APSSDC

Andhra Pradesh State Skill Development Corporation

(https://apssdc.in)

Day 11 Sets and Functions in Python

Recap

- Dictionary
- Dict Methods

Today Objectives

- Set
- Set Methods
- Functions in Python
    - Built-in Funtions -> print(), input(), sum(), min(), max(), type(),

      `id(), len(), range(), sorted(), reversed()`

    - User Defined Functions

Sets in Python

It is used for storing non-homogenous group of unique data on python

Properties

- `{}` for storing the data in comma seperated
- It is mutable data type
- It is an unordered
- we can't accessing the data from set using indexing
- It is iterable
- It doesn't allow duplicated data

In [1]:

```python
s1 = set()

print(type(s1))
```

```
<class 'set'>
```

In [2]:
```python
s1 = {2,1,5,6,'apssdc','Python', 2.55, 56, 0.26, 0.10}

print(type(s1))
```
```
<class 'set'>
```

In [3]:
```python
print(s1)
```
```
{0.26, 1, 2, 2.55, 0.1, 5, 6, 'apssdc', 'Python', 56}
```

In [4]:
```python
s2 = {1,2,3,4,4,3,2,1}

print(s2)
```
```
{1, 2, 3, 4}
```

In [5]:
```python
st = """Python is an interpreted high-level general-purpose programming l
Developer: Python Software Foundation
Stable release: 3.9.5 / 3 May 2021; 19 days ago
Preview release: 3.10.0b1 / 3 May 2021; 19 days ago
Typing discipline: Duck, dynamic, strong typing; gradual (since 3.5, but
First appeared: February 1991; 30 years ago
Paradigm: Multi-paradigm: object-oriented, procedural (imperative), funct
```

In [8]:
```python
c = {}

for char in st:
    # print(char, end = '  ')
    c[char] = st.count(char)

print(c)
```
```
{'P': 6, 'y': 15, 't': 28, 'h': 10, 'o': 25, 'n': 29, ' ': 67, 'i': 39,
's': 20, 'a': 38, 'e': 48, 'r': 29, 'p': 17, 'd': 19, 'g': 18, '-': 4, 'l':
17, 'v': 5, 'u': 13, 'm': 7, '.': 7, "'": 1, 'z': 1, 'c': 11, 'b': 7, 'w':
3, 'f': 5, 'W': 1, 'k': 2, '\n': 6, 'D': 2, ':': 7, 'S': 2, 'F': 3, '3': 6,
'9': 5, '5': 2, '/': 2, 'M': 3, '2': 4, '0': 5, '1': 8, ';': 4, 'T': 1,
',': 7, '(': 2, 'C': 1, ')': 2, 'j': 1}
```

In [9]:
```python
print(len(st))
```
```
565
```

In [10]:
```python
print(len(c))
```
```
49
```

In [11]:

```python
s3 = set(st)

print(s3)
```

```
{'b', '/', 'P', 'p', '2', 'c', 'j', 'i', ';', "'", 'n', 'm', 'M', ')', 'C',
'D', 'F', ':', '9', 'l', '\n', 'v', '.', 'y', 's', 'g', 'a', 'o', 'w', 'r',
'e', 'S', 'f', ' ', 'u', '5', '0', 'T', '(', 'W', 't', 'd', '3', 'h', ',',
'-', 'z', 'k', '1'}
```

In [12]:

```python
print(len(s3))
```

```
49
```

In [13]:

```python
c = {}

for char in set(st):
    # print(char, end = '  ')
    c[char] = st.count(char)

print(c)
```

```
{'b': 7, '/': 2, 'P': 6, 'p': 17, '2': 4, 'c': 11, 'j': 1, 'i': 39, ';': 4,
"'": 1, 'n': 29, 'm': 7, 'M': 3, ')': 2, 'C': 1, 'D': 2, 'F': 3, ':': 7,
'9': 5, 'l': 17, '\n': 6, 'v': 5, '.': 7, 'y': 15, 's': 20, 'g': 18, 'a': 3
8, 'o': 25, 'w': 3, 'r': 29, 'e': 48, 'S': 2, 'f': 5, ' ': 67, 'u': 13,
'5': 2, '0': 5, 'T': 1, '(': 2, 'W': 1, 't': 28, 'd': 19, '3': 6, 'h': 10,
',': 7, '-': 4, 'z': 1, 'k': 2, '1': 8}
```

In [14]:

```python
sli = st.split()


print(sli)
```

```
['Python', 'is', 'an', 'interpreted', 'high-level', 'general-purpose', 'pro
gramming', 'language.', "Python's", 'design', 'philosophy', 'emphasizes',
'code', 'readability', 'with', 'its', 'notable', 'use', 'of', 'significan
t', 'indentation.', 'Wikipedia', 'Developer:', 'Python', 'Software', 'Found
ation', 'Stable', 'release:', '3.9.5', '/', '3', 'May', '2021;', '19', 'day
s', 'ago', 'Preview', 'release:', '3.10.0b1', '/', '3', 'May', '2021;', '1
9', 'days', 'ago', 'Typing', 'discipline:', 'Duck,', 'dynamic,', 'strong',
'typing;', 'gradual', '(since', '3.5,', 'but', 'ignored', 'in', 'CPython)',
'First', 'appeared:', 'February', '1991;', '30', 'years', 'ago', 'Paradig
m:', 'Multi-paradigm:', 'object-oriented,', 'procedural', '(imperative),',
'functional,', 'structured,', 'reflective']
```

In [15]:

```python
print(len(sli))
```

```
74
```

In [16]:

```python
ssli = set(sli)

print(ssli, len(ssli))
```

```
{'days', 'gradual', 'object-oriented,', '/', 'Software', 'in', 'Develope
r:', 'Paradigm:', 'significant', 'Preview', 'Python', 'Wikipedia', 'notabl
e', '3.10.0b1', 'code', 'Duck,', 'functional,', 'but', '1991;', 'May', '3
0', 'indentation.', 'ago', 'high-level', 'structured,', 'general-purpose',
'(since', '3.5,', 'use', 'CPython)', 'interpreted', 'Stable', 'February',
'2021;', "Python's", 'its', 'Foundation', 'Typing', 'ignored', 'of', '(impe
rative),', 'dynamic,', 'typing;', 'language.', 'release:', 'an', 'Multi-par
adigm:', 'design', 'emphasizes', 'reflective', '3', 'philosophy', 'discipli
ne:', 'appeared:', 'with', 'is', 'strong', '19', 'years', 'procedural', 're
adability', 'First', 'programming', '3.9.5'} 64
```

In [17]:

```python
print(list(ssli))
```

```
['days', 'gradual', 'object-oriented,', '/', 'Software', 'in', 'Develope
r:', 'Paradigm:', 'significant', 'Preview', 'Python', 'Wikipedia', 'notabl
e', '3.10.0b1', 'code', 'Duck,', 'functional,', 'but', '1991;', 'May', '3
0', 'indentation.', 'ago', 'high-level', 'structured,', 'general-purpose',
'(since', '3.5,', 'use', 'CPython)', 'interpreted', 'Stable', 'February',
'2021;', "Python's", 'its', 'Foundation', 'Typing', 'ignored', 'of', '(impe
rative),', 'dynamic,', 'typing;', 'language.', 'release:', 'an', 'Multi-par
adigm:', 'design', 'emphasizes', 'reflective', '3', 'philosophy', 'discipli
ne:', 'appeared:', 'with', 'is', 'strong', '19', 'years', 'procedural', 're
adability', 'First', 'programming', '3.9.5']
```

In [18]:

```python
print(tuple(ssli))
```

```
('days', 'gradual', 'object-oriented,', '/', 'Software', 'in', 'Develope
r:', 'Paradigm:', 'significant', 'Preview', 'Python', 'Wikipedia', 'notabl
e', '3.10.0b1', 'code', 'Duck,', 'functional,', 'but', '1991;', 'May', '3
0', 'indentation.', 'ago', 'high-level', 'structured,', 'general-purpose',
'(since', '3.5,', 'use', 'CPython)', 'interpreted', 'Stable', 'February',
'2021;', "Python's", 'its', 'Foundation', 'Typing', 'ignored', 'of', '(impe
rative),', 'dynamic,', 'typing;', 'language.', 'release:', 'an', 'Multi-par
adigm:', 'design', 'emphasizes', 'reflective', '3', 'philosophy', 'discipli
ne:', 'appeared:', 'with', 'is', 'strong', '19', 'years', 'procedural', 're
adability', 'First', 'programming', '3.9.5')
```

In [23]:
```python
stli = str(ssli)

print(type(stli))
print(stli)
print(stli[0])
```

```
<class 'str'>
{'days', 'gradual', 'object-oriented,', '/', 'Software', 'in', 'Develope
r:', 'Paradigm:', 'significant', 'Preview', 'Python', 'Wikipedia', 'notabl
e', '3.10.0b1', 'code', 'Duck,', 'functional,', 'but', '1991;', 'May', '3
0', 'indentation.', 'ago', 'high-level', 'structured,', 'general-purpose',
'(since', '3.5,', 'use', 'CPython)', 'interpreted', 'Stable', 'February',
'2021;', "Python's", 'its', 'Foundation', 'Typing', 'ignored', 'of', '(impe
rative),', 'dynamic,', 'typing;', 'language.', 'release:', 'an', 'Multi-par
adigm:', 'design', 'emphasizes', 'reflective', '3', 'philosophy', 'discipli
ne:', 'appeared:', 'with', 'is', 'strong', '19', 'years', 'procedural', 're
adability', 'First', 'programming', '3.9.5'}
{
```

Set Methods

In [24]:
```python
se = set()
```

In [25]:
```python
print(se)
```

```
set()
```

In [26]:
```python
se.add(123)
se.add(2)
se.add(0.5)

print(se)
```

```
{0.5, 2, 123}
```

In [27]:
```python
se.add(input())
```

```
5656
```

In [28]:
```python
print(se)
```

```
{0.5, 2, 123, '5656'}
```

In [29]:
```python
se.add(0.5)

print(se)
```

```
{0.5, 2, 123, '5656'}
```

In [30]:
```python
se.add([1,2,3])

print(se)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-30-681ab51f9fe8> in <module>
----> 1 se.add([1,2,3])
      2 
      3 print(se)

TypeError: unhashable type: 'list'
```

In [31]:
```python
se.add((1,2,3))

print(se)
```

```
{0.5, 2, (1, 2, 3), '5656', 123}
```

In [32]:
```python
se.update({1,2,3,4,4})

print(se)
```

```
{0.5, 1, 2, 3, 4, (1, 2, 3), '5656', 123}
```

In [34]:
```python
se.update([1,2,3,4,4,5,6,7])


print(se)
```

```
{0.5, 1, 2, 3, 4, 5, 6, 7, (1, 2, 3), '5656', 123}
```

In [37]:
```python
print(se.pop())

print(se)
```

```
2
{3, 4, 5, 6, 7, (1, 2, 3), '5656', 123}
```

In [38]:
```python
se.remove((1,2,3))

print(se)
```

```
{3, 4, 5, 6, 7, '5656', 123}
```

In [39]:
```python
se.remove((1,2,3))


print(se)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-39-bdfa5ab0a593> in <module>
----> 1 se.remove((1,2,3))
      2
      3 print(se)

KeyError: (1, 2, 3)
```

In [40]:
```python
se.discard('5656')


print(se)
```

```
{3, 4, 5, 6, 7, 123}
```

In [41]:
```python
x = se.discard('5656')

print(x, se)
```

```
None {3, 4, 5, 6, 7, 123}
```

In [42]:
```python
x = se.discard(123)

print(x, se)
```

```
None {3, 4, 5, 6, 7}
```

In [43]:
```python
print(se[0])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-43-fb646ecf47b4> in <module>
----> 1 print(se[0])

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

In [44]:
```python
se2 = se.copy()


print(se2)
```

```
{3, 4, 5, 6, 7}
```

In [45]:
```
se2.clear()

print(se2)


del se2

print(se2)
```

```
set()

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-45-bebc9390be90> in <module>
      6 del se2
      7 
----> 8 print(se2)

NameError: name 'se2' is not defined
```

Math Sets

- Union
- Intersection

In [46]:
```
s1 = {1,2,3,4,5,6}
s2 = {4,5,6,7,8,9}


print(s1.union(s2))
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9}
```

In [47]:
```
print(s1 | s2)
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9}
```

In [48]:
```
print(s1.intersection(s2))

print(s1 & s2)
```

```
{4, 5, 6}
{4, 5, 6}
```

In [49]:
```
print(s1.difference(s2))
```

```
{1, 2, 3}
```

In [50]:
```python
print(s1-s2)
```
```
{1, 2, 3}
```

In [51]:
```python
print(s2 - s1)
```
```
{8, 9, 7}
```

In [52]:
```python
print(s1.symmetric_difference(s2))
```
```
{1, 2, 3, 7, 8, 9}
```

In [53]:
```python
print(s1 ^ s2)
```
```
{1, 2, 3, 7, 8, 9}
```

In [54]:
```python
print(s1.symmetric_difference_update(s2))

print(s1)
```
```
None
{1, 2, 3, 7, 8, 9}
```

In [55]:
```python
print(s1.difference_update(s2))

print(s1)
```
```
None
{1, 2, 3}
```

- Functions in Python
    - Built-in Funtions -> print(), input(), sum(), min(), max(), type(),

      `id(), len(), range(), sorted(), reversed()`

    - User Defined Functions

set of instruction given by the user to perform a task

- Reusable code to perform single related action
- it reduces the lines of code
- Reduces time complexity, memory

In [56]:
```python
print(abs(-5))
```
```
5
```

In [57]:

```python
li = [1,2,3,4,5,6]
li2 = [1,2,3,4,5,6,0]


print(all(li), all(li2))
```

```
True False
```

In [58]:

```python
li = [1,2,3,4,5,6,'0']
li2 = [1,2,3,4,5,6,0]


print(all(li), all(li2))
```

```
True False
```

In [59]:

```python
li = [1,2,3,4,5]
li2 = [0,0,0,0,0,0,0,1]
li3 = [0,0,0,0,0]
print(any(li), any(li2), any(li3))
```

```
True True False
```

In [60]:

```python
print(all([1,2,3,'']))
```

```
False
```

In [61]:

```python
print(bool([1,2]))
```

```
True
```

In [62]:

```python
help(str)
```

```
 |  errors is specified, then the object must expose a data buffer
 |  that will be decoded using the given encoding and error handler.
 |  Otherwise, returns the result of object.__str__() (if defined)
 |  or repr(object).
 |  encoding defaults to sys.getdefaultencoding().
 |  errors defaults to 'strict'.
 |
 |  Methods defined here:
 |
 |  __add__(self, value, /)
 |      Return self+value.
 |
 |  __contains__(self, key, /)

 |      Return key in self.
 |
 |  __eq__(self, value, /)
 |      Return self==value.
 |
 |  __format__(self, format_spec, /)
```

In [63]:

```python
dir(list)
```

Out[63]:

```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__gt__',
 '__hash__',
 '__iadd__',
 '__imul__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__reversed__',
 '__rmul__',
 '__setattr__',
 '__setitem__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'append',
 'clear',
 'copy',
 'count',
 'extend',
 'index',
 'insert',
 'pop',
 'remove',
 'reverse',
 'sort']
```

In [65]:
```python
print(str.__doc__)
```

```
str(object='') -> str
str(bytes_or_buffer[, encoding[, errors]]) -> str

Create a new string object from the given object. If encoding or
errors is specified, then the object must expose a data buffer
that will be decoded using the given encoding and error handler.
Otherwise, returns the result of object.__str__() (if defined)
or repr(object).
encoding defaults to sys.getdefaultencoding().
errors defaults to 'strict'.
```

In [66]:
```python
print(str.count.__doc__)
```

```
S.count(sub[, start[, end]]) -> int

Return the number of non-overlapping occurrences of substring sub in
string S[start:end].  Optional arguments start and end are
interpreted as in slice notation.
```

In [69]:
```python
li = [1,2,3,4]
li2 = [5,6,7,8]

li3 = zip(li, li2)

print(li3)
```

```
<zip object at 0x00000230581089C0>
```

In [70]:
```python
li3 = list(li3)

print(li3)
```

```
[(1, 5), (2, 6), (3, 7), (4, 8)]
```

In [71]:
```python
print(enumerate(li))
```

```
<enumerate object at 0x0000023058106F80>
```

In [72]:
```python
en = list(enumerate(li))

print(en)
```

```
[(0, 1), (1, 2), (2, 3), (3, 4)]
```

User-Defined functions

Syntax

```
def function_name(arg1, arg2, ....... argn): # arg are optional
    """Document for function""" # it is optional
    Block of code

    return result # return is optional
```

Types of Function

1. Based on arguments
   A. Positional argument/ required argument
   B. keyword argument
   C. default argument
   D. Variable length keyword arguments
2. Based on return and arguments
   A. With arg with return
   B. without arg with return
   C. with arg without return
   D. without arg without return
3. Call by value
4. Call by reference
5. Recursive Functions

In [76]:
```python
# with arg with return
def addition(a, b):
    print(a)
    print(b)
    return a + b
```

In [77]:
```python
addition(15, 20) # Function calling
```

```
15
20
```

Out[77]:
```
35
```

In [78]:
```python
def greet():
    return "Good Evening all"
```

In [79]:
```python
greet()
```

Out[79]:
```
'Good Evening all'
```

In [80]:
```python
# with arg without return
def addition(a, b):
    print(a)
    print(b)
    print(a + b)
```

In [81]:
```
addition(15,53)
```
```
15
53
68
```

In [82]:
```
def square(a):
    print(a ** 2)
    return a ** 2

# (5 + 6)

print(square(5) + square(6) + 2 * 5 * 6)
```
```
25
36
121
```

In [83]:
```
def greet():
    print("Good Evening all")
```

In [84]:
```
greet()
```
```
Good Evening all
```

Required arguments

In [86]:
```
def addition(a, b):
    """This function takes two args and returns addition of two args"""
    print(a)
    print(b)
    print(a + b)
```

In [88]:
```
addition.__doc__
```
Out[88]:
```
'This function takes two args and returns addition of two args'
```

In [89]:
```
addition(5, 5)
```
```
5
5
10
```

In [90]:
```
addition(5)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-90-ea28020bb458> in <module>
----> 1 addition(5)

TypeError: addition() missing 1 required positional argument: 'b'
```

In [102]:
```
def addition(a, b):
    return a + b
    print(a + b)
    print(a, b)
```

In [103]:
```
add = addition(5,5)
```

In [104]:
```
print(add)
```

```
10
```

In [105]:
```
def addition(a, b):
    return a + b, b + a, a ** b
    print(a + b)
    print(a, b)
```

In [106]:
```
add = addition(5,5)
```

In [107]:
```
print(add)
```

```
(10, 10, 3125)
```

In [97]:
```
def addition(a, b):
    print(a + b, b + a, a ** b)
    print(a + b)
    print(a, b)
```

In [98]:
```
add = addition(5,5)
```

```
10 10 3125
10
5 5
```

In [99]:
```
print(add)
```

```
None
```

In [108]:
```
print(range())
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-108-dc14fd2a0e83> in <module>
----> 1 print(range())

TypeError: range expected 1 argument, got 0
```

default argument

In [109]:
```
def addition(a, b = 0):
    return a + b
```

In [110]:
```
print(addition(5,10))
print(addition(5))
```

```
15
5
```

In [111]:
```
print(range(5))
```

```
range(0, 5)
```

Keyword arguments

In [112]:
```
def addition(a, b):
    return a + b
```

In [115]:
```
print(addition(a = 'abc', b = 'def'))
print(addition(b = 'abc', a = 'def'))
```

```
abcdef
defabc
```

In [116]:
```
print(addition('abc', 'def'))
```

```
abcdef
```

In [117]:
```
print(addition(b = 'abc', b = 'def'))
```

```
  File "<ipython-input-117-6f4f287dbb8f>", line 1
    print(addition(b = 'abc', b = 'def'))
                             ^
SyntaxError: keyword argument repeated
```

```
In [118]:  print(addition('def', b = 'abccccc'))
```

```
defabccccc
```

```
In [121]:  print(addition(b = 'abccccc', 'abc'))
```

```
  File "<ipython-input-121-e055ba18ad46>", line 1
    print(addition(b = 'abccccc', 'abc'))
                                 ^
SyntaxError: positional argument follows keyword argument
```

```
In [119]:  print(addition('def', a = 'abccccc'))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-119-aa7fa10b4f33> in <module>
----> 1 print(addition('def', a = 'abccccc'))

TypeError: addition() got multiple values for argument 'a'
```

```
In [120]:  print(addition(d = 'abc', a = 'abc'))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-120-63e559d6dade> in <module>
----> 1 print(addition(d = 'abc', a = 'abc'))

TypeError: addition() got an unexpected keyword argument 'd'
```

```
In [122]:  li = input().split()


           print(li)
```

```
1 2 3 4 5 6
['1', '2', '3', '4', '5', '6']
```

```
In [123]:  li = input()


           print(li)
```

```
['1', '2', '3', '4', '5', '6']
['1', '2', '3', '4', '5', '6']
```

```
In [124]:  li[0]
```

```
Out[124]: '['
```

In [125]:

```python
li = input()


print(li)
```

```
123
['1', '2', '3']
```

In [127]:

```python
li = []
for i in input():
    li.append(i)
```

```
456
```

In [128]:

```python
print(li)
```

```
['4', '5', '6']
```

In [131]:

```python
n=int(input())
li=[]
li.append(n)
for i in range(n):
    li.append(i)

print(li)
```

```
123
[123, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38,
39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57,
58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76,
77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95,
96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111,
112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122]
```

In [129]:

```python
range(55)
```

Out[129]:

```
range(0, 55)
```

Day Outcomes

- Sets and Set Methods in Python
- Functions in Python