AE4_ABPRO-Ejercicio grupal [Actividad Opcional]

AE4_ABPRO-Ejercicio grupal [Actividad Opcional]

Ejercicio grupal

Contexto

Un equipo de chefs está desarrollando un asistente digital para calcular los ingredientes y tiempos de cocción de sus recetas favoritas. El asistente debe estar programado en JavaScript usando funciones reutilizables. Tu equipo colaborará para construir las funciones necesarias y poner en práctica todos los conocimientos fundamentales sobre funciones, parámetros, alcance de variables y anidamiento de funciones.

Actividad

1. **¿Qué es una función y para qué sirve?**
   En equipo, redacten una definición con sus propias palabras sobre qué es una función en JavaScript y por qué es útil al programar.

2. **Definir funciones**
   Definan una función llamada **calcularTiempoTotal** que reciba tres parámetros: **preparacion**, **coccion** y **reposo**, y que devuelva el tiempo total de la receta.

3. **Paso de parámetros en una función**
   Definan otra función llamada **calcularPorciones** que reciba dos parámetros: **cantidadBase** y **numeroDePorciones**. La función debe multiplicar los ingredientes base por el número de porciones y devolver el resultado.

4. **Retorno de una función**
   Usen **console.log()** para mostrar el resultado del tiempo total de cocción y de la cantidad total de ingredientes, usando las funciones creadas anteriormente.

5. **Variables locales y variables globales**
   Declaren una variable global **unidad = "minutos"**. Luego, dentro de una función llamada **mostrarResumenReceta**, definan variables locales para los datos de una receta y usen **unidad** para mostrar un resumen como: **"La receta toma 90 minutos y rinde para 4 personas."**

6. **Invocación de una función**
   Llamen a todas sus funciones con valores reales y asegúrense de que sus resultados se vean correctamente en consola.

7. **Alcance de las variables locales**
   Dentro de una función llamada **pasosReceta**, definan una variable local llamada **instrucciones**. Intenten acceder a instrucciones fuera de la función y expliquen por qué no es posible. Comenten el error que aparece en consola.

8. **El problema de las variables globales**
   Definan una variable global **nivelDeDificultad = "fácil"**. Después, dentro de una función, cambien su valor a "**difícil**". Luego, invóquenla y analicen en equipo qué pasó con la variable. Discutan cómo esto puede afectar programas más grandes.

9. **Crear una función anidada**
   Construyan una función llamada **gestionarReceta** que incluya otra función interna llamada **imprimirDetalles**. La función principal debe recibir datos de la receta y la interna debe imprimir un mensaje como:

   "Para preparar Spaghetti necesitarás 40 minutos en total y 5 ingredientes."

Entrega:

- Archivo comprimido con respuestas y código.
- Duración: 1 jornada de clases.
- Entrega: Grupal.

Estado de la entrega

Estado de la entrega	Todavía no se han realizado envíos
Estado de la calificación	Sin calificar
Última modificación	-

DESARROLLO AE4_ABPRO-EJERCICIO GRUPAL (OPCIONAL)

1. ¿Qué es una función y para qué sirve?

Las funciones son bloques de código que se ejecutan solo cuando las llamamos, lo que nos permite reutilizar el código sin tener que escribirlo una y otra vez. Las funciones son herramientas poderosas que nos permiten estructurar de manera más organizada nuestro código.

2. Definir funciones

```
function calcularTiempoTotal(preparacion, coccion, reposo) {
        let tiempo = preparacion + coccion + reposo;
        return tiempo;
}
```

3. Paso de parámetros en una función

```
function calcularPorciones(cantidadBase, numeroDePorciones) {
        let porciones = cantidadBase * numeroDePorciones;
        return porciones;
  }
```

4. Retorno de una función

```
let tiempoTotal = calcularTiempoTotal(20, 60, 10);
let ingredientesTotales = calcularPorciones(2, 5);

console.log("Tiempo total de cocción:", tiempoTotal, "minutos");
console.log("Total de ingredientes:", ingredientesTotales);
```

5. Variables locales y globales

```
let unidad = "minutos";
function mostrarResumenReceta() {
        let receta = " Ravioles";
        let tiempo = calcularTiempoTotal(30, 45, 15);
        let porciones = 4;

        console.log("Resumen de receta:");
        console.log(`La receta toma ${tiempo} ${unidad} y rinde para ${porciones}
  personas.`);
  }
```

6. Invocación de funciones

```
let tiempoPasta = calcularTiempoTotal(10, 30, 5);
let ingredientesPasta = calcularPorciones(2, 3);
console.log("Tiempo para pasta:", tiempoPasta, unidad);
console.log("Ingredientes para pasta:", ingredientesPasta);
```

7. Alcance de las variables locales

```
function pasosReceta() {
        let instrucciones = "1. Hervir agua. 2. Colocar Espagueti. 3. Cocinar. 4. Dejar reposar.";
        console.log("Pasos dentro de la función:", instrucciones);
}
```

```
pasosReceta();

console.log(instrucciones);
```

ReferenceError: instrucciones is not defined
Instrucciones es una variable local y no se puede acceder fuera de la función.

8. **El problema de las variables globales**

```
let nivelDeDificultad = "fácil";
function cambiarDificultad() {
        nivelDeDificultad = "difícil";
        console.log("Dificultad dentro de la función:", nivelDeDificultad);
}

console.log("Dificultad antes:", nivelDeDificultad);
cambiarDificultad();
console.log("Dificultad después:", nivelDeDificultad);
```

El cambio del contenido de la variable puede generar errores, si otra parte del código utiliza lo que afectaría el resultado de alguna operación.

9. **Función anidada**

```
function gestionarReceta(nombreReceta, tiempoTotal, totalIngredientes) {
        function imprimirDetalles() {
            console.log(`Para preparar ${nombreReceta} necesitarás ${tiempoTotal} minutos
            en total y ${totalIngredientes} ingredientes.`);
        }
        imprimirDetalles();
}


gestionarReceta("Ravioles", 40, 5);
```