Un letrero de color negro

Descripción generada automáticamente con confianza baja

**Nombre:** Rivera Amanda

**NRC:** 10047

**Fecha:** 31/05/2023

**Prueba Unidad N° 1**

**1.2 Variables y operaciones**

**1.2.1 Responde las siguientes preguntas en la sección de comentarios:**

• ¿Qué es una variable y para qué sirve?

Una variable es un “contenedor” de datos, que se utiliza para guardar los datos, se utiliza variables para almacenar datos que serán de utilidad, o serán llamados dentro del código.

• ¿Cuál es la diferencia entre declarar e inicializar una variable?

La diferencia con respecto a declarar es indicar un nombre y un tipo de dato dentro del lenguaje en que estemos trabajando sin asignarle un valor en específico. En cambio, inicializar una variable es darle un valor específico al dato. Las 2 son muy importantes a la hora de programar.

• ¿Cuál es la diferencia entre sumar números y concatenar strings?

Sumar significa realizar una operación aritmética entre números con el fin de obtener otro como resultado, mientras que concatenar quiere decir juntar string o cadenas de manera que se obtiene una cadena mas grande.

• ¿Cuál operador me permite sumar o concatenar?

El símbolo de “+”

**1.2.2 Determina el nombre y tipo de dato para almacenar en variables la**

**siguiente información:**

• Nombre -> Variable: “Nombre” de tipo: “cadena” (string).

• Apellido -> Variable: “Apellido” de tipo: “cadena” (string).

• Nombre de usuario en ESPE -> Ejemplo: sntipanguano -> Variable:”sntipanguano” de tipo: “cadena” (string)

• Edad -> variable: “edad” de tipo “entero” (integer)

• Correo electrónico -> Variable “correoElectronico” de tipo “cadena” (string)

• Mayor de edad -> Variable “mayorEdad” de tipo booleano (boolean)

• Dinero ahorrado -> Variable “dineroAhorrado” de tipo flotante (float)

• Deudas -> Variable “deuda” de tipo flotante (float)

**1.2.3 Traduce a código JavaScript las variables del ejemplo anterior y deja**

**tu código en los comentarios.**

Texto

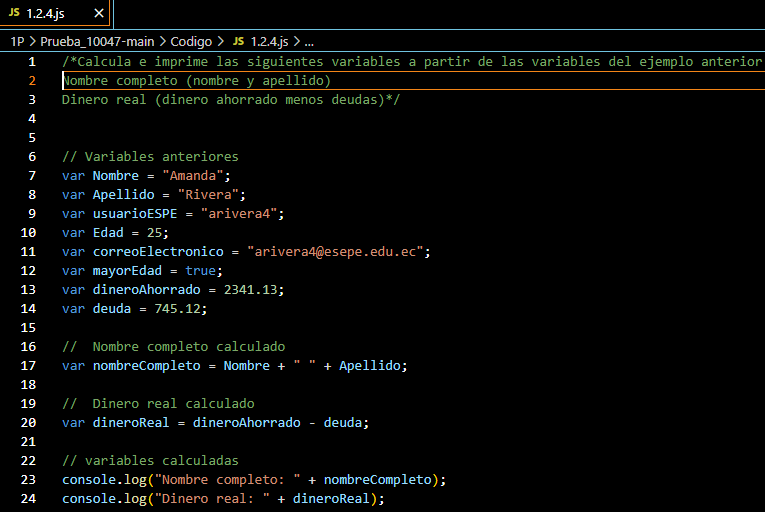
Descripción generada automáticamente

**1.2.4 Calcula e imprime las siguientes variables a partir de las variables del**

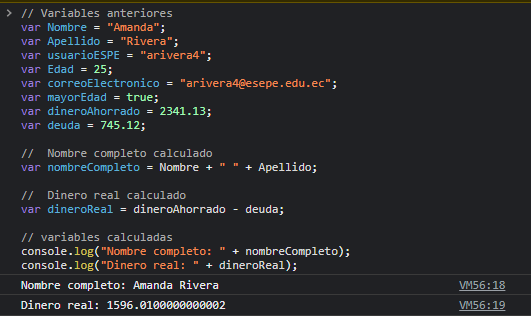
**ejemplo anterior:**

• Nombre completo (nombre y apellido)

• Dinero real (dinero ahorrado menos deudas)



Resultado:



**1.3 Funciones**

**1.3.1 Responde las siguientes preguntas en la sección de comentarios:**

• ¿Qué es una función?

Es como una herramienta creada para una función en específica y esta se puede utilizar en otras secciones del código

• ¿Cuándo me sirve usar una función en mi código?

Sirve para poder reutilizar esta función en varias partes del código dando así una buena estructura y organización del código

• ¿Cuál es la diferencia entre parámetros y argumentos de una función?

Los marcadores son parámetros de posición o el valor que la función espera recibir, mientras que los argumentos son valores reales que se pasa a la función cuando esta es llamada.

**1 Test de JavaScript.**

**1.3.2 Convierte el siguiente código en una función, pero, cambiando cuando**

**sea necesario las variables constantes por parámetros y argumentos**

**en una función:**

const name = "Diego Medardo";

const lastname = "Saavedra García";

const completeName = name + lastname;

const nickname = "statick";

console.log("Mi nombre es "

+ completeName

+ ", pero prefiero que me digas "

+ nickname + ".");

Texto

Descripción generada automáticamente

Resultado:

Texto

Descripción generada automáticamente

1.4 Condicionales

**1.4.1 Responde las siguientes preguntas en la sección de comentarios:**

• ¿Qué es un condicional?

Es una estructura de control que permite tomar decisiones al código en base a una condición establecida

• ¿Qué tipos de condicionales existen en JavaScript y cuáles son sus diferencias?

* **if**: Permite ejecutar un bloque de código si una condición es verdadera.
* **if-else:** Permite ejecutar un bloque de código si una condición es verdadera y otro bloque si es falsa.
* **if-else if-else:** Permite evaluar múltiples condiciones y ejecutar diferentes bloques de código según el resultado de cada una.
* **switch:** Permite evaluar una expresión y ejecutar diferentes bloques de código según el valor de la expresión.

• ¿Puedo combinar funciones y condicionales?

Si, si se puede para lograr dar condición a distintos bloques del código se puede crear un condicional dentro de cada función y se puede realizar lo mismo de forma inversa que sería llamar a la función dentro de una condición.

**1.4.2 Replica el comportamiento del siguiente código que usa la sentencia**

**switch utilizando if, else y else if:**

const tipoDeSuscripcion = "Basic";

switch (tipoDeSuscripcion) {

case "Free":

console.log("Solo puedes tomar los cursos gratis");

break;

case "Basic":

console.log("Puedes tomar casi todos los cursos de ESPE durante un mes");

break;

case "Expert":

console.log("Puedes tomar casi todos los cursos de ESPE durante un año");

break;

case "ExpertPlus":

console.log("Tú y alguien más pueden tomar TODOS los cursos

de ESPE durante un año");

break;

Texto

Descripción generada automáticamente

Resultado:

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

**1.4.3 Replica el comportamiento de tu condicional anterior con if, else y**

**else if, pero ahora solo con if (sin else ni else if).**

Texto

Descripción generada automáticamente

**Resultado:**

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

**1.5 Ciclos**

**1.5.1 Responde las siguientes preguntas en la sección de comentarios:**

• ¿Qué es un ciclo?

Es un bucle que se detiene solo cuando cmple una condición en específico

• ¿Qué tipos de ciclos existen en JavaScript?

* For: Se utiliza cuando se conoce el número de ciclos
* While: Se ejecuta cuando la condición es verdadera, antes de cada ciclo se evalúa si es Verdadero o falso
* Do While: Se ejecuta el proceso hasta que se cumpla una condición a diferencia del while este se ejecuta luego de cada ciclo

• ¿Qué es un ciclo infinito y por qué es un problema?

Es cuando un proceso  no termina y se repite infinitas veces y consume muchos recursos del sistema al consumir tiempo y memoria sin ningún fin

• ¿Puedo mezclar ciclos y condicionales?

Si se puede ya que con esto se puede condicionar distintos ciclos en ciertos bloques del código esto permite que no se creen bucles y el proceso sea más eficiente

**1.5.2 Replica el comportamiento de los siguientes ciclos for utilizando ciclos**

**while:**

for (let i = 0; i < 5; i++) {

console.log("El valor de i es: " + i);

}

for (let i = 10; i >= 2; i--) {

console.log("El valor de i es: " + i);

}

Texto

Descripción generada automáticamente

Resultado:

Texto

Descripción generada automáticamente

**1.5.3 Escribe un código en JavaScript que le pregunte a los usuarios cuánto**

**es 2 + 2. Si responden bien, mostramos un mensaje de felicitaciones,**

**pero si responden mal, volvemos a empezar.**

**fi Pista: puedes usar la función prompt de JavaScript.**

**4**

Texto

Descripción generada automáticamente

**Resultado:**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Captura de pantalla de un celular

Descripción generada automáticamente

**1 Test de JavaScript.**

**1.6 Listas**

**1.6.1 Responde las siguientes preguntas en la sección de comentarios:**

• ¿Qué es un array?

Se podría decir que es como una lista que puede estar formado por letras, números, cadenas u objetos

• ¿Qué es un objeto?

Es una entidad que une los métodos con propiedades usados para representar u organizar datos

• ¿Cuándo es mejor usar objetos o arrays?

Los objetos son útiles para almacenar datos con nombres descriptivos y asociar comportamientos a esos datos, mientras que los arrays son adecuados para almacenar listas ordenadas de elementos y realizar operaciones en conjunto con ellos

• ¿Puedo mezclar arrays con objetos o incluso objetos con arrays?

Si se puede mezclar ya que con esto se obtiene un funcionamiento más completo, se puede acceder a los elementos de un array dentro de un objeto y se puede lograr también de manera inversa

1.6.2 Crea un array con los siguientes elementos “Hola”, “¿”, “Cómo”, “estas”, “?”.

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

1.6.3 Accede al primer elemento de array que creaste en el ejercicio anterior e imprima en la consola

Pantalla negra con letras blancas

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1.6.4 Ahora supongamos que debes guardar datos de una persona. Crea un objeto con las siguientes propiedades: nombre, apellido, edad, hobbies (un array con varios elementos) y contacto (un objeto con las propiedades email y teléfono)

Captura de pantalla con la imagen de una pantalla

Descripción generada automáticamente

Texto

Descripción generada automáticamente

1.6.5 Accede al valor de la propiedad hobbies y imprímelo en la consola.

Captura de pantalla de un celular

Descripción generada automáticamente

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

1.6.6 Accede al valor de la propiedad email y imprímelo en la consola

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Captura de pantalla de un celular con texto e imagen

Descripción generada automáticamente

LINK DE GITHUB:

<https://github.com/alrivera4/Desarrollo_Repaso_Prueba.git>