Answer these questions in chat:

1. Have you done the homework quiz?
2. Have you read the materials? Was it helpful?
3. Have you practiced coding?
4. Which challenges you have encountered?

Python Control Flow & Loops

Python sessions | 04 FEB 2025

Introduction to Control Flow

- Control flow determines the order in which code executes.
- Python executes code sequentially but can make decisions using conditions and loops.
- Two main aspects:
  - Conditional Statements (if, elif, else)
  - Loops (for, while)

Conditional Statements

Used to make decisions in Python

SYNTAX:

```
if condition:
    # Code block executed if condition is True
elif another_condition:
    # Code block executed if another_condition is True
else:
    # Code block executed if no conditions are True
```

EXAMPLE:

```
age = 18
if age >= 18:
    print("You are an adult.")
else:
    print("You are a minor.")
```

Task Example for Conditional Statements:

Write a Python program that asks the user for their age and prints whether they are a child (0-12), teenager (13-19), or adult (20+).

The 'for' Loop

Used to iterate over a sequence (list, tuple, string, range, etc.).

SYNTAX:

```
for variable in sequence:
    # Code block to execute
```

EXAMPLE:

```
for i in range(5):
    print("Iteration:", i)
```

Task Example:

Write a Python program that prints the squares of numbers from 1 to 10 using a for loop.

The 'while' Loop

Repeats a block of code while a condition is True.

SYNTAX:

```
while condition:
    # Code block to execute
```

EXAMPLE:

```
count = 0
while count < 5:
    print("Count is", count)
    count += 1
```

Task Example:

Write a Python program that keeps asking the user for input until they type "exit".

Loop Control Statements

Modify the behavior of loops:

- break: Stops the loop completely
- continue: Skips the current iteration and moves to the next.
- pass: Placeholder for future code.

EXAMPLE:

```
for num in range(10):
    if num == 5:
        break  # Stops loop at 5
    print(num)
```

Practice Time! Exercise 1

Print numbers from 1 to 20, but skip multiples of 3.

Hint: Use continue inside a loop.

Practice Time! Exercise 2

Task: Create a guessing game where the user has to guess a number between 1-10.

Hint: Use a while loop to keep asking until the correct number is guessed.

Practice Time! Exercise 3

Task: Write a program that prints a multiplication table for numbers 1 to 5 using nested loops.

Hint: Use a for loop inside another for loop.

Any Questions?

Python sessions | 04 FEB 2025