Match Case

Match-Case

- The **match-case** statement is used to compare a value against multiple possible patterns and execute different code depending on which pattern matches.
- It's like a more advanced version of the **if-elif-else** construct.
- The **match-case** statement in Python was introduced in Python 3.10
- It is similar to **switch-case** statements in other languages like **C++** and **Java**.
- However, it's more powerful and flexible as it allows more than just matching fixed values—it can match patterns like **sequences, structures, and specific conditions.**

Match-Case - syntax

Python

```
match variable:
    case pattern1:
        # Code to execute if pattern1 matches
    case pattern2:
        # Code to execute if pattern2 matches
    case _:
        # Default case (if no patterns match)
```

Match-Case – Matching Features

- **Literal Matching**: Match specific values.
- **Variable Binding**: Bind matched values to variables for later use.
- **Sequence Matching**: Match against sequences like lists or tuples.
- **Structured Matching**: Match against structured data like objects. [ADVANCED]

Match-Case – Literal Matching Example

Python

```python
status = 200
match status:
    case 200:
        print("OK")
    case 404:
        print("Not Found")
    case 500:
        print("Internal Server Error")
    case _:
        print("Unknown Status")
# Output: OK
```

Match-Case – Variable Matching Example

Python

```python
command = ("add", 3, 4)

match command:
    case ("add", x, y):
        result = x + y
    case ("subtract", x, y):
        result = x - y
    case _:
        result = "Invalid Command"

print(result)  # Output: 7
```

Match-Case – Sequence Matching Example

Python

```python
data = [1, 2]  # or [1, 2, 3]

match data:
    case [x, y]:
        result = f"Two items: {x}, {y}"
    case [x, y, z]:
        result = f"Three items: {x}, {y}, {z}"
    case _:
        result = "Unknown format"

print(result)
```

Knowledge Reinforcement

1

Question

What is the correct way to create a default case in a **match** statement?

Answer

A) case *:

B) case None:

C) case _:

D) case default:

1

Question

What is the correct way to create a default case in a **match** statement?

Answer

A) case *:

B) case None:

C) case _:

D) case default:

2

Question

What will be the output of the following code?

Python

```
status = 404
match status:
    case 200:
        print("Success")
    case 404:
        print("Not Found")
    case _:
        print("Unknown Status")
```

Answer

A) Success

B) Not Found

C) Unknown Status

D) Error

2

Question

What will be the output of the following code?

Python

```
status = 404
match status:
    case 200:
        print("Success")
    case 404:
        print("Not Found")
    case _:
        print("Unknown Status")
```

Answer

A) Success

B) Not Found

C) Unknown Status

D) Error

3

Question

You can use **match-case** to destructure lists and tuples.

Answer

A) True

B) False

3

Question

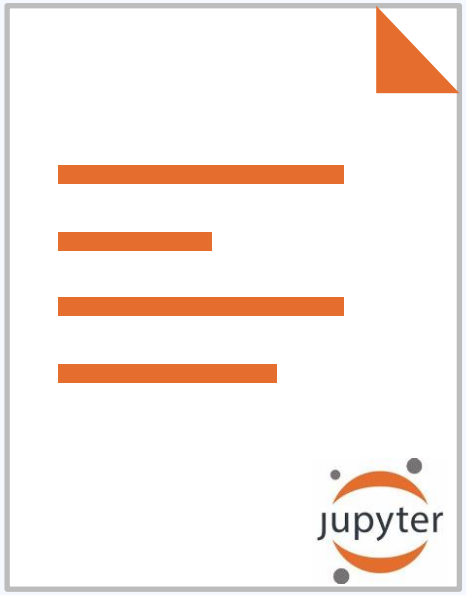
You can use **match-case** to destructure lists and tuples.

Answer

A) True

B) False

Practice Set

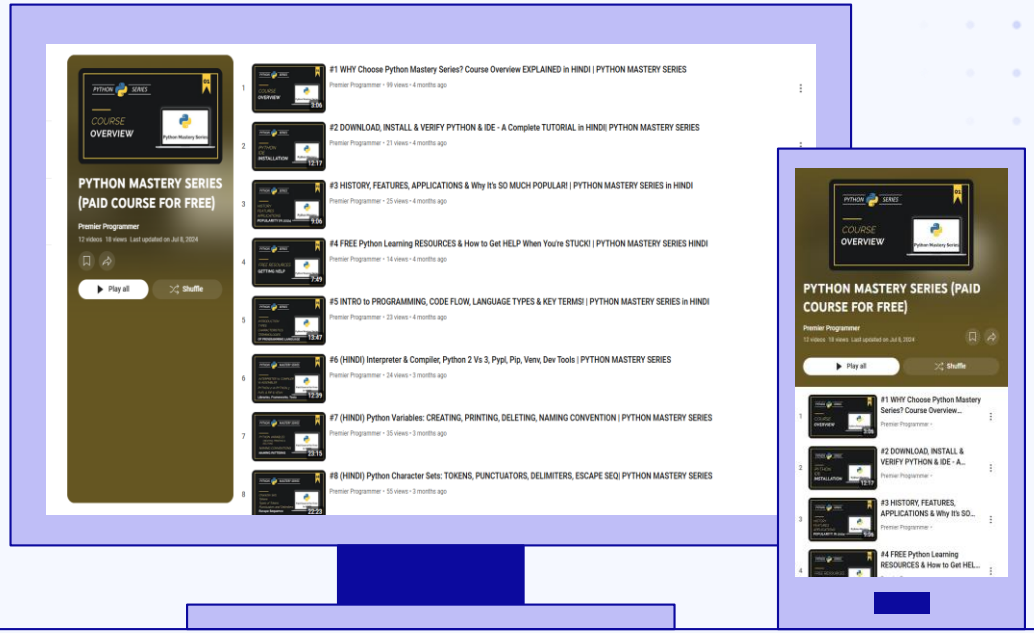


Match - case based Questions

Download Link in **Description**

WATCH

Level up your coding with each episode in this focused Python series.



Next Video!

Practice Set - 2 Solution