

Ejercicio final - evaluación M03 C09

THE COLOR GAME

Paso 1: Crear las tarjetas

1. En un documento HTML nuevo, crea un `<div>` con la Clase `container`.
Dentro suyo, crea otros 6 con la Clase `square`.

Paso 2: Dar estilo a las tarjetas

1. Cree un documento `CSS` y vinculelo con su `HTML`.
2. A su Clase `square` dele las siguientes propiedades:
 - a. `float: left;` Para que se apilen uno al lado del otro.
 - b. `width de 30%;`
 - c. `height: 0;`
 - d. `padding-bottom: 30%;`
 - e. `Color` de fondo.
 - f. `Margen` para separar los cuadrados.

3. Para su contenedor:

Para que los cuadrados no ocupen todo el ancho de la página, dale al contenedor un ancho `máximo de 600px y centralo`. Recuerde que para centrar un elemento `block` debe usar la propiedad `margin: 0 auto;` que distribuye el margen de forma equidistante.

4. Agregue un color de fondo al `body: background-color: #232323`.

Paso 3: Dar color a las tarjetas

1. Use JavaScript para darle a cada cuadrado un color distinto:
 - a. Cree un Arreglo llamado `colors` y escriba 6 colores en formato RGB.

Importante: Escriba los colores respetando esta sintaxis (con los espacios) "rgb(240, 14, 128)". Escriba los Strings con una misma sintaxis. Un espacio de más (o de menos) puede hacer que el juego no funcione.

2. Asigne a cada cuadrado un color:
 - a. Seleccione todos los cuadrados usando `querySelectorAll()`.
 - b. Recorra usando un for Loop. En cada vuelta del Loop, **asigne un color** del **Arreglo colors**.

*Aproveche la **Variable i**, no solo para acceder al cuadrado específico, sino también al color.*

*Asegúrese de que los 6 cuadrados tengan los colores del Arreglo **colors**.*

Acerca De Los Selectores

*Cada vez que seleccione un elemento utilice una Variable para guardarlo. Por ejemplo, escribir la **Variable square**, es más rápido que*

`document.querySelectorAll(".square")`

Paso 4: Contenido de la página

En esta parte del proyecto, creará el título del juego: "**The Great Guessing Game**". Deberá lograr que se muestre el código RGB del color ganador.

Para hacerlo, siga estos pasos:

1. Cree un `<h1>` que diga "**The Great Guessing Game**".

-
- a. Entre medio de "Great" y "Guessing" agregá un span `#colorDisplay`. Aquí irá el código RGB del color ganador.
 - b. Cambie el color de la tipografía del `<h1>` a blanco para que se vea mejor en la página.
2. Cree la Variable `pickedColor`. Dele un color específico (por ejemplo `colors[3]`) y su valor al texto de `#colorDisplay`.

Más adelante, la Variable tendrá un color aleatorio (seleccionado del Arreglo).

3. Compruebe que el color aparezca correctamente en el `<h1>`.

Paso 5: Un poco del juego

Ahora empecemos con la lógica del juego.

Cuando el usuario haga click en un cuadrado incorrecto, éste desaparecerá (cambiando su color de fondo al mismo que el del `<body>`) y, al mismo tiempo, aparecerá el mensaje "Inténtalo nuevamente".

Si el usuario adivina, en cambio, todos los cuadrados y el `<h1>` deberán tener el color del cuadrado ganador y aparecerá el mensaje "¡Correcto!".

Para lograr esta dinámica, siga estos pasos:

1. Cree el lugar donde se mostrará el mensaje: entre el título y el contenedor, agregue un nuevo `<div>` y, dentro suyo, un `span#message`.
2. Agregue la regla `CSS` para que el texto sea de color blanco.
3. Agregue un evento de `click` a cada uno de los cuadrados.

TIP: Cuando el usuario haga click en un cuadrado necesitará guardar su código RGB en una variable `clickedColor`. Luego, lo comparará con `pickedColor` y comprobará si el color clickeado es el mismo que el color ganador previamente seleccionado.

¿Qué sucede si los colores son distintos?

Si son distintos, le dará a ese cuadrado el color de fondo del `<body>` y le mostrará el mensaje "Intentalo Nuevamente" en el ``.

Fíjese si funciona correctamente cuando hace click en los colores equivocados.

¿Qué sucede si los colores son iguales?

Si son el mismo color, siga estos pasos:

1. Muestre el mensaje "¡Correcto!" en el ``.
2. Asigne al `<h1>` el color de fondo del ganador.

TIP: Para cambiar el color de los cuadrados, haga una Función por fuera. De esta manera, mantendrá el código ordenado.

3. Dentro de una nueva Función `changeColors`, que tomará como Argumento un color, itere sobre todos los cuadrados y dele el color de fondo del Argumento.
4. Ejecuta la Función en el momento que el usuario hace click en el color correcto y pase como Argumento ese color.

FIJATE SI FUNCIONA CUANDO HACÉS CLICK EN EL COLOR CORRECTO.

5. Cree una Función para asignarle a `pickedColor` un color aleatorio del Arreglo de colores. Para eso, cree la Función `pickColor` que devuelva un color random:
 - a. Dentro de la Función `pickColor` cree un número aleatorio entre los Índices del Arreglo de colores.
 - b. Use ese número para devolver un color del Arreglo.
 - c. Asigne lo que devuelve la Función a la Variable `pickedColor`.

FIJATE SI CADA VEZ QUE ACTUALIZAS LA PÁGINA EL COLOR DEL `<h1>` ES DISTINTO Y SI EL JUEGO SIGUE FUNCIONANDO.

6. Cree la Función `randomColor` para generar y devolver un color aleatorio.

Recuerde: RGB son siempre 3 números entre 0 y 255.

7. Cree la Función `generateRandomColors` que generará de forma aleatoria el Arreglo de colores usando la `randomColor`.

- La Función tomará como Argumento un número que indique la longitud del Arreglo (6 o 3 cuadrados, dependiendo si se está jugando en `Easy` o `Hard Mode`).
- El Arreglo que devuelve esta Función se lo asigna a la Variable `colors`.

POR ÚLTIMO, FIJATE SI CADA VEZ QUE CARGÁS LA PÁGINA APARECEN COLORES RANDOM EN LOS CUADRADOS Y FUNCIONA EL JUEGO.

El botón Play Again

En esta parte del ejercicio creará el botón de "Play Again?".

Este botón tiene 2 funcionalidades:

- Cuando se está jugando el juego, ofrece pedir nuevos colores.
- Si se ganó el juego, ofrece jugar otra vez.

En ambos casos el efecto es el mismo: `reinicia el juego` con nuevos colores.

Para crear el botón "Play Again?", sigue estos pasos:

- Dele el ID `stripe` al `<div>` donde está el mensaje "`¡Correcto!`" e "`Intentalo Nuevamente`".
- Dentro del `<div>`, agregue un `button#reset` que diga "`Nuevos Colores`".
- En el documento CSS, seleccioná `#stripe` y dale:
 - `Color` de fondo blanco.
 - `Altura` de 30px.
 - `text align center`.

Importante: Cambiá el `` negro a blanco para poder verlo.

-
4. Seleccione el botón Reset y dale un event listener para que, cuando se haga click:
 - a. Genere nuevos colores en el Arreglo.
 - b. Vuelva a elegir un color de ese Arreglo (para que sea el ganador) y aparezca en #colorDisplay.
 - c. Cambie los colores de los cuadrados con los nuevos colores.

FÍJESE SI TODO SIGUE FUNCIONANDO CORRECTAMENTE.

5. Agregue un evento para que, cuando el usuario gane, #reset cambie su texto a "Play Again?".
6. Agregue en el listener de #reset un evento para que, cuando el usuario haga click, vuelva a decir "Nuevos Colores" y desaparezca el mensaje "Intentalo Nuevamente" y ";Correcto!".

***ANTES DE SEGUIR AVANZANDO, VUELVE A RESETEAR EL COLOR DE FONDO DEL <H1>
Y FIJATE SI TODO ESTÁ FUNCIONANDO COMO DEBERÍA.***

Los botones EASY y HARD

En esta parte del ejercicio creará los botones Easy y Hard. Para hacerlo, agregue los 2 botones al final del div#stripe y siga los pasos indicados para cada botón.

El Botón Hard

1. Dele al botón de Hard la clase selected para que, al cargar la página, indique que está seleccionado.
2. Dele a la clase selected el color de fondo que quiera.
3. Dele a cada botón un ID distinto. Luego, cree un event listener a cada uno para que, cuando le haga click, le agregue la clase selected y se la quite al otro.

FÍJESE SI FUNCIONA CORRECTAMENTE CUANDO HACE CLICK EN AMBOS BOTONES.

El Botón Easy

Cuando hacemos click en el botón Easy queremos que se generen solo 3 colores con un color ganador y que se muestre en el título cuál es.

Para lograrlo, le asignaremos a los primeros 3 cuadrados los 3 colores del Arreglo y ocultaremos los otros 3 a través de la Propiedad `hidden`.

PISTA: Debe ser un String con el color. Si el String es verdadero, le asignará el color al cuadrado, sino (si no hay ningún elemento en ese Índice y nos devuelve `undefined`) le cambiará al cuadrado la propiedad CSS `display` a `none`. De esta manera, el cuadrado se esconderá.

FIJATE QUE CUANDO HAGAS CLICK EN EASY LOS 3 PRIMEROS CUADRADOS CAMBIEN DE COLOR Y LOS OTROS TRES SE ESCONDAN.

El último tramo

En esta parte del ejercicio, se enfocará nuevamente en el botón `Hard`.

No se preocupe si el código se empieza a repetir. Antes de terminar el proyecto lo podrá refactorizar.

Ahora deberá:

- Generar nuevos colores (que vuelvan a ser 6).
- Elegir un color ganador.
- Mostrar en el título cuál es ese color.

Para hacerlo, siga estos pasos:

- Haga Loop a todos los cuadrados y asigne a todos un color del Arreglo.
- Para poder volver a mostrar los cuadrados escondidos, dele a los cuadrados el Valor `block` a la propiedad CSS `display`.

FÍJESE SI DESPUÉS DE IR A EASY Y VOLVER A HARD SE MUESTRAN LOS CUADRADOS ESCONDIDOS Y SI APARECEN NUEVOS COLORES. TAMBIÉN CORROBORE SI EL JUEGO FUNCIONA CORRECTAMENTE EN AMBOS MODOS.

PISTA

Un Bug En El Juego

En este momento, tiene un bug en el juego: cuando está en el modo `Easy` y hace click en `#reset` se generan 6 nuevos colores en el Arreglo que no se muestran en la pantalla.

Para solucionar este problema, creará la Variable `numberOfSquares` que registre con cuántos cuadrados se está jugando en ese momento:

- La Variable iniciará siendo 6.
- Cuando se haga click en Easy cambiará a 3.
- Cuando se haga click en Hard volverá a 6.
- Luego, cada vez que se ejecute `generateNumberColors` ya no se pasará el número Hardcodeado como Argumento sino que se pasará la Variable `numberOfSquares`.

FÍJESE SI SE SOLUCIONA EL BUG MOSTRANDO POR CONSOLA CUÁNTOS ELEMENTOS HAY EN EL ARREGLO COLORS.

Estilo

Ahora que el juego funciona, nos enfocaremos en el estilo. En este punto, existen 2 caminos:

1. Seguir el maquetado del ejemplo.
2. Crear su propia versión.

Si elige la segunda opción, tiene que respetar los lineamientos generales:

1. Los botones de "Nuevos Colores", "Easy" y "Hard", no deben parecer botones clásicos ni tener una línea azul alrededor (en inglés, outline) cuando le hacemos click.
2. El mensaje de ";Correcto!" y de "Intentalo Nuevamente" no tienen que mover a los otros elementos cuando aparecen.
3. Los cuadrados deben desaparecer con un efecto de transición de CSS. Investigue cómo hacerlo.

Conclusión

Bien sabemos que es un proyecto largo, pero todo lo que necesita para realizarlo, lo viene trabajando desde el comienzo del módulo 02.

Con este proyecto ya va a poder mejorar su habilidad en JavaScript Vanilla.

Recuerde que si necesita repasar algún concepto, simplemente debe mencionarlo al relator o a su asistente.

Como siempre, desde OTEC Plataforma 5 Chile le deseamos el mayor de los éxitos.

OTEC PLATAFORMA 5 CHILE