Tutorial 9

Date: Feb 9, 2021

Email: mohdakram.ansari@ucalgary.ca

Agenda

1. Sequence in python
2. For Loop in python

1. Sequence

A sequence in python is a collection of items in an order.

Strings

```
# A String (sequence of characters)
text = "Python"
print(list(text))
```

'P'	'y'	't'	'h'	'o'	'n'

List *(more on this later)*

```
# A list of languages
languages = ["English", "French", "Spanish"]
print(languages)
```

"English"	"French"	"Spanish"

Range

Range in python is used for generating a sequence of integers.

Syntax

```
range(start, stop, step)
```

Note: The start is optional with a default value of 0. Step is optional with a default value of 1

Example

```
# A range of numbers from 0 to 9
numbers = range(0, 10)
print(list(numbers))
```

0	1	2	3	4	5	6	7	8	9

Examples

```
# empty range
print(list(range(0)))
# Output: []

# using range(stop)
print(list(range(10)))
# Output: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

# using range(start, stop)
print(list(range(5, 10)))
# Output: [5, 6, 7, 8, 9]

# using range(start, stop, step)
print(list(range(1, 10, 2)))
# Output: [1, 3, 5, 7, 9]

# Negative step
print(list(range(2, -2, -1)))
# Output: [2, 1, 0, -1]

# Bad range
print(list(range(2, -2)))
# Output: []
```

For loop

A for loop is used to iterate over a sequence.

Syntax

```
for variable in sequence:
    # loop body
```

Example

```
# Print numbers from 0 to 9

# Using while loop
i = 0
while i < 10:
    print(i)
    i += 1

# Using for loop
for i in range(0, 10):
    print(i)
```

Example

Print the multiplication table of a number entered by the user

```
number = int(input("Enter an integer: "))

# Count from 1 to 10
for count in range(1, 11):
    product = number * count
    print("{0} x {1} = {2}".format(number, count, product))
```

Example

Write a program to check whether an integer entered by the user is prime or not.

```
number = int(input("Enter an integer: "))

# The number is assumed to be prime
is_prime = True
for i in range(2, number):
    if number % i == 0:
        is_prime = False
        break

# Print result
if is_prime == True:
    print("PRIME")
else:
    print("COMPOSITE")

```

Nested for loop

We can write a for/while loop inside the body of another for/while loop. This is known is nesting of loops.

Example

Print prime numbers from 1 to 99

```
# 1 is not a prime number
for number in range(2, 100):
    # The number is assumed to be prime
    is_prime = True
    for i in range(2, number):
        if number % i == 0:
            is_prime = False
            break

    # Only print the number if it is prime
    if is_prime == True:
        print("{} is a prime number!".format(number))
```

Exercise

Can you write a program to find the sum of odd numbers from 1 to 99.

Result should be equal to:

result = 1 + 3 + 5 + 7 + ... + 97 + 99

Solution

```
result = 0
for i in range(1, 100, 2):
    result += i

print("Sum of odd numbers from 1 to 99: {}".format(result))
```

Extra

Python is pretty powerful and has handy utilities that can make your program much compact.

The above program can be written in just 1 line! Pretty cool right?

```
print(sum(range(1, 100, 2)))
```