**Python Programming Basic Assignment-20**

**Question1**

Create a function that takes a list of strings and integers, and filters out the list so that it returns a list of integers only.

Examples

filter\_list([1, 2, 3, "a", "b", 4]) ➞ [1, 2, 3, 4]

filter\_list(["A", 0, "Edabit", 1729, "Python", "1729"]) ➞ [0, 1729]

filter\_list(["Nothing", "here"]) ➞ []

In [4]:

*#way 1 by using filter function*

In [5]:

l**=**[1, 2, 3, "a", "b", 4]

result**=**list(filter(**lambda** a:type(a)**==**int, l))

print(result)

[1, 2, 3, 4]

In [10]:

*#way 2 defining function*

In [6]:

**def** filter\_list(l):

result**=**[]

**for** i **in** l:

**if** type(i)**==**int:

result**.**append(i)

**return** result

In [7]:

filter\_list([1, 2, 3, "a", "b", 4])

Out[7]:

[1, 2, 3, 4]

In [8]:

filter\_list(["A", 0, "Edabit", 1729, "Python", "1729"])

Out[8]:

[0, 1729]

In [9]:

filter\_list(["Nothing", "here"])

Out[9]:

[]

**Question2**

Given a list of numbers, create a function which returns the list but with each element's index in the list added to itself. This means you add 0 to the number at index 0, add 1 to the number at index 1, etc...

Examples

add\_indexes([0, 0, 0, 0, 0]) ➞ [0, 1, 2, 3, 4]

add\_indexes([1, 2, 3, 4, 5]) ➞ [1, 3, 5, 7, 9]

add\_indexes([5, 4, 3, 2, 1]) ➞ [5, 5, 5, 5, 5]

In [32]:

**def** add\_indexes(l):

new**=**[]

**for** i **in** range(len(l)):

new**.**append(l[i]**+**i)

**return** new

In [33]:

add\_indexes([0, 0, 0, 0, 0])

Out[33]:

[0, 1, 2, 3, 4]

In [34]:

add\_indexes([1, 2, 3, 4, 5])

Out[34]:

[1, 3, 5, 7, 9]

In [35]:

add\_indexes([5, 4, 3, 2, 1])

Out[35]:

[5, 5, 5, 5, 5]

**Question3**

Create a function that takes the height and radius of a cone as arguments and returns the volume of the cone rounded to the nearest hundredth. See the resources tab for the formula.

Examples

cone\_volume(3, 2) ➞ 12.57

cone\_volume(15, 6) ➞ 565.49

cone\_volume(18, 0) ➞ 0

In [22]:

*# V=(πr\*\*2h)/3*

**import** numpy **as** np

**def** cone\_volume(height,radius):

pie**=**np**.**pi

volume**=**(pie**\***radius**\*\***2**\***height)**/**3

**return** round(volume,2)

In [23]:

cone\_volume(3,2)

Out[23]:

12.57

In [24]:

cone\_volume(15, 6)

Out[24]:

565.49

In [25]:

cone\_volume(18, 0)

Out[25]:

0.0

**Question4**

This Triangular Number Sequence is generated from a pattern of dots that form a triangle. The first 5 numbers of the sequence, or dots, are:

1, 3, 6, 10, 15

This means that the first triangle has just one dot, the second one has three dots, the third one has 6 dots and so on.

Write a function that gives the number of dots with its corresponding triangle number of the sequence.

Examples

triangle(1) ➞ 1

triangle(6) ➞ 21

triangle(215) ➞ 23220

In [40]:

*# 1, 1+2, 1+2+3, 1+2+3+4, 1+2+3+4+5, 1+2+3+4+5+6 etc......*

**def** triangle(num):

sum**=**0

**for** i **in** range(num**+**1):

sum**=**sum**+**i

**return** sum

In [41]:

triangle(1)

Out[41]:

1

In [42]:

triangle(6)

Out[42]:

21

In [43]:

triangle(215)

Out[43]:

23220

**Question5**

Create a function that takes a list of numbers between 1 and 10 (excluding one number) and returns the missing number.

Examples

missing\_num([1, 2, 3, 4, 6, 7, 8, 9, 10]) ➞ 5

missing\_num([7, 2, 3, 6, 5, 9, 1, 4, 8]) ➞ 10

missing\_num([10, 5, 1, 2, 4, 6, 8, 3, 9]) ➞ 7

In [45]:

**def** missing\_num(l):

**for** i **in** range(1,11):

**if** i **not** **in** l:

print(i)

In [46]:

missing\_num([1, 2, 3, 4, 6, 7, 8, 9, 10])

5

In [47]:

missing\_num([7, 2, 3, 6, 5, 9, 1, 4, 8])

10

In [48]:

missing\_num([10, 5, 1, 2, 4, 6, 8, 3, 9])

7

In [ ]: