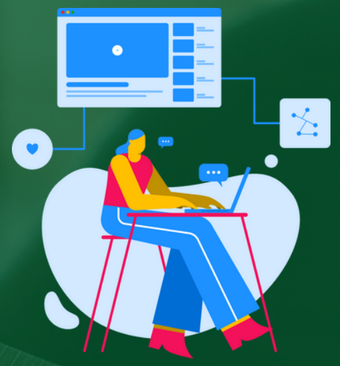
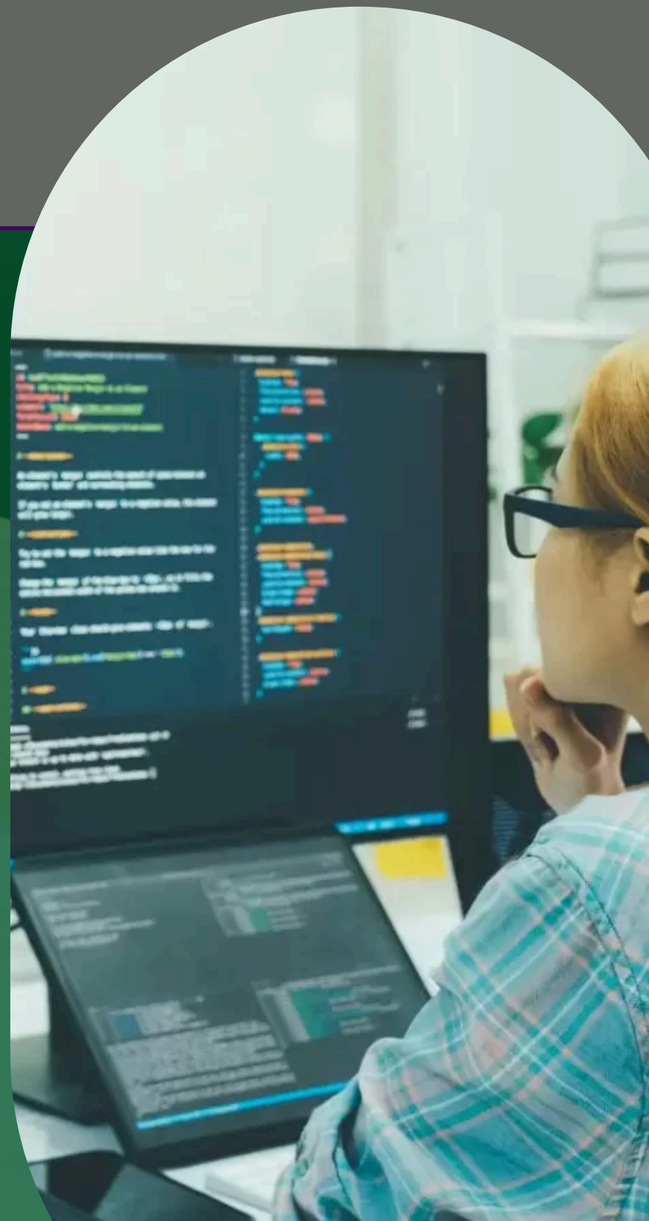


Primeros Pasos en Python: Curso Básico para Principiantes



DOCENTE: JEAN MARCOS CABALLERO REQUIZ,
BACHILLER EN INGENIERIA ELECTRONICA

¿QUÉ ES EL CONTROL DE FLUJO?

Se refiere a la capacidad de dirigir el flujo de ejecución de un programa mediante la toma de decisiones y la repetición de ciertas acciones. Esto se logra a través de diferentes estructuras de control, como condicionales y bucles, que permiten ejecutar ciertos bloques de código bajo ciertas condiciones o repetirlos varias veces.

¿QUÉ ES LA FUNCIÓN INPUT?

Es una función integrada que se utiliza para obtener la entrada del usuario desde la consola, permitiendo al usuario ingresar datos y proporcionarlos al programa en tiempo de ejecución.

INPUT

Ejemplos para enter mejor:

```
name = input("Quién es usted")
print("Bienvenido", name)
```

Si queremos leer un número del usuario, debemos convertir de una cadena a un número utilizando

```
X = int(input("Ingrese un valor:") # para poder ingresar solo valores enteros
print( "DATO DE VALOR INT:", X)

Y= float(input("Ingrese un valor:") # para poder ingresar solo valores flotantes
print( "DATO DE VALOR FLOAT:", Y)
```

¿QUE SON LAS CONDICIONES?

Las condiciones son expresiones que se evalúan como verdaderas o falsas y se utilizan para controlar el flujo de un programa. Se utilizan principalmente en estructuras de control como los condicionales y los bucles para tomar decisiones basadas en ciertas condiciones.

CONDICIONES

IF

Se utiliza para ejecutar un bloque de código si una condición específica es verdadera.

ELIF

Se utiliza para verificar múltiples condiciones después de la primera instrucción if.

ESLE

Se ejecuta si ninguna de las condiciones anteriores es verdadera.

IF

```
1 #QUE ES LA CONDICION IF:
2
3 U = 9
4 if U < 10:
5   print(U, "Es menor que 10")
6
7 print("Fin del programa")
```

```
9 Es menor que 10
Fin del programa
```

ELSE

```
1 #QUE ES LA CONDICION ELSE:
2
3 U = 10
4 if U < 10:
5   print(U, "Es menor que 10")
6 else:
7   print(U, "Es mayor o igual que 10")
8
9 print("Fin del programa")
```

```
10 Es mayor o igual que 10
Fin del programa
```

ELIF

```
1 #QUE ES UNA CONDICION ELIF:
2
3 U = 10
4
5 if U < 10:
6   print(U, "Es menor que 10")
7 elif U == 10:
8   print(U, "Es igual a 10")
9 else:
10   print(U, "Es mayor que 10")
11
12 print("Fin del programa")
```

```
10 Es igual a 10
Fin del programa
```

EJERCICIO:

La variación de la temperatura en el Perú es muy irregular durante las fechas de abril y mayo. Usted es un técnico de laboratorio y le solicitan que realice un código donde sabiendo la temperatura del ambiente que se diga si es alta o baja la temperatura.

DATOS DE TEMPERATURAS:

30° - Muy alto
25° - Alto
20° - Perfecto
10° - Muy Baja

¿QUÉ SON LOS BUCLES?

los bucles son estructuras de control que permiten repetir un bloque de código múltiples veces, con base en una condición dada. Los bucles son fundamentales en la programación, ya que facilitan la automatización de tareas repetitivas y la iteración sobre colecciones de datos.

CREACIÓN DE BUCLES FOR

Se utiliza para iterar sobre una secuencia de elementos, como una lista, una cadena de texto, una tupla o un rango numérico, y ejecutar un bloque de código para cada elemento en la secuencia.

```
1 #Como utilizar el for
2
3 for num in range(1,9):
4   print(num)
```

Mostrar el resultado oculto

```
1 Nombre = "Marcos Caballero"
2 for letra in Nombre:
3     print(letra)
```

Mostrar el resultado oculto

CREACIÓN DE BUCLES FOR

Mas ejemplos de For:

```
1 numeros = range(1,6)
2 suma = 0
3 for numero in numeros:
4     suma += numero
5 print("La suma de los números es:", suma)
```

Mostrar el resultado oculto

CREACIÓN DE BUCLES WHILE

En Python se utiliza para ejecutar repetidamente un bloque de código mientras se cumpla una condición especificada.

La condición es una expresión booleana que se evalúa en cada iteración del bucle. Mientras la condición sea verdadera, el bloque de código dentro del bucle se seguirá ejecutando. Si en algún momento la condición se vuelve falsa, el bucle se detendrá y la ejecución del programa continuará con la siguiente instrucción después del bucle.

CREACIÓN DE BUCLES WHILE

Mas ejemplos de WHILE:

```
1 contador = 1
2 while contador <= 5:
3     print(contador)
4     contador += 1
```

Mostrar el resultado oculto

```
1 suma = 0
2 numero = int(input("Ingrese un número (negativo para terminar): "))
3
4 while numero >= 0:
5     suma += numero
6     numero = int(input("Ingrese un número (negativo para terminar): "))
7
8 print("La suma de los números es:", suma)
```

Mostrar el resultado oculto

CREACIÓN DE BUCLES WHILE

Mas ejemplos de WHILE:

```python
contraseña_correcta = "python"
contraseña = input("Ingrese la contraseña: ")

while contraseña != contraseña_correcta:
    print("Contraseña incorrecta, intente nuevamente.")
    contraseña = input("Ingrese la contraseña: ")

print("Contraseña correcta, acceso concedido.")
```

Mostrar el resultado oculto

CONTROL DE FLUJO

Se refiere a cómo se controla la ejecución de bloques de código en un programa. Las instrucciones break y continue son dos herramientas esenciales para manejar el flujo de los bucles (for y while).

CONTROL DE FLUJO BREAK

La instrucción break se utiliza para salir inmediatamente de un bucle, independientemente de la condición del bucle. Cuando se encuentra break dentro de un bucle, el control del programa salta a la primera línea de código después del bloque del bucle.

```
1 contador = 0
2
3 while True:
4     print(contador)
5     contador += 1
6     if contador == 45:
7         break
8
9 print("Bucle terminado.")
```

Mostrar el resultado oculto

```
1 numeros = range(1,9)
2
3 for numero in numeros:
4     if numero == 5:
5         break
6     print(numero)
7
8 print("Bucle terminado.")
```

Mostrar el resultado oculto

CONTROL DE FLUJO CONTINUE

La instrucción continue se utiliza para saltar la iteración actual de un bucle y pasar a la siguiente iteración. Cuando se encuentra continue dentro de un bucle, el control del programa salta inmediatamente al comienzo de la siguiente iteración del bucle.

```
1 contador = 0
2 while contador < 10:
3     contador += 1
4     if contador % 2 == 0:
5         continue
6     print(contador)
```

Mostrar el resultado oculto

```
1 for numero in range(1, 11):
2     if numero % 2 == 0:
3         continue
4     print(numero)
```

Mostrar el resultado oculto

Ejercicio:

Imagina que estás desarrollando un sistema básico para gestionar el inventario de una tienda. Necesitas escribir un programa que permita al usuario:

1. **Agregar productos al inventario.**
2. **Ver la lista de productos en el inventario.**
3. **Buscar un producto en el inventario.**
4. **Salir del programa.**

GRACIAS

¡...POR SU PARTICIPACION POR EL DIA DE HOY...!

LOS ESPERO EN LA PROXIMA CLASE

AVANZA A LA SIGUIENTE CLASE