

Example

```
spaces=4
for i in range(1,6): # 1 2 3 4 5
    for s in range(1,spaces+1):
        print(" ",end=")
    for j in range(1,i+1):
        print("* ",end=")
```

```
spaces=spaces-1
print()
```

```
spaces=1
for i in range(4,0,-1):
    for s in range(1,spaces+1):
        print(" ",end=")
    for j in range(1,i+1):
        print("* ",end=")
    print()
    spaces=spaces+1
```

Output

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

Branching Statements

1. break
2. continue
3. pass (Empty Blocks)
4. return

break

break is branching statement, which is used to terminate the loop in between or unconditionally.

continue

This statement is used to move execution control to the beginning of the loop.

Example:

write a program to print sum first 10 numbers
which are divisible with 7

```
num=7
s=0
c=0
while True:
    if num%7==0:
        print(num)
        s=s+num
        c=c+1
    num=num+1
    if c>10:
        break
```

```
print(f'Sum is {s}')
```

Output:

```
7
14
21
28
35
42
49
56
63
70
77
Sum is 462
```

Example:

Write a program to find input number is prime or not

```
num=int(input("Enter any number")) #6
```

```
c=0
for i in range(1,num+1): # 1 2 3 4 5 6
    if num%i==0:
        c=c+1
    if c>2:
        break

if c==2:
    print(f'{num} is prime')
else:
    print(f'{num} is not prime')
```

Output:

Enter any number6
6 is not prime

Enter any number5
5 is prime

Example:

Example of continue statement

```
for num in range(1,21):
    if num%2==0:
        continue
    print(num)
```

Output:

1
3
5
7
9
11
13
15
17
19

Data Structures or Collection Types

Python data types are classified into two categories

1. Scalar Types

- a. Int
- b. Float
- c. Complex
- d. Bool
- e. NoneType

2. Collection Types

a. Sequences (Ordered Collection)

- i. List → Mutable
- ii. Tuple → Immutable
- iii. Range → Immutable
- iv. String → Immutable
- v. Bytes → Immutable
- vi. Bytearray → mutable

b. Sets (Unordred Collections)

- i. Set → Mutable
- ii. Frozenset → Immutable

c. Mapping (Key based collections)

- i. Dictionary → Mutable

Scalar types are used to represent one value.

Collection types are used to represent more than one value.

Collection types allow to group individual values/objects and represent as one object.

Collections are containers; containers are used to store more than one value.

Every collection uses one data structure for storing or organizing data. Data structure defines set of rules and regulations for organization of data in memory.

Advantage of collection types

- 1. Avoid declaring number of variables for storing more than one value
- 2. Every collection is one data structure, complex problems can be solved easily
- 3. It allows perform aggregate operations.

Sequences

Sequence data types allow organizing data in memory in sequential order. By storing in sequential order, the data can be read and write sequential and randomly.

Sequences are index based collection.

In application development, sequences are used to group individual objects and allow duplicates and reading and writing is done sequential or random.

List

List is a sequence data type.

List is a mutable sequence data type. After creating list changes can be done.

Lists are mutable sequences, typically used to store collections of homogeneous items and heterogeneous items.

How to create list?

Lists may be constructed in several ways:

1. Using a pair of square brackets to denote the empty **list**: []
2. Using square brackets, separating items with commas: [a], [a, b, c]
3. Using a **list** comprehension: [x for x in iterable]
4. Using the type constructor or function: **list**() or **list**(iterable)

