LOOPS

loops are used to repeatedly execute a block of code. There are two main types of loops in Python: `for` and `while`. Each has its own use cases and syntax.

1. `for` loop:

The `for` loop is used for iterating over a sequence (that is either a list, tuple, dictionary, string, or other iterable objects). The basic syntax of a `for` loop is as follows:

```python
for variable in sequence:
    # Code to be executed for each element in the sequence
```

Here's a simple example using a list:

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

for num in numbers:
    print(num)
```

This will print each number in the `numbers` list.

2. `while` loop:

The `while` loop is used to repeatedly execute a block of code as long as a condition is true. The basic syntax of a `while` loop is as follows:

```python
while condition:
    # Code to be executed as long as the condition is true
```

Here's a simple example using a `while` loop to print numbers from 1 to 5:

```python
count = 1

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

This will print numbers 1 through 5.

Control Statements in Loops:

1. **`break` statement:** The `break` statement is used to exit the loop prematurely. It is typically used when a certain condition is met.

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

for num in numbers:
    if num == 3:
        break
    print(num)
```

This will print 1 and 2, and then the loop will terminate when `num` becomes 3.

2.`continue` statement: The `continue` statement is used to skip the rest of the code inside the loop for the current iteration and move to the next iteration.

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

for num in numbers:
    if num == 3:
        continue
    print(num)
```

1. This will print all numbers except 3.

Looping through a Range:

You can use the `range()` function to generate a sequence of numbers that you can iterate over.

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

This will print numbers from 0 to 4.

Nested Loops:

You can also have loops inside loops, known as nested loops. Here's an example:

```python
for i in range(3):
    for j in range(2):
        print(i, j)
```

This will print combinations of `i` and `j` for each iteration.

Understanding loops is fundamental in programming, as they allow you to efficiently perform repetitive tasks.