Programming Skills - Python

Learn the basics of Python Programming

Day - 2

Control Structures

Agenda

2.1. Brief Recap

A brief recap of what we discussed yesterday.

2.2. The `if` statement

What it is, what it does, why it is used, and how to use it.

2.3. `else` and `elif`

What they are, how they differ from if and the difference between both.

2.4. Comparison Operators

Common comparison operators, syntax, and use cases.

Agenda

2.5. Logical Operators

`and`, `or`, and `not`.

2.6. Introduction to Loops

Definition, types, and use cases.

2.7. `for` and `while`

What they are, how they differ from if and the difference between both.

2.8. Breaking Out

Exiting a loop conditionally.

2.1. Brief Recap

2.2. The `if` statement

`if:`

What is it?

The if statement checks a condition and runs code if the condition is true.

Why is it used?

It is used to control the flow of the program conditionally and make decisions.

How to use it?

```
if condition:
    # execute code
```

Replit →

2.3. The `else` and `elif`

`else:`

Contains a block of code that is executed when the condition checked by if is false.

How to use it?

```
if condition:
    # executed when condition
    # is true

else:
    # executed when condition
    # is false
```

Replit →

`elif:`

Used when more than 1 condition need checking.

How to use it?

```
if condition_1:
    # executed when condition_1
    # is true

elif condition_2:
    # executed when condition_2
    # is true

else:
    # executed when both are
    # false
```

Replit →

2.4. Comparison Operators

Comparison Operators

Used to compare values in conditions.

Types of Comparison Operators

Greater Than	>
Less Than	<
Greater Than or Equal to	> =
Less Than or Equal to	< =
Equal to	= =
Not Equal to	! =

2.5. Logical Operators

Logical Operators

Used to combine multiple conditions.

Types of Comparison Operators

`and`

Checks for truth in both conditions.

`or`

Checks for truth in at least one condition.

`not`

Negates the condition.

How to use it?

```
if a > b and b > c:
    print (“A is greater”)
elif b > a and b > c:
    print (“B is greater”)
else:
    print (“C is greater”)
```

Replit →

2.6. Introduction to Loops

Introduction to Loops

Loops are used to run one particular part of a code multiple times.

There are 2 types of loops: **`for`** and **`while`**.

2.7. `for` and `while`

for

Loops over a sequence until it reaches the end of it.

```
for variable in sequence:
    print(variable)
```

while

Loops as long as a condition is true.

```
while condition:
    print(“condition is true”)
```

2.8. Breaking Out

Breaking out

To exit a loop before the loop condition is met, the **`break`** keyword is used.

```
for i in range(10):
    if i == 5:
        break
    print(i)
```

Skipping an Iteration

To skip an iteration a loop, the **`continue`** keyword is used.

```
for i in range(10):
    if i == 5:
        continue
    print(i)
```

Q&A

Thank You!