Contenido

```
#Variables
1.Asignación directa
variable = 30
2.Asignación múltiple
a , b ,c = 1,2,3
3.Asignación con el mismo valor a varias variables
x=y=z=0
4.Declaración de variables sin asignarles un valor inicial
variable
5.Declaración de variables con valores por defecto
variable_con_valor = 42 if 1>5 else 0
6.Declaración de variables con tipos de datos
variable_con_tipo: int = 10
7.Declaración de variables en múltiples líneas:
variable_larga = (5 + 15+ 5)
8.Uso de variables globales
global variable_global
variable_global = 100
9.Variables Locales
def ejemplo():
    variable_local = 5
```

```
#Condicionales
1.If statement (Declaración if):
x = 10

if x > 5:
    print("x es mayor que 5")
2.If-else statement
y = 3

if y % 2 == 0:
    print("y es un número par")
else:
    print("y es un número impar")
3.If-elif-else statement
z = 0
if z > 0:
    print("z es positivo")
elif z < 0:
    print("z es negativo")
else:
    print("z es igual a cero")
```

```
4.Expresiones condicionales
a = 15
b = 10
max_value = a if a > b else b
print("El valor máximo es:", max_value)
5.Condicionales anidadas
temperatura = 25

if temperatura > 30:
    print("Hace mucho calor")
else:
    if temperatura > 20:
        print("Hace calor")
    else:
        print("Hace fresco")
6.Short-circuit evaluation
es_hombre = True
tiene_barba = False

if es_hombre and tiene_barba:
    print("Es un hombre con barba")
else:
    print("No es un hombre con barba")
```

```
#Bucles
FOR
1.For itera sobre una lista
frutas = ["manzana", "banana", "cereza"]
for fruta in frutas:
    print(fruta)
2.For itera sobre un rango
for i in range(5):
    print(i)
3.For itera sobre una cadena de texto
mensaje = "Hola"
for letra in mensaje:
    print(letra)
4.For itera sobre un diccionario
mi_diccionario = {"a": 1, "b": 2, "c": 3}
for clave, valor in mi_diccionario.items():
    print(clave, valor)
```

5.For función enumerate()
Es especialmente útil cuando necesitas realizar un seguimiento del índice mientras iteras.

```
frutas = ["manzana", "plátano", "uva"]

for indice, fruta in enumerate(frutas):
    print(f"Índice: {indice}, Fruta: {fruta}")
```

6.Bucles Anidados:
Puedes tener bucles for dentro de otros bucles para iterar

```
matriz = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

for fila in matriz:
    for elemento in fila:
        print(elemento)
```

7.Utilizar la función zip()
La función zip() combina elementos de dos o más secuencias en tuplas.

```
nombres = ["Juan", "María", "Pedro"]
edades = [25, 30, 22]
for nombre, edad in zip(nombres, edades):
    print(f"{nombre} tiene {edad} años")
```

8.Uso de break y continue:
break: Termina prematuramente el bucle.

```
for i in range(10):
    if i == 3:
        break  # Termina el bucle cuando i es 3
    print(i)
```

continue: Salta a la siguiente iteración sin ejecutar el resto

```
for i in range(5):
    if i == 2:
        continue  # Salta a la siguiente iteración cuando i es 2
    print(i)
```

WHILE
1.Mientras una condición sea verdadera

```
contador = 0
while contador < 5:
    print(contador)
    contador += 1
```

```
#Funciones:
1.Puedes definir funciones con la palabra clave def
def suma(a, b):
    return a + b
resultado = suma(3, 4)
print(resultado)
```

```
#Listas y Diccionarios:
Las listas almacenan elementos ordenados, mientras que los diccionarios
almacenan pares clave-valor
mi_lista = [1, 2, 3, 4]
mi_diccionario = {"clave": "valor", "nombre": "Juan"}

print(mi_lista[0])        # Acceder al primer elemento de la lista
print(mi_diccionario["nombre"])  # Acceder al valor asociado a la clave
"nombre"
```

```
#Apertura y Cierre de Archivos:
1.Para abrir un archivo, puedes utilizar la función open():
1.1 Abre un archivo en modo lectura
archivo_lectura = open('archivo.txt', 'r')

1.2 Abre un archivo en modo escritura
archivo_escritura = open('archivo.txt', 'w')

1.3 Abre un archivo en modo apendizaje (añade al final)
archivo_append = open('archivo.txt', 'a')
2.Recuerda cerrar el archivo después de usarlo con el método close():
archivo_lectura.close()
archivo_escritura.close()
archivo_append.close()
3.Lectura de Archivos:
Para leer el contenido de un archivo, puedes utilizar varios métodos. Uno de
los más comunes es read():
with open('archivo.txt', 'r') as archivo:
    contenido = archivo.read()
    print(contenido)
También puedes leer el archivo línea por línea con readline() o todas las
líneas a la vez con readlines():
with open('archivo.txt', 'r') as archivo:
    # Lee una línea
```

```
    linea = archivo.readline()
    print(linea)

    # Lee todas las líneas en una lista
    lineas = archivo.readlines()
    print(lineas)
```

4.Escritura de Archivos:

5.Para escribir en un archivo, puedes usar el método write()

```
with open('archivo.txt', 'w') as archivo:
    archivo.write('Hola, mundo!\n')
    archivo.write('Esta es otra línea.')
```

6.Si deseas agregar contenido al final de un archivo, utiliza el modo de apendizaje ('a'):

```
with open('archivo.txt', 'a') as archivo:
    archivo.write('Esta línea se añade al final.')
```