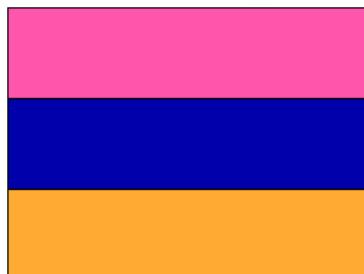


## Práctica Eventos 14-oct

### 1. evento onclick

Crea una bandera tricolor con 3 `<div>`, de manera que al hacer click sobre cada de las bandas, ésta cambie a un color aleatorio.



### 2. evento onmouseenter

Crea una tabla con al menos 50 celdas. Haz que al pasar el ratón por encima de las celdas, éstas también de color a un color aleatorio.

red	purple	teal	brown	pink	yellow	white	green	brown	yellow
light blue	tan	blue	gold	red	dark blue	teal	gold	cyan	dark red
grey	dark green	light green	gold	yellow	cyan	dark blue	dark green	green	light green
dark purple	yellow	grey	purple	grey	teal	dark red	dark blue	green	red
light blue	grey	blue	green	grey	yellow	purple	orange	pink	green
dark blue	red	yellow	grey	grey	dark blue	purple	dark red	grey	green
light blue	dark red	cyan	dark blue	purple	yellow	pink	purple	dark red	purple

### 3. evento oninput

a) Crea una página web con una cabecera `<h1>`, un `<div>` con borde o algún color de fondo, y dos deslizadores (`input-range`).

- Uno de los deslizadores modificará el **ancho** del div conforme lo desplazamos, y el otro el **alto**.
- En la cabecera se mostrará constantemente el valor de los deslizadores (i.e., las dimensiones del div).

Ejemplos de deslizador:

```
<input type="range" min="0" max="255" value="0" orient="vertical">
<input type="range" min="0" max="1" value="1" step="0.05">
```

Deslizador en MDN - Deslizador en w3school

b) Añade otros dos `input-number` para mover el div a lo ancho y alto de la página.

```
<input type="number" min="-100" max="1200" value="1" step="1">
```

Necesitas los estilos:

```
position : absolute;
top : <distanza-borde-superior>
left : <distanza-borde-izquierdo>
```

Si sólo nos interesa el evento final, podemos usar `onchange`.

#### 4. evento onchange

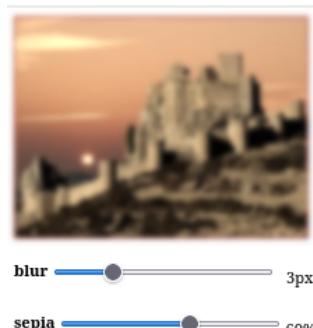
Introduce varios colores en un cuadro de lista desplegable (`select`, `option`) de manera que al elegir uno de ellos, un `div` que habrá en la página cambiará a ese color.



[<select> + <option> \(MDN\)](#) - [<select> + <option> \(w3schools\)](#)

#### 5. filtros para imágenes

Utiliza dos filtros CSS para imágenes para aplicarlos mediante deslizadores `input-range`.



`filter: tipo(valor)` [MDN](#) - [w3schools](#)  
`<input type="range"` [MDN](#) - [w3schools](#)

- Utiliza el evento `input` que va filtrando según mueves el deslizador.  
Al finalizar el deslizamiento, se genera un único evento `change`.

#### 6. chat para solitarios

Crea un formulario como el de la figura, tendrá un `textarea` un `input-text` y un botón.

Lo que se vaya escribiendo en el cuadro de texto se irá añadiendo al textarea cada vez que pulsemos el botón.

### chat para solitarios

```
Hola!  
Estoy tomando un café  
aquí en la estación
```

- Cada vez que se pulse el botón también se vaciará el cuadro de texto, y se le devolverá el enfoque.

Para recuperar el foco en el cuadro de texto cuando se puls "Enviar", el input text debe invocar su método `focus()`:

```
inpTxt.focus();
```

## 7. keydown

Cada vez que pulsamos una tecla se generan los eventos `keydown` y `keyup`.  
[\(keydown en MDN\)](#).

Modifica el chat de manera que, cada vez que se pulse la tecla `Enter` en el cuadro de texto se lleven a cabo las mismas acciones que cuando se pulsa el botón "Enviar".

Abajo tienes un ejemplo de uso de `keydown` y detección de la tecla Enter:

```
// en La propiedad "key" del evento, se guarda La tecla pulsada
inpTxt.addEventListener( "keydown", ev => {
    if(ev.key == "Enter") alert("Has pulsado Enter");
})
```