**Lesson 4: Control Flow**

**🔹 Introduction**

Control flow allows your program to make decisions and repeat actions — just like how we make choices in real life. In this lesson, we’ll cover the basics of:

* Conditional statements: if, elif, else
* Loops: for and while
* Control statements: break, continue, and pass

By the end of this lesson, you’ll be able to control the direction of your Python programs with logic and repetition.

**🔹 1. Conditional Statements (if, elif, else)**

* These statements let your code respond to different situations.
* Example

age = 18

if age >= 18:

print("You are an adult.")

elif age > 12:

print("You are a teenager.")

else:

print("You are a child.")

* ✅ Use case: Checking conditions like age, scores, choices, etc.

**🔹 2. for Loops**

* A for loop is used to iterate over a sequence like a list, string, or range.
* Example

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

for fruit in fruits:

print(f"I love {fruit}")

* With range():

for i in range(1, 6):

print(i) # Prints numbers from 1 to 5

**🔹 3. while Loops**

* A while loop runs as long as a condition is true.
* Example

counter = 0

while counter < 3:

print(f"Count: {counter}")

counter += 1

* ⚠️ Be careful — infinite loops happen if the condition never becomes False.

**🔹 4. Control Statements**

* break – Exit a loop early:

for i in range(10):

if i == 5:

break

print(i) # Stops at 4

* continue – Skip to the next loop iteration:

for i in range(5):

if i == 2:

continue

print(i) # Skips 2

* pass – Do nothing (used as a placeholder):

if True:

pass # Still valid Python code

**🔹 Mini Challenge**

* Write a program that:

1. Loops from 1 to 20.
2. Prints "Fizz" if the number is divisible by 3.
3. Prints "Buzz" if divisible by 5.
4. Prints "FizzBuzz" if divisible by both.

**🔹 Outro**

Awesome work! 🚀 In this lesson, you learned how to control the logic of your program using:

* if, elif, else for decisions
* for and while for repetition
* break, continue, and pass for extra control

These tools help you write smart, interactive programs.

👉 Next up: Functions in Python — how to create reusable blocks of code for cleaner and better programs.