

4. EVENTOS

Desarrollo Web en entorno cliente

IES Clara del Rey

- Qué son los eventos.
- Manejadores de eventos
- El objeto evento
- Tipos de eventos

Los eventos son la manera que tiene Javascript para controlar las acciones de los usuarios. Al interactuar se producen los eventos y con Javascript definimos los que queremos que ocurra cuando se producen los eventos.

- **Evento:** Es algo que ocurre. Provocado por el usuario (scroll) o no (un elemento que no carga)
- **Tipo de evento:** Lo que ha ocurrido (click, move, ...)
- **Manejador de evento:** comportamiento asignado como respuesta a un evento. Se especifica asociando el evento a una función JS. Cada vez que ocurre el evento, se ejecuta el manejador.

Los eventos es la forma que tiene el desarrollador de responder a las acciones de los usuarios.

Cómo definir un evento en JS (I)

Especificados en el código HTML

El manejador de eventos se coloca en la etiqueta HTML del elemento de la página.

Se usan atributos especiales en las etiquetas del HTML, que tienen el prefijo **on** seguido del tipo de evento.

Ejemplos:

```
<INPUT type="button" value="pulsame"  
      onclick="sentencias_javascript...">
```

```
<SELECT onchange="window.alert('Cambiaste la selección')">  
<OPTION value="opcion1">Opcion 1  
<OPTION value="opcion2">Opcion 2  
</SELECT>
```

Uso no recomendado de eventos

Funciona pero no es escalable ni mantenible

```
<input type=button value="Pulsame"
      onclick="x=30; window.alert(x);
      window.document.bgColor = 'red'">
```

Mejor opción:

```
<script>
function ejecutaOnclick(){
    var x = 30;
    window.alert(x);
    window.document.bgColor = 'red';
}
</script>
<FORM>
<input type="button" value="Pulsame"
      onclick="ejecutaOnclick()">
</FORM>
```

Cómo definir un evento en JS (II)

Asociados con addEventListener

Mejora la mantenibilidad del código. Permite separar el código de la funcionalidad (JS) del código del contenido (HTML).

Son necesarios dos pasos:

- 1 Acceder al objeto sobre el que queremos definir el evento.
- 2 Aplicar `addEventListener()` al objeto, indicando el tipo de evento y la función manejadora.

```

<script>
var miImagen = document.getElementById('imagen');
miImagen.addEventListener('mouseover', function() {
    alert('Has pasado el ratón encima de la imagen')
})
</script>
```

El objeto Evento

El objeto evento incluye multitud de datos útiles sobre las condiciones en las cuales un evento se ha producido.

Se genera automáticamente por parte del navegador y se envía como parámetro a la función asociada al evento. Si no es necesario obtener ningún dato se puede omitir, y si por contra se necesita se coloca al definir el manejador.

```
document.addEventListener('click', function(event) {  
    alert(`Coordenadas:  
        X: ${event.clientX}, Y: ${event.clientY}`);  
})  
//este evento se asocia al objeto document directamente
```

- Cada evento tiene un nombre, “click”
- Los atributos a usar en el HTML tienen siempre el prefijo “on” seguido del nombre del evento, “onclick”.

Del documento

- abort(onabort)
- error (onerror)
- load (onload)
- unload (onunload)
- blur (onblur)
- focus (onfocus)

De objetos

- move (onmove)
- resize (onresize)

De formularios

- change (onchange)
- select (onselect)
- reset (onreset)
- submit (onsubmit)

De teclado

- keydown (onkeydown)
- keypress (onkeypress)
- keyup (onkeyup)

De ratón

- click (onclick)
- DbClick (ondblclick)
- dragdrop (ondragdrop)
- mousedown (onmousedown)
- mousemove (onmousemove)
- mouseout (onmouseout)
- mouseover (onmouseover)
- mouseup (onmouseup)
- scroll (onscroll)

Métodos de eventos

- **preventDefault()**: cancela el evento si es cancelable (submit, link)
- **getModifierState()**: obtiene si una tecla “modificadora” está pulsada

Propiedades de eventos

- **MouseEvent**
 - button
 - clientX
 - clientY
 - metaKey
 - screenX
 - screenY
 - relatedTarget

- **KeyboardEvent** Propiedades

- code
- key
- location
- altKey
- ctrlKey
- metaKey
- shiftKey

- **wheelEvent**

- deltaX
- deltaY
- deltaZ
- deltaMode

- **Focus Event**
 - relatedTarget
- **Progress Events**
 - loaded
 - total
 - lengthComputable
- Clipboard Events
- Animation Events
- PageTransition Events
-

❶ Evento onblur

Comprobar la validez de un dato al salir del objeto de formulario

❷ Evento onunload

Abrir una ventana secundaria cuando el usuario abandone la página

❸ Evento onload

Ejecutar acciones al terminar de cargar todos los elementos de la página

Soluciones en desarrolloweb.com

