

Desafío 3 - Funciones

En este desafío validaremos nuestros conocimientos en transformación y modificación de funciones y códigos en función de instrucciones dadas.

Lee todo el documento antes de comenzar el desarrollo individual, para asegurarte de tener el máximo de puntaje y enfocar bien los esfuerzos.

Descripción

Aplicando los conceptos y herramientas aprendidas hasta ahora, desarrolla el siguiente desafío que consiste en un conjunto de ejercicios que nos permitirán poner en práctica los elementos clave de la unidad.

Requerimientos

1. Transforma esta declaración de función a una función de expresión. El ejercicio debe quedar en un archivo nombrado ``1_funcion.js`` (1 Punto)

```
function example(a, b, c){  
  return a+b+c  
}
```

2. Transforma la siguiente función a una arrow function de una línea. Este ejercicio debe quedar en un archivo llamado ``2_arrow.js`` (1 Punto)

```
let suma = function(a, b){  
  return a + b  
}
```



Tip: Para los requerimientos 1 y 2 recuerda que tienes material asociado en la “Guía - Funciones” en el capítulo de **Arrow functions**.

Se tiene como base el siguiente ejercicio que cambia el color de fondo de un elemento de HTML al hacerle click **(3 Puntos)**

```
<div id="ele1"> hello </div>

<script>
function pintar(){
  ele.style.backgroundColor = 'yellow'
}

const ele = document.getElementById("ele1")
ele.addEventListener("click", pintar);
</script>
```

2.1. El código debe estar en 2 archivos: **pintar.html** y **script.js**. El script debe estar dentro de la carpeta **assets/js** **(1 Punto)**.
Para obtener el puntaje debes entregar los archivos cumpliendo la estructura pedida en conjunto con los requerimientos de 3.2 y 3.3

2.2. Modifica la función para que reciba el elemento clickeado de forma de no tener que seleccionarlo nuevamente dentro de la función **(1 Punto)**.
Para obtener el puntaje, debes entregar los archivos **pintar.html** y **script.js** funcionando en conjunto con el código modificado.



Tip: revisa en la “Guía - Funciones” las funciones anónimas.

2.3. Modifica el código anterior para poder pasarle un color como argumento a la función **pintar**. El color debe ser verde (green) por defecto, al hacer clic en el párrafo se debe pasar amarillo como color. **(1 Punto)**.

3. Construye una página web **4_colores.html** con las siguientes características **(Total: 5 puntos)**

- Crea 4 divs que tengan alto y ancho de 200px y de colores azul, rojo, verde y amarillo (insertar estilos con atributo style). Cada uno de los divs debe tener un identificador único. **(1 Punto)**
- Dentro del script agrega el evento para que al hacer click a cada uno de los divs, este cambie de color al color negro. Utiliza **addEventListener** para agregar el evento. **(1 Punto)**

- Crea una página junto a un script que guarde dentro de una variable global un color dependiendo de la letra del teclado presionada. **(2 Puntos)**.
 - La letra **a** guardará el color rosado.
 - La letra **s** guardará el color naranja.
 - La letra **d** guardará el color celeste.
 - Para guardar el color revisa el tip al final del enunciado.
 - Crea un nuevo div con el id “key”, de 200 px de ancho y alto y de color blanco y borde negro.
 - Al presionar las teclas **a**, **s** o **d**, se deberá cambiar el color del div “key” a rosado, naranja o celeste respectivamente.
- Siguiendo con la lógica del punto anterior, al presionar las teclas **q**, **w** o **e** se deberá crear un div nuevo de las mismas dimensiones antes mencionadas con los colores morado, gris y café respectivamente. **(1 Punto)**



Tip: El siguiente script te ayudará a realizar acciones en función de que se presione una tecla, además en la guía de estudio encontrarás un ejemplo.

```
document.addEventListener('keydown', function (event) {  
  if (event.key === 'a') {  
    /* Cambiar a color 1 */  
  } else if (event.key === 's') {  
    /* Cambiar a color 2 */  
  }  
})
```



¡Mucho éxito!