Welcome to the Python Conditional Statements Lecture

The session will start shortly...

CoGrammar

Johannesburg Team Housekeeping

- Please be mindful and respectful to everyone in this supportive learning environment. Mutual respect and tolerance are fundamental values we uphold.
- There are no bad or silly questions—feel free to ask anything! You can ask Sashlin or myself questions at any time, regardless of the situation. Even if you find yourself in a dire situation—like stuck in quicksand—you're still welcome to ask us a question (though we recommend calling or shouting first!).
- A few additional reminders for onsite behavior:
  - Keep shared spaces tidy—clean up after yourself in the break areas.
  - Please mute your devices during sessions to minimize distractions.
  - Avoid making personal phone calls in common areas—use designated quiet zones if you need to step away.
- Additionally, please remember to put any dishes in the sink before 2 p.m., as Lizbeth will have already finished for the day. If you’re feeling unwell, kindly inform Ingrid or myself via email.

Learning Objectives

- Understand the purpose of conditional statements in Python.
- Write Python programs using if-elif-else statements.
- Debug programs that utilize conditional logic.

What are Conditional Statements ?

- Conditional statements control the flow of a program.
- They help the computer make decisions based on different conditions
- These statements allow a program to choose different actions based on the situation.
- Python provides if, elif, and else to handle different conditions.

Conditional Statements Cont.

- **if:** Checks a condition (e.g., "Is it raining?"). If it's true, it does something.
- **elif:** Checks another condition if the first one wasn’t true (e.g., "Is it sunny?").
- **else:** If none of the conditions are true, it does something else (e.g., "I’ll go out without an umbrella or sunglasses").

Syntax Breakdown

```
if condition:
    # code to run if condition is True
elif another_condition:
    # code to run if another_condition is True
else:
    # code to run if all conditions are False
```

Simple Condition Example

```python
number = int(input("Enter a number: "))

if number > 0:
    print("The number is positive.")
elif number == 0:
    print("The number is zero.")
else:
    print("The number is negative.")
```

Common Conditional Mistakes

- Debugging conditional logic involves ensuring that conditions are checked in the correct sequence and that indentation is properly followed in Python, as blocks depend on it.
  - Misplaced indentation.
  - Incorrect condition order.
  - Forgetting to test all cases.

Key Things to Note

- Indentation is Crucial:
  - Python uses indentation (spaces or tabs) to define the blocks of code that belong to an if, elif, or else statement.
- Conditions Must Evaluate to **True** or **False**:
  - The conditions in if and elif statements need to be expressions that return True or False.
- Order of Conditions Matters:
  - The if and elif conditions are evaluated in order from top to bottom. As soon as a True condition is found, the corresponding block runs, and the rest of the conditions are ignored.

Key Things to Note

- **else** is Optional but Handy:
  - The else block is not required, but it's useful for catching all other cases when none of the previous conditions are met.
- Use **elif** for Multiple Conditions:
  - If you need to check more than one condition, use elif instead of multiple if statements. This prevents unnecessary checks once a condition is found to be True, making your program more efficient.

Q & A SECTION

Please use this time to ask any questions relating to the topic, should you have any.

Recap of Conditional Statements

- Conditional statements control the flow of a program.
- if, elif, and else allow us to check conditions.
- Careful use of conditions helps create accurate and effective programs.

Thank you for attending