

eventos JS

UniCen / FCE / TUDAI

INTERFACES DE USUARIO E INTERACCIÓN | TUDAI

Clasificación de eventos

Podemos clasificar los eventos en varios tipos, dependiendo de la manera que el usuario tiene de interactuar con ellos.

- **Eventos del mouse:**
Aquellos en los que el usuario utiliza el mouse.
- **Eventos de teclado:**
Aquellos en los que el usuario utiliza el teclado.
- **Eventos de página:**
Cambian el estado de la página (se carga, se descarga, redimensiona, etc).
- **Eventos de formulario:**
Cambian el estado de algún elemento de un formulario.

.. hay más

Propiedades y métodos de MouseEvent

Property/Method	Description
<u>altKey</u>	Returns whether the "ALT" key was pressed when the mouse event was triggered
<u>button</u>	Returns which mouse button was pressed when the mouse event was triggered
<u>buttons</u>	Returns which mouse buttons were pressed when the mouse event was triggered
<u>clientX</u>	Returns the horizontal coordinate of the mouse pointer, relative to the current window, when the mouse event was triggered
<u>clientY</u>	Returns the vertical coordinate of the mouse pointer, relative to the current window, when the mouse event was triggered
<u>ctrlKey</u>	Returns whether the "CTRL" key was pressed when the mouse event was triggered
<u>getModifierState()</u>	Returns true if the specified key is activated
<u>metaKey</u>	Returns whether the "META" key was pressed when an event was triggered
<u>movementX</u>	Returns the horizontal coordinate of the mouse pointer relative to the position of the last

Eventos del mouse en JavaScript

mousemove

mouseenter

mouseover

mouseleave

mouseout

click

dblclick

mousedown

mouseup

contextmenu

auxclick

wheel

JS

En Javascript existen diversos eventos que reaccionan a la interacción del usuario mediante el mouse.

Clasificación de eventos

Podemos clasificar los eventos en varios tipos, dependiendo de la manera que el usuario tiene de interactuar con ellos.

- **Eventos del mouse:**
Aquellos en los que el usuario utiliza el mouse.
- **Eventos de teclado:**
Aquellos en los que el usuario utiliza el teclado.
- **Eventos de página:**
Cambian el estado de la página (se carga, se descarga, redimensiona, etc).
- **Eventos de formulario:**
Cambian el estado de algún elemento de un formulario.

.. hay más

Propiedades y métodos de KeyboardEvent

Property/Method	Description
<u>altKey</u>	Returns whether the "ALT" key was pressed when the key event was triggered
<u>charCode</u>	Returns the Unicode character code of the key that triggered the event
<u>code</u>	Returns the code of the key that triggered the event
<u>ctrlKey</u>	Returns whether the "CTRL" key was pressed when the key event was triggered
<u>getModifierState()</u>	Returns true if the specified key is activated
<u>isComposing</u>	Returns whether the state of the event is composing or not
<u>key</u>	Returns the key value of the key represented by the event
<u>keyCode</u>	Returns the Unicode character code of the key that triggered the onkeypress event, or the Unicode key code of the key that triggered the onkeydown or onkeyup event
<u>location</u>	Returns the location of a key on the keyboard or device
<u>metaKey</u>	Returns whether the "meta" key was pressed when the key event was triggered

https://www.w3schools.com/jsref/obj_keyboardevent.asp

index.html

```
<h1>Eventos del mouse en JavaScript</h1>

<div class="events" id="event_mousemove">
  <p>mousemove</p>
</div>
```

events.js

```
document.addEventListener('DOMContentLoaded', () => {

  // obtiene el elemento mediante su id
  let event_mousemove = document.getElementById('event_mousemove');

  // agrega evento de escucha
  event_mousemove.addEventListener('mousemove', () => {
    // modifica su color accediendo a la propiedad style
    event_mousemove.style.background = '#FF6C00';
  });

});
```

Interfaces Drag & Drop

Eventos disparados en el objetivo arrastrable (el elemento fuente) :

- ondragstart: ocurre cuando el usuario comienza a arrastrar un elemento.
- ondrag : ocurre cuando se arrastra un elemento.
- ondragend : ocurre cuando el usuario ha terminado de arrastrar el elemento.

Interfaces Drag & Drop

Eventos disparados en el objetivo de caída:

- ondragenter - ocurre cuando el elemento arrastrado ingresa al destino.
- ondragover - ocurre cuando el elemento arrastrado está sobre el destino.
- ondragleave - ocurre cuando el elemento arrastrado abandona el destino de colocación.
- ondrop - ocurre cuando el elemento arrastrado se deja caer en el destino.

Interfaces Drag & Drop

Las interfaces Drag-and-Drop posibilitan arrastrar y soltar archivos en una página web. Los pasos principales para configurar Drag-and-drop son:

- Hacer que un elemento se pueda arrastrar seteando el atributo “*draggable*” en true”.
- Especificar qué sucede cuando se arrastra el elemento, especificando que tipo de datos se arrastrarán y su valor.
- Especificar dónde se pueden dejar los elemento arrastrados, debemos evitar el manejo predeterminado del elemento.
- Tener en cuenta que soltar el elemento se produce un evento de caída.

index.html

```

```

```
<div class="drop" id="event_drop"
  ondrop="drop(event)"
  ondragover="allowDrop(event)">
</div>
```

events.js

```
function allowDrop(ev) {
  ev.preventDefault();
}
```

```
function drag(ev) {
  ev.dataTransfer.setData("text", ev.target.id);
}
```

```
function drop(ev) {
  ev.preventDefault();
  let data = ev.dataTransfer.getData("text");
  ev.target.appendChild(document.getElementById(data));
}
```

index.html

```

```

1

2

Evita el manejo
predeterminado del elemento

events.js

```
function allowDrop(ev) {
  ev.preventDefault();
}
```

```
<div class="drop" id="event_drop"
  ondrop="drop(event)"
  ondragover="allowDrop(event)">
</div>
```

3

Tipo de dato

Valor del dato

```
function drag(ev) {
  ev.dataTransfer.setData("text", ev.target.id);
}
```

4

- Llame a `preventDefault ()` para evitar que el navegador maneje los datos por defecto (el valor predeterminado es abrir como enlace al soltar)
- Obtenga los datos arrastrados con el método `dataTransfer.getData ()`. Devolverá los datos que se establecieron en el mismo tipo en el método `setData ()`
- Los datos arrastrados son la identificación del elemento arrastrado ("drag1")
- Agregue el elemento arrastrado al elemento de colocación