**Case Statements in Python**

Python does not have a built-in `case` or `switch` statement like some other programming languages (e.g., C, Java, or JavaScript). However, starting with \*\*Python 3.10\*\*, a new feature called \*\*structural pattern matching\*\* was introduced, which provides similar functionality using the `match` and `case` keywords.  
  
Here’s how you can use `match` and `case` in Python 3.10+:  
  
### Syntax of `match` and `case`  
```python  
def http\_status\_code\_handler(status\_code):  
 match status\_code:  
 case 200:  
 return "OK"  
 case 404:  
 return "Not Found"  
 case 500:  
 return "Internal Server Error"  
 case \_:  
 return "Unknown Status Code"  
```  
  
### Explanation:  
- `match` is used to start the pattern matching block.  
- `case` is used to define each condition.  
- `\_` is the wildcard pattern, which matches anything (similar to a default case in other languages).  
  
---  
  
### Example: Using `match` with Patterns  
You can also use more complex patterns with `match` and `case`. For example:  
  
```python  
def process\_command(command):  
 match command:  
 case ("move", x, y):  
 return f"Moving to coordinates ({x}, {y})"  
 case ("say", message):  
 return f"Saying: {message}"  
 case ("quit",):  
 return "Quitting the program"  
 case \_:  
 return "Unknown command"  
```  
  
Here:  
- Tuples are matched against specific patterns.  
- You can extract values (e.g., `x`, `y`, `message`) directly from the pattern.  
  
---  
  
### Alternative for Older Python Versions  
If you're using a version of Python \*\*before 3.10\*\*, you can simulate a `case` statement using a dictionary or `if-elif-else` statements.  
  
#### Using a Dictionary:  
```python  
def http\_status\_code\_handler(status\_code):  
 responses = {  
 200: "OK",  
 404: "Not Found",  
 500: "Internal Server Error"  
 }  
 return responses.get(status\_code, "Unknown Status Code")  
```  
  
#### Using `if-elif-else`:  
```python  
def http\_status\_code\_handler(status\_code):  
 if status\_code == 200:  
 return "OK"  
 elif status\_code == 404:  
 return "Not Found"  
 elif status\_code == 500:  
 return "Internal Server Error"  
 else:  
 return "Unknown Status Code"  
```  
  
---  
  
### When to Use `match` and `case`  
- Use `match` and `case` when working with Python 3.10+ for cleaner and more readable code, especially when dealing with complex patterns.  
- For older versions of Python, stick to dictionaries or `if-elif-else` chains.