For Loop

Dr. Seema Gupta Bhol

For Loops

- A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).
- With the for loop we can execute a set of statements, once for each item in a list, tuple, set etc.

Example : Loop through the letters in the word "banana":

```
for x in "banana":
  print(x)
```

The for loop does not require an indexing variable to set beforehand.

Loop Lists

One can loop through the list items by using a for loop

- Print all items in the list, one by one:

```
thislist = ["apple", "banana", "cherry"]
for x in thislist:
  print(x)
```

A short hand for loop that will print all items in a list:

```
thislist = ["apple", "banana", "cherry"]
[print(x) for x in thislist]
```

Loop Through the Index Numbers

One can also loop through the list items by referring to their index number.

Use the range() and len() functions to create a suitable iterable.

Example: Print all items by referring to their index number:

```
thislist = ["apple", "banana", "cherry"]
for i in range(len(thislist)):
  print(thislist[i])
```

The range() Function

To loop through a set of code a specified number of times, we can use the range() function,

The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

Example : Using the range() function:

```
for x in range(6):
print(x)
```

Note : The range(6) is not the values of 0 to 6, but the values 0 to 5.

The range() function defaults to 0 as a starting value, however it is possible to specify the starting value by adding a parameter: range(2, 6), which means values from 2 to 6 (but not including 6):

Example :Using the start parameter:

```
for x in range(2, 6):
  print(x)
```

The `range()` function defaults to increment the sequence by 1, however it is possible to s
parameter: `range(2, 30, 3)`:

Example: Increment the sequence with 3

```
for x in range(2, 30, 3):
  print(x)
```

The break Statement

With the break statement we can stop the loop before it has looped through all the items:

Example 1: Exit the loop when x is "banana":

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
  if x == "banana":
    break
```

Example2 :Exit the loop when x is "banana", but this time the break comes before the print:

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  if x == "banana":
    break
  print(x)
```

The continue Statement

With the continue statement we can stop the current iteration of the loop, and continue with the next interation.

Example: Do not print banana

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  if x == "banana":
    continue
  print(x)
```

Else in For Loop

The else keyword in a for loop specifies a block of code to be executed when the loop is finished:

Example: Print all numbers from 0 to 5, and print a message when the loop has ended

```
for x in range(6):
  print(x)
else:
  print("Finally finished!")
```

The else block will NOT be executed if the loop is stopped by a break statement.

Example: Break the loop when x is 3, and see what happens with the else block:

```
for x in range(6):
  if x == 3: break
  print(x)
else:
  print("Finally finished!")
```

Nested Loops

A nested loop is a loop inside a loop.

The "inner loop" will be executed one time for each iteration of the "outer loop":

Example: Print each adjective for every fruit

```
adj = ["red", "big", "tasty"]
fruits = ["apple", "banana", "cherry"]
for x in adj:
  for y in fruits:
    print(x, y)
```

Exercise

1. Write a program to print all even numbers between 1 and 20.

2. Write a program to print the multiplication table of a given number.

Sample Input:5
Sample Output:

```
5 x 1 = 5
5 x 2 = 10
...
5 x 10 = 50
```

3. Write a program to count the number of digits in a given number using a for loop

4.Write a program to find the factorial of a number.

5.Write a program to count the number of vowels in a given string.

6. Write a program to find the largest number in a list using a for loop (without using max()).

Exercise

7. Write a program to check whether a given number is prime.

8. Write a program to print N terms of Fibonacci series.

9. Write a program to reverse a string using a for loop.

10. Write a program to print the following pattern:

```
*
**
***
****
*****
```