Looping Statement in Python Programming

Looping statements in Python, such as `for` and `while`, facilitate code repetition based on conditions. The `for` loop is commonly used to iterate over sequences, while the `while` loop continues as long as its condition remains `True`. These loops enable efficient and repetitive execution without redundancy.

The for loop

In [1]:
```
# The for loop in Python is used to iterate over a sequence (like a li

# Example: Print each fruit in a list
fruits = ["apple", "banana", "cherry"]
for fruit in fruits:
    print(fruit)
```

```
apple
banana
cherry
```

The range() function

In [2]:
```
# To loop through a set of code a specified number of times, we can us

# Example: Print numbers 0 to 4
for num in range(5):
    print(num)
```

```
0
1
2
3
4
```

3. The while loop

In [3]:
```
# The while loop lets you execute a set of statements as long as a cor

# Example: Print numbers 1 to 5 using while loop
num = 1
while num <= 5:
    print(num)
    num += 1
```

```
1
2
3
4
5
```

Nested Loops

In [4]:
```
# A nested loop is a loop inside a loop.

# Example: Matrix traversal
matrix = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]

for row in matrix:
    for element in row:
        print(element, end=" ")
    print()  # Move to the next line after each row
```

```
1 2 3
4 5 6
7 8 9
```

In [5]:
```
# Note: The `break` statement can be used to exit a loop prematurely,
# and the `continue` statement can be used to skip the current iterati
```

Understanding break and continue in Python

The break statement

In [6]:
```
# The `break` statement is used to exit a loop prematurely when a cert
```

In [7]:
```
# Example: Find the first number divisible by 5 in a list
numbers = [1, 3, 7, 9, 10, 13, 15]
for num in numbers:
    if num % 5 == 0:
        print(f"The first number divisible by 5 is: {num}")
        break
```

```
The first number divisible by 5 is: 10
```

The continue statement

In [8]:
```
# The `continue` statement is used to skip the current iteration of th
```

In [9]:
```
# Example: Print all numbers except those divisible by 5
for num in numbers:
    if num % 5 == 0:
        continue
    print(num)
```

```
1
3
7
9
13
```

In [10]:
```
Note:
- The `break` statement will completely exit the loop.
- The `continue` statement will only skip the current iteration and t
```

Pass Statement

In [11]:
```
# The pass statement in Python is essentially a "do nothing" statement
```

In [12]:
```
# Defining a function, but not implementing it yet
def not_implemented_function():
    pass

# This will not raise any error when called
not_implemented_function()
```

In [ ]: