Pensamiento computacional: Proceso de resolución de problemas que se vale de la organización lógica y el análisis de datos. Posibilita el uso de una computadora u otro recurso para resolverlos.

El pensamiento computacional se trata del proceso mental a través del cual una persona se plantea un problema y para su posible solución utiliza una secuencia de instrucciones ejecutadas por una computadora, un humano o ambos. Es decir, aplica habilidades propias de la computación y del pensamiento crítico.

Programar: Proceso de diseñar y escribir una secuencia de instrucciones en un lenguaje que pueda ser entendido por una computadora.

Elementos:

Descomposición: Consiste en el procedimiento por el cual un problema de mayor complejidad se desarticula en series pequeñas (más manejables).

Patrones: Luego de la desarticulación del problema complejo, las series pequeñas son enfrentadas de forma individual. Esto permite que puedan ser resueltas de forma similar a problemas frecuentados anteriormente.

Descomposición: Despejar la información irrelevante y analizar, en forma lógica, el problema

Algoritmos: Se presentan pasos para la resolución del problema. Procedimiento por el cual un problema de mayor complejidad se desarticula en series pequeñas (más manejables).

Visual Studio Code: Es un IDE (Entorno de desarrollo integrado); cuando hablamos de un IDE nos referimos a un conjunto de herramientas diseñadas para facilitarnos la creación y el desarrollo de nuestros programas o aplicaciones.

Node JS (V8): Es un entorno de ejecución que nos permite ejecutar JavaScript por fuera de un navegador web. Todos los navegadores presentan un motor de JavaScript para leer y renderizar (representar gráficamente) código de JavaScript.

Las variables son espacios de memoria donde podemos almacenar distintos tipos de datos.

* Var
* Let
* Const

Funciones: Una función es un *bloque de código que nos permite realizar una tarea en particular*. Para que la función se ejecute, "algo" debe invocarla.

Una función es un bloque de código que nos permite agrupar una funcionalidad para usarla todas las veces que necesitemos. Normalmente realiza una tarea específica y retorna un valor como resultado

Funciones declaradas: Son aquellas que se declaran usando la estructura básica. Pueden recibir un nombre, escrito a continuación de la palabra reservada function, a través del cual podemos invocarla.

Las funciones con nombres son funciones nombradas.

Funciones expresadas: Son aquellas que se asignan como valor de una variable. En este caso, la función en si no tienen nombre, es una función anónima. Para invocarla podremos usar el nombre de la variable que declaremos.

Guardar resultados:



Parámetros: Son las variables que escribimos cuando definimos la función

Argumentos: Son los valores que enviamos cuando invocamos una función.

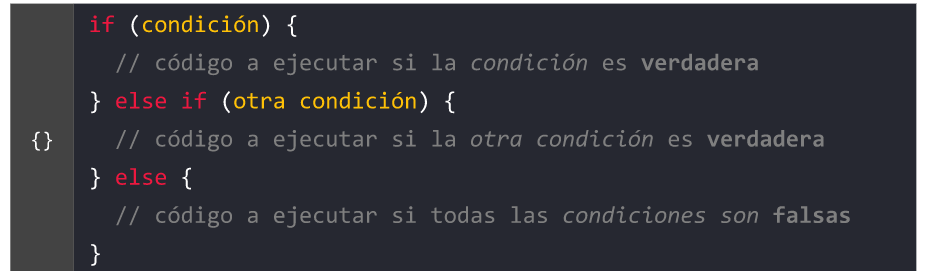
Scope : Se refiera al alcance que tiene una variable, es decir, desde donde podemos acceder a ella. En JavaScript

* Scope Local (dentro del bloque de la función)
* Scole Global (fuera de la función, se puede acceder a ella desde cualquier lugar del código

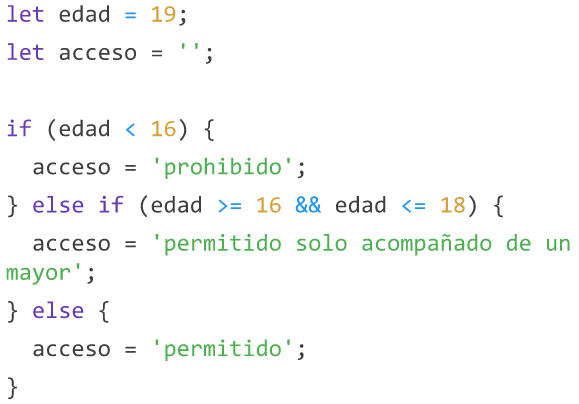
Operadores:

1. *DE ASIGNACIÓN =*
2. *ARITMÉTICOS + - \* /*
3. *DE COMPARACIÓN SIMPLE == o ¡=*
4. *DE COMPARACIÓN ESTRICTA === o ¡===*
5. *OPERADORES RELACIONALES < , > , <= , >=*
6. *LÓGICOS AND (&&) OR (||) NOT (¡)*
7. *Concatenacion +*

IF ; ELSE IF ; ELSE



Ej:



Swicht :



Resolucion 14: //typeof n1 !== "number"//

let num1= 10

let num2= 5

function dividir (num1 , num2) {

    if ( typeof num1 == 'number' && typeof num2 == 'number' && num2 != 0) {

    return num1 / num2;

    } else {

    return 'No es posible realizar la división';

    }

}

console.log(dividir(num1,num2));