Control Flow in Python: A Beginner's Guide

Welcome to this session on **Control Flow in Python**! This guide will help you understand and apply control structures in your Python programs, focusing on **if-else statements** and **loops.**

What Are Control Structures?

Control structures allow programs to make decisions and execute different paths based on conditions. In Python, the main control structures are:

1. **Conditional Statements (if-else)**
2. **Loops (for, while)**

---

1. Conditional Statements (If-Else)

What is an If-Else Statement?

An **if-else statement** allows the program to execute certain code only if a specific condition is met. If the condition is not met, the program can execute a different set of instructions.

Syntax of If-Else in Python

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

Examples:

1. **Voting Eligibility Check**

```python
age = 18

if age >= 18:
    print("You are eligible to vote.")
else:
```

```
    print("You are not eligible to vote.")
```

2. **Temperature Check**

```python
temperature = 30

if temperature > 25:
    print("It's warm outside.")
else:
    print("It's cold outside.")
```

3. **Number Comparison**

```python
a = 10
b = 20

if a > b:
    print("a is greater than b")
else:
    print("a is not greater than b")
```

---

2. Loops

What is a Loop?

A **loop** allows you to repeat a block of code multiple times. There are two main types of loops in Python:

1. **For Loop**
2. **While Loop**

2.1 For Loop

A **for loop** is used when you know in advance how many times you want to repeat a block of code.

Syntax of For Loop:

```python
for item in sequence:
    # Code to execute for each item
```

Examples:

1. **Iterating Over a List**

```python
fruits = ["apple", "banana", "cherry"]

for fruit in fruits:
    print(fruit)
```

2. **Range Loop**

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

3. **String Iteration**

```python
word = "Python"

for letter in word:
    print(letter)
```

2.2 While Loop

A **while loop** is used when you want to repeat a block of code as long as a certain condition is true.

Syntax of While Loop:

```python
while condition:
    # Code to execute while the condition is True
```

Examples:

1. **Counting with While Loop**

```python
count = 1

while count <= 5:
    print(count)
    count += 1
```

2. **User Input Loop**

```python
user_input = ""

while user_input != "exit":
    user_input = input("Type 'exit' to stop: ")
```

3. **Decrementing Loop**

```python
n = 5

while n > 0:
    print(n)
    n -= 1
```

Using If-Else and Loops Together

You can combine **if-else statements** and **loops** to create more complex programs.

Example:

```python
numbers = [1, 2, 3, 4, 5]

for number in numbers:
    if number % 2 == 0:
        print(f"{number} is even")
    else:
        print(f"{number} is odd")
```

Activity: Create a Python Program Using If-Else and Loops

Task:

Write a Python program that asks the user to input a number. The program should then print whether the number is positive, negative, or zero. After that, the program should print all the numbers from 1 to the input number using a loop.

Solution:

```python
# Get user input
number = int(input("Enter a number: "))

# Check if the number is positive, negative, or zero
if number > 0:
    print("The number is positive.")
elif number < 0:
    print("The number is negative.")
else:
    print("The number is zero.")

# Print numbers from 1 to the input number
if number > 0:
    for i in range(1, number + 1):
        print(i)
```

Conclusion

In this session, we covered the basics of **control structures** in Python, focusing on **if-else statements** and **loops.** These tools allow you to control the flow of your program, making it more dynamic and responsive to different conditions.

- **If-else statements** help your program make decisions.
- **Loops** allow your program to repeat actions multiple times**.**

By mastering these concepts, you'll be well on your way to writing more complex and efficient Python programs**.**

Time Required to Read:

- **Introduction to Control Structures**: 2 minutes
- **If-Else Statements**: 3 minutes
- **For Loops**: 3 minutes
- **While Loops**: 3 minutes
- **Combining If-Else and Loops**: 2 minutes
- **Activity**: 5 minutes

Total: **18 minutes**

Happy coding! 😊