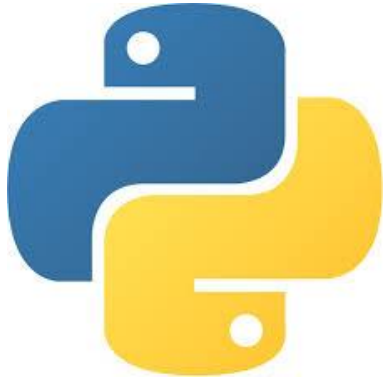
Python -
Conditions and Branching



Shahriar's Intelligence Academy

Condition

Definition:

A condition is a statement or an expression that **evaluates to either True or False** in Python.

Characteristics:

- **Boolean Nature:** Conditions return either True or False.
- **Control Flow:** They decide the execution of code blocks.
- **Operators:** Conditions often use comparison (==, !=, >, <) and logical operators (and, or, not).
- **Used in Statements:** Conditions are mainly used in control flow statements like if, if-else, and loops.

Example:

```
x = 10
if x > 5:
    print("x is greater than 5")
# Output: x is greater than 5
```

Comparison Operators

Comparison Operators:

1. `==` : Equal to
   - Example: `5 == 5` → `True`
2. `!=` : Not equal to
   - Example: `5 != 3` → `True`
3. `>` : Greater than
   - Example: `7 > 4` → `True`
4. `<` : Less than
   - Example: `3 < 8` → `True`
5. `>=` : Greater than or equal to
   - Example: `6 >= 6` → `True`
6. `<=` : Less than or equal to
   - Example: `2 <= 5` → `True`

Example:

```
a = 10
b = 20
print(a == b)   # False
print(a < b)    # True
```

Branching

Definition:

Branching is the process of **making decisions to execute specific code paths** based on conditions.

Characteristics:

- **Decision Making:** Executes different blocks of code based on the conditions.
- **Control Statements:** Branching uses statements like if, if-else, and if-elif-else.
- **Dynamic Execution:** Enables programs to respond dynamically to different inputs or states.

Example:

```
age = 18
if age >= 18:
    print("Eligible to vote")
else:
    print("Not eligible to vote")
# Output: Eligible to vote
```

If Statement

Definition:

The if statement is used to **test a condition**. If the condition is True, the code block inside the if statement is executed.

Syntax:

```
if condition:
    # Code to execute if condition is True
```

Example:

```
number = 7
if number % 2 == 1:
    print("The number is odd")
# Output: The number is odd
```

If-else Statement

Definition:

The if-else statement **provides two paths of execution**. If the condition is True, one block is executed; otherwise, the other block is executed.

Syntax:

```
if condition:
    # Code to execute if condition is True
else:
    # Code to execute if condition is False
```

Example:

```
marks = 70
if marks >= 50:
    print("Pass")
else:
    print("Fail")
# Output: Pass
```

If-elif-else Statement

Definition:

The if-elif-else statement allows checking **multiple conditions sequentially**. The first condition that evaluates to True will execute its block of code.

Syntax:

```
if condition1:
    # Code for condition1
elif condition2:
    # Code for condition2
else:
    # Code if no condition is True
```

Example:

```
score = 85
if score >= 90:
    print("Grade A")
elif score >= 75:
    print("Grade B")
else:
    print("Grade C")
# Output: Grade B
```

If – else Statement + or

Definition:

The or operator **combines two conditions** and **evaluates to True if at least one condition is True**.

Syntax:

```
if condition1 or condition2:
    # Code to execute if either condition is True
else:
    # Code to execute if both conditions are False
```

Example:

```
temperature = 35
if temperature < 0 or temperature > 30:
    print("Extreme Weather")
else:
    print("Normal Weather")
# Output: Extreme Weather
```

If – else Statement + and

Definition:

The and operator combines two conditions and **evaluates to True only if both conditions are True**.

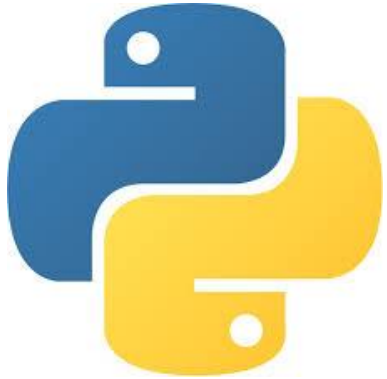
Syntax:

```
if condition1 and condition2:
    # Code to execute if both conditions are True
else:
    # Code to execute if any condition is False
```

Example:

```
age = 25
income = 50000
if age > 18 and income > 30000:
    print("Eligible for loan")
else:
    print("Not eligible for loan")
# Output: Eligible for loan
```

Thank You!



Shahriar's Intelligence Academy