**VARIABLES Y OPERACIONES**

1. **Responde las siguientes preguntas en la sección de comentarios:**

* ¿Qué es una variable y para qué sirve?

Es la representación de algún lugar en la memoria, que vamos a reservar para guardar algún valor de cualquier tipo

**Cajitas(espacios en memoria) donde podemos guardar información (dependiendo de los tipos y estructuras que soporte nuestro lenguaje)**

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

Declarar es reserva un espacio en JS (var edad;). Inicializar esa signarle un valor a la variable (edad=30); es decir que la variable ya tiene un valor, ya fue declarada, ya se puede usar.

**Var hoy en día tiene muchos problemas es por ello que es mejor usar let o const.**

**Let: permite cambiar el valor de la variable, cuando se cambia el valor de la carible se llama reinicializar**

**Const no lo permite, son variables constantes**

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

La suma de números da como resultado un valor y concatenar String da como resultado la unió de palabras que forma un texto

* ¿Cuál operador me permite sumar o concatenar?

El símbolo de más (+),

**suma numero y une palabras (strings)**

**2️. Determina el nombre y tipo de dato para almacenar en variables la siguiente información:**

* Nombre: string
* Apellido: String
* Nombre de usuario en Platzi: string
* Edad: Numérico
* Correo electrónico: String
* Mayor de edad:

**Boolean, porque es como una pregunta de true or false, porque lo que queremos saber es si la persona con la que estamos trabajando es mayor de edad o no**

* Dinero ahorrado: Numérico
* Deudas: Numérico

### 3. Traduce a código JavaScript las variables del ejemplo anterior y deja tu código en los comentarios.

**No se ponen con mayúscula porque eso esta reservado para otras cosas**

Var nombre= “pamela”;

Var apellido=“cortes”;

Var usuario= “pcortesj”;

Var edad=33;

Var correo = “[pamelacortesj@gmail.com](mailto:pamelacortesj@gmail.com)”

Var esMayorDeEdad= true;

Var ahorros = 10000

Var deudas = 1000

### 4️. Calcula e imprime las siguientes variables a partir de las variables del ejemplo anterior:

* Nombre completo (nombre y apellido)

**Var nombreCompleto= nombre + “ “ + apellido ( se deja un string con un espacio para que no queden los dos strings juntos es decir no quede pamelacortes sino pamela cortes**

* Dinero real (dinero ahorrado menos deudas)

Var dineroReal= ahorros - deudas

**FUNCIONES**

1. Responde las siguientes preguntas:

¿Que es una función?

Son un conjunto de sentencias que podemos utilizar, para generar ciertas acciones, con los valores que ya guardamos antes en las variables.

**Formas en las que encapsulamos bloques de código para que sean reutilizables en el futuro.**

**Un bloque de código son líneas de código, donde tenemos variables, ciclos condicionales, ejecuciones de otras funciones, etc.**

**Cuando hay un ;, un salto de línea o un cierre de corchetes significa que termino una línea de código, por lo tanto las funciones nos ayudan a guardalos en una burbuja o una caja, la cual se puede mandar a llamar, para ejecutarla cuando lo necesitemos**

¿Cuando me sirve una función en mi código?

Es un procedimiento o tareas, este conjunto de sentencias que vamos a hacer dentro de la función para hacer ciertos pasos o cálculos para poder validar, regresar o generar algo mas.

**Sirve cuando tenemos variables o bloques de códigos muy parecidos (con cambios que podrían ser parametros y argumentos) que podemos encapsular para reutilizar más de una vez en el futuro.**

**También nos sire para ordenar y mejorar la legibilidad de nuestro código.**

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

Los parametros son los datos que necesita para ejecutarse. Los parámetros son los nombres que aparecen en la definición de una función. Por su parte, los argumentos son los valores que le pasamos (y que, por tanto, recibe) una función.

**Parámetros: son lo que recibe una función cuando la estamos creando, las funciones recibe parámetros.**

**Argumentos: cuando la ejecutamos, es decir le estamos dando argumentos a una función.**

**Las funciones reciben parámetros cuando las estamos creando. Y les enviamos argumentos cuando las ejecutamos**

2. Convierte el siguiente código en una función, pero cambiando cuando sea necesario las variable contantes por parámetro y argumentos en una función:

**const** name = "Juan David";

**const** lastname = "Castro Gallego";

**const** completeName = name + lastname;

**const** nickname = "juandc";

console.log("Mi nombre es " + completeName + ", pero prefiero que me digas " + nickname + ".");

fuction saludo (name, lastname, nickname) {

const completeName = nombreCompleto (name, lastname);

console.log (“Mi nombre es” +completeName +, “pero prefiero que me digas”+ username + “.”);

}.

**CONDICIONALES**

¿Que es un condicional?

Es una condición para que algo se cumpla, son reglas para poder validar si algo es verdadero o falso y poder generar varias acciones con estas.

**Son la forma en la que ejecutamos un bloque de código u otro dependiendo de alguna condición o validación**

Que tipos de condicionales existen en Javascript y cuales son sus diferencias?

If

else

Else if: es validación, si no es esto entonces podría ser esto

**IF (else y else if), SWITCH**

**El condicional if con else y else if, nos permite hacer validaciones completamente distintas (si asi queremos) en cada validación o condicional.**

**El switch todos los cases (casos condicionales, valiaciones) se comparan con la misma variable o condicion que definimos en el switch.**

**El else if es: si no entonces aplícame esta otra validación**

Se pueden combinar funciones y condicionales?

**Si, las funciones pueden encapsular cualquier tipo de código incluyendo condicionales, de hecho los condicionales son muy buenos para las funciones porque, como las funciones deben hacer una cosa u otra dependiendo de lo parámetros que reciban, habrá casos en los que un console.log no será suficiente sino que tengamos que hacer algún tipo de validación extraña con condicionales, para dar una respuesta correscta incluso si tenemos parámetros concretamente diferentes.**

Replica el comportamiento del siguiente codigo que usa la sentencia que 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 Platzi durante un mes");

**break**;

**case** "Expert":

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

**break**;

**case** "ExpertDuo": console.log("Tú y alguien más pueden tomar TODOS los cursos de Platzi durante un año");

**break**;

}

**const tipoDeSuscripcion = "Basic";**

**if (tipodeSuscripcion == "Free"){**

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

**} else if (tipodeSuscripcion == "basic"){**

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

**} else if (tipodeSuscripcion == "Expert"){**

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

**} else if (tipodeSuscripcion == "ExpertDuo"){**

**console.log ("Tú y alguien más pueden tomar TODOS los cursos de Platzi durante un año");**

Replica el comportamientode tu condional anterior con if, el se, else if, pero ahora solo con if (sin else ni else if)

**function conseguirTipoDeSucripcion (suscripcion){**

**if (suscripcion == "Free"){**

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

**}**

**if (suscripcion == "basic"){**

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

**}**

**if (suscripcion == "Expert"){**

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

**}**

**if (suscripcion == "ExpertDuo"){**

**console.log ("Tú y alguien más pueden tomar TODOS los cursos de Platzi durante un año");**

**}**

**}**

**BONUS: Si ya eres un experto en el lenguaje, te desafío a comentar como replicar este comportamiento con array u objetos y un solo condicional**

**const tiposDeSuscripciones = {**

**free: "Solo puedes tomar los cursos gratis",**

**basic: "Puedes tomar casi todos los cursos de Platzi durante un mes",**

**expert: "Puedes tomar casi todos los cursos de Platzi durante un año",**

**expertduo: "Tú y alguien más pueden tomar TODOS los cursos de Platzi durante un año"**

**}**

**function conseguirTipoDeSuscripcion (suscripcion){**

**if tiposDeSuscripciones[suscripcion]{**

**console.log (tiposDeSuscripciones[suscripcion]);**

**return;**

**}**

**console.warn ("este tipo de suscripcion no existe")**

**}**

**CICLOS**

 Responde las siguientes preguntas en la sección de comentarios:

¿Qué es un ciclo?

**La forma de ejecutar un bloque de codigo hasta que se cumpla cierta condicion.**

**Es la forma de ejecutar un mismo código hasta que se cumpla una condición, esa condición puede ser que un número llegue a 10, que un número pase de cierto numero a cierto otro, que una variable pase a tener otro valor, que no necesariamente sea un numero, que una variable pase de ser true a false o false a true**

¿Qué tipos de ciclos existen en JavaScript?

**While, for, do while( es igual al while pero la primera vez no pregunta, no hace la validación, primero ejecuta el código y después hace la validación y dependiendo de esa validación vuelve a ejecutar el ciclo.)**

**El for= automáticamente nos obliga adefinir que pasa al principio y al final de cada ejecución, de codigo en nuestro ciclo, es decir el while hace una validacion y después esta el bloque de codigo, pero el bloque no esta obligado a cambiar esa condicion en algún momento para que pare el ciclo, en cambio el for automáticamente pide que inicialicemos una variable, que definamos algún momento en que nuestro ciclo lo debe terminar la validacion como tal y además alguna variable que deba cambiar**

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

**Es cuando la condición o validacion para terminar de validar un ciclo nunca se cumple, por lo tanto los ciclos, siguen siguen y siguen, hasta que el navegados dice “NO MAS” y termina toteando (dañando) la aplicación**

¿Puedo mezclar ciclos y condicionales?

**Por su puesto que SI, aunque los ciclos son una especie de condicionales nada nos impide agregar mas condicionales dentro del ciclo**

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); }

**let i= 0;**

**while (i < 5) {**

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

**i++;**

**}**

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

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

**let i =10**

**while (i >= 2){**

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

**1--;**

**}**

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.

💡 Pista: puedes usar la función prompt de JavaScript.

**Let respuesta;**

**while (respuesta != "4")**

**{**

**let pregunta = prompt("cuanto es 2+2?")**

**repuesta = pregunta;**

**}**

**ARRAY Y OBJETOS**

Responde las siguientes preguntas en la sección de comentarios:

¿Qué es un array?

Es una estructura de datos que va a guardar cierto valores, valores de numero, de string incluso objetos.

**Es una lista de elementos, los cuales no tienen que ser de un mismo tipo.**

**Ejm: const array= [1, “jaja”, true, false ];**

¿Qué es un objeto?

**Una lista de elementos PERO que cada elemento tiene un nombre clave**

**Ejm: const obj = {**

**Nombre: “pamela”,**

**Edad: 33,**

**};**

¿Cuándo es mejor usar objetos o arrays?

**Arrays: cuando lo que haremos en un elemento es lo mismo que en todos los demás (la regla se puede incumplir).**

**Objeto: cuando los nombres de cada elemento son importantes para nuestros algoritmo**

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

**Si. Los arrays pueden guardar objetos y cualquier otro tipo de dato y los objetos pueden guardar arrays entre sus propiedades**

Crea una función que pueda recibir cualquier array como parámetro e imprima su primer elemento.

**Function imprimirPrimerElementoArray (arr){**

**console.log (arr [0])**

**}**

Crea una función que pueda recibir cualquier array como parámetro e imprima todos sus elementos uno por uno (no se vale imprimir el array completo).

**Function imprimirElementoPorElemento (arr) {for (let i=0; i<**

**arr.length; i++)**

**{console.log (arr[i])}**

**}**

**imprimirElementoPorElemento (["rigoberto", "juanita", "natilla"])**

**cuando trabajamos con array no debemos trabajar con el igual**

Crea una función que pueda recibir cualquier objeto como parámetro e imprima todos sus elementos uno por uno (no se vale imprimir el objeto completo).

**Function imprimirElementoPorElementoObjeto(obj) {**

**Const arr=object.value(obj);**

**for (let i=0; i<**

**arr.length; i++)**

**{console.log (arr[i])}**

**}**

**Bonus**

**function conseguirTipoDeSucripcion {**

**if (Suscripcion == "Free"){**

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

**}**

**if (Suscripcion == "basic"){**

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

**}**

**if (Suscripcion == "Expert"){**

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

**}if (Suscripcion == "ExpertDuo"){**

**console.log ("Tú y alguien más pueden tomar TODOS los cursos de Platzi durante un año");**

**}**

**}**