git config --global core.autocrlf true

type NUL > .gitignore

|  |
| --- |
| 1.MainPrincipal.py |
| if \_\_name\_\_ == '\_\_main\_\_':  print("Hola mundo") |

|  |
| --- |
| 2.Variables.py |
| if \_\_name\_\_ == "\_\_main\_\_":  a=10  b=20  c=a+b  print(f"la suma de {a} + {b} = {c}") |

|  |
| --- |
| 3.TiposDeDatos.py |
| if \_\_name\_\_ == "\_\_main\_\_":  *# Tipos de datos básicos* entero = 42 *# int* decimal = 3.14 *# float* complejo = 2 + 3j *# complex* booleano = True *# bool* cadena = "Hola, Python!" *# str* binario = bytes([50, 100, 150]) *# bytes* print("Tipos básicos:")  print("Entero:", entero)  print("Decimal:", decimal)  print("Complejo:", complejo)  print("Booleano:", booleano)  print("Cadena:", cadena)  print("Binario:", binario, "\n")   *# Estructuras de datos avanzadas* lista = [1, 2, 3, "cuatro"] *# list* tupla = (5, 6, 7, "ocho") *# tuple* conjunto = {9, 10, "once", "doce"} *# set* diccionario = {"clave1": "valor1", "clave2": 20} *# dict* print("Estructuras avanzadas:")  print("Lista:", lista)  print("Tupla:", tupla)  print("Conjunto:", conjunto)  print("Diccionario:", diccionario, "\n")   *# Otros tipos especiales* nulo = None *# NoneType* rango = range(3) *# range (equivale a [0, 1, 2])* print("Tipos especiales:")  print("Nulo:", nulo)  print("Rango:", list(rango)) *# Convertimos a lista para ver los valores   # Ejemplo de iteración con el tipo range* print("\nIterando sobre el rango:")  for numero in rango:  print(numero) |

|  |
| --- |
| 4.If.py |
| if \_\_name\_\_ == "\_\_main\_\_":  a=int(input("Proporciona el valor de A "))  b=int(input("Proporciona el valor de B "))  c=int(input("Proporciona el valor de C "))   if a>b:  if a>c:  print(f"El mayor es {a}")  else:  print(f"El mayor es {b}")  elif b>c:  print(f"El mayor es {b}")  else:  print(f"El mayor es {c}") |

|  |
| --- |
| 5.CicloWhile.py |
| *#Muestra la potencia de un numero X a la Y* if \_\_name\_\_ == "\_\_main\_\_":  print("Este programa calcula la potencia de x a la y")  x=int(input("Proporciona el valor de la base "))  y=int(input("Proporciona el valor de la potencia "))   i=1  resultado=1  while i<=y:  print(f"Conteo {i}")  resultado\*=x  i+=1;   print(f"El resultado de {x} a la {y} es {resultado}") |

|  |
| --- |
| 6.CicloFor.py |
| if \_\_name\_\_ == "\_\_main\_\_":  for i in range(10):  print(f"{i}")   print("--------------------")  for j in range(1,20):  print(f"{j}", end=" ")   print("\n--------------------")  lista={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16}  for lista in lista:  print(lista) |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

git config --global core.autocrlf true

type NUL > .gitignore

1. Tipos de datos básicos

Estos son los tipos de datos fundamentales en Python:

int: Representa números enteros de cualquier tamaño, como 42 o -7.

float: Representa números de punto flotante (decimales), como 3.14 o -0.001.

complex: Representa números complejos, con una parte real y una imaginaria, como 3 + 4j.

bool: Representa valores booleanos, True o False.

str: Representa cadenas de texto, como "hello" o "123".

bytes: Representa datos binarios inmutables, usados principalmente para manipulación de datos binarios en archivos o redes.

2. Estructuras de datos avanzadas

Python proporciona tipos de datos más complejos para almacenar colecciones de elementos:

list: Una colección ordenada y mutable de elementos, como [1, 2, 3].

tuple: Una colección ordenada e inmutable de elementos, como (1, 2, 3).

set: Una colección desordenada de elementos únicos, como {1, 2, 3}.

dict: Una colección desordenada de pares clave-valor, como {"key1": "value1", "key2": "value2"}.

Otros tipos especiales

Python incluye algunos tipos de datos adicionales útiles en contextos específicos:

NoneType: Representa el valor None, usado para indicar la ausencia de valor o un valor nulo.

range: Representa una secuencia de números, usado comúnmente en bucles, como range(5)