1.

\* Que es una variable: Es un espacio de memoria, que se le asigna a un nombre asociado a dicho espacio, en el cual se pueden almacenar datos que varían según la función asignada.

\* Diferencia entre Declarar e inicializar una Variable: En la declaración creamos la variable, el nombre simbólico es añadido a un espacio de memoria, y el la inicialización se le asigna un valor especifico que es almacenado en el espacio de memoria.

\* Diferencia entre sumar números y concatenar strings: Sumar números se hace como su nombre lo indica con variables numéricas , con lo cual cada sumando va haciendo cambiar el resultado, encambio la concatenación lo que hace es ir uniendo las partes, formando un estilo de cadena que consta de todos los strings que se le han añadido a la operación, uno tras de otro.

\* El operador que permite sumar o concaternar es el símbolo +.

2. y 3.

Let Nombre = ´yeferson´; STRING

Let Apellido =’lasso ‘; string

Let user=’yeferess’; string

Let Edad=’23’; string

Let Email=’yeferess@...’; string

Let Mayor\_de\_edad= ´true’; booleana

Let Dinero\_ahorrado = 3000; number

Let Deudas = 5; number

Let Dinero = Dinero\_ahorrado – Deudas;

Console.log(“ Mi nombre es” + nombre + Apellido)

Console.log(“mi dinero es: “ + Dinero);

FUNCIONES

* definición de función: una función es un pedazo de código, con el cual puedo trabajar las veces que sea necesario, sin necesidad de estar repitiéndolo, la puedo usar con solo llamar el nombre de la función y los debidos datos de entrada.
* Diferencia entre parámetros y argumentos de una función: los parámetros, son la lista de variables que tiene una función y los argumentos son los valores que entran a la función cuando es invocada.

2 funcion:

Function nameFunction(nombre, lastname, nickname) {

Let nombre = nombre;

Let apellido= lastname;

Let apodo= nickname;

Let complete= nombre +” “ + apellido;

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

}

CONDICIONALES

* Definición de condicional: es un criterio que debe cumplir dentro del programa para que la sentencia del condicional se cumpla o no se cumpla según sea el caso.
* Condicionales: if, else, else if, ternario, switch.
* Si puedo combinar funciones y condicionales, todo depende la necesidad y lo eficaz del código.

Replica de código con if, else y else if

const tipoDeSuscripcion = "exp";

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=="expert+"){

    console.log("Tu y alguien mas pueden tomar todos los cursos de Platzi durante un año");

}

else {

    console.log("Aun no tienes una suscribción en platzi, adquierela ahora!");

}

* Codigo solo con if

const tipoDeSuscripcion = "free";

if(tipoDeSuscripcion=="free") {

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

}

if(tipoDeSuscripcion=="Basic") {

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

}

if(tipoDeSuscripcion=="expert") {

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

}

if (tipoDeSuscripcion=="expert+"){

    console.log("Tu y alguien mas pueden tomar todos los cursos de Platzi durante un año");

}

if(tipoDeSuscripcion != "free" && tipoDeSuscripcion!= "basic" && tipoDeSuscripcion!="expert" && tipoDeSuscripcion!="expert+") {

console.log("Aun no tienes una suscribción en platzi, adquierela ahora!");

}

CICLOS:

* Definición de ciclo: un ciclo es el retorno o reevaluación que se le da a ciertos parámetros, hasta que se cumpla una condición especifica.

\* Ciclos de javascript:

\* Ciclo For

\* Ciclo For Of

\* Ciclo For in

\* Ciclo While

\* Ciclo Do While

\* Un ciclo infinito, es un ciclo que no para de iterar y el problema es que al no salir de ese ciclo, el programa no para y no podrá cumplir con las demás instancias de código que tenga.

\* si se puede mezclar ciclos y condicionales.

Ciclos con While

let i= 0;

while (i < 5) {

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

    i += 1;

}

let v = 10;

while ( v > 2) {

     console.log("el valor de v es: " +v);

     v -=2;

}

* suma de 2+2

let c = 1;

while(c ==1) {

let a = prompt("cuanto es 2 + 2" );

if(a==4) {

alert("eso es correcto, Felicidades!!");

c = 0;

}

else {

alert("Incorrecto vuelve intentarlo!");

}

}

LISTAS

* Definición de Array: un Array es un tipo de dato que permite almacenar un conjunto de datos homogéneos.
* Definición de Objetos: es un conjunto de propiedades
* ¿Que es mejor un Array o un Objeto?: depende los requerimientos de programa.
* ¿Puedo mezclar arrays con objetos o incluso objetos con arrays? : esto no lo tengo muy claro, pero me imagino que si y lo voy aprender, o me daré cuenta que no se puede.
* Función que recibe un array como parámetro e imprime su primer valor

let motos = ["velocidad", "enduro", "motocrcoss", "Trial"];

function mifun(array){

let f= array;

console.log("el primer elemento del array es: " +f[0]);

}

mifun(motos);

* Función recibe array e imprime cada valor, uno por cada ciclo

let motos = ["velocidad", "enduro", "motocrcoss", "Trial"];

function mifun(array){

let f= array.length;

let l=0;

while( l<f) {

console.log("el primer elemento del array es: " + array[l]);

l += 1;

}

}

mifun(motos);

.