
The output of expression, `k = [print(i) for i in "maverick" if i not in "aeiou"]` is _____.
prints all characters that are not vowels

The output of the expression `[chr(i) for i in [65, 66, 67]]` is _____.
['A', 'B', 'C']

Generators consume more space in memory than the lists. State if the statement is True or false.
False

A generator function can have multiple yield expressions. State if the statement is True or False.
True

The output of the expression `{ ord(i) for i in 'apple' }` is _____.
{97, 112, 108, 101}

What is the return type of function 'id'?
int

Which of the following are present in a function header?
function name and parameter list

The elements of an iterator can be accessed multiple times. State if the statement is True or False.
False

The output of the expression `{0 if i%2 ==0 else 1 for i in range(8)}` is _____.
{0, 1}

What is the default return value of a Python function?
None

Which of the following types of arguments can be passed to a function?
All the options mentioned

The output of the expression `itertools.takewhile(lambda x: x<5, [1,4,6,4,1])` is _____.
[1,4]

Which of the following statement sets the metaclass of class A to B?
`class A:
 __metaclass__ = B`

What is the output of the following code?
`2 2 3 2`

What is the output of the following code?
(<class '__main__.child'>, <class '__main__.mother'>, <class '__main__.father'>,
<class '__main__.grandpa'>, <class 'object'>)

What is the output of the following code?
False
`A(x: 12, y: 3)`

What is the output of the following code?
"4 - 10"

What is the output of the following code?
Results in Error

Which of the following keyword is used for creating a method inside a class ?

def

What is the output of the following code ?

two
0.0
-0x2a

Which of the following method is used by a user defined class to support '+' operator?

__add__

Which methods are invoked on entering into and exiting from the block of code written in 'with' statement?

__enter__, __exit__

The output of the expression '2' == 2 is _____.

False

Which of the following exception occurs, when an undefined object is accessed?

NameError

Which of the keyword is used to display a customised error message to the user?

raise

In which of the following scenarios, finally block is executed?

always

Which of the following exception occurs, when a number is divided by zero?

ZeroDivisionError

Which of the following exception occurs, when an integer object is added to a string object?

TypeError

When will the else part of try-except-else be executed?

when no exception occurs

If a list has 5 elements, then which of the following exceptions is raised when 8th element is accessed?

IndexError

Can one block of except statements handle multiple exception?

Yes, like except NameError, SyntaxError, ...

How many except statements can a try-except block have?

more than zero

Any Python Script can act like a Module. State if the statement is True or False?

True+

In Python, which of the following files is mandatory to treat a folder as a package?

init.py(bold)+

Which of the following expression can be used to check if the file 'C:\Sample.txt' exists and is also a regular file?

os.path.isfile(C:\Sample.txt)+

Which of the following modules is used to manage installation, upgrade, deletion of other packages automatically?

pip+

Which of the following methods of 'random' module is used to pick a single

element, randomly, from a given list of elements?
choice+

Which of the following statement retrieves names of all builtin module names?
import sys; sys.builtin_module_names+

Which of the following is not a way to import the module 'm1' or the functions 'f1' and 'f2' defined in it?
import f1, f2 from m1+

Which of the following modules are used to deal with Data compression and archiving?
All of those mentioned+

Which of the following module is not used for parsing command line arguments automatically?
cmdparse+

Which of the following statement retrieves names of all builtin objects?
import builtins; builtins.dict.keys()+

//////////Asses 1
Which methods are defined in an iterator class?
iter, next

Which of the following keyword is used for creating a method inside a class ?
def

What is the output of the following code?
(<class '__main__.child'>, <class '__main__.mother'>, <class '__main__.father'>,
<class '__main__.grandpa'>, <class 'object'>)

Which methods are invoked on entering into and exiting from the block of code written in 'with' statement?
__enter__, __exit__

In Python, which of the following files is mandatory to treat a folder as a package?
init.py*

Which of the following statement sets the metaclass of class A to B?
class A:
 __metaclass__ = B

Which of the following variables stores documentation of a function?
docstr*

In which of the following scenarios, finally block is executed?
always

Which of the following statement retrieves names of all builtin objects?
import builtins; builtins.__dict__.keys()

Which keyword is used for defining a function?
def

Which of the following modules contain functions that create iterators for efficient looping?
itertools

Which of the following module is not used for parsing command line arguments automatically?
cmdparse

What is the output of the following code?

```
2 2 3 2
```

Which of the following brackets are used to define a set comprehension?

```
{}
```

The output of the expression `{i:j for i in "abcd" for j in "kiwi"}` is _____.
`{'a': 'kiwi', 'd': 'kiwi', 'c': 'kiwi', 'b': 'kiwi'}*`

When will the else part of try-except-else be executed?

when no exception occurs

Which of the following statement retrieves names of all builtin module names?

```
import sys; sys.builtin_module_names
```

What is the output of the following code ?

```
two  
0.0  
-0x2a
```

How many except statements can a try-except block have?

more than zero

Which of the following exception occurs, when an integer object is added to a string object?

`TypeError`

//////////Exercise1

```
import math  
class Point:  
    def __init__(self,x,y,z):  
        self.x=x  
        self.y=y  
        self.z=z  
    def __str__(self):  
        display="point : (" +str(self.x)+", "+str(self.y)+", "+str(self.z)+")."  
        return display  
    def __add__(self,other):  
        return Point(self.x+other.x,self.y+other.y,self.z+other.z)  
  
def distance(Point1,Point2):  
    distance=math.sqrt( (Point1.x-Point2.x)**2 + (Point1.y-Point2.y)**2 +  
(Point1.z -Point2.z)**2 )  
    return distance  
p1=Point(5,5,5)  
p2=Point(3,4,5)  
print(p1+p2)
```

//////////Exercise2

```
import unittest  
def isEven(x=2):  
    return x%2==0  
class TestIsEvenMethod(unittest.TestCase):  
    def test_isEven1(self):  
        self.assertEqual(isEven(5),False)  
    def test_isEven2(self):  
        self.assertEqual(isEven(10),True)  
    def test_isEven3(self):  
        self.assertRaises(TypeError,isEven("hello"))  
if __name__ == '__main__':  
    unittest.main()
```

//////////Exercise3

```
import sys
class Circle:
    def __init__(self,radius):
        if not (isinstance(radius,int)):
            raise RadiusInputError(radius+" is not a number")
        self.radius=radius
class RadiusInputError(Exception):
    def __init__(self,value):
        self.value=value
    def __str__(self):
        return str(self.value)
try:
    #n = int(sys.stdin.readline())
    #if not 0 <= n <= 100:
    #    raise ValueError('the number is not between 0 and 100')

    #string=sys.stdin.readline()
    #if(len(string)>10):
    #    raise ValueError('the string has more than 10 #characters')

    #open("hello.txt")

    Circle("7")

except ValueError as e:
    # print(e)
except IOError as e:
    # print("File not found")
except RadiusInputError as e:
    print(e)
```

//////////Exercise4

```
import itertools as it
n=list([10,13,16,22,9,4,37])
even=[]
odd=[]
group=it.groupby(n)
for key,graph in group:
    if key%2==0:
        even.append(key)
    else:
        odd.append(key)
print(even,odd)
```

//////////Exercise5

```
import os
print ("*" * 20)
path="/home"
lstFiles=[]
lstDir=os.walk(path)
for root, dirs, files in lstDir:
    for fichero in files:
        (nombreFichero,extension)=os.path.splitext(fichero)
        if(extension==".py"):
            lstFiles.append(nombreFichero+extension)

print(lstFiles)
print("LISTADO FINALIZADO")
print ("longitud de la lista = ", len(lstFiles))
```

//////////Exercise6

```
import calendar
months_ans=[]
for month in range(1,13):
    mycal=calendar.monthcalendar(2019,month)
    if mycal[4][6]!=0:
        months_ans.append(month)
print(months_ans)
```

//////////Exercise7

```
import timeit
def f1():
    x=list(range(1,21))
    y=[i**2 for i in x]
    return y
def f2():
    x=list(range(1,21))
    g=(i**2 for i in x)
    return g
print(timeit.timeit(f1,number=100000))
print(timeit.timeit(f2,number=100000))
```

//////////Exercise8

```
import cProfile
def f1():
    x=list(range(1,200001))
    y=[i**2 for i in x]
    return y
def f2():
    x=list(range(1,200001))
    g=(i**2 for i in x)
    return g
print(cProfile.runctx("f1()",globals(),locals()))
print("\n"*2+"*"*20+"\n"*2)
print(cProfile.runctx("f2()",globals(),locals()))
```