

Actividad: Diseño de reloj analógico

Utilizando lenguajes de marcación HTML y CSS, crear un reloj analógico formado por un solo elemento estructura, teniendo en cuenta:

- Un fondo cuadrado opaco con profundidad 0, ocupando el 60% de la ventana.
 - Tres imágenes superpuestas equivalentes a las tres manecillas del reloj (profundidad)
 - Crear un evento que dispare una transformación del segundero (z-index mayor)
 - Efectuar un disparo de evento en cascada de una transformación de las capas inferiores (z-index menor) por cada vuelta de la imagen superior.
 - Añadir una propiedad que efectúe las tres transformaciones de forma lineal (incluyendo z-index mayor)
 - Cambiar el evento de disparo manual por un evento programado.
 - Capturar la hora del sistema, y aplicando a las tres imágenes (capas)
 - Añadir un intervalo Delay que lleve el elemento a la esquina inferior derecha, aplicando una transformación de tamaño (10%)
 - Una vez situado el elemento según punto anterior, disparar un evento que vuelva a situarlo en el centro de la pantalla y con las dimensiones iniciales.
 - Explicar brevemente el proceso de cada etapa del diseño.
-

- **Un fondo cuadrado opaco con profundidad 0, ocupando el 60% de la ventana.**

Clase CSS reloj.

Definimos la propiedad height al 60%, no asignamos profundidad al elemento por lo que su valor es 0.

```
.reloj {  
    height: 60%;  
    width: (16*60%)/9;  
    left: 10%;  
    position: absolute;  
}
```

- **Tres imágenes superpuestas equivalentes a las tres manecillas del reloj (profundidad)**

Cuatro clases CSS que hacen referencia a la esfera horaria, al segundero, al minuterero y a la manecilla encargada de la hora respectivamente, de forma que la esfera queda situada con mas profundidad que las manecillas, sobre ella tenemos la clase #horario con una profundidad de 4, sobre ella minuterero, y finalmente como imagen mas externa y menos profunda el segundero.

```
#segundero {  
    i-index: 2  
}  
#minuterero {  
    i-index: 3  
}  
#horario {  
    i-index: 4  
}  
#esfera {  
    i-index: 5  
}
```

- **Crear un evento que dispare una transformación del segundero (z-index mayor)**

El evento onload de window llama a la variable intervalo que almacena un intervalo que dispara la transformación del segundero llamando a la función actualizar() cada 10ms.

```
var intervalo = setInterval(actualizar, 10);  
window.onload = function(){  
    degs = 0;  
    degm = 0;  
    degs = 0;  
    intervalo;  
}
```

```
function actualizar(){
```

```

    degs+=0.06;
    segundero.style.transform=`rotate(${degs}deg)`
    degm+=0.06/60;
    minuterero.style.transform=`rotate(${degm}deg)`
    degh+=(0.001/12);
    horario.style.transform=`rotate(${degh}deg)`
  }

```

- **Efectuar un disparo de evento en cascada de una transformación de las capas inferiores (z-index menor) por cada vuelta de la imagen superior.**

Degs, degm y degh son directamente proporcionales. En el mismo instante todos rotan un ángulo directamente proporcional al anterior de forma que se produce una cascada. Aquí disponemos del código y su correspondencia.

```

    degs+=0.06;
    segundero.style.transform=`rotate(${degs}deg)`
    degm=degs/60;
    minuterero.style.transform=`rotate(${degm}deg)`
    degh=(degm/12);
    horario.style.transform=`rotate(${degh}deg)`

    degs+=0.06;
    segundero.style.transform=`rotate(${degs}deg)`
    degm+=0.06/60;
    minuterero.style.transform=`rotate(${degm}deg)`
    degh+=(0.001/12);
    horario.style.transform=`rotate(${degh}deg)`

```

- **Añadir una propiedad que efectúe las tres transformaciones de forma lineal (incluyendo z-index mayor)**

Los elementos con el id igual a “horario”, “minuterero” o “segundero” respectivamente implementan este bloque CSS encargado de que la rotación sea lineal.

```

#horario:active,#minuterero:active,#segundero:active{
  transform: rotate(360deg) linear;
}

```

Aunque realmente al segundero no es necesario, aplicamos la propiedad linear.

- **Cambiar el evento de disparo manual por un evento programado.**

Asignamos un evento a un botón que nos permite actualizar la hora tomando de referencia la hora actual.

```

window.onload = function(){
  boton.addEventListener('click',horaActual);
}

```

- **Capturar la hora del sistema, y aplicando a las tres imágenes (capas).**

Al igual que el anterior. El botón anterior dispara la función horaActual() que captura la hora del sistema y actualiza cada manecilla con una precisión exacta. A su vez esta función cuenta con un timeout que llama a una función externa minimizar() que veremos mas adelante.

```
function horaActual(){
  clearInterval(intervalo);
  var d=new Date();
  var segundos = d.getSeconds();
  var minutos= d.getMinutes()
  var horas = (d.getHours()*60+d.getMinutes())/60;
  degs=6*segundos;
  degm=6*minutos+degs*1/60;
  deggh=(360/12)*horas;
  setInterval(actualizar, 10);
  setTimeout(minimizar, 10000);
}
```

- **Añadir un intervalo Delay que lleve el elemento a la esquina inferior derecha, aplicando una transformación de tamaño (10%)**

setTimeout(minimizar, 10000) definido en la función anterior se encarga de esto.

```
function minimizar(){
  document.getElementById('esfera').setAttribute("class","di");
  document.getElementById('segundero').setAttribute("class","di");
  document.getElementById('minutero').setAttribute("class","di");
  document.getElementById('horario').setAttribute("class","di");
  eliminarBoton();
  var b=document.createElement('button');
  var texto = document.createTextNode(`Cambiar tamaño`);
  b.appendChild(texto);
  b.addEventListener('click',maximizar)
  document.getElementById('botones').appendChild(b);
}
```

minimizar() se encarga de asignar a cada imagen la clase “di”, encargada de reducir y mover el elemento a la esquina inferior derecha.

```
.di{
  height:10%;
  width:(16*10%)/9;
  left: 600px;
  top: 100%;
  position: absolute;
}
```

Ademas elimina un botón asignado a cambiar la proporción del reloj y lo define llamando a la función maximizar() que se encarga de volver a darle las proporciones iniciales.

- **Una vez situado el elemento según punto anterior, disparar un evento que vuelva a situarlo en el centro de la pantalla y con las dimensiones iniciales.**

Maximizar() vuelve a situar el elemento en su posición inicial.

Accede a cada elemento que forma el reloj y le asigna la clase “reloj” que es que forma al bloque en el momento inicial.

```
function maximizar(){
    document.getElementById('esfera').setAttribute("class","reloj");
    document.getElementById('segundero').setAttribute("class","reloj");
    document.getElementById('minutero').setAttribute("class","reloj");
    document.getElementById('horario').setAttribute("class","reloj");
    eliminarBoton();
    var b=document.createElement('button');
    var texto = document.createTextNode(`Cambiar tamaño`);
    b.appendChild(texto);
    b.addEventListener('click',minimizar)
    document.getElementById('botones').appendChild(b);
}
```

Le asigna la clase inicial “reloj” y redefine el botón encargado de redefinir el elemento, asignándole la función minimizar() al clicar sobre él.

- **Explicar brevemente el proceso de cada etapa del diseño.**

He documentado todo el código con la función que lleva a cabo cada bloque de código.