**Nested if-else conditions**

In Python, a nested if-else condition involves placing one or more if, elif, or else statements inside another if, elif, or else block. This allows for hierarchical decision-making, where a condition is evaluated, and if it's true, a further set of conditions are then evaluated within that specific block.

The key aspect of nested if-else statements in Python is indentation, which defines the scope and relationship between the outer and inner conditional blocks.

Syntax Example:

if outer\_condition:  
 *# Code executed if outer\_condition is True*  
 if inner\_condition\_1:  
 *# Code executed if outer\_condition and inner\_condition\_1 are True*  
 pass  
 elif inner\_condition\_2:  
 *# Code executed if outer\_condition is True and inner\_condition\_2 is True*  
 pass  
 else:  
 *# Code executed if outer\_condition is True and neither inner\_condition\_1 nor inner\_condition\_2 are True*  
 pass  
else:  
 *# Code executed if outer\_condition is False*  
 pass

Explanation:

* Outer if statement:

This is the primary condition. If outer\_condition evaluates to True, the code block indented beneath it is executed.

* Inner if, elif, else statements:

These are placed within the indented block of the outer if or else. They are only evaluated if the condition of their parent block is met.

* **Indentation:**

Each level of nesting requires an additional level of indentation (typically 4 spaces). This visual structure clearly indicates which else belongs to which if and which conditions are dependent on others.

Example:

score = 85  
attendance = 92  
  
if score >= 60:  
 print("Student passed the exam.")  
 if attendance >= 90:  
 print("Student also has excellent attendance.")  
 else:  
 print("Student's attendance is below excellent.")  
else:  
 print("Student did not pass the exam.")