5.3.2 Python While Loop: Your Beginner-Friendly Step-by-Step Guide

- A while loop in python is a control flow statement that repeatedly executes a block of code until a specified condition is met.

Syntax:

```
while condition:
    # code block
```

Here, condition is a boolean expression that is evaluated before each iteration of the loop. If the condition is `True`, the code block is executed. The loop continues to execute as long as the condition remains True.

video: Python while loop example - Learn how to use while loop

Example #5.18: Print numbers from 1 to 10 using while loop

Question: Write a Python program to print the numbers from 1 to 10, using a while loop?

```
count = 1  # Start counting at 1
while count <= 10:  # Keep counting as long as we're less than or equal to 10
    print(count)  # Print the current number
    count += 1  # Add 1 to the count for the next round
```

Example #5.19: Print "Hello, world!" 5 times using while loop

Question: Write a Python program to print the string "Hello, world!" 5 times, using a while loop?

```
i = 1
while i <= 10:
    print('Hello, world!')
    i += 1
```

Task #5.12: Display Multiplication Tables**

Description:

Write a Python program that asks the user to input a number and displays its multiplication table up to 10 using a `while` loop.

Input:

- The program should prompt the user to enter an integer value.

Output:

- The program should display the multiplication table for the entered number in the following format:

```
n x 1 = n
n x 2 = 2n
...
n x 10 = 10n
```

Example:

Input:

```
Enter a number: 5
```

Output:

```
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

Requirements:

- Use a `while` loop to generate and display the multiplication table.

Task #5.13: Sum Integers from 1 to 100 Using While Loop

Write a Python program that calculates the sum of all integers from 1 to 100. The program should use a `while` loop to iterate through these integers and accumulate their sum.

Input:

- The program initializes a variable `i` to 1 and a variable `sum` to 0.
- The loop continues while `i` is less than or equal to 100.

Expected Output:

- The program will print the total sum of integers from 1 to 100 in the following format:
  `Sum = {sum}`

Example Output:

```
Sum = 5050
```

Explanation: This code uses a `while` loop to iterate through the integers from 1 to 100. In each iteration, the current value of `i` is added to the variable `sum`, and `i` is incremented by 1. After the loop completes, the total sum is printed.

Example #5.20: Sum of even numbers from 2 to 20 using while loop

Question: Write a Python program to calculate the sum of the even numbers from 2 to 20 using a while loop.

```
sum = 0  # Initialize a variable to store the sum
number = 1

while number <= 20:
    if number%2 == 0:
      sum += number  # Add the current number to the sum
    number += 1

print(f'The sum of even numbers from 1 to 20 is: {sum}')
```

Task #5.14: Calculate Squares of Numbers from 1 to 4 Using While Loop

Write a Python program that calculates and prints the square of the numbers from 1 to 4. The program should use a `while` loop to iterate through these numbers.

Input:

- The program initializes a variable `i` to 1 and continues looping while `i` is less than 5.

Expected Output:

- The program will print the square of each number from 1 to 4 in the following format:
  `Square of {i} is {square}`

Example Output:

```
Square of 1 is 1
Square of 2 is 4
Square of 3 is 9
Square of 4 is 16
```

Explanation: This code uses a `while` loop to calculate the square of the variable `i`, which starts at 1 and increments by 1 in each iteration until it reaches 5. The squares of the numbers 1 through 4 are printed during each iteration.

Example #5.21: Prompt User for Input Until Blank Line is Entered

Question: Write a Python program to prompt the user to enter lines of text until the user enters a blank line. The program should then display the message "You entered a blank line.".

```
inputStr = 'Start'

while inputStr != "":
  inputStr = input("Enter a line of text:")

print('You entered a blank line.')
```

Example #5.22: Sum User-Entered Numbers Until Zero is Entered

Question: Write a Python program to add all the numbers entered by the user until the user enters zero. The program should display the sum of the numbers.

```
sum = 0; # Initialize the sum

# Prompt the user to enter a number
number = int(input('Enter a number: ')) # int() Convert string input to integer

# While the number entered is not zero, add the number to the sum and prompt the
user to enter another number
while number != 0:
    sum += number
    number = int(input('Enter another number: ')) # int() Convert string input to
integer

# Display the sum of the numbers
print(f'The sum of the numbers is: {sum}')
```

Task #5.15: Sum User-Entered Numbers Until a Negative Number is Entered

Write a Python program that prompts the user to enter numbers. The program should keep accepting numbers until the user enters a negative number. Once a negative number is entered, the program should stop and display the sum of all the numbers entered (excluding the negative number).

Sample Input:

```
Enter a number (negative number to stop): 10
Enter another number (negative number to stop): 20
```

```
Enter another number (negative number to stop): 5
Enter another number (negative number to stop): -1
```

Sample Output:

```
The sum of all numbers is: 35
```

Task #14: Number Guessing Game

Instructions:

- Write a Python program where the computer picks a random number between 1 and 10, and the user has to guess it.
- The program should continue asking the user for a guess until they guess the correct number.
- After each incorrect guess, the program should give a hint whether the guess is too low or too high.

Example Code:

```
import random

# Step 1: Computer picks a random number between 1 and 10
secret_number = random.randint(1, 10)

# Step 2: Initialize the guess variable to None
guess = None

# Step 3: Use a while loop to keep asking the user for input

# write your code here
```

Sample Input and Output:

```
Guess a number between 1 and 10: 4
Too low! Try again.
Guess a number between 1 and 10: 9
Too high! Try again.
Guess a number between 1 and 10: 7
Too low! Try again.
Guess a number between 1 and 10: 8
Congratulations! You guessed the correct number.
```

This sample demonstrates:

- How the user is prompted repeatedly.

- Feedback on whether their guess is too low or too high.
- A congratulatory message when the correct guess is made.
-

Example #24: Infinite Loop Printing Messages with Counter

Video: Learn how to use INFINITE while loop)

```
x = 1
while True:
    print("To infinity and beyond! We're getting close, on %d now!" % (x))
    x += 1
```

while loop examples:

See also:

- Video: Learn how to use while loops
- Video: Python while loop

video: Python Loops Performance Comparison: For vs. While | Which is Faster?