



SDEV 1001

Programming Fundamentals

Making Decisions in Programming - 3

A LEADING POLYTECHNIC COMMITTED TO YOUR SUCCESS

Expectations - What I expect from you

- No Late Assignments
- No Cheating
- Be a good classmate
- Don't waste your time
- Show up to class

Agenda

On the right is what we will cover today.

Introduction to Match Case in Python

Basic Match Case Syntax

Example: Day of the Week

Why Use Match Case?

Summary

Introduction to Match Case in Python

- `match` is a new statement introduced in Python 3.10.
- It allows you to match a value against a series of patterns, similar to switch/case in other languages.
- Useful for replacing long chains of `if / elif` statements.

Basic Match Case Syntax

```
value = "apple"

match value:
    case "apple":
        print("It's an apple!")
    case "banana":
        print("It's a banana!")
    case _:
        print("Unknown fruit")
```

- The `_` case is a wildcard (default) if no other case matches.

Example: Day of the Week

```
day = input("Enter a day of the week: ").lower()

match day:
    case "monday":
        print("Start of the work week!")
    case "friday":
        print("Almost the weekend!")
    case "saturday" | "sunday":
        print("It's the weekend!")
    case _:
        print("Just another day.")
```

- You can match multiple values in a single case using `|`.

Why Use Match Case?

- Cleaner and more readable than multiple `if / elif` statements.
- Easier to maintain when checking many possible values.
- Supports pattern matching for more complex data (beyond simple values).
- Note: in this course we're not going to use this much but it's good to know about it, and this is very common in other languages.

Summary

- Use `match` for clear, concise multi-way branching.
- Always include a wildcard (`case _:`) for unmatched cases.
- Available in Python 3.10 and above.



Example

Let's go run a few examples together

A LEADING POLYTECHNIC COMMITTED TO YOUR SUCCESS