**Variables y tipos de datos en Python**

**Variables:** Son contenedores para almacenar datos. No se declara el tipo de la variable, Python lo deduce automáticamente.

**Ejemplo:** x = 5, nombre = "Diego"

**Tipos de datos:**

Enteros (int): Números enteros. **Ej.:** x = 10

Flotantes (float): Números decimales. **Ej.:** y = 10.5

Cadenas (str): Texto. **Ej.:** mensaje = "Hola"

Booleanos (bool): True o False. **Ej.:** is\_active = True

**Operadores aritméticos y de comparación Aritméticos:**

Suma: +, Resta: -, Multiplicación: \*, División: /

División entera: //, Módulo: %, Exponente: \*\*

**Ej.:** resultado = 10 + 5, potencia = 2 \*\* 3

**Comparación:**

Igual: ==, Diferente: !=, Mayor: >, Menor: <

Mayor o igual: >=, Menor o igual: <=

**Ej.:** x == 5, y != 3

**Estructuras de control**

**Condicionales:**

if: Ejecuta un bloque de código si la condición es verdadera.

elif: (else if) Verifica otra condición si la anterior es falsa.

else: Ejecuta un bloque si ninguna condición anterior se cumple.

**Ej.:**

if x > 10:

print("Mayor que 10")

elif x == 10:

print("Es 10")

else:

print("Menor que 10")

**Bucles:**

for: Itera sobre una secuencia (lista, cadena, etc.).

Ej.: for i in range(5): print(i)

while: Repite un bloque de código mientras una condición sea verdadera.

**Ej.:**

i = 0

while i < 5:

print(i)

i += 1

**Listas**

**Creación y manipulación:**

**Lista vacía:** lista = []

**Lista con elementos:** numeros = [1, 2, 3, 4]

**Acceso:** primero = numeros[0]

**Modificación:** numeros[1] = 10

**Métodos comunes:**

append(): Añade un elemento al final. **Ej.:** numeros.append(5)

len(): Devuelve la longitud de la lista. **Ej.:** longitud = len(numeros)

remove(): Elimina el primer elemento igual al valor dado. **Ej.:** numeros.remove(2)

**Funciones -Definición y llamada:**

Se definen con def.

**Ej.:**

def saludar(nombre):

return "Hola " + nombre

print(saludar("Diego"))

**Parámetros y retorno de valores:**

Las funciones pueden recibir parámetros y retornar valores usando return.

Conceptos básicos de Programación Orientada a Objetos

**Clases y objetos:**

**Clase:** Es un plano para crear objetos (instancias). **Ej.:**

class Perro:

def \_\_init\_\_(self, nombre, raza):

self.nombre = nombre

self.raza = raza

**Objeto: Instancia de una clase.**

**Ej.:** mi\_perro = Perro("Rex", "Labrador")

**Métodos y atributos:**

**Atributos:** Variables asociadas a un objeto.

**Ej.:** mi\_perro.nombre

**Métodos:** Funciones asociadas a un objeto.

**Ej.:**

def ladrar(self):

return "Guau!"

**Módulos básicos de Python**

**Importar módulos:** Usando import.

**Ej.:**

import random

numero = random.randint(1, 10)

**Módulos comunes:**

**date:** Manejo de fechas. **Ej.:** from datetime import date

**random:** Generación de números aleatorios. **Ej.:** random.random()

**Entrada y salida básica**

**Entrada:** Usar input() para recibir datos del usuario. **Ej.:** nombre = input("¿Cómo te llamas?")

**Salida:** Usar print() para mostrar datos. **Ej.:** print("Hola, " + nombre)