```python
# strings - what is string, how we create strings
# '', "", ''''''
# str are sequence of characters
s = '''python'''
# indexing - positive indexing, negative indexing
# positive indexing - starts with 0, left to the right
#negative indexing start with -1 , from the end of the string
# s[0]
# s[2]
# s[-4]
# slicing s[start index   :   end index]  # end is exclusive , end index will not be included
# s[   start index : end index : step ]
# s[ 0 : 4  : 2]  #pt

# upper, lower, title, capitalize, isupper , islower, isdigit, istitle
# isspace, isascii, count, replace, split, join, strip, isalnum

s = '10000a'
s.isdigit()
```

```
False
```

```python
ord('a')
```

```
97
```

```python
bin(97)
```

```
'0b1100001'
```

```python
s = 'python for data science and data analytics'
s.count('data')
```

```
2
```

```python
s.replace('python' , 'SQL')
```

```
'SQL for data science and data analytics'
```

```python
a = s.split()
a
```

```
['python', 'for', 'data', 'science', 'and', 'data', 'analytics']
```

```python
'-'.join(a)
```

```
'python-for-data-science-and-data-analytics'
```

```python
s = '***python***'
s.strip('*')
```

```
'python'
```

```
# index, find
s = 'python'
s.index('y')
```

```
1
```

```
# index, find
s = 'python'
s.find('o')
```

```
4
```

```
# string concatenation
# string self replication
10 + 10
```

```
20
```

```
"python" + "java"  # when we use + on strings , it does string
conncatenation (merge)
```

```
'pythonjava'
```

```
"python" * 3  # self replication
```

```
'pythonpythonpython'
```

```
10 * 2
```

```
20
```

```
s = 'python'
s
```

```
'python'
```

```
len(s)
```

```
6
```

```
# list, tuple, set, dictionary

#list
# What are list ?
# List is a type of data structure in python,
# which can store multiple values of different data types

list_of_products = [ 'Keyboard' ,  'Monitor' ,  'Mouse' , 'Hard Drive'
]

marks_of_students = [ 23, 24, 18, 45 ]

type(list_of_products)
```

```
list
```

```
type(marks_of_students)
```

```
list
```

```
list_1 = []  # the square brackets always creates a list
print(list_1)
```

```
[]
```

```
type(list_1)
```

```
list
```

```
# a list can store any type of values
# str, bool, int, float, complex
```

```
detail_of_s1 = ['John' , 24 , True , 10.5]
detail_of_s1
```

```
['John', 24, True, 10.5]
```

```
detail_of_s1 = [ 'John' , 24 , True , 10.5 , [24, 45, 32]   ]
detail_of_s1
```

```
['John', 24, True, 10.5, [24, 45, 32]]
```

```
# length of list
len(list_of_products)
```

```
4
```

```
len(marks_of_students)
```

```
4
```

```
len(detail_of_s1)
```

```
5
```

```
# indexing and slicing on list
# indexing
list_of_products
```

```
['Keyboard', 'Monitor', 'Mouse', 'Hard Drive']
```

```
list_of_products[2]
```

```
'Mouse'
```

```
list_of_products[-4]
```

```
'Keyboard'
```

```
marks_of_students
```

```
[23, 24, 18, 45]
```

```
marks_of_students[1]
```

```
24
```

```
marks_of_students[-3]
```

```
24
```

```
detail_of_s1
```

```
['John', 24, True, 10.5, [24, 45, 32]]
```

```
detail_of_s1[-1]
```

```
[24, 45, 32]
```

```
detail_of_s1[-2]
```

```
10.5
```

```
# indexing and slicing can be done on str, list, tuple
a = 10  #int , indexing cannot be performed
a[0]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_14820/2569799649.py in <module>
      1 # indexing on list and strings
      2 a = 10
----> 3 a[0]

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

```
list_of_products[0][0]
```

```
'K'
```

```
list_of_products[1][-1]
```

```
'r'
```

```
marks_of_students[0][0]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_14820/2598207722.py in <module>
```

```
----> 1 marks_of_students[0][0]

TypeError: 'int' object is not subscriptable

detail_of_s1[-1][0]

24

detail_of_s1[-1][-1]

32

# list is mutable , strings are immutable
s[0] = 's'

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_14820/3127928240.py in <module>
      1 # list is mutable , strings are immutable
----> 2 s[0] = 's'

TypeError: 'str' object does not support item assignment

marks_of_students

[23, 24, 18, 45]

marks_of_students[0] = 25

marks_of_students

[25, 24, 18, 45]

# list slicing -
detail_of_s1[ 0 : 2 ]

['John', 24]

detail_of_s1

['John', 24, True, 10.5, [24, 45, 32]]

detail_of_s1[2 : 4 ]

[True, 10.5]

detail_of_s1[-3 : -1 ]

[True, 10.5]

detail_of_s1[ : 3]
```

```
['John', 24, True]
```

```
detail_of_s1[ 2 : ]
```

```
[True, 10.5, [24, 45, 32]]
```

```
detail_of_s1[ :  ]
```

```
['John', 24, True, 10.5, [24, 45, 32]]
```

```
detail_of_s1[ :  : 2 ]
```

```
['John', True, [24, 45, 32]]
```

```
detail_of_s1[ :  : -1 ]
```

```
[[24, 45, 32], 10.5, True, 24, 'John']
```

```
# list functions -
#'append', 'clear', 'copy', 'count', 'extend', 'index',
# 'insert', 'pop', 'remove', 'reverse', 'sort'
print(dir(list))
```

```
['__add__', '__class__', '__class_getitem__', '__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']
```

```
list_of_products
```

```
['Keyboard', 'Monitor', 'Mouse', 'Hard Drive']
```

```
# append - append the value at the end of the list
# append is used to insert only 1 value
list_of_products.append('HDMI Cable')
```

```
print(list_of_products)
```

```
['Keyboard', 'Monitor', 'Mouse', 'Hard Drive', 'HDMI Cable']
```

```
marks_of_students
```

```
[25, 24, 18, 45]
```

```
# if we want to insert more than 1 value we will use extend
new_stud = [44,42,32]
```

```
marks_of_students.extend(new_stud)
```

```
print(marks_of_students)
```

```
[25, 24, 18, 45, 44, 42, 32]
```

```
# insert - to insert values at a particular index
list_of_products.insert(2, 'Tablets')
```

```
list_of_products
```

```
['Keyboard', 'Monitor', 'Tablets', 'Mouse', 'Hard Drive', 'HDMI
Cable']
```

```
# remove is used to remove a particular value from the list
list_of_products.remove('Monitor')
```

```
print(list_of_products)
```

```
['Keyboard', 'Tablets', 'Mouse', 'Hard Drive', 'HDMI Cable']
```

```
list_of_products.pop()  # pop by default removes the last value and
also prints it
```

```
'HDMI Cable'
```

```
print(list_of_products)
```

```
['Keyboard', 'Tablets', 'Mouse', 'Hard Drive']
```

```
list_of_products.pop(0) # we can also specify the index value to be
removed
```

```
'Keyboard'
```

```
print(list_of_products)
```

```
['Tablets', 'Mouse', 'Hard Drive']
```

```
# remove - we have to give the value to be removed
# pop - by default removes last value from the list, also we can
specify the index to  be removed
```

```
list_of_products.clear()
```

```
print(list_of_products)
```

```
[]
```

```
# clear - it removes everything from the list
```

```
# copy, count, reverse, sort
marks_of_students
```

```
[25, 24, 18, 45, 44, 42, 32]
```

```
marks_of_students.count(24)
```

```
1
```

```
marks_of_students.sort()
print(marks_of_students)
```

```
[18, 24, 25, 32, 42, 44, 45]
```

```
marks_of_students.sort(reverse=True)
print(marks_of_students)
```

```
[45, 44, 42, 32, 25, 24, 18]
```

```
detail_of_s1
```

```
['John', 24, True, 10.5, [24, 45, 32]]
```

```
detail_of_s1[ : : -1]
```

```
[[24, 45, 32], 10.5, True, 24, 'John']
```

```
detail_of_s1.reverse()
print(detail_of_s1)
```

```
[[24, 45, 32], 10.5, True, 24, 'John']
```

```
#copy
list_1 = [10,14,18]
list_2 = list_1

list_1[0] = 50

list_1
```

```
[50, 14, 18]
```

```
print(list_2)
```

```
[50, 14, 18]
```

```
list_2[-1] = 100

list_2
```

```
[50, 14, 100]
```

```
print(list_1)
```

```
[50, 14, 100]
```

```
#copy
list_1 = [10,14,18]
list_2 = list_1.copy()

list_1[0] = 40

list_1
```

```
[40, 14, 18]
```

```
list_2
```

```
[10, 14, 18]
```

```
#copy
list_1 = [10,14,18]
list_2 = list_1

id(list_1)
```

```
2472471997568
```

```
id(list_2)
```

```
2472471997568
```

```
#copy
list_1 = [10,14,18]
list_2 = list_1.copy()

id(list_1)
```

```
2472472226304
```

```
id(list_2)
```

```
2472470122432
```

```
# append, extend, insert(index, new value)
# remove, pop , clear
# count, sort, copy, reverse

student_names = ['john', 'max', 'steve', 'roy']
student_names.sort()
print(student_names)
```

```
['john', 'max', 'roy', 'steve']
```

```
mixed_list = [10,'john', True]
mixed_list.sort()
print(mixed_list)
```

```
---------------------------------------------------------------------------
-----
```

```
TypeError                                 Traceback (most recent call
last)
~\AppData\Local\Temp/ipykernel_14820/432215814.py in <module>
      1 mixed_list = [10,'john', True]
----> 2 mixed_list.sort()
      3 print(mixed_list)

TypeError: '<' not supported between instances of 'str' and 'int'
```

```
10 < "john"
```

```
---------------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
~\AppData\Local\Temp/ipykernel_14820/112258202.py in <module>
----> 1 10 < "john"

TypeError: '<' not supported between instances of 'int' and 'str'
```

```
# tuple
# tuple is immutable(cannot be changed)
# creating tuple
tuple1 = ()
type(tuple1)
```

```
tuple
```

```
tuple1 = (10, 20, 5, 6, True, 10.6, 'python' , [10,12] , (10,2) )
type(tuple1)
```

```
tuple
```

```
tuple1 = (10, 20, 5, 6, True, 10.6, 'python' , [10,12] , (10,2) )
print(tuple1)
```

```
(10, 20, 5, 6, True, 10.6, 'python', [10, 12], (10, 2))
```

```
tuple1[0]
```

```
10
```

```
tuple1[-9]
```

```
10
```

```
tuple1[0 : 4]
```

```
(10, 20, 5, 6)
```

```
t1 = ("python",)
type(t1)
```

```
tuple
```

```
tuple1[0] = 500
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_14820/464280791.py in <module>
----> 1 tuple1[0] = 500

TypeError: 'tuple' object does not support item assignment
```

```
# count, index
tuple1.count(5)
```

```
1
```

```
tuple1.index("python")
```

```
6
```

```
# + , *
# list concatenation
list_1
```

```
[10, 14, 18]
```

```
mixed_list
```

```
[10, 'john', True]
```

```
list_1 + mixed_list
```

```
[10, 14, 18, 10, 'john', True]
```

```
list_1 * 3
```

```
[10, 14, 18, 10, 14, 18, 10, 14, 18]
```

```
# tuple concatenation
tuple1
```

```
(10, 20, 5, 6, True, 10.6, 'python', [10, 12], (10, 2))
```

```
t1
```

```
('python',)
```

```
tuple1 + t1
```

```
(10, 20, 5, 6, True, 10.6, 'python', [10, 12], (10, 2), 'python')
```

```
t1 * 3
```

```
('python', 'python', 'python')
```

```
len(tuple1)
```

```
9
```

```
len(t1)
```

```
1
```

```
# min, max, sum, sorted
list_1
```

```
[10, 14, 18]
```

```
min(list_1)
```

```
10
```

```
max(list_1)
```

```
18
```

```
sum(list_1)
```

```
42
```

```
sorted(list_1, reverse = True)
```

```
[18, 14, 10]
```

```
t2 = (10,7,12)
min(t2)
```

```
7
```

```
max(t2)
```

```
12
```

```
sum(t2)
```

```
29
```

```
sorted(t2)  # sorted will always return the output in list form
```

```
[7, 10, 12]
```

```
sorted(t2, reverse = True)
```

```
[12, 10, 7]
```

```
min(mixed_list)
```

```
---------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
~\AppData\Local\Temp/ipykernel_14820/2204442338.py in <module>
----> 1 min(mixed_list)

TypeError: '<' not supported between instances of 'str' and 'int'
```

```
max(mixed_list)
```

```
---------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
~\AppData\Local\Temp/ipykernel_14820/2291670481.py in <module>
----> 1 max(mixed_list)

TypeError: '>' not supported between instances of 'str' and 'int'
```

```
sorted(mixed_list)
```

```
---------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
~\AppData\Local\Temp/ipykernel_14820/3885089925.py in <module>
----> 1 sorted(mixed_list)

TypeError: '<' not supported between instances of 'str' and 'int'
```

```
# dictionaries
# dict - is a python data structure which stores the data in form of
key value pair
dictonary_1 = {}
type(dictonary_1)
```

```
dict
```

```
dictonary_2 = {"John" :  24, "Roy" : 32 }
type(dictonary_2)
```

```
dict
```

```
dictonary_2
```

```
{'John': 24, 'Roy': 32}
```

```
# what can be the keys in dict?
# Only immutable data types can be the key
```

```
dictonary_3 = {101  : 24, 102 : 45}
dictonary_3
```

```
{101: 24, 102: 45}
```

```
dictonary_3 = { 1.1  : 'Python 101', 1.2 : 'Advanced python'}
dictonary_3
```

```
{1.1: 'Python 101', 1.2: 'Advanced python'}
```

```
dictonary_3 = { [0,1]  : 'Python 101', [0,1]: 'Advanced python'}
dictonary_3
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_14820/2401943406.py in <module>
----> 1 dictonary_3 = { [0,1]  : 'Python 101', [0,1]: 'Advanced python'}
      2 dictonary_3

TypeError: unhashable type: 'list'
```

```
# only immutable data types can be the key in the dict
```

```
dictonary_3 = { (0,1)  : 'Python 101', (0,2): 'Advanced python'}
dictonary_3
```

```
{(0, 1): 'Python 101', (0, 2): 'Advanced python'}
```

```
# What can be the values in the dict? values can be any data type
dictonary_4 = {"Academy Name" : "ArivUpro Academy",
               'Institute Age' : 7,
               'Course Offered' : ["Data Analytics" , "CA", "CFA"],
               "Students Placed" : {"year 1" : 40, "year_2" : 120}
               }
dictonary_4
```

```
{'Academy Name': 'ArivUpro Academy',
 'Institute Age': 7,
 'Course Offered': ['Data Analytics', 'CA', 'CFA'],
 'Students Placed': {'year 1': 40, 'year_2': 120}}
```

```
# What can be the values in the dict? values can be any data type
dictonary_4 = {"Academy Name" : "ArivUpro Academy",
               'Institute Age' : 7,
               'Course Offered' : ["Data Analytics" , "CA", "CFA"],
               "Students Placed" : {"year 1" : 40, "year_2" : 120},
                "Institute Age" : 8
               }
dictonary_4
```

```
{'Academy Name': 'ArivUpro Academy',
 'Institute Age': 8,
 'Course Offered': ['Data Analytics', 'CA', 'CFA'],
 'Students Placed': {'year 1': 40, 'year_2': 120}}

# When we have duplicate keys only the last value is stored

# on dict we cannot perform indexing and slicing
dictonary_4[0]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_14820/1467084134.py in <module>
      1 # on dict we cannot perform indexing and slicing
----> 2 dictonary_4[0]

KeyError: 0
```

```
dictonary_4["Academy Name"]
```

```
'ArivUpro Academy'
```

```
dictonary_2
```

```
{'John': 24, 'Roy': 32}
```

```
dictonary_2['Roy']
```

```
32
```

```
dictonary_4['Course Offered']
```

```
['Data Analytics', 'CA', 'CFA']
```

```
dictonary_2['Roy'] = 40
```

```
dictonary_2
```

```
{'John': 24, 'Roy': 40}
```

```
dictonary_2['steve']
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_14820/3948404455.py in <module>
----> 1 dictonary_2['steve']

KeyError: 'steve'
```

```
dictonary_4
```

```
{'Academy Name': 'ArivUpro Academy',
 'Institute Age': 8,
 'Course Offered': ['Data Analytics', 'CA', 'CFA'],
 'Students Placed': {'year 1': 40, 'year_2': 120}}
```

```
dictonary_4['Course Offered'].append('Data Science')
```

```
dictonary_4
```

```
{'Academy Name': 'ArivUpro Academy',
 'Institute Age': 8,
 'Course Offered': ['Data Analytics', 'CA', 'CFA', 'Data Science'],
 'Students Placed': {'year 1': 40, 'year_2': 120}}
```

```
# keys, values, items, get, popitem, pop
dictonary_4.keys()
```

```
dict_keys(['Academy Name', 'Institute Age', 'Course Offered',
'Students Placed'])
```

```
dictonary_4.values()
```

```
dict_values(['ArivUpro Academy', 8, ['Data Analytics', 'CA', 'CFA',
'Data Science'], {'year 1': 40, 'year_2': 120}])
```

```
dictonary_4.items()
```

```
dict_items([('Academy Name', 'ArivUpro Academy'), ('Institute Age',
8), ('Course Offered', ['Data Analytics', 'CA', 'CFA', 'Data
Science']), ('Students Placed', {'year 1': 40, 'year_2': 120})])
```

```
# ('Academy Name', 'ArivUpro Academy')
# ('Institute Age', 8)
#  ('Course Offered', ['Data Analytics', 'CA', 'CFA', 'Data Science'])
```

```
dictonary_4['Academy Name']
```

```
'ArivUpro Academy'
```

```
dictonary_4.get('Academy Name')
```

```
'ArivUpro Academy'
```

```
dictonary_4['Year Est.']
```

```
---------------------------------------------------------------------
-----
KeyError                                  Traceback (most recent call
last)
~\AppData\Local\Temp/ipykernel_14820/350487931.py in <module>
----> 1 dictonary_4['Year Est.']
```

```
KeyError: 'Year Est.'
```

```
dictonary_4.get('Year Est.'  ,  2018 )
```

```
2018
```

```
dictonary_4.get('Academy Name'  ,  'Key not found' )
```

```
'ArivUpro Academy'
```

```
# keys, values, items, get, pop
dictonary_4.pop("Academy Name")
```

```
'ArivUpro Academy'
```

```
dictonary_4
```

```
{'Institute Age': 8,
 'Course Offered': ['Data Analytics', 'CA', 'CFA', 'Data Science'],
 'Students Placed': {'year 1': 40, 'year_2': 120}}
```

```
dictonary_2.pop('John')
```

```
24
```

```
dictonary_2
```

```
{'Roy': 40}
```

```
# popitem
```

```
dictonary_4
```

```
{'Institute Age': 8,
 'Course Offered': ['Data Analytics', 'CA', 'CFA', 'Data Science'],
 'Students Placed': {'year 1': 40, 'year_2': 120}}
```

```
dictonary_4.popitem()  # by default removes last key and value pair
```

```
('Students Placed', {'year 1': 40, 'year_2': 120})
```

```
dictonary_4
```

```
{'Institute Age': 8,
 'Course Offered': ['Data Analytics', 'CA', 'CFA', 'Data Science']}
```

```
# adding new key and value in dict
dictonary_4['Institute Age'] = 7
dictonary_4
```

```
{'Institute Age': 7,
 'Course Offered': ['Data Analytics', 'CA', 'CFA', 'Data Science']}

dictonary_4['Address'] = 'Kormangla, Bengaluru'

dictonary_4

{'Institute Age': 7,
 'Course Offered': ['Data Analytics', 'CA', 'CFA', 'Data Science'],
 'Address': 'Kormangla, Bengaluru'}

dictonary_4['State'] = 'Karnataka'

dictonary_4

{'Institute Age': 7,
 'Course Offered': ['Data Analytics', 'CA', 'CFA', 'Data Science'],
 'Address': 'Kormangla, Bengaluru',
 'State': 'Karnataka'}

# update
dictonary_4['Institute Age'] = 10

dictonary_4

{'Institute Age': 10,
 'Course Offered': ['Data Analytics', 'CA', 'CFA', 'Data Science'],
 'Address': 'Kormangla, Bengaluru',
 'State': 'Karnataka'}

dictonary_4.update( {'Address' : 'HSR,Bengaluru', 'Institute Age' :
7 } )

dictonary_4

{'Institute Age': 7,
 'Course Offered': ['Data Analytics', 'CA', 'CFA', 'Data Science'],
 'Address': 'HSR,Bengaluru',
 'State': 'Karnataka'}

# clear
dictonary_4.clear()

dictonary_4

{}

# copy
d_1 = {"A" : 10,
       "B" : 20}
d_2 = d_1

d_2
```

```
{'A': 10, 'B': 20}
```

```
d_1
```

```
{'A': 10, 'B': 20}
```

```
d_1['A'] = 100
d_1
```

```
{'A': 100, 'B': 20}
```

```
d_2
```

```
{'A': 100, 'B': 20}
```

```
# copy
d_1 = {"A" : 10,
       "B" : 20}
d_2 = d_1.copy()
d_1
```

```
{'A': 10, 'B': 20}
```

```
d_2
```

```
{'A': 10, 'B': 20}
```

```
d_1['A'] = 100
d_1
```

```
{'A': 100, 'B': 20}
```

```
d_2
```

```
{'A': 10, 'B': 20}
```

```
# fromkeys
stud_name = ['john', 'roy' , 'steve']
```

```
dict.fromkeys(stud_name)
```

```
{'john': None, 'roy': None, 'steve': None}
```

```
# fromkeys
stud_name = ['john', 'roy' , 'steve']
```

```
dict.fromkeys(stud_name, 0)
```

```
{'john': 0, 'roy': 0, 'steve': 0}
```

```
len(dictonary_3)
```

```
2
```