😊 For loops

A **for loop** in Python is used for iterating over a sequence (such as a list, tuple, dictionary, string, or range). It allows you to execute a block of code multiple times without manually repeating it.

How the For Loop Works

1. It iterates through each element in a sequence.
2. Executes the block of code for each element.
3. Stops when there are no more elements left.

Use Cases of For Loop

1. **Iterating Over a List or Tuple**
    - Example: Processing items in a shopping cart.
2. **Looping Through a String**
    - Example: Checking each character in a password for validation.
3. **Using the range() Function**
    - Example: Running a loop a specific number of times, like generating numbers from 1 to 10.
4. **Iterating Over a Dictionary**
    - Example: Accessing keys and values of a dictionary to process data.
5. **Reading Files Line by Line**
    - Example: Processing large text files efficiently.

Benefits of For Loop

1. **Simplifies Iteration**
    - Makes looping over sequences easy and readable.
2. **Reduces Code Complexity**
    - Eliminates the need for manually tracking indexes or counters.
3. **Memory Efficient**
    - Iterates over sequences without storing extra data in memory.
4. **Supports Various Data Structures**
    - Works with lists, tuples, strings, dictionaries, and even generators.

😄 To begin with the Lab

1. In our first example, we are iterating through the list of numbers from 1 to 10.

```
[1]: list1 = [1,2,3,4,5,6,7,8,9,10]
```

```
[2]: for num in list1:
         print(num)
```

```
1
2
3
4
5
6
7
8
9
10
```

2. In example 2, you can see that we are saying that in the list 1 print the even number by dividing the number with 2 and if we get 0 as the output then print the number.
3. But then we changed our approach and added an else statement and told it to print the odd number as well.

[7]:
```
for num in list1:
    if num % 2 == 0:
        print(num)
```

```
2
4
6
8
10
```

We could have also put an `else` statement in there:

[8]:
```
for num in list1:
    if num % 2 == 0:
        print(num)
    else:
        print('Odd number')
```

```
Odd number
2
Odd number
4
Odd number
6
Odd number
8
Odd number
10
```

[9]:
```
for num in list1:
    if num % 2 == 0:
        print(num)
    else:
        print(f'Odd number: {num}')
```

```
Odd number: 1
2
Odd number: 3
4
Odd number: 5
6
Odd number: 7
8
Odd number: 9
10
```

4. Another common idea during a for loop is keeping some sort of running tally during multiple loops.

[19]:
```
# Start sum at zero
list_sum = 0

for num in list1:
    list_sum = list_sum + num

print(list_sum)
```

```
55
```

[21]:
```
# Start sum at zero
list_sum = 0

for num in list1:
    list_sum = list_sum + num

    print(list_sum)
```

```
1
3
6
10
15
21
28
36
45
55
```

[23]:
```
# Start sum at zero
list_sum = 0

for num in list1:
    list_sum += num

print(list_sum)
```

```
55
```

5. Below we have used for loop for a string.

[11]:
```
for letter in 'This is a string.':
    print(letter)
```

```
T
h
i
s

i
s

a

s
t
r
i
n
g
.
```

6. If want to use for loop for tuples then we can do that too.

[12]:
```
tup = (1,2,3,4,5)

for t in tup:
    print(t)
```

```
1
2
3
4
5
```

[13]:
```
list2 = [(2,4),(6,8),(10,12)]
```

•••

[14]:
```
for tup in list2:
    print(tup)
```

```
(2, 4)
(6, 8)
(10, 12)
```

[15]:
```
# Now with unpacking!
for (t1,t2) in list2:
    print(t1)
```

```
2
6
10
```