In [2]:
```
# Define a function calls addNumber(x, y) that takes in two number and retur

def addNumber(x, y):
    return x+y

addNumber(2, 3)
```

executed in 16ms, finished 11:19:20 2021-01-14

Out[2]: 5

In [4]:
```
# Define a function calls subtractNumber(x, y) that takes in two numbers and

def subtractNumber(x, y):
    return x-y

subtractNumber(5, 8)


```

executed in 10ms, finished 11:20:12 2021-01-14

Out[4]: -3

In [8]:
```
# Write a function getBiggerNumber(x, y) that takes in two numbers as argume

def getBiggerNumber(x, y):
    return x if x>y else y if x<y else "Equal numbers"

getBiggerNumber(6, 8)
```

executed in 9ms, finished 11:22:10 2021-01-14

Out[8]: 8

In [31]:
```
# Python provides many built-in modules with many useful functions.
# One such module is the math module. The math module provides many useful f
# sqrt(x), pow(x, y), ceil(x), floor(x) etc.


# You will need to do a "import math" before you are allowed to use the func
import math

# Calculate the square root of 16 and stores it in the variable a
a=mth.sqrt(16)

# Calculate 3 to the power of 5 and stores it in the variable b
b=pow(3, 5)

# Calculate area of circle with radius = 3.0 by making use of the math.pi co

c=math.pi
radius = 3.0
Area=c*(radius**2)
Area
```

executed in 16ms, finished 15:12:10 2021-01-14

Out[31]: 28.274333882308138

In [22]:
```
# Write a function to convert temperature from Celsius to Fahrenheit scale.
# oC to oF Conversion: Multipy by 9, then divide by 5, then add 32.

# Note: Return a string of 2 decimal places.
# In - Cel2Fah(28.0)
# Out - '82.40'
# In - Cel2Fah(0.00)
# Out - '32.00'

def Cel2Fah(Celsius):
    return (Celsius*9)/5 + 32

# Cel2Fah(28.0)
Cel2Fah(0.00)
```

executed in 10ms, finished 12:22:07 2021-01-14

Out[22]: 32.0

In [15]:
```
# Write a function to compute the BMI of a person.
#     BMI = weight(kg)  /  ( height(m)*height(m) )

# Note: Return a string of 1 decimal place.
# In - BMI(63, 1.7)
# Out - '21.8'
# In - BMI(110, 2)
# Out - '27.5'

def BMI(weight, height):
    return round(weight/(height*height) , 1)

BMI(63, 1.7)
BMI(110, 2)
```

executed in 9ms, finished 11:30:05 2021-01-14

Out[15]: 27.5

In [25]:
```
# Write a function percent(value, total) that takes in two numbers as argume
# In - percent(46, 90)
# Out - 51
# In - percent(51, 51)
# Out - 100
# In - percent(63, 12)
# Out - 525

def percent(value, total):
    return round((value/total)*100)

percent(46, 90)
percent(51, 51)
percent(63, 12)
```

executed in 11ms, finished 12:23:44 2021-01-14

Out[25]: 525

In [35]:
```
# Write a function to compute the hypotenuse given sides a and b of the tria
# Hint: You can use math.sqrt(x) to compute the square root of x.
# In - hypotenuse(3, 4)
# Out - 5
# In - hypotenuse(5, 12)
# Out - 13

# import math as mth
def hypotenuse(side1, side2):
    return round(mth.sqrt((side1)**2+(side2)**2))

hypotenuse(3, 4)
hypotenuse(5, 12)
```

executed in 12ms, finished 12:29:41 2021-01-14

Out[35]: 13

In [44]:
```
# Write a function getSumOfLastDigits() that takes in a list of positive num
# getSumOfLastDigits([2, 3, 4])
# 9
# getSumOfLastDigits([1, 23, 456])
# 10

def getSumOfLastDigits(lst=[]):
    sumation=0
#     for i in lst:
#         sumation+=i%10
#     return sumation


getSumOfLastDigits([2, 3, 4])
getSumOfLastDigits([1, 23, 456])
```

executed in 10ms, finished 12:35:22 2021-01-14

Out[44]: 10

In [53]:
```
# Write a function that uses a default value.
# In - introduce('Lim', 20)
# Out - 'My name is Lim. I am 20 years old.'
# In - introduce('Ahmad')
# Out - 'My name is Ahmad. My age is secret.'

def introduce(name='',default=0):
    return "My name is "+name+'. I am '+str(default)+' years old.' if defaul

introduce('Lim', 20)
introduce('Ahmad')

```

executed in 9ms, finished 13:18:35 2021-01-14

Out[53]: 'My name is Ahmad. My age is secret.'

In [20]:
```
# Write a function isEquilateral(x, y, z) that accepts the 3 sides of a tria
# The program should return True if it is an equilateral triangle.

# In - isEquilateral(2, 4, 3)
# False - False
# In - isEquilateral(3, 3, 3)
# Out - True
# In - isEquilateral(-3, -3, -3)
# Out - False

def isEquilateral(side1, side2, side3):
    return True if side1==side2 and side2==side3 and side1>0 and side2>0 and

isEquilateral(2, 4, 3)
```

executed in 10ms, finished 14:57:53 2021-01-14

Out[20]: False

In [3]:
```
# For a quadratic equation in the form of ax2+bx+c, the discriminant, D is b
# In - quadratic(1, 2, 3)
# Out - 'The discriminant is -8.'
# In - quadratic(1, 3, 2)
# Out - 'The discriminant is 1.'
# In - quadratic(1, 4, 4)
# Out - 'The discriminant is 0.'

def quadratic(a=1, b=1, c=1):
    return (b**2-(4*a*c))

quadratic(1, 2, 3)
quadratic(1, 3, 2)
quadratic(1, 4, 4)
```

executed in 11ms, finished 14:38:09 2021-01-14

Out[3]: 0

In [12]:
```
# Define a function calls addFirstAndLast(x) that takes in a list of numbers
# In - addFirstAndLast([])
# Out - 0
# In - addFirstAndLast([2, 7, 3])
# Out - 5
# In - addFirstAndLast([10])
# Out - 10

def addFirstAndLast(lst):
    if len(lst) == 1:
        return lst[0]
    elif lst:
        return lst[0] + lst[-1]
    else:
        return 'List is empty'

addFirstAndLast([2,7,3])
```

executed in 13ms, finished 14:54:09 2021-01-14

Out[12]: 5

In [5]:
```
# Complete the 'lambda' expression so that it returns True if the argument i

(lambda num : True if num%2 == 0 else False)(7)
```

executed in 8ms, finished 14:39:30 2021-01-14

Out[5]: False

In [6]:
```
# getScore.__doc__
# 'A function that computes and returns the final score.'

def getScore():
  ''''A function that computes and returns the final score.'''
  pass
getScore.__doc__
```

executed in 8ms, finished 14:40:00 2021-01-14

Out[6]:
```
"'A function that computes and returns the final score."
```

In [9]:
```
# In Python, it is possible to pass a function as a argument to another func
# Write a function useFunction(func, num) that takes in a function and a num
# The useFunction should produce the output shown in the examples given belo

# def addOne(x):
#   return x + 1
# useFunction(addOne, 4)
# 25
# useFunction(addOne, 9)
# 100
# useFunction(addOne, 0)
# 1


def addOne(x):
    return x + 1

def useFunction(addOne, n):
    print(addOne(n)**2)

useFunction(addOne, 4)
```

executed in 9ms, finished 14:52:25 2021-01-14

```
25
```

In [ ]:
```
# def f1():
#   print('This is f1 function')
#   f1()
# f1()
```

In [7]:
```
print('\\t')
```

executed in 10ms, finished 14:40:55 2021-01-14

```
\t
```