Introduction to If-Else

There are certain points in our code when we need to make some decisions and then based on the outcome of those decisions we execute the next block of code. Such conditional statements in programming languages control the flow of program execution.

Most commonly used conditional statements in Python are

- Simple If statements
- If-Else statements
- If-Elif statements
- Nested Conditionals

Simple If statements

These are the most simple decision-making/conditional statements. It is used to decide whether a certain statement or block of statements will be executed or not.

- The most important part of any conditional statement is a condition or a boolean.
- And the second important thing is the code block to be executed.

In the case of simple `If` statements, if the conditional/boolean is true then the given code block is executed, else the code block is simply skipped and the flow of operation comes out of this If condition.

The **general syntax** of such statements in Python is:

```
if <Boolean/Condition>:
      <Code Block to be executed in case the Boolean is True>
<Code Block to be executed in case the Boolean is False>
```

An **example** of a simple `if` statement can be as follows:

```
Val = False
if Val == True:
      print("Value is True") # Statement 1
print("Value is False") # Statement 2
```

In the above code, the variable Val has a boolean value **False**, and hence the condition is not satisfied. Since the condition is not satisfied, it skips the `If` statement, and instead, the next statement is executed. Thus the output of the above code is:

```
Value is False
```

Importance of Indentation in Python

To indicate a block of code and separate it from other blocks in Python, you must indent each line of the block by the same amount. The two statements in our example of Simple if-statements are both indented four spaces, which is a typical amount of indentation used in the case of Python.

In most other programming languages, indentation is used only to improve the readability of the code. But in Python, it is required for indicating what block of code, a statement belongs to. **For instance**, *Statement 1* which is indented by 4 spaces is a part of the `if` statement block. On the other hand, *Statement 2* is not indented, and hence it is not a part of the `if` block. This way, indentation indicates which statements from the code belong together.

Any deviation from the ideal level of indentation for any statement would produce an indentation error. **For example:** On running the given script:

```
Val = False
if Val == True:
        print("Value is True") # Statement 1
    print("Value is False") # Statement 2
```

We get the output as:

```
IndentationError: unindent does not match any outer indentation level
```

This error is because *statement 1* is not in the indentation line for the `if` statement.

Else-If statements

The simple `if` statement, tells us that if a condition is true it will execute a block of statements, and if the condition is false it won’t. But what if we want some other block of code to be executed if the condition is false. Here comes the *else* statement. We can use the `else` statement with the `if` statement to execute a block of code when the condition is false. The general Syntax for the If-Else statement is:

```
if (Condition/Boolean):
    <Code block to be executed in case the condition is True>
else:
    <Code block to be executed in case the condition is False>
```

*** Keep in mind the indentation levels for various code blocks.