Wexdi SoftSkills

Conditional Statements and Loops

If-else conditions

- **if Statement**
  - **if statement** is the most simple decision-making statement. If the condition evaluates to True, the block of code inside the if statement is executed.

```
i = 10


 # Checking if i is greater than 15
if (i > 15):
    print("10 is less than 15")


print("I am Not in if")
```

If-else conditions

Wexdi SoftSkills

- **if....else Statement**
  - **if...else statement** is a control statement that helps in decision-making based on specific conditions. When the if condition is **False**. If the condition in the if statement is not true, the else block will be executed.

```
i = 20

 # Checking if i is greater than 0
if (i > 0):
   print("i is positive")
else:
   print("i is 0 or Negative")
```

If-else conditions

- **Nested If Else Statement**
  - **Nested if...else statement** occurs when **if...else** structure is placed inside another **if** or **else** block. Nested If..else allows the execution of specific code blocks based on a series of conditional checks.

```
i = 10
if (i == 10):

    #  First if statement
    if (i < 15):
        print("i is smaller than 15")

    # Nested - if statement
    # Will only be executed if statement above
    # it is true
    if (i < 12):
        print("i is smaller than 12 too")
    else:
        print("i is greater than 15")

else:
  print("i is not equal to 10")
```

For and While loops with real-world

Wexdi SoftSkills

- **Why Use Loops in Python?**
  - **Loops are vital for:**
  - Processing data sets.
  - Simplify a complex process: self-performance and avoiding doing repetitive actions. Coordinate multiple steps in an operation.
  - The use of procedures that call for more than a single loop.

For and While loops with real-world

Wexdi SoftSkills

- **Types of Loops in Python**
  - **For Loop**
    - A for loop goes through a sequence, which may be a list, tuple, dictionary, or range of values. It is especially useful for situations when the number of iterations is known beforehand. For instance, in cases of performing operations for every record in a database or names in a list.
    - **Applications:**
      - Business analytics data collection processing, such as in banking and selling.
      - This is because in order to iterate through the contents of the files for analysis, the files must be opened in the current session.
      - Automating report generation.

For and While loops with real-world

Wexdi SoftSkills

- **Types of Loops in Python**
  - **While Loop**
    - The while loop executes while a certain condition exists and retains its true value. It is most effective when used with data sets whose number of iterations cannot be predetermined, for instance, waiting for a user's response or analyzing a connection to a network.
    - **Applications:**
      - Real-time monitoring systems.
      - Automating tasks adopting the time-based task model.
      - Calculation procedures are evaluated employing dynamic algorithms.

For and While loops with real-world

Wexdi SoftSkills

- **The primary elements of a loop exist: The for-loop’s main components**
- **1. Initialization**
  - Operations may start with conditioning parameters for the loop or defining the range of a specific loop. For example, setting a counter in a while loop.
- **2. Condition**
  - Every loop is dependent on a condition. With while loops, it is a logical expression, and with for loops, it is the sequence through which to iterate.

For and While loops with real-world

Wexdi SoftSkills

- **The primary elements of a loop exist: The for-loop’s main components**
- **3. Iteration**
  - Every loop proceeds through the block of code and, if necessary, changes a variable or gets the next value in its list.
- **4. Termination**
  - The termination of a loop is based on the condition used in the named structure; as long as this condition is valid, the loop continues in its execution. One area of special attention is terminating techniques to avoid having endless recursion and iteration loops.

For and While loops with real-world

Wexdi SoftSkills

- **4. Real-World Scenarios**
  - Loops are used in different areas of study. Here's how they excel in real-life situations:
- **4.1 Data Processing**
  - Loops are one of the most popular Python's tools to manipulate data. For example:
  - Scrolling through numerous fields within every row.
  - Sweeping activities such as filling the missing values can also be automated.
  - Summarizing statistics.
  - There are other libraries available for the looping efficiency of the data, such as Pandas and NumPy for Python.

For and While loops with real-world

Wexdi SoftSkills

- **4.2 Automation Tasks**
  - Loops automate mundane and repetitive processes.
  - A mode of emailing that involves communicating with postings of significant quantity to numerous individuals at a given time.
  - Data scraping using web scraping tools such as BeautifulSoup.
  - Restructurating the cabinet’s shelves or drawers and renaming their folders.
  - Check out this simple browser project for an idea on how to go about it.

For and While loops with real-world

- **4.3 Game Development**
  - Loops guarantee the game's rendering and updating (mostly at the screen-refresh rate). These involve:
  - Processing of events (keyboard and mouse).
  - Managing character movement.
  - Changing screen graphic displays after some time.

For and While loops with real-world

- **4.4 Machine Learning**
  - In AI and ML, loops iterate through datasets to:
  - Train models in batches.
  - Verify the measure variables for the purposes of assessing the performance.
  - Fine-tune hyperparameters.

For and While loops with real-world

Wexdi SoftSkills

- **5. Best Practices**
  - **Avoid Infinite Loops:** It is very important that loop conditions are clearıy stated.
  - **Use Enumerate for Indexing:** A neater way of referring to indices in sequences.
  - **Leverage Comprehensions:** Eliminate all the uses of loops wherever they are present and replace them with list or dictionary comprehension appropriately.