



(<https://apssdc.in>)

APSSDC

Andhra Pradesh State Skill Development Corporation



Day06 Python Programming

Day06 Objectives

Loops in Python

1. What is a Loop
2. Requirements to Construct a Loop
3. Types of Loops in Python
 - A. For Loop
 - B. While Loop
4. Loop Termination Keyword (Break)
5. Iteration Termination Keyword (Continue)

Loops in Python

To execute sequence of instructions for multiple times

- For
- While

```
for(i = 0; i < condition; inc/dec)
```

Requirements to construct a loop

1. Initialization variable/ iterative variable -> it a variable declared to iterate the loop
2. Condition
3. Increment/decrement
4. To get the finite result/ expected result
 - Condition
 - inc/dec of iterative variable

For Loop Syntax

```

for iterative_variable in sequenceOfData:
    statement1
    statement2
    .
    .
    .
    .
    .
    .
    statementn

```

Sequence of Data

Data --> Characters, Elements, SeqOfNumbers

- Integers
- Strings
- Lists
- Tuple
- Dictionary
- Sets

Iterating sequence of integers

`range()` #- 1 req arguments, 2 default arguments

```

# req = Condition, init = 0, inc = 1
# condition, init = 5, inc = 1
# init, condition, inc

```

```

In [1]: 1 for num in range(10):
        2     print(num, end = '\t')

```

0 1 2 3 4 5 6 7 8 9

```

In [2]: 1 range()

```

TypeError

Traceback (most recent call last)

<ipython-input-2-5bcbe005bf48> in <module>

----> 1 range()

TypeError: range expected 1 argument, got 0

```
In [3]: 1 for num in range(1, 10):
        2     print(num, end = '\t')
```

1 2 3 4 5 6 7 8 9

```
In [4]: 1 for num in range(0, 10, 2):
        2     print(num, end = '\t')
```

0 2 4 6 8

```
In [5]: 1 for num in range(10, 0, -1):
        2     print(num, end = '\t')
```

10 9 8 7 6 5 4 3 2 1

```
In [6]: 1 for num in range(1000000, 0, -1):
        2     print(num, end = '\t')
```

1000000	999999	999998	999997	999996	999995	999994	999993	999992	9
999991	999990	999989	999988	999987	999986	999985	999984	999983	9
999982	999981	999980	999979	999978	999977	999976	999975	999974	9
999973	999972	999971	999970	999969	999968	999967	999966	999965	9
999964	999963	999962	999961	999960	999959	999958	999957	999956	9
999955	999954	999953	999952	999951	999950	999949	999948	999947	9
999946	999945	999944	999943	999942	999941	999940	999939	999938	9
999937	999936	999935	999934	999933	999932	999931	999930	999929	9
999928	999927	999926	999925	999924	999923	999922	999921	999920	9
999919	999918	999917	999916	999915	999914	999913	999912	999911	9
999910	999909	999908	999907	999906	999905	999904	999903	999902	9
999901	999900	999899	999898	999897	999896	999895	999894	999893	9
999892	999891	999890	999889	999888	999887	999886	999885	999884	9
999883	999882	999881	999880	999879	999878	999877	999876	999875	9
999874	999873	999872	999871	999870	999869	999868	999867	999866	9
999865	999864	999863	999862	999861	999860	999859	999858	999857	9
999856	999855	999854	999853	999852	999851	999850	999849	999848	9
999847	999846	999845	999844	999843	999842	999841	999840	999839	9
999838	999837	999836	999835	999834	999833	999832	999831	999830	9
999829	999828	999827	999826	999825	999824	999823	999822	999821	9

Task

1. write a program to print multiplication table for the number given by the user from 1 to 50

output - enter a number 5

$$5 \times 1 = 5$$

.

.

.

.

.

.

.

.

.

.

$$5 \times 50 = 250$$

In [8]:

```
1 num = input("enter a number to get multiple table: ")
2 for i in range(1, 51):
3     print(num, 'x', i, '=', num*i)
```

enter a number to get multiple table: 5

```
5 x 1 = 5
5 x 2 = 55
5 x 3 = 555
5 x 4 = 5555
5 x 5 = 55555
5 x 6 = 555555
5 x 7 = 5555555
5 x 8 = 55555555
5 x 9 = 555555555
5 x 10 = 5555555555
5 x 11 = 55555555555
5 x 12 = 555555555555
5 x 13 = 5555555555555
5 x 14 = 55555555555555
5 x 15 = 555555555555555
5 x 16 = 5555555555555555
5 x 17 = 55555555555555555
5 x 18 = 555555555555555555
5 x 19 = 5555555555555555555
5 x 20 = 55555555555555555555
5 x 21 = 555555555555555555555
5 x 22 = 5555555555555555555555
5 x 23 = 55555555555555555555555
5 x 24 = 555555555555555555555555
5 x 25 = 5555555555555555555555555
5 x 26 = 55555555555555555555555555
5 x 27 = 555555555555555555555555555
5 x 28 = 5555555555555555555555555555
5 x 29 = 55555555555555555555555555555
5 x 30 = 555555555555555555555555555555
5 x 31 = 5555555555555555555555555555555
5 x 32 = 55555555555555555555555555555555
5 x 33 = 555555555555555555555555555555555
5 x 34 = 5555555555555555555555555555555555
5 x 35 = 55555555555555555555555555555555555
5 x 36 = 555555555555555555555555555555555555
5 x 37 = 5555555555555555555555555555555555555
5 x 38 = 55555555555555555555555555555555555555
5 x 39 = 555555555555555555555555555555555555555
5 x 40 = 5555555555555555555555555555555555555555
5 x 41 = 55555555555555555555555555555555555555555
5 x 42 = 555555555555555555555555555555555555555555
5 x 43 = 5555555555555555555555555555555555555555555
5 x 44 = 55555555555555555555555555555555555555555555
5 x 45 = 555555555555555555555555555555555555555555555
5 x 46 = 5555555555555555555555555555555555555555555555
5 x 47 = 55555555555555555555555555555555555555555555555
5 x 48 = 555555555555555555555555555555555555555555555555
5 x 49 = 5555555555555555555555555555555555555555555555555
5 x 50 = 55555555555555555555555555555555555555555555555555
```

```
In [9]: 1 num = int(input("enter a number to get multiple table: "))
        2 for i in range(1, 51):
        3     print(num, 'x', i, '=', num*i)
```

enter a number to get multiple table: 5

5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
5 x 11 = 55
5 x 12 = 60
5 x 13 = 65
5 x 14 = 70
5 x 15 = 75
5 x 16 = 80
5 x 17 = 85
5 x 18 = 90
5 x 19 = 95
5 x 20 = 100
5 x 21 = 105
5 x 22 = 110
5 x 23 = 115
5 x 24 = 120
5 x 25 = 125
5 x 26 = 130
5 x 27 = 135
5 x 28 = 140
5 x 29 = 145
5 x 30 = 150
5 x 31 = 155
5 x 32 = 160
5 x 33 = 165
5 x 34 = 170
5 x 35 = 175
5 x 36 = 180
5 x 37 = 185
5 x 38 = 190
5 x 39 = 195
5 x 40 = 200
5 x 41 = 205
5 x 42 = 210
5 x 43 = 215
5 x 44 = 220
5 x 45 = 225
5 x 46 = 230
5 x 47 = 235
5 x 48 = 240
5 x 49 = 245
5 x 50 = 250

```
In [10]: 1 num = int(input("enter a number to get multiple table:"))
2 i = int(input("starting range"))
3 p = int(input("ending range"))
4
5 for i in range(i, p + 1):
6     print(num, 'x', i, '=', num * i)
7
```

```
enter a number to get multiple table:10
starting range1
ending range10
10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
10 x 10 = 100
```

Task 2

Write a program to find all the leap years between 1900 to 2020

- Conditions to be a leap year
 - it should be divisible by 4 as a leap year
 - 100 is not a leap year
 - 400 yr is a leap year

```
In [11]: 1 24 * 0.25
```

Out[11]: 6.0

```
In [12]: 1 365/(6 * 60)
```

Out[12]: 1.0138888888888888

```
In [14]: ▶ 1 year = int(input("Enter a year: "))
2 for I in range (1900,2000):
3     if (year % 4) == 0:
4         if (year % 100) == 0:
5             if (year % 400) == 0:
6                 print("{0} is a leap year".format(year))
7             else: print("{0} is not a leap year".format(year))
8         else: print("{0} is a leap year".format(year))
9     else: print("{0} is not a leap year".format(year))
```

File "<tokenize>", line 8

```
    else: print("{0} is a leap year".format(year))
    ^
```

IndentationError: unindent does not match any outer indentation level

```
In [15]: ▶ 1 for year in range(1800,2020):
2     if (year % 4 == 0 and year % 100 != 0) or year % 400 == 0:
3         print(year, end = ' ')
```

```
1804 1808 1812 1816 1820 1824 1828 1832 1836 1840 1844 1848 1852 1856 1860
1864 1868 1872 1876 1880 1884 1888 1892 1896 1904 1908 1912 1916 1920 1924
1928 1932 1936 1940 1944 1948 1952 1956 1960 1964 1968 1972 1976 1980 1984
1988 1992 1996 2000 2004 2008 2012 2016
```

iterating sequence of strings or characters

```
for iter in Strings:
    statements
```



```
In [16]: 1 s1 = 'APSSDC Day06 python programming'
          2
          3 for char in s1:
          4     print(char)
```

```
A
P
S
S
D
C

D
a
y
0
6

p
y
t
h
o
n

p
r
o
g
r
a
m
m
i
n
g
```

Task

Write a program to count lower case, upper case, numbers and special characters available in string

- U = 7
- L = 19
- num = 2
- sep = 3

```
In [17]: 1 s1
```

```
Out[17]: 'APSSDC Day06 python programming'
```

```
In [18]: 1 u,l,n,s = 0,0,0,0
2
3 for char in s1:
4     if char.isupper():
5         u += 1
6     elif char.islower():
7         l += 1
8     elif char.isdigit():
9         n += 1
10    else:
11        s += 1
12
13 print("Upper case characters {}, Lower case characters {}".format(u, l))
14 print("Numbers {}, Special characters {}".format(n, s))
```

Upper case characters 7, Lower case characters 19
Numbers 2, Special characters 3

```
In [28]: 1 string = ""In the Gregorian calendar, the standard calendar in most of t
2 4 are leap years. In each leap year, the month of February has 29 days in
3 calendar every four years compensates for the fact that a period of 365 c
4 almost 6 hours.[10] Some exceptions to this basic rule are required since
5 less than 365.25 days. The Gregorian reform modified the Julian calendar
```

```
In [29]: 1 string
```

Out[29]: "In the Gregorian calendar, the standard calendar in most of the world, most years that are multiples of 4 are leap years. In each leap year, the month of February has 29 days instead of 28. Adding one extra day in the calendar every four years compensates for the fact that a period of 365 days is shorter than a tropical year by almost 6 hours.[10] Some exceptions to this basic rule are required since the duration of a tropical year is slightly less than 365.25 days. The Gregorian reform modified the Julian calendar's scheme of leap years as follows:"

Task 3

- Counting no of words in string

```
In [30]: 1 string.count()
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-30-d7df178771b5> in <module>
----> 1 string.count()

TypeError: count() takes at least 1 argument (0 given)
```

```
In [31]: 1 len(string.split())
```

```
Out[31]: 100
```

```
In [32]: 1 string.count(' ') + 1
```

```
Out[32]: 100
```

While Loop

```
init_variable
while condition:
    statement1
    statement2
    .
    .
    .
    .
    .
    .
    statementn
    inc/dec
```

```
In [33]: 1 init = 1
2
3 while init < 51:
4     print(init ** 2, end = " ")
5     init += 1
```

```
1 4 9 16 25 36 49 64 81 100 121 144 169 196 225 256 289 324 361 400 441 484
529 576 625 676 729 784 841 900 961 1024 1089 1156 1225 1296 1369 1444 1521
1600 1681 1764 1849 1936 2025 2116 2209 2304 2401 2500
```

Termination conditions

- Break - stops the execution of the current loop
- continue - The continue keyword is used to end the current iteration in a loop, and continues to the next iteration.

```
In [34]: 1 for num in range(1, 50):
2         if num == 5 or num == 10 or num == 15:
3             continue
4         elif num == 40:
5             print("Supply is terminated")
6             break
7         else:
8             print(num, end = ' ')
```

```
1 2 3 4 6 7 8 9 11 12 13 14 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
32 33 34 35 36 37 38 39 Supply is terminated
```

- Take the input from the user and display to the user until unless the user given input as end

```
In [35]: 1 while True:
2         inp = input()
3         if inp == 'end':
4             break
```

```
jdbvsbvmnd
hdbvhsbvj
sjdbvjsbv
jshvbsjdvbsds
sjhdbvshvdj
end
```

```
In [36]: 1 inp = 0
2 while inp != 'end':
3     inp = input("Enter something")
4     print(inp)
```

```
Enter somethingbdsj
bdsj
Enter somethingsjbgjds
sjbgjds
Enter somethingsjhbdjsdv
sjhbdjsdv
Enter somethingsdfjsdhbv
sdfjsdhbv
Enter somethingsjdbvsdjvb
sjdbvsdjvb
Enter somethingend
end
```

Day06 Outcomes

- Loops in Python
 - For Loop
 - While Loop
- Loop Termination Keyword (Break)
- Iteration Termination Keyword (Continue)