Python Programming

Unit 01 – Lecture 05: Decision Making and Looping Structures

Tofik Ali

School of Computer Science, UPES Dehradun

February 14, 2026

Repository: `https://github.com/tali7c/Python-Programming`

Quick Links

Core Concepts Demo Interactive Summary

Agenda

1 Core Concepts

2 Demo

3 Interactive

4 Summary

Learning Outcomes

- Write decisions using `if/elif/else` and nested conditions

Learning Outcomes

- Write decisions using `if/elif/else` and nested conditions
- Use `match-case` for multi-way branching (Python 3.10+)

Learning Outcomes

- Write decisions using `if/elif/else` and nested conditions
- Use `match-case` for multi-way branching (Python 3.10+)
- Write `for` and `while` loops using `range()`

Learning Outcomes

- Write decisions using `if/elif/else` and nested conditions
- Use `match-case` for multi-way branching (Python 3.10+)
- Write `for` and `while` loops using `range()`
- Control loops using `break`, `continue`, `pass`, and loop `else`

Decision Making: if/elif/else

```python
marks = int(input("Marks: "))
if marks >= 90:
    print("A+")
elif marks >= 75:
    print("A")
else:
    print("Needs improvement")
```

Nested if (Example)

```
age = int(input("Age: "))
if age >= 18:
    if age >= 60:
        print("Senior")
    else:
        print("Adult")
else:
    print("Minor")
```

match-case (Python 3.10+)

- Useful for clean multi-way choices

```python
day = int(input("Enter day number (1-7): "))
match day:
    case 1:
        print("Monday")
    case 2:
        print("Tuesday")
    case _:
        print("Other day")
```

`match-case` (Python 3.10+)

- Useful for clean multi-way choices
- Alternative to long `elif` ladders

```
day = int(input("Enter day number (1-7): "))
match day:
    case 1:
        print("Monday")
    case 2:
        print("Tuesday")
    case _:
        print("Other day")
```

Loops: for with range

- range(n) generates 0..n-1

```
for i in range(1, 6):
    print(i)
```

Loops: `for` with `range`

- `range(n)` generates 0..n-1
- `range(start, stop, step)` is flexible

```
for i in range(1, 6):
    print(i)
```

Loops: while

- Use when you do not know the exact number of iterations

```
n = 5
while n > 0:
    print(n)
    n -= 1
```

Loops: `while`

- Use when you do not know the exact number of iterations
- Always ensure the loop condition eventually becomes false

```python
n = 5
while n > 0:
    print(n)
    n -= 1
```

Loop Control

- `break`: stop the loop

Loop Control

- `break`: stop the loop
- `continue`: skip to next iteration

Loop Control

- `break`: stop the loop
- `continue`: skip to next iteration
- `pass`: do nothing (placeholder)

Loop Control

- `break`: stop the loop
- `continue`: skip to next iteration
- `pass`: do nothing (placeholder)
- `else` in loops: runs only if loop ends normally (no `break`)

else with a for Loop

```
for i in range(1, 6):
    if i == 3:
        break
else:
    print("Loop finished without break")
```

- Here `else` does not run because we used `break`.

Demo: Menu-Driven Program

- File: `demo/menu_driven_number_analyzer.py`

Demo: Menu-Driven Program

- File: `demo/menu_driven_number_analyzer.py`
- Uses a `while True` loop + `match-case`

Demo: Menu-Driven Program

- File: `demo/menu_driven_number_analyzer.py`
- Uses a `while True` loop + `match-case`
- Demonstrates clean branching + repetition until user exits

Checkpoint 1

Question: When does the `else` part of a loop execute?

Checkpoint 2

Question: What is the difference between `break` and `continue`?

Think-Pair-Share

Choose the better loop and justify:

- Printing 1..100
- Repeating until the user enters “exit”

Key Takeaways

- Use `if/elif/else` for decisions and `match-case` for clean multi-way choices

Key Takeaways

- Use `if/elif/else` for decisions and `match-case` for clean multi-way choices
- Use `for` when you have a count/range; use `while` for condition-based repetition

Key Takeaways

- Use `if/elif/else` for decisions and `match-case` for clean multi-way choices
- Use `for` when you have a count/range; use `while` for condition-based repetition
- `break` stops a loop; `continue` skips to next iteration

Key Takeaways

- Use `if/elif/else` for decisions and `match-case` for clean multi-way choices
- Use `for` when you have a count/range; use `while` for condition-based repetition
- `break` stops a loop; `continue` skips to next iteration
- Loop `else` runs only when no `break` happens

Exit Question

Write a loop that prints numbers 1..10 but skips multiples of 3.