento se define sobre todos los elementos.
2. El tipo de evento puede ser `click()`, `dblclick()`, `focus()`, `keydown()`, etc.
3. Como parámetro tenemos que enviar una función, con el código a ejecutar cuando se produzca el evento en cuestión.
4. La función tiene otro parámetro que es el propio evento.
5. `mievento.preventDefault()` evita el comportamiento por defecto de un enlace.

# Manejadores de eventos

## Relacionados con el ratón

- **click()**: Cuando se produce un clic en un elemento de la página.
- **dblclick()**: Cuando se produce un doble clic sobre un elemento.
- **hover()**: Sirve para manejar dos eventos, cuando el ratón entra y sale de encima de un elemento. Espera recibir dos funciones en vez de una.
- **mousedown()**: Al hacer clic, en el momento que presiona el botón, independientemente de si lo suelta o no. Tanto el derecho como el izquierdo.
- **mouseup()**: Cuando suelta un botón del ratón.
- **mouseenter()**: Al situar el ratón encima de un elemento.
- **mouseleave()**: El ratón sale de encima de un elemento.
- **mousemove()**: Al mover el ratón sobre un elemento de la página.
- **mouseout()**: Cuando el usuario sale de la superficie de un elemento.
- **mouseover()**: Cuando está sobre un elemento, puede producirse varias veces mientras se mueve el ratón sobre el elemento, sin necesidad de haber salido.
- **toggle()**: Para indicar dos o más funciones a ejecutar cuando el usuario realiza clics, esas funciones se van alternando con cada clic.

# Manejadores de eventos

## Relacionados con el teclado

- **keydown():** En el momento de presionar una tecla, independientemente de si se libera la presión o se mantiene. Se produce una única vez en el momento exacto de la presión.
- **keypress():** Ocurre cuando se digita un carácter, o se presiona otro tipo de tecla. Es como el evento keypress de Javascript, por lo que se entiende que keypress() se ejecuta una vez, como respuesta a una pulsación e inmediata liberación de la tecla, o varias veces si se pulsa una tecla y se mantiene pulsada.
- **keyup():** Se ejecuta en el momento de liberar una tecla, es decir, al dejar de presionar una tecla pulsada.

Si se pulsa y suelta una tecla, primero se produce un evento keydown, al presionar la tecla, luego un keypress y por último un keyup al soltarla.

Si se hace una pulsación prolongada el esquema varía, se produce un keydown y luego un keypress. Mientras se mantiene pulsada la tecla en bucle se van produciendo eventos keydown y keypress, repetidas veces hasta que finalmente se suelta la tecla y se produce un keyup.

En el caso de las teclas CTRL, Mayúsculas o ALT, se producen múltiples keydown hasta que se suelta la tecla y se produce un keyup. Es decir, al pulsar una de estas teclas no se produce el evento keypress.

# Manejadores de eventos

combinados teclado o ratón

- **focusin()**: El elemento gana el foco de la aplicación, al hacer clic o al presionar el tabulador y situar el foco en ese elemento.
- **focusout()**: El elemento pierde el foco de la aplicación, cuando el foco está en ese elemento y nos movemos a otro elemento.
- **focus()**: Definir acciones cuando se produce el evento focus de Javascript, cuando el elemento gana el foco de la aplicación.

# Ejemplos de Eventos

- Averiguar la posición del ratón al hacer clic

```
$("#mielemento").click(function(e){  
    $("#mielemento").html("X: " + e.pageX + " - Y: " + e.pageY)  
});
```

- Ver el tipo de evento al pulsar una tecla, el código de la tecla y el carácter escrito.

```
$("#mitexto").keypress(function(e){  
    e.preventDefault();  
    $("#loescrito").html(+ evento.type + ", " + e.which + ": " + String.fromCharCode(e.which));  
});
```

Escribe el tipo de evento, con type, el código de la tecla pulsada, con which y el carácter pulsado con String.fromCharCode(e.which)

# Definir eventos genericos y eliminarlos.

- El método bind() sirve para definir eventos de manera genérica, lo que sería como una manera genérica de definir eventos
- bind(tipo\_de\_evento, manejador)

primer parámetro → tipo de evento que queremos definir. Se pueden especificar varios eventos separados por un espacio y así asignar un mismo manejador de evento para varios tipos de situaciones.

Segundo parámetro → El manejador o función a ejecutar.

```
$(".miclase").bind("click", function(){  
    alert("Has hecho clic");  
});
```

```
$("p").bind("click mouseenter mouseleave", function(e){  
    if ($(this).css("color")!="rgb(250, 100, 0)")  
        $(this).css("color", "rgb(250, 100, 0)");  
    else  
        $(this).css("color", "rgb(150, 0, 255)");  
})
```

# Definir eventos genericos y eliminarlos.

- El método `unbind()` elimina un evento cualquiera indicado con `bind()`.
- Si invocamos a `unbind()` sin ningún parámetro, eliminamos todos los manejadores de eventos, de cualquier tipo de evento.

`$("#p").unbind();` //Así hemos eliminado todos los eventos asociados con los párrafos de la página.

- Para eliminar sólo los eventos de un tipo hay que indicar como parámetro ese tipo de evento.
- `$("#p").unbind("click");`
- Se puede especificar la función que se desea descartar en la lista de parámetros de la llamada a `unbind()`.

```
var funcionManejador = function(e) {  
  // cualquier código  
}  
$("#p").bind('click', funcionManejador);  
$("#p").unbind('click', funcionManejador);
```

# Eventos definidos con `live()` ///obsoleto usar `on()`

- Definir eventos "live". Es una definición de evento "en directo" o durante toda la "vida" de la página, una definición de un evento sobre los elementos actuales y futuros que cuadren con un selector.
- `live()` funciona de manera similar a `bind()`, pero la asignación del evento es "dinámica" y afecta a los elementos en el momento de la invocación, y a los elementos que se puedan definir en un futuro.

`$(".miclase").bind("click", mifuncion);` ///definiendo un "click" sobre los elementos de clase "miclase".

`$(".miclase").live("click", mifuncion);` //sirve igual que con `bind()`, pero afectará también a todos los elementos con clase "miclase" en el futuro.

\*web que con un desplegable eligues a quien colocarle el click con otro boton creo elementos de la clase que elijo.