**Module 3: Python Program Flow**

Indentation

Indentation in Python refers to the spaces at the beginning of a code line. Unlike many other programming languages, Python uses indentation to define blocks of code instead of curly braces {}.

Indentation is used to:

* Define the **body of loops, functions, conditionals, and classes**.
* Show the **logical structure** of the code.
* Make the code more **readable and consistent**.

Example

def greet(name):

if name:

print("Hello,", name)

else:

print("Hello, Guest")

greet("Amit")

print("Hello,", name) is indented to indicate it belongs to the if block.

print("Hello, Guest") is indented to indicate it belongs to the else block.

Example

def square(n):

result = n \* n

return result

print(square(5))

The If statement and its’ related statement

**if statement** is used to execute a block of code only **if a specific condition is true**. It's a fundamental part of decision-making in programming.

if – checks the condition.

elif – checks another condition if the previous if or elif was false.

else – executes if none of the if or elif conditions are true.

Example

age = 18

if age < 13:

print("You are a child.")

elif age < 18:

print("You are a teenager.")

else:

print("You are an adult.")

Example

number = 7

if number % 2 == 0:

print("Even number")

else:

print("Odd number")

The while loop

The while loop in Python is used to repeatedly execute a block of code as long as a given condition is **True**.

Example

i = 1

while i <= 5:

print(i)

i += 1

Example Using break to exit early

i = 1

while True:

print(i)

if i == 3:

break

i += 1

Example Using continue to skip an iteration

i = 0

while i < 5:

i += 1

if i == 3:

continue

print(i)

The for loop

Example 1: Looping through a list

fruits = ["apple", "banana", "cherry"]

for fruit in fruits:

print(fruit)

Example 2: Using range() function

for i in range(5):

print(i)

Example 3: Looping through a string

for letter in "hello":

print(letter)

Example 4: Using for loop with if condition

numbers = [1, 2, 3, 4, 5]

for num in numbers:

if num % 2 == 0:

print(f"{num} is even")

**The range statement**

The range() statement in Python is used to generate a sequence of numbers. It is commonly used in for loops to iterate(repeating again and again) over a block of code a specific number of times.

range(start, stop, step)

* start: The starting number (optional, default is 0)
* stop: The ending number (exclusive)
* step: The increment (optional, default is 1)

**Examples:**

**1. Using range(stop)**

for i in range(5):

print(i)

**2. Using range(start, stop)**

for i in range(2, 6):

print(i)

**3. Using range(start, stop, step)**

for i in range(1, 10, 2):

print(i)

**4. Using negative step (counting backwards)**

for i in range(10, 0, -2):

print(i)

**Break & Continue**

The **Break** statement is used to exit the loop prematurely, even if the loop condition is still true.

Example

for number in range(1, 10):

if number == 5:

print("Breaking the loop at number", number)

break

print("Number:", number)

The **continue** statement is used to skip the current iteration and move to the next one.

Example

for number in range(1, 6):

if number == 3:

print("Skipping number", number)

continue

print("Number:", number)