

# The while Loop

In Python, the `while` loop is used to repeatedly execute a block of code as long as a specified condition remains true. The syntax for the `while` loop is as follows:

```
```python
while condition:
    # code to be executed while the condition is True
```
```

The loop will continue to execute the indented block of code as long as the `condition` remains true. If at any point the condition becomes false, the control exits the loop, and the program continues with the next statement after the `while` block. Here are some examples to illustrate the usage of the `while` loop:

### Example 1: Simple Counter

```
```python
# Print numbers from 1 to 5 using a while loop
counter = 1

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

Output:

```
```
1
2
3
4
5
```
```

### Example 2: User Input

```
```python
# Ask the user to enter a number until a valid input is provided
user_input = input("Enter a positive number: ")

while not user_input.isdigit() or int(user_input) <= 0:
    print("Invalid input. Please enter a positive number.")
    user_input = input("Enter a positive number: ")

print("You entered:", user_input)
```
```

This example demonstrates how to use a `while` loop to repeatedly prompt the user for input until a valid positive number is provided.

#### ### Example 3: Sum of Digits

```
```python
# Calculate the sum of digits of a number
number = int(input("Enter a number: "))
sum_of_digits = 0

while number > 0:
    digit = number % 10
    sum_of_digits += digit
    number //= 10

print("Sum of digits:", sum_of_digits)
```
```

This example calculates the sum of the digits of a number using a `while` loop.

#### ### Example 4: Infinite Loop with User Break

```
```python
# Demonstrate a while loop with a break statement
while True:
    user_input = input("Enter 'quit' to exit: ")

    if user_input.lower() == 'quit':
        break

    print("You entered:", user_input)
```
```

In this example, the `while` loop runs indefinitely until the user enters "quit," at which point the `break` statement is used to exit the loop.

#### ### Note:

- It's important to ensure that the condition in a `while` loop eventually becomes `False` to prevent infinite loops.
- Care should be taken to avoid conditions that never become `False`, leading to an infinite loop.

The `while` loop is useful when the number of iterations is not known beforehand and depends on a dynamic condition. It provides flexibility in handling various scenarios where repetitive execution is required.