

## Invernadero Virtual.

Para realizar el invernadero he tenido que crear tres archivos:

HTML, CSS y JavaScript.

He creado en HTML un desplegable de plantas para poder seleccionar cuantas flores (plantas) coger.

En el Css le he dado el estilo que quería tanto para el botón como para las letras, además de poner las letras en una posición y el borde del botón.

```
select {
  margin-bottom: 25%;
  margin-top: 15px;
  font-family: cursive, sans-serif;
  outline: 0;
  background: ■#8edfb0;
  color: ■#8050da;
  border: 1px solid □rgb(9, 9, 9);
  padding: 7px;
  border-radius: 9px;
}
```

En el JavaScript, se esta utilizando JQuery.

En este bloque de Código estoy haciendo una función para el desplegable en la cual utilizo un for para poder crear un array con las posiciones que el cliente selecciona en el desplegable, a su vez se incluye la foto de la planta o flor deseada.

```
/** Escuchar el cambio en el selector */
$('#numeroPlantas').change(function () {
    var numPlantas = parseInt($(this).val());
    var htmlPlantas = "";
    /** Crear un string con las imagenes que se van a mostrar, en funcion del valor del selector */
    for (var i = 0; i < numPlantas; i++) {
        htmlPlantas += '<img class="fotoFlor" id="flor' + i + '" src="img/img/' + plantas[i] + '.jpg' + '">';
    }
    /** Apadir el string al div destinado para contener las plantas */
    $("#plantas").html(htmlPlantas);
});
```

En este bloque de código comprobamos que al hacer clic en botón 1= Iniciar lo primero comprobara si se encuentra en los valores que he puesto, mayor que 0 en este caso.

Cuando hacemos clic en el botón 2 = Reiniciar, desaparecerá la información obtenida y se resetea para empezar de nuevo.

```
$("#boton1").click(function () {
    /** Comprobamos que el selector tenga un valor valido para empezar la animación */
    if ($('#numeroPlantas').val() > 0) {
        animacionAleatoria();
        accionclick();
    }
});

$("#boton2").click(function () {
    /** reseteamos */
    resetplants();
});
```

En el CSS he puesto la clase fotoFlor dar la separación entre las imágenes de las flores, además he creado un borde para la imagen.

```
.fotoFlor {
  width: 100px;
  font-size: 40px;
  text-align: center;
  margin: 20px;
  border-style:solid;
  border-width: 5px;
  border-color: □black;
  border-radius: 25px;
  display: inline-block;
}
```

He creado el id = topelnvernadero para poder establecer una altura para que las flores sobrepasen, además le he puesto estilo para que quede con un poco de transparencia y color. Este se encuentra a 30 vh de altura superior.

```
C: > Users > maria > Desktop > PAC_UF2_SierraMaria > # index.css > ...

#topeInvernadero {

position: absolute;
width: 100%;
height: 30vh;
border: solid;
top: 0px;
background-color: □rgb(68, 85, 85);
opacity: 0.45;

position: absolute;
width: 100%;
position: absolute;
width: 100%;
position: absolute;
width: 100%;
position: absolute;
position: a
```

En este bloque de código he creado una función para poder incrementar la subida de las flores hacia el topelnvernadero, en el he puesto la función math.random para que elija de forma aleatoria los valores de 1-10 y que vaya sumando añadiendo al marginbotton para que la flor vaya subiendo hacia arriba hasta rebasar la línea de topelnvernadero.

```
/** Recorrer div de las imagenes */
$("#plantas img").each(function (index) {
    /** Obtener margin-bottom para comparar con la diferencia (al ser un string con px, se lo quitamos y lo transformamos en float)*/
let margin = parsefloat($(this).css('margin-bottom').slice(0,-2));

if ( margin < diferencia) {
    /** obtenemos "diero aleatorio entre 0 y 10 para aliadirle al margin-bottom de la planta */
    var rand = Math.floor(Math.random() * 10);
    $(this).animate( | "margin-bottom": "+=" + rand | ], 5);
} else {
    /** Si el margen es mayor que la diferencia y en el array no este la planata, la metemos, obteniendo en el array las planatas en orden de llegada */
    if (!marginPlants.includes(plantas[index])) {
        marginPlants.push(plantas[index]);
    }
});
```

Se crea un div para los controles en el cual se incluirá botón 1 y botón 2 y en el CSS le daremos estilos.

```
#controles{
  position: absolute;
 top: 5px;
  display: block;
  left: 40%;
  display: inline-block;
  margin: 40px;
#boton1 {
  background: ■rgb(6, 131, 131);
  font-family: cursive, sans-serif;
 color: white;
 text-align: center;
  margin: top 0; ;
  margin-left: 0%;
#boton2 {
 background: ■rgb(157, 196, 186);
 font-family: cursive, sans-serif;
  color: #ebe8ee;
  text-align: center;
  font-size: 18px;
  display: none
```

En el CSS le he dado los estilos a los botones, dándole color alineando el texto y poniendo estilo de letra.

En el Js he creado la función para que permita cambiar la visibilidad mostrando u ocultando los botones.

```
function togglesBotones() {

/** Funcion que cambia el estado (mostrado/oculto) de los botones */

$("#boton1").toggle();

$("#boton2").toggle();

37

38
```

He creado esta función para que las flores(plantas) cuando termine la ejecución de subir compruebe que flores (plantas) han llegado y cual es su posición.

```
/** llamar recursivamente a la funcion hasta que lleguen todas las plantas o mostrar la tabla, con un timeout, para esperar a que termine el animate para hacer la comprobacion */
setTimeout(function(){
    if (manginPlants.length < $("#plantas img").length) {
        animacionAleatoria();
    } else {
        | show[able();
    }
}, 200)
}
```

Esta función la he creado para que aparezca la tabla con las posiciones de llegada de las flores (plantas).

```
function showTable() {

/** function showTable() {

/** function showTable() {

/** function que inserta en la tabla las celdas con el orden de la llegada de las plantas */

var htmlToAdd = '';

$(marginPlants).each(function(index, element) {

index++;

| htmlToAdd += '
?(td>' + element + '
?);

$('#tablaFlores tbody').html(htmlToAdd);

}

}

}

}

}

9

}

9

99
```