Python
Conditional Statements

Conditional Statements

Conditional statements helps in decision making process,

Example,

"Are you hungry? If so, then eat" is a conditional statement

Phrased as: - ***If you're hungry, then eat***

In Python, there are three forms of the if...else statement.

1. if statement
2. if...else statement
3. if...elif...else statement

Python if Statement

The if statement evaluates condition.

- If condition is evaluated to True, the code inside the body of if is executed.
- If condition is evaluated to False, the code inside the body of if is skipped.

Syntax

```
if condition:
    # body of statement
```

Example

```
hungry = True
if hungry:
    print("Eat Food")
```

Python if...else Statement

If the condition evaluates to True,

- The code inside if is executed
- The code inside else is skipped

Syntax

```
if condition:
        # Block of code if condition is True
else:
        # Block of code if condition is False
```

Example

```
num = 2
if num >= 20:
        print("Number Greater or Equal to 20")
else:
        print("The Number is less than 20")
```

Python if...elif...else Statement

The **if...else** statement is used to execute a block of code among two alternatives.

However, if we need to make a choice between more than two alternatives, then we use the **if...elif...else** statement

Syntax

```
if condition1:
        # Code block 1
elif condition2:
        # Code block 2
else:
        # Code block 3
```

If condition1 evaluates to true, code block 1 is executed.

If condition1 evaluates to false, then condition2 is evaluated.

If condition2 is true, code block 2 is executed.

If condition2 is false, code block 3 is executed.

Example:

Write an **if/elif/else** statement for a college with a grading system as shown below:

- If grade is 90 or higher, print "A"
- Else if grade is 80 or higher, print "B"
- Else if grade is 70 or higher, print "C"
- Else if grade is 60 or higher, print "D"
- Else, print "F"

```
marks = 92

if marks >= 90:
        print("A")
elif marks >= 80:
        print("B")
else:
        print("C")
```

Python Loops

In computer programming, loops are used to repeat a block of code. We perform a process of *iteration* (repeating tasks).

Two types of loops in Python:

- for loop
- while loop

For Loop

In a **for** loop, we will know in advance how many times the loop will need to iterate because we will be working on a collection with a predefined length.

With for loops, on each iteration, we will be able to perform an action on each element of the collection.

Example

```
names = ["Habibi", "Lucy", "George", "Wendy"]

for name in names:
    print(name)
```

In above example:

- name is the **<temporary variable>**.
- **names** is our **<collection>.**
- print(name) was the **<action>** performed on every iteration using the temporary variable of name.

For Loops: Using Range

A range is a series of values between two numeric intervals.

We use Python's built-in function range() to define a range of values.

Example,

```
five_steps = range(5)
    # five_steps is now a collection with 5 elements:
    # 0, 1, 2, 3, 4
```

Example

```
welcome_message = "Welcome To PLP"

for i in range(5):
    print(welcome_message)
```

While Loop