



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

ÍNDICE

- Introducción.
- Event Bubbling (la propagación de los eventos).
- Captura de los eventos.
- Eventos con “Nombre”.
- El objeto EVENT.

1. Introducción



Empieza lo bueno, hemos visto muchas cosas hasta ahora....

Ahora vamos a **REACCIONAR** a lo que sucede en mi página y a **DARLE VIDA**.

1. Introducción



Y para esto vamos a CAPTURAR EVENTOS y RESPONDER a esos eventos.

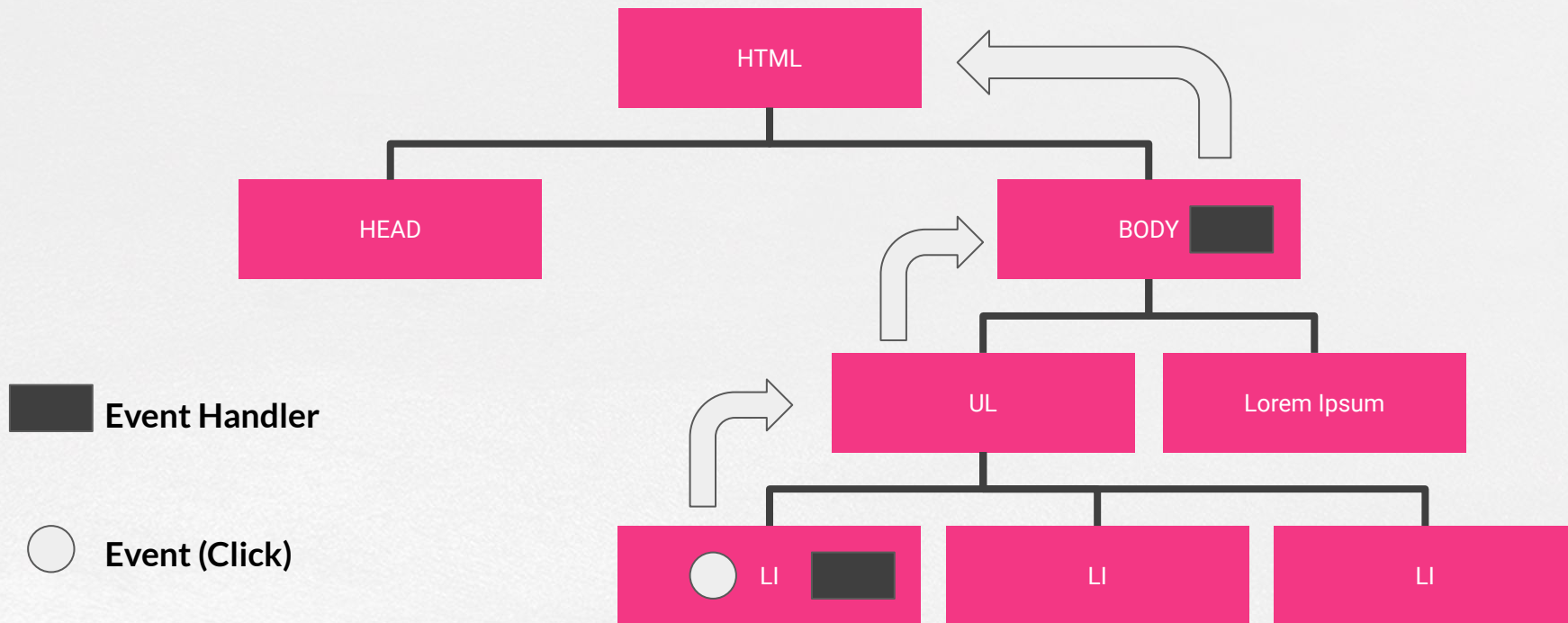
EVENTO: Cualquier cosa que “sucede dentro de nuestra web”.

1. Introducción



Hay muchas posibilidades pero en este curso vamos a ver las más utilizadas.

2. Event Bubbling



3. Capturas de Eventos

- Utilizando funciones generales (yo indico el nombre). Permite ASOCIACIÓN DIRECTA y ASOCIACIÓN DELEGADA
- Utilizando funciones de eventos (con nombre propio). Sólo permite ASOCIACIÓN DIRECTA

3. Capturas de Eventos

(funciones generales)

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

//ASOCIACIÓN DIRECTA (cuando evento sucede en elemento o hijos)

//event es opcional.

```
$(“selector”).on(“event1 event2....eventN”,function(event) {
```

//Tengo acceso a \$(this) (elemento donde ocurrió el evento)

```
});
```

//También puedo pasar datos a la función

```
});
```


3. Capturas de Eventos

(funciones generales)

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

```
  // ASOCIACIÓN DELEGADA (cuando evento sucede en hijos que cumplen  
  // selector)
```

```
  //event es opcional
```

```
  $("selector1").on("event1 event2....eventN","selector2",function(event) {  
    //Tengo acceso a $(this) (elemento donde ocurrió el evento)  
  });
```

```
  //También puedo pasar datos a la función
```

```
});
```

3. Capturas de Eventos

(funciones generales)

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

```
//Análogo a on() pero se ejecuta el handler como mucho una vez
```

```
//Luego se ELIMINA EL HANDLER.
```

```
$(“selector”).one(“event1 event2....eventN”,function(event) {
```

```
//Tengo acceso a $this (elemento donde ocurrió el evento)
```

```
});
```

```
//También puedo pasar datos a la función
```

```
});
```

3. Capturas de Eventos

(funciones generales)

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

```
    var numVeces = 0;
```

```
    //Con asociación directa
```

```
    $("li#ejemplo").on("click mouseenter",function(event) {  
        $(this).html("Número de clicks:"+(++numVeces));  
    });
```

```
    //Con asociación delegada
```

```
    $("li#ejemplo").on("click mouseenter","li#ejemplo",function(event) {  
        $(this).html("Número de clicks:"+(++numVeces));  
    });
```

```
});
```

3. Capturas de Eventos

(funciones con nombres)

```
$(function() {  
    //El funcionamiento es análogo pero los eventos “importantes”  
    // tienen funciones propias  
    $("selector").nombreEvento(funcion(event) {  
        //Tengo acceso a $this (elemento donde ocurrió el evento)  
    });  
  
    var numVeces = 0;  
    $("li#ejemplo").click(funcion(event) {  
        $(this).html("Número de clicks:"+(++numVeces));  
    });  
});
```


4. Eventos con “nombre”

- .change()
- .click()
- .dblclick()
- .focus()
- .focusout()
- .hover()
- .keypress()
- .keydown()
- .keyup()
- .mousedown()
- .mouseenter()
- .mouseleave()
- .mousemove()
- .mouseout()
- .mouseover()
- .resize()
- .scroll()
- .select()
- .submit()

5. El objeto Event



Si ponemos a disposición del **HANDLER el objeto **EVENT** este objeto puede proporcionarnos mucha información adicional sobre el evento, parar el event bubbling y evitar comportamientos por defecto....entre otras cosas.**

5. El objeto Event

- e.currentTarget
- e.stopPropagation()
- e.delegatedTarget
- e.Target
- e.pageX
- e.timeStamp
- e.metaKey
- e.pageY
- e.type
- e.preventDefault()
- e.which

THANKS!

Any questions?

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