1. **Variables**

* Una variable es una palabra que representa algo que cambia o experimenta algún tipo de modificación y carece de estabilidad, constancia y mutabilidad. Las variables son la manera como los programadores le dan nombre a un valor, para poder reusarlo, actualizarlo o simplemente registrarlo.
* Declarar una variable significa informar al sistema de la existencia de ella y la intención de hacer uso de ella.

Inicializar una variable significa asignarle un valor inicial.

* Al sumar 2 números se realiza una operación matemática que arroja como resultado otro número. Al concatenar dos Strings, lo que se hace es juntarlos para formar una nueva cadena de texto.
* El operador usado para concatenar o sumar es el operador “+”.

1. **Tipos de datos**

Nombre: string

Apellido: string

Nombre de usuario en Platzi: string

Edad: numero

Correo electrónico: string

Mayor de edad: booleano

Dinero ahorrado: numero

Deudas: numero

1. **Funciones**

* En JavaScript, una función es un bloque de código que se define una vez y se puede reutilizar en diferentes partes de un programa. Las funciones son un elemento fundamental en la programación ya que permiten encapsular una tarea específica, de manera que se pueda ejecutar múltiples veces sin necesidad de repetir el código. Una función puede ser invocada desde cualquier parte del código, lo que ayuda a modularizar y organizar el programa.
* Si tienes una tarea que se repite en varios lugares de tu programa, encapsular esa tarea en una función te permite escribir el código una vez y reutilizarlo en múltiples partes de tu programa. Utilizar funciones en tu código puede mejorar la claridad, la eficiencia y la facilidad de mantenimiento, al tiempo que permite una mayor reutilización del código.
* Argumentos: Son los valores reales que se pasan a una función cuando esta es invocada. Parámetros: Son los nombres utilizados dentro de la definición de una función para representar los valores que se espera recibir cuando se llama a esa función.

1. **Condicionales**

* Es una estructura de control que permite tomar decisiones en función de una condición. Estos condicionales se utilizan para ejecutar diferentes bloques de código según si una expresión lógica es verdadera o falsa.
* else if: Permite verificar condiciones adicionales si la condición del if no se cumple.
* Operador ternario: Es una forma abreviada de escribir un condicional if en una sola línea.
* La combinación de condicionales y funciones es fundamental en la programación, ya que permite escribir código más modular y reutilizable. Se pueden diseñar funciones que realicen tareas específicas y luego utilizar condicionales para tomar decisiones basadas en los resultados de esas funciones.

1. **Ciclos**

* es una estructura de control que permite ejecutar repetidamente un bloque de código mientras se cumpla una condición específica. Los ciclos son útiles para realizar tareas repetitivas de manera eficiente sin tener que escribir el mismo código una y otra vez.
* For, while, do while,
* Un ciclo infinito sucede cuando la condición nunca se vuelve falsa, es un problema porque puede provocar problemas de rendimiento en el programa o en el sistema, bloqueo del programa
* Se puede utilizar y combinar ciclos y condicionales para controlar el flujo de ejecución en un determinado caso. Para realizar acciones específicas según una condición dentro de cada iteración del ciclo.

1. **Listas**

* Un array es una estructura de datos que se utiliza para almacenar una colección ordenada de elementos. Estos elementos pueden ser de cualquier tipo de dato, como números, cadenas, booleanos, objetos, funciones, u otros arrays, permitiendo así crear estructuras de datos más complejas y flexibles.
* Un objeto es una estructura de datos que permite almacenar y organizar información de manera más compleja que un simple array. Un objeto puede contener propiedades y métodos, donde las propiedades son pares clave-valor que representan características o atributos del objeto, y los métodos son funciones asociadas al objeto.
* Los objetos son ideales cuando quieres organizar datos utilizando claves descriptivas (o propiedades) en lugar de índices numéricos.
* Los objetos son útiles para modelar entidades del mundo real que tienen propiedades y comportamientos asociados. Puedes incluir funciones (métodos) dentro del objeto.
* Los arrays son ideales cuando trabajas con una secuencia de elementos ordenados y necesitas acceder a ellos por índices numéricos.
* Los arrays son eficientes para realizar operaciones en bloques de datos, como filtrar, mapear o reducir elementos.
* Cuando todos los elementos en la colección son del mismo tipo o representan valores similares, los arrays proporcionan una estructura de datos simple y eficiente.
* En JavaScript puedes mezclar objetos con arrays y arrays con objetos. Esto significa que puedes tener arrays que contienen objetos, objetos que contienen arrays o incluso arrays que contienen objetos que a su vez contienen arrays.