1. **Variable:** es un espacio de almacenamiento que tiene un nombre simbólico (un identificador) y se utiliza para contener un valor que puede cambiar a lo largo de la ejecución del programa.
2. **Constante:** es un valor que no cambia durante la ejecución del programa. A diferencia de las variables, una vez que se asigna un valor a una constante, este no puede ser modificado.
3. **Tipos de datos comunes en Python:**

* **Enteros (int):** Números sin decimales.
  + Ejemplo: 1, -5, 42.
* **Flotantes (float):** Números con decimales.
  + Ejemplo: 3.14, -0.001, 2.0.
* **Cadenas de texto (str):** Secuencias de caracteres.
  + Ejemplo: "Hola", "1234".
* **Booleanos (bool):** Verdadero o falso.
  + Ejemplo: True, False.
* **Listas (list):** Colecciones de elementos de diferentes tipos y tamaño dinámico.
  + Ejemplo: [1, "Hola", 3.14].
* **Tuplas (tuple):** Colecciones inmutables.
  + Ejemplo: (1, "Hola", 3.14).
* **Conjuntos (set):** Elementos únicos y desordenados.
  + Ejemplo: {1, 2, 3}.
* **Diccionarios (dict):** Pares clave-valor.
  + Ejemplo: {"nombre": "Juan", "edad": 30}.

1. **Los operadores aritméticos:**
   * **Suma (+):** 3 + 2 → 5
   * **Resta (-):** 5 - 3 → 2
   * **Multiplicación (\*):** 4 \* 2 → 8
   * **División (/):** 10 / 2 → 5.0
   * **División entera (//):** 10 // 3 → 3
   * **Módulo (%):** 10 % 3 → 1
   * **Exponenciación (\*\***): 2 \*\* 3 → 8
2. **Los operadores relacionales:**
   * **Igual a (==):** 5 == 5 → True
   * **Distinto de (!=):** 5 != 3 → True
   * **Mayor que (>):** 5 > 3 → True
   * **Menor que (<):** 3 < 5 → True
   * **Mayor o igual que (>=):** 5 >= 5 → True
   * **Menor o igual que (<=):** 3 <= 5 → True
3. **Los operadores lógicos:**
   * **Y lógico (and):** True and False → False
   * **O lógico (or):** True or False → True
   * **No lógico (not):** not True → False
4. **Los tipos de datos complejos:**

* **Listas (list):** Colecciones ordenadas y mutables. [1, "Hola", 3.14]
* **Tuplas (tuple):** Colecciones ordenadas e inmutables. (1, "Hola", 3.14)
* **Conjuntos (set):** Colecciones desordenadas y mutables de elementos únicos. {1, 2, 3}
* **Diccionarios (dict):** Colecciones de pares clave-valor. {"nombre": "Juan", "edad": 30}
* **Objetos (object):** Instancias de clases para estructuras personalizadas.
* **Nombres de módulo (module):** Permiten importar y usar código externo.
* **Funciones (function):** Bloques de código reutilizables.

1. La **indentación:** es crucial y se utiliza para definir la estructura del código. La indentación indica qué bloques de código pertenecen a una misma estructura de control, como bucles, condicionales y funciones.
2. **Cometarios:** Se puede usar una cadena de texto de varias líneas (también llamada cadena de docstring) delimitada por triple comillas """ o '''.
3. **Encoding** (codificación): se refiere al proceso de convertir datos de un formato a otro, particularmente en relación con la representación de caracteres. Esto es crucial para asegurar que los datos se interpreten correctamente en diferentes sistemas y plataformas.