

Tuple

```
In [ ]: tuple1 = (12,15,16,7)
print(tuple1[1])
print(tuple1[3])
```

```
In [5]: tuple1 = (12,15,16,20)
print( len(tuple1) )
```

4

```
In [6]: tuple1 = (23,45,7,23)
for item in (tuple1):
    print(item)
```

23
45
7
23

```
In [8]: tuple1 = (12,34,22,32)
index=0
while index<len(tuple1):
    print(tuple1[index])
    index = index+1
```

12
34
22
32

```
In [11]: my_tuple = (5 , 6 , 8)
print(my_tuple)
```

(5, 6, 8)

```
In [12]: my_tuple = (1, "WELCOME TO KPI", 2.3)
print(my_tuple)
```

(1, 'WELCOME TO KPI', 2.3)

) print(my_tuple)

```
In [14]: my_tuple = 3, 4.6, "dog"
print(my_tuple)
```

```
(3, 4.6, 'dog')
```

```
In [15]: a, b, c = my_tuple

print(a)      # 3
print(b)      # 4.6
print(c)      # dog
```

```
3
4.6
dog
```

```
In [16]: my_tuple = ("kpi")
print(type(my_tuple))
```

```
<class 'str'>
```

```
In [17]: my_tuple = ("kpi",)
print(type(my_tuple))
```

```
<class 'tuple'>
```

```
In [19]: my_tuple = "hello",
print(type(my_tuple))
```

```
<class 'tuple'>
```

```
In [21]: my_tuple = ('s','e','s','s','i','o','n')

print(my_tuple[4])
print(my_tuple[5])
print(my_tuple[6])
```

```
i
o
n
```

```
In [24]: n_tuple = ("honeybee", [8, 4, 6], (1, 2, 3))
print(n_tuple[0][3])
print(n_tuple[1][1])
```

```
e
4
```

```
In [25]: my_tuple = ('f', 'u', 'n', 'c', 't', 'i', 'o', 'n')
print(my_tuple[1:4])
```

```
('u', 'n', 'c')
```

```
In [26]: print(my_tuple[:-4])
```

```
('f', 'u', 'n', 'c')
```

```
In [27]: print(my_tuple[6:])
```

```
('o', 'n')
```

```
In [28]: print(my_tuple[:])
```

```
('f', 'u', 'n', 'c', 't', 'i', 'o', 'n')
```

```
In [2]: my_tuple = ('m', 'a', 'n', 'g', 'o',)
print('a' in my_tuple)
print('o' in my_tuple)
```

```
True
True
```

```
In [ ]:
```

Dictionary

```
In [47]: my_dict = {1: 'grapes', 2: 'bat'}
```

```
In [48]: my_dict = {'name': 'sai', 1: [2, 4, 3]}
```

```
In [44]: my_dict = {'name': 'vamsi', 1: [3, 4, 3]}
```

```
In [45]: my_dict = dict([(1,'apple'), (2,'ball')])
```

```
In [1]: my_dict = {'name': 'Jack', 'age': 26}
```

```
In [49]: print(my_dict.get('age'))
```

```
None
```

```
In [50]: my_dict = {'name': 'Jack', 'age': 26}
```

```
In [54]: print(my_dict['address'])
```

```
KeyError
```

```
Traceback (most recent call last)
```

```
~\AppData\Local\Temp\ipykernel_3056/1422516528.py in <module>
```

```
----> 1 print(my_dict['address'])
```

```
KeyError: 'address'
```

```
In [53]: print(my_dict.get('address'))
```

```
None
```

```
In [55]: my_dict = {'name': 'Jack', 'age': 26}
```

```
In [2]: my_dict['age'] = 27
```

```
In [3]: print(my_dict)
```

```
{'name': 'Jack', 'age': 27}
```

```
In [4]: my_dict['address'] = 'Downtown'
```

```
In [5]: print(my_dict)
```

```
{'name': 'Jack', 'age': 27, 'address': 'Downtown'}
```

```
In [6]: squares = {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
```

```
In [7]: print(squares.pop(4))
```

```
16
```

```
In [8]: print(squares)
```

```
{1: 1, 2: 4, 3: 9, 5: 25}
```

```
In [9]: print(squares.popitem())
```

```
(5, 25)
```

```
In [10]: print(squares)
```

```
{1: 1, 2: 4, 3: 9}
```

```
In [11]: print(squares)
```

```
{1: 1, 2: 4, 3: 9}
```

```
In [3]: squares = {1: 1, 3: 9, 5: 25, 7: 49, 9: 81}
```

```
In [4]: print(7 in squares)
```

```
True
```

```
In [5]: squares = {0: 0, 1: 1, 5: 9, 5: 25, 6: 49, 9: 81}
```

```
In [6]: print(all(squares))
```

```
False
```

```
In [7]: print(any(squares))
```

```
True
```

```
In [8]: print(len(squares))
```

```
5
```

```
In [9]: print(sorted(squares))
```

```
[0, 1, 5, 6, 9]
```

SETS

```
In [11]: my_set = {4, 5, 8}  
print(my_set)
```

```
{8, 4, 5}
```

```
In [13]: my_set = {1, "welcome to kpi", (1, 2, 3)}  
print(my_set)
```

```
{1, (1, 2, 3), 'welcome to kpi'}
```

```
In [14]: my_set = {23,23,24,76,76,21,25}  
print(my_set)
```

```
{21, 23, 24, 25, 76}
```

```
In [15]: my_set = set([2, 4, 5, 5])           // sets from the list  
print(my_set)
```

```
{2, 4, 5}
```

```
In [19]: a = {}  
print(type(a))
```

```
<class 'dict'>
```

```
In [20]: a = ()  
print(type(a))
```

```
<class 'tuple'>
```

```
print(my_set) my_set[0]
```

```
In [26]: my_set.add(4)  
print(my_set)
```

```
{1, 2, 3, 4}
```

```
In [28]: my_set.update([4,6,8])  
print(my_set)
```

```
{1, 2, 3, 4, 6, 8}
```

lambda

```
In [42]: square = lambda a: a*a
result = square(99)
print(result)
```

9801

```
In [31]: mul = lambda a,b: a*b
result = mul(7,9)
print(result)
```

63

```
In [33]: six = lambda : 8
result = six()
print(result)
```

8

```
In [37]: fact = lambda a: a*fact(a-1) if (a>1) else 1
result = fact(9)
print(result)
```

362880

```
In [38]: import math
def myfunc(n):
    return lambda a : math.pow(a, n)

square = myfunc(3)
cube = myfunc(5)
squareroot = myfunc(0.8)
print(square(3))
print(cube(3))
print(squareroot(3))
```

27.0

243.0

2.4082246852806923

```
In [39]: my_list = [1, 5, 4, 6, 8, 11, 3, 12]

new_list = list(map(lambda x: x * 2 , my_list))

print(new_list)
```

```
[2, 10, 8, 12, 16, 22, 6, 24]
```

functions

```
In [40]: def call(i):
    return i
print(callable(call))
```

```
True
```

```
In [48]: def foo():
    pass
print(type(foo))
print("foo name attribute: ",foo.__name__)
```

```
<class 'function'>
foo name attribute: foo
```

```
In [45]: z = lambda x:x*2
print(z(5))
lst = [1,5,7,14]
newlst = list(filter(lambda a:a%7==0,lst))
print(newlst)
```

```
10
[7, 14]
```

methods

```
In [47]: class Abc:
    def m1(self):
        pass
```

```
In [50]: def __init__(self):
    pass
```

```
In [55]: class sample:  
    def __init__(self):  
        self.a = 25  
        self.b = 20  
s = sample()  
print(s.a)
```

25

```
In [54]: class sample:  
    def __call__(self):  
        pass  
s = sample()  
print(callable(s))
```

True

functions and closures

```
In [56]: def local_function():  
    print("This is a local function")  
local_function()
```

This is a local function

```
In [62]: def a():  
    def b():  
        print("inside b")  
    print("inside a")  
    b()  
a()
```

inside a
inside b

```
In [64]: def outer_func():  
    def inner_func():  
        print("This is a nested function")  
    inner_func()  
outer_func()
```

This is a nested function

```
In [67]: def outer_func(a):
    def inner_func():
        print("This is a nested function",a)
    inner_func()
outer_func('hi!')
```

This is a nested function 'hi!'

```
In [69]: def mno(x):
    return x**2
def pqr(func):
    num=20
    return func(num)
pqr(mno)
```

Out[69]: 400

In []:

```
In [16]: def mno(x):
    return x**5
def pqr(func):
    num=50
    return func(num)
pqr(mno)
```

Out[16]: 312500000

sorting

```
In [41]: def merge_sort(unsorted_list):
    if len(unsorted_list) <= 1:
        return unsorted_list

    middle = len(unsorted_list) // 2
    left_list = unsorted_list[:middle]
    right_list = unsorted_list[middle:]

    left_list = merge_sort(left_list)
    right_list = merge_sort(right_list)
    return list(merge(left_list, right_list))

def merge(left_half,right_half):
    res = []
    while len(left_half) != 0 and len(right_half) != 0:
        if left_half[0] < right_half[0]:
            res.append(left_half[0])
            left_half.remove(left_half[0])
        else:
            res.append(right_half[0])
            right_half.remove(right_half[0])
    if len(left_half) == 0:
        res = res + right_half
    else:
        res = res + left_half
    return res

unsorted_list = [64, 34, 25, 12, 22, 11, 90]
print(merge_sort(unsorted_list))
#print(unsorted_list)
```

```
-----  
TypeError                                     Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_32/176996967.py in <module>  
      27     return res  
      28 unsorted_list = [64, 34, 25, 12, 22, 11, 90]  
----> 29 print(merge_sort(unsorted_list))  
      30 #print(unsorted_list)  
  
~\AppData\Local\Temp\ipykernel_32/176996967.py in merge_sort(unsorted_list)  
      7     right_list = unsorted_list[middle:]  
      8  
----> 9     left_list = merge_sort(left_list)  
     10     right_list = merge_sort(right_list)  
     11     return list(merge(left_list, right_list))  
  
~\AppData\Local\Temp\ipykernel_32/176996967.py in merge_sort(unsorted_list)  
      8  
      9     left_list = merge_sort(left_list)  
----> 10    right_list = merge_sort(right_list)  
     11    return list(merge(left_list, right_list))  
     12  
  
~\AppData\Local\Temp\ipykernel_32/176996967.py in merge_sort(unsorted_list)  
      9     left_list = merge_sort(left_list)
```

```
10     right_list = merge_sort(right_list)
---> 11     return list(merge(left_list, right_list))
12
13
```

TypeError: 'list' object is not callable

```
In [42]: def insertion_sort(InputList):
    for i in range(1, len(InputList)):
        j = i-1
        nxt_element = InputList[i]
    # Compare the current element with next one
        while (InputList[j] > nxt_element) and (j >= 0):
            InputList[j+1] = InputList[j]
            j=j-1
            InputList[j+1] = nxt_element
list = [19,2,31,45,30,11,121,27]
insertion_sort(list)
print(list)
```

[19, 2, 31, 45, 30, 11, 27, 121]

```
In [39]: def bubblesort(list):

    for iter_num in range(len(list)-1,0,-1):
        for idx in range(iter_num):
            if list[idx]>list[idx+1]:
                temp = list[idx]
                list[idx] = list[idx+1]
                list[idx+1] = temp
list = [19,2,31,45,6,11,121,27]
bubblesort(list)
print(list)
```

[2, 6, 11, 19, 27, 31, 45, 121]

```
In [16]: def linear_search(values, search_for):
    search_at = 0
    search_res = False

    while search_at < len(values) and search_res is False:
        if values[search_at] == search_for:
            search_res = True
        else:
            search_at = search_at + 1
    return search_res
l = [64, 34, 25, 12, 22, 11, 90]
print(linear_search(l, 12))
print(linear_search(l, 91))
```

True
False

range

```
In [2]: for i in range(20):
    print(i, end = " ")
```

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

```
In [3]: for i in range(1, 2, 4):
    print(i, end = " ")
```

1

directory

```
In [7]: import os

with os.scandir('my_directory/') as entries:
    for entry in entries:
        print(entry.name)
```

```
-----
FileNotFoundError                                     Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_8636\1030001133.py in <module>
      1 import os
      2
----> 3 with os.scandir('my_directory/') as entries:
      4     for entry in entries:
      5         print(entry.name)

FileNotFoundError: [WinError 3] The system cannot find the path specified: 'my_directory/'
```

```
In [10]: import os

entries = os.listdir

for x in entries:
    print(x)
```

```
-----
TypeError                                         Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_8636\3027327760.py in <module>
      3 entries = os.listdir
      4
----> 5 for x in entries:
      6     print(x)

TypeError: 'builtin_function_or_method' object is not iterable
```

```
In [9]: os.makedirs('./monisha',exit_ok=true)
```

```
-----
NameError                                         Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_8636\1462565001.py in <module>
----> 1 Os.makedirs('./monisha',exit_ok=true)

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

```
In [11]: import os
```

```
In [14]: entries = os.listdir('my_directory/')
```

```
-----  
FileNotFoundException                                     Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_8636\559847432.py in <module>  
----> 1 os.listdir('./data')  
      2  
  
FileNotFoundException: [WinError 3] The system cannot find the path specified: './d  
ata'
```

Decorators

```
In [66]: def addexclamation(function):  
    def add():  
        func = function()  
        return func + " !!!"  
    return add  
def sentence():  
    return "hello all"  
msg = addexclamation(sentence)  
print(msg())
```

```
hello all !!!
```

```
In [61]: def addexclamation(function):  
    def add():  
        func = function()  
        return func + " !!!"  
    return add  
@addexclamation  
def sentence():  
    return "hello all"  
print(sentence())
```

```
hello all !!!
```

```
In [72]: def addstar(func):
    def star():
        return "*" + func() + "*"
    return star
@addstar
@addexclamation
def sentence():
    return "welcome to kpi"
print(sentence())
```

welcome to kpi !!!

```
In [91]: def args_function(func):
    def getargs(arg1,arg2):
        print(arg1,arg2)
        func(arg1,arg2)
    return getargs
@args_function
def decorator_with_args(num1,num2):
    print("arguments are {} and {}".format(num1,num2))
decorator_with_args(10,11)
```

10 11
arguments are 10 and 11

Comprehensions

```
In [93]: list1 = [x**10 for x in range(20)]
print(list1)
```

[0, 1, 1024, 59049, 1048576, 9765625, 60466176, 282475249, 1073741824, 34867844
01, 10000000000, 25937424601, 61917364224, 137858491849, 289254654976, 57665039
0625, 1099511627776, 2015993900449, 3570467226624, 6131066257801]

```
In [94]: list2 = [5,2,7,8,14,22]
list3 = [x for x in list2 if x%2==0]
print(list3)
```

[2, 8, 14, 22]

```
In [95]: x = lambda a:a**2
print(x(5))

y = lambda a,b:a-b
print(y(10,6))
```

25
4

```
In [96]: l = [1,2,3,4,5,6,7]
newl = list(filter(lambda x:x%2==0,l))
print(newl)
```

[2, 4, 6]

```
In [97]: num = []
for i in range(10):
    num.append(i)
newnum = list(map(lambda x:x+2,num))
print(newnum)
```

[2, 3, 4, 5, 6, 7, 8, 9, 10, 11]

```
In [98]: s='1 2 3 4'
l=s.split()
m=list(map(int,l))
n=list(map(lambda x:x**3,m))
o=list(filter(lambda x:(x%2==0),n))
print(o)
```

[8, 64]

```
In [99]: lst = [2,5,6,3,8]
dict = {x:x+5 for x in lst if x%2==0}
print(dict)
```

{2: 7, 6: 11, 8: 13}

```
In [100]: l1 = [1,2,3]
l2 = [5,8,7]
dict1 = {key:value for (key,value) in zip(l1, l2)}
print(dict1)
```

{1: 5, 2: 8, 3: 7}

```
In [104]: dict = {'jack': 38, 'michael': 48, 'guido': 57, 'sai': 33}
new_dict = {k: v for (k, v) in dict.items() if v%2!=0 if v<40}
print(new_dict)
```

{'sai': 33}

```
In [105]: dct = {'jack': 38, 'michael': 48, 'guido': 57, 'sai': 33}
new_dct = {k: ('old' if v > 40 else 'young')
           for (k, v) in dct.items()}
print(new_dct)
```

{'jack': 'young', 'michael': 'old', 'guido': 'old', 'sai': 'young'}

```
In [106]: dictionary = {k1: {k2: k1 * k2 for k2 in range(1, 6)} for k1 in range(2, 5)}
print(dictionary)
```

```
{2: {1: 2, 2: 4, 3: 6, 4: 8, 5: 10}, 3: {1: 3, 2: 6, 3: 9, 4: 12, 5: 15}, 4:
{1: 4, 2: 8, 3: 12, 4: 16, 5: 20}}
```

```
In [107]: lst = [1,2,3,2,5,3]
set1 = {x for x in lst}
print(set1)
```

```
{1, 2, 3, 5}
```

```
In [108]: input_list = [1, 2, 3, 4, 4, 5, 6, 6, 6, 7, 7]
set2 = {var for var in input_list if var % 2 == 0}
print(set2)
```

```
{2, 4, 6}
```

```
In [109]: lst = [2,7,5,0,4,6]
gen = (x for x in lst if x%3==0)
for i in gen:
    print(i)
```

```
0
6
```

```
In [110]: gen2 = (i**2 for i in range(5))
for item in gen2:
    print(item)
```

```
0
1
4
9
16
```

Object-orientation programming

```
In [113]: class Abc:
    pass
```

```
In [116]: class College:  
    def __init__(self):  
        print("Welcome")  
  
    s1 = College()
```

Welcome

```
In [118]: class Values:  
    def __init__(self):  
        self.a = 13  
        self.b = 15  
    s = Values()  
    print(s.a)  
    print(s.b)  
    s.b = 25  
    print(s.b)
```

13
15
25

```
In [120]: class Values2:  
    def __init__(hi,a,b):  
        hi.a = a  
        hi.b = b  
    s = Values2(int(input()),int(input()))  
    print(s.a,s.b)
```

23
23
23 23

```
In [130]: class Student:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def printname(hi):
        print("my name is :" + hi.name)
S1 = Student("barathi", 22)
print(S1.name)
print(S1.age)

S2 = Student("surya", 20)
S2.printname()

del S2
print(S2.name)

del S1.name
print(S1.name)
```

barathi
22

```
-----  
AttributeError                                     Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_8636\2152120122.py in <module>  
      11  
      12     S2 = Student("surya", 20)  
----> 13     S2.printname()  
      14  
      15 del S2
```

AttributeError: 'Student' object has no attribute 'printname'

```
In [137]: class Campus():
    def __init__(self, code, name):
        self.code = code
        self.name = name

    def show(self):
        print(self.code, self.name)
c = Campus(123, 'ASIET')
c.show()
```

123 ASIET

```
In [138]: class Student1(Campus):
    pass
s = Student1(456, 'ABC')
s.show()
```

456 ABC

```
In [141]: class Student2(Campus):  
  
    def __init__(self,code,name,address): #using init in child overrides parents init  
  
        Campus.__init__(self,code,name)  
        self.address = address  
  
    def all3(self):  
        print(self.code,self.name,self.address)  
  
st = Student2(156,'QWE','kerala')  
st.show()  
st.all3()
```

```
156 QWE  
156 QWE kerala
```

```
In [146]: class A:  
    def __init__(self):  
        self.str1="hi"  
    print("A")  
class B:  
    def __init__(self):  
        self.str2 = "hello"  
    print("B")  
  
class C(A,B):  
    def __init__(self):  
        A.__init__(self)  
        B.__init__(self)  
        self.str3 = "bye"  
    print("C")  
    def printstr(self):  
        print(self.str1,self.str2,self.str3)  
  
cobj = C()  
cobj.printstr()
```

```
A  
B  
C  
hi hello bye
```

In [148]:

```
class A:  
    def __init__(self,num1):  
        self.num1=num1  
    def printnum1(self):  
        print(self.num1)  
class B(A):  
    def __init__(self,num1,num2):  
        A.__init__(self,num1)  
        self.num2 = num2  
    def printnum2(self):  
        print(self.num2)  
  
class C(B):  
    def __init__(self,num1,num2,num3):  
        B.__init__(self,num1,num2)  
        self.num3 = num3  
    def printnum3(self):  
        print(self.num3)  
  
cobj = C(1,2,3)  
cobj.printnum1()  
cobj.printnum2()  
cobj.printnum3()
```

```
1  
2  
3
```

In [153]:

```
class MyDecorator:  
    def __init__(self,function):  
        self.function = function  
  
    def __call__(self): #instance called in the form of function->call is executed  
        var = self.function()  
        print(var,"all")  
  
@MyDecorator  
def function():  
    return "Hi"  
  
function()
```

```
Hi all
```

```
In [154]: class MyDecorator:
    def __init__(self, function):
        self.function = function

    def __call__(self, *args, **kwargs):
        self.function(*args, **kwargs)

@MyDecorator
def function(name, message ='Hello'):
    print("{} {}".format(message, name))
function("arathi")
function("arathi","Hi")
```

Hello arathi!
Hi arathi!

```
In [157]: class CubeDecorator:

    def __init__(self, function):
        self.function = function

    def __call__(self, *args, **kwargs):
        result = self.function(*args, **kwargs)
        return result

@CubeDecorator
def get_cube(n):
    print("given number is:", n)
    return n*n*n

print("Cube of number is:", get_cube(5))
```

given number is: 5
Cube of number is: 125

```
In [159]: class Makeupper:
    def __init__(self,func):
        self.func = func

    def __call__(self):
        string = self.func()
        return string.upper()

@Makeupper
def enter_str():
    return "arathi"
enter_str()
```

Out[159]: 'ARATHI'

Iterators

```
In [160]: my_list = [4, 7, 0, 3]
# get an iterator using iter()
my_iter = iter(my_list)
# iterate through it using next()
# Output: 4
print(next(my_iter))
# Output: 7
print(next(my_iter))
# next(obj) is same as obj.__next__()
# Output: 0
print(my_iter.__next__())
# Output: 3
print(my_iter.__next__())
# This will raise error, no items left
next(my_iter)
```

4
7
0
3

```
StopIteration                                     Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_8636\1654642474.py in <module>
      13 print(my_iter.__next__())
      14 # This will raise error, no items left
---> 15 next(my_iter)

StopIteration:
```

```
In [161]: print(dir(my_iter))
```

```
['__class__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__',
 '__getattribute__', '__gt__', '__hash__', '__init__', '__init_subclass__',
 '__iter__', '__le__', '__length_hint__', '__lt__', '__ne__', '__new__', '__next__',
 '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__setstate__',
 '__sizeof__', '__str__', '__subclasshook__']
```

```
In [162]: for city in ["Berlin", "Vienna", "Zurich"]:
    print(city)
print("\n")
for city in ("Python", "Perl", "Ruby"):
    print(city)
for char in "Iteration is easy":
    print(char, end = " ")
```

Berlin
Vienna
Zurich

Python
Perl
Ruby
I t e r a t i o n i s e a s y

```
In [164]: cities = ["Berlin", "Vienna", "Zurich"]
# initialize the object
iterator_obj = iter(cities)
print(next(iterator_obj))
print(next(iterator_obj))
print(next(iterator_obj))
```

Berlin
Vienna
Zurich

```
In [168]: def iterable(obj):
    try:
        iter(obj)
        return True

    except TypeError:
        return False

#Driver Code
for element in [34, [4, 5], (4, 5),
{'a':4}, "dfsdf", 4.5]:

    print(element, " is iterable : ", iterable(element))
```

34 is iterable : False
[4, 5] is iterable : True
(4, 5) is iterable : True
{'a': 4} is iterable : True
dfsdf is iterable : True
4.5 is iterable : False

collections

```
In [169]: from collections import Counter
lst=[1,3,2,5,7,9,3,2,6,9]
Counter(lst)
```

```
Out[169]: Counter({1: 1, 3: 2, 2: 2, 5: 1, 7: 1, 9: 2, 6: 1})
```

```
In [170]: cnt = Counter(lst)
print(cnt[1])
print(cnt.most_common())
```

```
1
[(3, 2), (2, 2), (9, 2), (1, 1), (5, 1), (7, 1), (6, 1)]
```

```
In [171]: deduct={1:1,4:1,2:1}
cnt.subtract(deduct)
print(cnt)
```

```
Counter({3: 2, 9: 2, 2: 1, 5: 1, 7: 1, 6: 1, 1: 0, 4: -1})
```

```
In [172]: from collections import namedtuple
a = namedtuple('details','name, age')
s = a('barathi',22)
print(s)
```

```
details(name='barathi', age=22)
```

```
In [173]: s = a._make(['abcd',21])
print(s)
```

```
details(name='abcd', age=21)
```

```
In [174]: from collections import namedtuple
a=namedtuple('courses','name,technology')
s=a('data science','python')
print(s)
```

```
courses(name='data science', technology='python')
```

```
In [175]: for i in s:
    print(i)
```

```
data science
python
```

```
In [176]: from collections import deque
lst = ['a','e','i','o','u']
dequelist = deque(lst)
print(dequelist)
```

```
deque(['a', 'e', 'i', 'o', 'u'])
```

```
In [177]: dequelist.append('b')
print(dequelist)
```

```
deque(['a', 'e', 'i', 'o', 'u', 'b'])
```

```
In [178]: dequelist.appendleft('z')
print(dequelist)
```

```
deque(['z', 'a', 'e', 'i', 'o', 'u', 'b'])
```

```
In [179]: dequelist.pop()
print(dequelist)
```

```
deque(['z', 'a', 'e', 'i', 'o', 'u'])
```

```
In [188]: itr=iter(a)
while True:
    try:
        print(next(itr),end=' ')
    except StopIteration:
        break
```

TypeError

Traceback (most recent call last)

~\AppData\Local\Temp\ipykernel_8636/2322106475.py in <module>

```
----> 1 itr=iter(a)
      2 while True:
      3     try:
      4         print(next(itr),end=' ')
      5     except StopIteration:
```

TypeError: 'type' object is not iterable

```
In [189]: from collections import ChainMap
a={1:'m',2:'n'}
b={3:'x',4:'y'}
d=ChainMap(a,b)
print(d)
```

```
ChainMap({1: 'm', 2: 'n'}, {3: 'x', 4: 'y'})
```

```
In [190]: for i in d.items():
    print(i)
```

```
(3, 'x')
(4, 'y')
(1, 'm')
(2, 'n')
```

```
In [191]: itr=iter(d.items())
while True:
    try:
        print(next(itr))
    except StopIteration:
        break
```

```
(3, 'x')
(4, 'y')
(1, 'm')
(2, 'n')
```

```
In [192]: from collections import OrderedDict
d = OrderedDict()
d[1] = 'a'
d[2] = 'r'
d[3] = 'a'
d[4] = 't'
d[5] = 'h'
d[6] = 'i'
print(d)
```

```
OrderedDict([(1, 'a'), (2, 'r'), (3, 'a'), (4, 't'), (5, 'h'), (6, 'i')])
```

```
In [193]: d[2]='a'
print(d)
```

```
OrderedDict([(1, 'a'), (2, 'a'), (3, 'a'), (4, 't'), (5, 'h'), (6, 'i')])
```

```
In [194]: print(d.keys())
for key,val in d.items(): #similar to dictionary
    print(key,val,end=' ')
```

```
odict_keys([1, 2, 3, 4, 5, 6])
1 a 2 a 3 a 4 t 5 h 6 i
```

```
In [195]: itr=iter(d.items())
while True:
    try:
        print(next(itr))
    except StopIteration:
        break
```

```
(1, 'a')
(2, 'a')
(3, 'a')
(4, 't')
(5, 'h')
(6, 'i')
```

```
In [196]: from collections import defaultdict
d = defaultdict(int)
d[1]='abc'
d[2]='efg'
print(d)
print(d[3])
```

```
defaultdict(<class 'int'>, {1: 'abc', 2: 'efg'})
0
```

```
In [197]: from collections import defaultdict
d=defaultdict(int)
d[1]='AI'
d[2]='ML'
print(d)
```

```
defaultdict(<class 'int'>, {1: 'AI', 2: 'ML'})
```

```
In [198]: for i in d.items():
    print(i)
```

```
(1, 'AI')
(2, 'ML')
```

numeric types

```
In [199]: a = 10
print("Type of a: ", type(a))
#int(x) -> converts x to integer type
print(int(2.5))
```

```
Type of a: <class 'int'>
2
```

```
In [200]: b = 24.0
print("Type of b: ", type(b))
#use float(x) -> converts x to float type
print(float(37))
```

```
Type of b: <class 'float'>
37.0
```

```
In [201]: c = 3 + 5j
print("Type of c: ", type(c))
#complex(x) is used to convert x to complex type
myComplex = complex(21)
print(myComplex)
```

```
Type of c: <class 'complex'>
(21+0j)
```

```
In [203]: import decimal
print(decimal.Decimal(0.2))
```

```
0.20000000000000011102230246251565404236316680908203125
```

```
In [204]: from decimal import Decimal as D
print(D('1.1') + D('2.2'))
print(D('1.2') * D('2.50'))
```

```
3.3
3.000
```

```
In [205]: from fractions import Fraction
print(Fraction(0.125))
print(Fraction(1.1))
```

```
1/8
2476979795053773/2251799813685248
```

```
In [206]: from fractions import Fraction
print(Fraction('1.1'))
```

```
11/10
```

```
In [207]: from fractions import Fraction as F  
print(F(1, 3) + F(1, 3))  
print(1 / F(5, 6))
```

```
2/3  
6/5
```

```
In [208]: import datetime as dt  
dtobj = dt.datetime.now()  
print(dtobj)  
print(dtobj.year)  
print(dtobj.strftime("%A"))
```

```
2021-12-17 16:10:37.796667  
2021  
Friday
```

```
In [209]: import datetime  
x = datetime.datetime(2020, 12, 18)  
print(x)
```

```
2020-12-18 00:00:00
```

```
In [210]: x = min(5, 10, 25)  
y = max(5, 10, 25)  
print(x)  
print(y)
```

```
5  
25
```

```
In [211]: a = abs(-7.25)  
print("abs(-7.25) is: ",a)  
#The pow(x, y) function returns the value of x to the power of y (xy).  
b = pow(4, 3)  
print("pow(4, 3) is: ",b)
```

```
abs(-7.25) is: 7.25  
pow(4, 3) is: 64
```

```
In [212]: print(math.pi)
print(math.exp(2))
print(math.log10(1000))
print(math.factorial(6))
print(math.sqrt(64))
print(math.ceil(1.4))
print(math.floor(1.4))
```

NameError

Traceback (most recent call last)

```
~\AppData\Local\Temp\ipykernel_8636/3869402749.py in <module>
----> 1 print(math.pi)
      2 print(math.exp(2))
      3 print(math.log10(1000))
      4 print(math.factorial(6))
      5 print(math.sqrt(64))
```

NameError: name 'math' is not defined

```
In [213]: x = 100
y = 3.256
print(x + y)
print(x - y)
print(x * y)
print(x / y)
print(x // y) #floor division
print(x % 7) #modulus operator
print(12 ** 3)
```

```
103.256
96.744
325.5999999999997
30.712530712530715
30.0
2
1728
```

Exceptions

```
In [37]: try:
    f = open('demo1.txt')
    if f.name == 'demo123.txt':
        raise Exception
except IOError as e:
    print('First!')
except Exception as e:
    print('Second')
else:
    print(f.read())
    f.close()
finally:
    print("Executing Finally...")
print('End of program')
```

First!
Executing Finally...
End of program

```
In [216]: amount = 10000
# check that You are eligible to
if(amount>2999):
    print("You are eligible to purchase Dsa Self Paced")
```

You are eligible to purchase Dsa Self Paced

```
In [220]: def AbyB(a , b):
    try:
        c = ((a+b) / (a-b))
    except ZeroDivisionError:
        print("a/b result in 0")
    else:
        print(c)
# Driver program to test above function
AbyB(2.0, 3.0)
AbyB(3.0, 3.0)
```

-5.0
a/b result in 0

```
In [221]: try:
    num = int(input("Enter a number: "))
    assert num % 2 == 0
except:
    print("Not an even number!")
else:
    reciprocal = 1/num
    print(reciprocal)
```

Enter a number: 20
0.05

```
In [222]: try:  
    num = int(input("Enter a number: "))  
    assert num % 2 == 0  
except:  
    print("Not an even number!")  
else:  
    reciprocal = 1/num  
    print(reciprocal)
```

```
Enter a number: 10  
0.1
```

```
In [223]: try:  
    raise NameError("Hi there") # Raise Error  
except NameError:  
    print("An exception")
```

```
An exception
```

```
In [40]: try:  
    x=int(input("Enter First Number: "))  
    y=int(input("Enter Second Number: "))  
    print(x/y)  
except ZeroDivisionError:  
    print("ZeroDivisionError:Can't divide with zero")  
except:  
    print("Default Except:Plz provide valid input only")
```

```
Enter First Number: 20  
Enter Second Number: 40  
0.5
```

```
In [38]: try:  
    print("Hi")  
    print(10/5)  
except zeroDivisionError:  
    print("10 is not divisible by 5")
```

```
Hi  
2.0
```

files

```
In [11]: import os  
f=open('C:\\exceptions.txt','r')  
print(f.read())  
f.close()
```

```
SQL> DECLARE  
2      emp employee%ROWTYPE;  
3  BEGIN  
4      SELECT * INTO emp FROM employee WHERE deptno=77;  
5  END;  
6 /  
DECLARE  
*  
ERROR at line 1:  
ORA-01403: no data found  
ORA-06512: at line 4
```

```
SQL> DECLARE  
2      emp employee%ROWTYPE;  
3  BEGIN  
4      SELECT * INTO emp FROM employee WHERE deptno=77;  
5  EXCEPTION  
6    WHEN NO_DATA_FOUND THEN  
7      DBMS_OUTPUT.PUT_LINE('No data found');
```

```
In [13]: import os  
os.listdir('D:\\New folder\\')
```

```
Out[13]: ['KPI Python algorithms.docx',  
'KPI Python files.docx',  
'KPI Python.docx',  
'Python.docx',  
'Python.pdf']
```

```
In [15]: import os  
f=open('Downloads\\MOVIE.txt','r')  
print(f.read())  
f.close()  
  
create table MOVIE (  
    MOV_ID number not null constraint movie_pk primary key,  
    MOV_TITLE varchar2(20) not null,  
    MOV_YEAR number not null,  
    MOV_TIME number not null,  
    MOV_LANG varchar2(20) not null,  
    MOV_DT_REL date,  
    MOV_REL_COUNTRY varchar2(20) not null  
);  
Insert into movie values(901,'Vertigo',1958,128,'English',TO_DATE('1958-08-2  
4','YYYY-MM-DD'),'UK');  
  
Insert into movie values(902,'The Innocents',1961,100,'English',TO_DATE('1962-0  
2-19','YYYY-MM-DD'),'SW');  
  
Insert into movie values(903,'Lawrence of Arabia',1962,216,'English',TO_DATE('1  
962-12-11','YYYY-MM-DD'),'UK');  
  
Insert into movie values(904,'The Deer Hunter',1978,183,'English',TO_DATE('1979  
-03-08','YYYY-MM-DD'),'UK');  
  
Insert into movie values(905,'Amadeus',1984,160,'English',TO_DATE('1985-01-0  
7','YYYY-MM-DD'),'UK');  
  
Insert into movie values(906,'Blade Runner',1982,117,'English',TO_DATE('1982-09  
-09','YYYY-MM-DD'),'UK');  
  
Insert into movie values(907,'Eye Wide Shut',1999,159,'English',NULL,'UK');  
  
Insert into movie values(908,'The Usual Suspects',1995,106,'English',TO_DATE('1  
995-08-25','YYYY-MM-DD'),'UK');  
  
Insert into movie values(909,'Chinatown',1974,130,'English',TO_DATE('1974-08-0  
9','YYYY-MM-DD'),'UK');  
  
Insert into movie values(910,'The Boogie Nights',1997,155,'English',TO_DATE('19  
98-02-16','YYYY-MM-DD'),'UK');  
  
Insert into movie values(911,'Annie Hall',1977,93,'English',TO_DATE('1977-04-2  
0','YYYY-MM-DD'),'USA');  
  
Insert into movie values(912,'Princess Mononoke',1997,134,'Japanese',TO_DATE('2  
001-10-19','YYYY-MM-DD'),'UK');  
  
Insert into movie values(913,'The Shawshank Redemption',1994,142,'English',TO_D  
ATE('1995-02-17','YYYY-MM-DD'),'UK');  
  
Insert into movie values(914,'American Beauty',1999,122,'English',NULL,'UK');  
  
Insert into movie values(915,'Titanic',1997,194,'English',TO_DATE('1998-01-2  
3','YYYY-MM-DD'),'UK');
```

```
Insert into movie values(916,'Good Will Hunting',1997,126,'English',TO_DATE('1998-06-03','YYYY-MM-DD'),'UK');

Insert into movie values(917,'Deliverance',1972,109,'English',TO_DATE('1982-10-05','YYYY-MM-DD'),'UK');

Insert into movie values(918,'Trainspotting',1996,94,'English',TO_DATE('1996-02-23','YYYY-MM-DD'),'UK');

Insert into movie values(919,'The Prestige',2006,130,'English',TO_DATE('2006-11-10','YYYY-MM-DD'),'UK');

Insert into movie values(920,'Donnie Darko',2001,113,'English',NULL,'UK');

Insert into movie values(921,'Slumdog Millionaire',2008,120,'English',TO_DATE('2009-01-09','YYYY-MM-DD'),'UK');

Insert into movie values(922,'Aliens',1986,137,'English',TO_DATE('1986-08-29','YYYY-MM-DD'),'UK');

Insert into movie values(923,'Beyond the Sea',2004,118,'English',TO_DATE('2004-11-26','YYYY-MM-DD'),'UK');

Insert into movie values(924,'Avatar',2009,162,'English',TO_DATE('2009-12-17','YYYY-MM-DD'),'UK');

Insert into movie values(925,'Braveheart',1995,178,'English',TO_DATE('1995-09-08','YYYY-MM-DD'),'UK');

Insert into movie values(926,'Seven Samurai',1954,207,'Japanese',TO_DATE('1954-04-26','YYYY-MM-DD'),'JP');

Insert into movie values(927,'Spirited Away',2001,125,'Japanese',TO_DATE('2003-09-12','YYYY-MM-DD'),'UK');

Insert into movie values(928,'Back to the Future',1985,116,'English',TO_DATE('1985-12-04','YYYY-MM-DD'),'UK');
```

```
In [18]: f=open('Documents\\data.txt','r')
print(f.read())
f.close()
```

sai

```
In [22]: f=open('Documents\\data.txt','w')
f.write("mouni")
```

Out[22]: 5

```
In [23]: f=open('Documents\\data.txt','a')
f.write('mounisha')
f.close()
```

```
In [25]: f=open('Documents\\data.txt','r')
print(f.read())
f.close()
```

mounimounisha

```
In [27]: f=open('Documents\\data.txt','r')
print(f.read(6))
f.close()
```

mounim

```
In [28]: f=open('Documents\\data.txt','r')
print(f.readlines())
f.close()
```

['mounimounisha']

```
In [29]: f=open('Documents\\data.txt','r')
for x in f:
    print(x)
```

mounimounisha

arrays

```
In [30]: food = ["fat", "protein", "vitamin"]
food[0] = "mineral"
print(food)
```

['mineral', 'protein', 'vitamin']

```
In [31]: food = ["fat", "protein", "vitamin"]
a = len(food)
print(a)
```

3

```
In [32]: cars = ["Ford", "Volvo", "BMW"]
for x in cars:
    print(x)
```

Ford
Volvo
BMW

```
In [35]: import socket
ip = socket.gethostbyname('www.google.com')
print(ip)
```

142.250.76.36

```
In [41]: import http.client
import json
conn = http.client.HTTPSConnection('www.httpbin.org')
headers = {'Content-type': 'application/json'}
foo = {'text': 'Hello HTTP #1 **cool**, and #1!'}
json_data = json.dumps(foo)
conn.request('POST', '/post', json_data, headers)
response = conn.getresponse()
print(response.read().decode())
```

```
{
  "args": {},
  "data": "{\"text\": \"Hello HTTP #1 **cool**, and #1!\"}",
  "files": {},
  "form": {},
  "headers": {
    "Accept-Encoding": "identity",
    "Content-Length": "43",
    "Content-Type": "application/json",
    "Host": "www.httpbin.org",
    "X-Amzn-Trace-Id": "Root=1-61c02154-0b01a7873f17edbb54287db2"
  },
  "json": {
    "text": "Hello HTTP #1 **cool**, and #1!"
  },
  "origin": "157.47.31.140",
  "url": "https://www.httpbin.org/post"
}
```

srtings

```
In [1]: print("welcome to kpi")
a = "welcome to kpi"
print(a)
```

```
welcome to kpi
welcome to kpi
```

```
In [3]: a = "hi, mouni!"
print(a[1])
```

```
i
```

```
In [4]: for x in "mango":
    print(x)
```

```
m
a
n
g
o
```

```
In [5]: a = "Hello, World!"
print(len(a))
```

```
13
```

```
In [6]: txt = "The best things in life are free!"
if "free" in txt:
    print("Yes, 'free' is present.")
txt = "The best things in life are free!"
if "expensive" not in txt:
    print("Yes, 'expensive' is NOT present.")
```

```
Yes, 'free' is present.
Yes, 'expensive' is NOT present.
```

```
In [7]: b = "Hello, World!"
print(b[2:5])

b = "Hello, World!"
print(b[-5:-2])
```

```
llo
orl
```

```
In [8]: a = "Hello, World!"  
print(a.upper())  
  
a = "Hello, World!"  
print(a.lower())  
  
a = " Hello, World! "  
print(a.strip())  
  
a = "Hello, World!"  
print(a.replace("H", "J"))  
  
a = "Hello, World!"  
print(a.split(","))
```

```
HELLO, WORLD!  
hello, world!  
Hello, World!  
Jello, World!  
['Hello', ' World!']
```

lists

```
In [9]: thislist = ["pink", "blue", "yellow"]  
print(thislist)
```

```
['pink', 'blue', 'yellow']
```

```
In [10]: thislist = ["apple", "banana", "cherry"]  
print(len(thislist))
```

```
3
```

```
In [11]: thislist = list(("apple", "banana", "cherry"))  
print(thislist)
```

```
['apple', 'banana', 'cherry']
```

```
In [12]: thislist = ["pink", "blue", "yellow"]  
print(thislist[1])
```

```
blue
```

```
In [13]: thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[2:5])
```

```
['cherry', 'orange', 'kiwi']
```

```
In [14]: thislist = ["apple", "banana", "cherry"]
if "apple" in thislist:
    print("Yes, 'apple' is in the fruits list")
```

```
Yes, 'apple' is in the fruits list
```

```
In [15]: thislist = ["apple", "banana", "cherry"]
thislist.insert(2, "watermelon")
print(thislist)
```

```
['apple', 'banana', 'watermelon', 'cherry']
```

```
In [16]: thislist = ["apple", "banana", "cherry"]
thislist.remove("banana")
print(thislist)
```

```
['apple', 'cherry']
```

```
In [17]: thislist = ["apple", "banana", "cherry"]
thislist.pop(1)
print(thislist)
```

```
['apple', 'cherry']
```

```
In [19]: thislist = ["apple", "banana", "cherry"]
thislist.pop()
print(thislist)
```

```
['apple', 'banana']
```

```
In [20]: thislist = ["apple", "banana", "cherry"]
del thislist
```

```
[]
```

```
In [22]: thislist = ["apple", "banana", "cherry"]
for x in thislist:
    print(x)
```

```
apple
banana
cherry
```

```
In [23]: thislist = ["apple", "banana", "cherry"]
for i in range(len(thislist)):
    print(thislist[i])
```

```
apple
banana
cherry
```

```
In [27]: thislist = ["apple", "banana", "cherry"]
i = 0
while i < len(thislist):
    print(thislist[i])
    i = i + 1
```

```
apple
banana
cherry
```

```
In [28]: thelist = ["apple", "banana", "cherry"]
[print(x) for x in thelist]
```

```
apple
banana
cherry
```

```
Out[28]: [None, None, None]
```

loops

```
In [29]: colours = ["red", "blue", "yellow"]
for x in colours:
    print(x)
```

```
red
blue
yellow
```

```
In [31]: for x in "cherry":  
    print(x)
```

c
h
e
r
r
y

```
In [33]: for x in range(10):  
    print(x)  
print("\n")
```

0
1
2
3
4
5
6
7
8
9

```
In [35]: for x in range(2, 30, 3):  
    print(x)
```

2
5
8
11
14
17
20
23
26
29

```
In [36]: i = 1  
while i < 6:  
    print(i)  
    i += 1
```

1
2
3
4
5

In []: