

Clase 2.4

Sentencias de Repetición

Las sentencias de repetición For y While, son herramientas esenciales en programación que permiten ejecutar un conjunto de instrucciones de manera iterativa.

- ▶ El bucle `while` realiza repeticiones mientras una condición específica sea verdadera.
- ▶ El bucle `for` se utiliza para iteraciones con un número conocido de repeticiones, se debe dar datos de inicialización y de condición.

Ciclo for

El ciclo FOR permite ejecutar una tarea, un determinado número de veces (finito por lo general)

```
# Definimos range(x): genera valores de 0 a x-1
for i in range(9):
    # i es el indice de la iteracion
    # no necesita ser definido
    print(f"{i}.No debo botar la basura")
```

Ciclo While

While permite ejecutar las fases mientras una condición se cumpla

```
# Se entrega valor inicial que cumpla condicion
# Para que el loop pueda iniciar
a = 5
# Mientras a sea positivo
while(a>0):
    # Solicita input al usuario
    a = int(input("ingrese un valor"))
    # Imprime numero ingresado por el usuario
    print(f"el valor de a es :{a}")
```

Excepciones

Una excepción es un evento anormal o inesperado que ocurre durante la ejecución de un programa y que afecta el flujo normal de ejecución.

Cuando ocurre una excepción, el programa genera un proceso que encapsula la información sobre el error. Este objeto se pasa a un administrador de excepciones.

Excepciones (Sintaxis)

```
try:
    #Codigo que debe intentar funcionar
except <tipo_error>:
    # Que hacer si tenemos un error como tipo_error
except <tipo_error2>:
    # Idem
# .....
else:
    # Que hacer si no hubo errores
finally:
    # Corre siempre (haya error o no)
```

Excepciones más comunes

Divisiones por cero (10/0) → `ZeroDivisionError`

Tipos de datos incorrectos (Aparece cuando intentas concatenar una cadena con un número.) → `TypeError`

Error de valores (Se genera cuando hay un problema con el tipo o valor de los datos que estás manipulando, como convertir una cadena no numérica a un número) → `ValueError`

Archivos no encontrados (Dar ubicaciones de archivo no existentes) → `FileNotFoundError`

Error de sintaxis (Se genera cuando hay un error de sintaxis en tu código) → `SyntaxError`

Excepciones

```
# EJEMPLO
try:
    resultado = 10/ 0
except ZeroDivisionError as nombre_de_excepcion:
    print(f"Se produjo una excepcion:
          {nombre_de_excepcion}")
else:
    print("No se produjo ninguna excepcion")
finally:
    print("Este bloque se ejecuta siempre")
```


Ejercicios

Ejercicio 1

Construya programa en python que solicite input al usuario:
¿hasta que número quiere ver los números impares? e imprima
la serie de números impares separada por saltos de línea

```
# Solicitar al usuario un numero
num = int(input(' Hasta que numero quiere ver los
numeros impares?'))

# Imprimir los numeros impares hasta ese numero
for i in range(num+1):
    if i % 2 != 0:
        print(i)
```

Ejercicio 2

Escribe un programa que pida al usuario que introduzca su nombre. Sin embargo, el nombre debe ser al menos de 3 caracteres y no más de 50 caracteres. Si el usuario introduce un nombre no válido, pídele que introduzca su nombre de nuevo.

```
while True:
    name = input("Por favor, introduce tu nombre: ")
    if 3 <= len(name) <= 50:
        print(f"Hola, {name}!")
        break # Salir del bucle
    else:
        print("El nombre debe tener entre 3 y
        50 caracteres. Por favor,
        intenta de nuevo.")
```

Ejercicio 3

Escribe un programa que pida al usuario que introduzca un número entero y que imprima la tabla de multiplicar de ese número desde el 1 hasta el 10.

```
num = int(input('Introduce un numero para mostrar  
su tabla de multiplicar: '))  
  
# Imprimir la tabla de multiplicar de ese numero  
for i in range(10):  
    print(f"{num} x {i+1} = {num*(i+1)}")
```

Ejercicio 4

Escribe un programa que pida al usuario que introduzca un número. El programa debe convertir la entrada del usuario a un float. Si la conversión es exitosa, el programa debe imprimir el número inverso (1 dividido por el número).

```
try:
    # Solicitar al usuario un numero
    num = input('Por favor, introduce un numero: ')
    num = float(num)
except ValueError:
    print("Eso no es un numero valido")
else:
    print(f"El inverso de tu numero es {1/num}")
finally:
    print("Fin")
```