python-3

January 11, 2024

1 list

In Python, a lis is an inbuilt data type in Python which is versatile and mutable ordered collection of elements. Lists are defined by enclosing a comma-separated sequence of values within square brackets []. Lists can contain elements of different data types, and each element is indexed starting from 0.

[2]:
```
L =[1,2,3,4,5,'ashish','s',12.5]

L.append(1)   # to append somthing at the end of the list
```

[6]:
```
type(L)
```

[6]:
```
list
```

[7]:
```
print(type(L))
```

```
<class 'list'>
```

[8]:
```
len(L)
```

[8]:
```
8
```

[3]:
```
L[5]   # fetching element by its index value
```

[3]:
```
'ashish'
```

[12]:
```
L.remove('ashish')      # to remove an element
```

[13]:
```
L
```

[13]:
```
[1, 2, 3, 4, 5, 's', 12.5]
```

[4]:
```
L.insert(5,'ashish')            # to add something at specific index value
```

[5]:
```
L
```

[5]:
```
[1, 2, 3, 4, 5, 'ashish', 'ashish', 's', 12.5, 1]
```

```
[6]: L.pop(0)           # delete something by its index value
```

```
[6]: 1
```

```
[19]: L
```

```
[19]: [2, 3, 4, 5, 'ashish', 's', 12.5, 'ashish']
```

2 tuple

A tuple is another built-in data type in Python. It is an ordered, immutable collection of elements. Tuples are similar to lists, but the key difference is that once a tuple is created, its elements cannot be modified or changed. Tuples are defined using parentheses ()

```
[18]: T=(1,2,4)
```

```
[19]: type(T)
```

```
[19]: tuple
```

```
[20]: #T.append()
      # L.remove()
      # L.pop()
      # L.insert()
      T[0]
```

```
[20]: 1
```

3 sets

A set in Python is an unordered, mutable collection of unique elements. Sets are defined using curly braces {} or by using the set() constructor. Unlike lists or tuples, sets do not allow duplicate elements.

```
[11]: S=set()
```

```
[12]: type(S)
```

```
[12]: set
```

```
[13]: s={1,2,3,4,5,1,2,3,4,5,6,7,89,0,10,'ashish'}
```

```
[14]: type(s)
```

```
[14]: set
```

[15]:

```
s[0]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[15], line 1
----> 1 s[0]

TypeError: 'set' object is not subscriptable
```

[ ]:

```

```

4 dictinory

A dictionary in Python is an unordered, mutable collection of key-value pairs. Dictionaries are defined using curly braces {}, and each key-value pair is separated by a colon :.

[47]:

```
D= {'a':'ashish',
   'b':'Boy',1:10,1000:100}
```

[48]:

```
type(D)
```

[48]:

```
dict
```

[49]:

```
D.keys()
```

[49]:

```
dict_keys(['a', 'b', 1, 1000])
```

[50]:

```
D.values()
```

[50]:

```
dict_values(['ashish', 'Boy', 10, 100])
```

[ ]:

```
def fruits():
```

[53]:

```
L=[]
for i in range(5):
 a=input()
 L.append(a)
print(L)
```

```
 a
 b
 c
 d
 f
['a', 'b', 'c', 'd', 'f']
```

5 libraries

In Python, a library (also known as a module or package) is a collection of pre-written code that can be reused for specific tasks or functionalities.

```
[26]: l1=[1,2]
      l2=[3,4]
```

```
[58]: l1+l2
```

```
[58]: [1, 2, 3, 4]
```

```
[59]: l1-l2   # list does not support arithmetic operations
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[59], line 1
----> 1 l1-l2

TypeError: unsupported operand type(s) for -: 'list' and 'list'
```

6 NumPy is a powerful numerical computing library for Python.

# It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.

```
[27]: import numpy as np
      a =np.array(l1)
```

```
[29]: b =np.array(l2)
```

```
[30]: a+b
```

```
[30]: array([4, 6])
```

```
[31]: A= np.array([12,3,434,55])
```

```
[32]: type(A)
```

```
[32]: numpy.ndarray
```

```
[35]: A.shape  # to get the shape of the array
```

```
[35]: (4,)
```

```
[38]: arr = np.random.rand(3,3)  # to generate an array of 3,3 with random element
```

```
[37]: arr
```

```
[37]: array([[0.16840036, 0.01876406, 0.65371598],
             [0.60006366, 0.09170061, 0.27588615],
             [0.81571777, 0.54486994, 0.2454437 ]])
```

```
[78]: arr.shape
```

```
[78]: (3, 3)
```

```
[75]: len(arr)
```

```
[75]: 9
```

```
[79]: arr[0,0]
```

```
[79]: 0.524404208129184
```

```
[82]: arr[2,2]
```

```
[82]: 0.26942334871666185
```

```
[83]: arr
```

```
[83]: array([[0.52440421, 0.96373655, 0.72140629],
             [0.43497587, 0.20989841, 0.97717195],
             [0.49617606, 0.59053274, 0.26942335]])
```

```
[84]: arr[0]   # row
```

```
[84]: array([0.52440421, 0.96373655, 0.72140629])
```

```
[91]: arr[:,0] # column
```

```
[91]: array([0.52440421, 0.43497587, 0.49617606])
```

```
[92]: arr
```

```
[92]: array([[0.52440421, 0.96373655, 0.72140629],
             [0.43497587, 0.20989841, 0.97717195],
             [0.49617606, 0.59053274, 0.26942335]])
```

```
[93]: # slicing
```

```
[94]: arr
```

```
[94]: array([[0.52440421, 0.96373655, 0.72140629],
             [0.43497587, 0.20989841, 0.97717195],
             [0.49617606, 0.59053274, 0.26942335]])
```

```
[110]: arr[0:2,0:2]
```

```
[110]: array([[0.52440421, 0.96373655],
              [0.43497587, 0.20989841]])
```

```
[111]: arr[1:,1:]
```

```
[111]: array([[0.20989841, 0.97717195],
              [0.59053274, 0.26942335]])
```

```
[ ]:
```