

Day 5: Loops and Iterations in Python

Topics Covered:

- *Introduction to Loops*
- *for Loop*
- *while Loop*
- *Nested Loops*
- *Loop Control Statements*
- *Range Function*
- *Iterating over Sequences*

1. Introduction to Loops:

Loops are used to execute a block of code repeatedly, as long as a specified condition is true. Python provides two main types of loops: the `for` loop and the `while` loop. Loops are essential for automation and handling repetitive tasks.

2. for Loop:

The `for` loop is used to iterate over a sequence (such as a list, tuple, or range) and execute a block of code for each item in the sequence.

Example:

```
for i in range(5):  
    print(i) # Output: 0 1 2 3 4
```

3. while Loop:

The `while` loop repeatedly executes a block of code as long as the specified condition is true. If the condition is false, the loop stops.

Example:

```
i = 0  
while i < 5:  
    print(i)  
    i += 1 # Output: 0 1 2 3 4
```

4. Nested Loops:

You can place one loop inside another loop. This is called a nested loop. Nested loops are useful when dealing with multi-dimensional data structures.

Example:

```
for i in range(3):  
    for j in range(3):  
        print(i, j) # Output: (0, 0) (0, 1) (0, 2) (1, 0) (1, 1) (1, 2) (2, 0) (2, 1) (2, 2)
```

5. Loop Control Statements:

Loop control statements are used to alter the flow of a loop. The three main loop control statements are `'break'`, `'continue'`, and `'pass'`.

Example using break:

```
for i in range(5):  
    if i == 3:  
        break  
    print(i) # Output: 0 1 2
```

Example using continue:

```
for i in range(5):  
    if i == 3:  
        continue  
    print(i) # Output: 0 1 2 4
```

Example using pass:

```
for i in range(5):  
    if i == 3:  
        pass  
    print(i) # Output: 0 1 2 3 4
```

6. Range Function:

The `'range()'` function is used to generate a sequence of numbers. It is commonly used in `'for'` loops to iterate a specific number of times.

Example using range:

```
for i in range(1, 6):  
    print(i) # Output: 1 2 3 4 5
```

7. Iterating over Sequences:

You can use loops to iterate over various data structures such as lists, tuples, and dictionaries.

Example with list:

```
my_list = [1, 2, 3, 4, 5]  
for item in my_list:  
    print(item) # Output: 1 2 3 4 5
```

Example with tuple:

```
my_tuple = (1, 2, 3, 4, 5)  
for item in my_tuple:  
    print(item) # Output: 1 2 3 4 5
```

Example with dictionary:

```
my_dict = {"a": 1, "b": 2, "c": 3}
for key, value in my_dict.items():
    print(key, value) # Output: a 1 b 2 c 3
```

Summary of Day 5:

Today, we explored loops and iterations in Python. We learned about `for` and `while` loops, and how to use them to iterate over sequences of data. We also learned how to control loop execution with control statements like `break`, `continue`, and `pass`, and how to use the `range()` function for generating number sequences.