```
#BEGINNER PRACTICE OF PYTHON AND IMPORTANT POINTS.
#NOTE: Python uses indentation to indicate a block of code.
# Note: we can comment out multiple lines using """ below and above the block (Double Quotation 3 times) [INDENTATION MATTERS]

print('hello world')
#How to assign values to multiple variables in one line
var1, var2=19,2
print("I am ",var1+var2," Years old.") # should print 21 (as 19+2=21)

fltvar=21.7
print("The FLOAT VALUE IS: ",fltvar, " but the INTEGER VALUE IS: ",int(fltvar))

#Cast to string
y,z=str(30.0),str("20")
print(y)
print(z)

# Sum two numbers using typecast
num_int = 123
num_str = "456"
print("Data type of num_int:",type(num_int))
print("Data type of num_str before Type Casting:",type(num_str))
num_str = int(num_str)
print("Data type of num_str after Type Casting:",type(num_str))
num_sum = num_int + num_str
print("Sum of num_int and num_str:",num_sum)
print("Data type of the sum:",type(num_sum))


#Notice the type() function tells the data type of variable

#Practising the math operators:
print("IF YOU GET 10rs EVERY DAY FOR A MONTH THEN BY THE END OF THE MONTH YOU SHOULD HAVE: ",10*30," rs")
print("4 to the power 2 is: ",4**2)
print ("REMAINDER OF 13 / 2 : ",13%2)
print ("Integer division OF 13 / 2 : ",13//2)
print ("Division OF 13 / 2 : ",13/2)
print ("five minus two is: ",5-2)
```

```
I am  21  Years old.
The FLOAT VALUE IS:  21.7  but the INTEGER VALUE IS:  21
30.0
20
Data type of num_int: <class 'int'>
Data type of num_str before Type Casting: <class 'str'>
Data type of num_str after Type Casting: <class 'int'>
Sum of num_int and num_str: 579
Data type of the sum: <class 'int'>
IF YOU GET 10rs EVERY DAY FOR A MONTH THEN BY THE END OF THE MONTH YOU SHOULD HAVE:  300  rs
4 to the power 2 is:  16
REMAINDER OF 13 / 2 :  1
Integer division OF 13 / 2 :  6
Division OF 13 / 2 :  6.5
five minus two is:  3
```

```
#USING RANDOM NUMBER GENERATOR:-(Return a random number between, and included, 20 & 60) import random then on next line write print(random.un
import random
ran=int(random.uniform(0,5))
print("RANDOM NUMBER GENERATED IS: ",ran)

#-----> Practising Comparison Operators & If-else Conditions
#Python uses indentation instead of curly brackets to define the scope in the code.:
#EQUAL TO,NOT EQUAL TO,LESS THAN,GREATER THAN, LESS THAN OR EQUAL TO, GREATER THAN OR EQUAL TO
if ran<=2 and ran!=0:
  print("RAN LESS THAN OR EQUAL TO 2! ")
  if (ran==2):
     print("RAN EQUAL TO 2")
  if(ran!=2):
    print("RAN NOT EQUAL TO 2")
  if(ran<2):
    print("RAN LESS THAN TO 2! ")
elif ran>=3:
  print("RAN GREATER THAN OR EQUAL TO 3!")
  if (ran==3):
     print("RAN EQUAL TO 3")
  if(ran!=3):
    print("RAN NOT EQUAL TO 3")
  if(ran>3):
    print("RAN GREATER THAN TO 3! ")
else:
  print("RAN IS ZERO!")
```

```
RANDOM NUMBER GENERATED IS:  1
RAN LESS THAN OR EQUAL TO 2!
RAN NOT EQUAL TO 2
RAN LESS THAN TO 2!
```

```
#Practising Boolean Operators
# AND
print("and TRUTH TABLE: ",True and True,True and False,False and True,False and False)
# OR
print("or TRUTH TABLE: ",True or True,True or False,False or True,False or False)
# NOT
print("not TRUTH TABLE: ",not True, not False)
```

```
and TRUTH TABLE:  True False False False
or TRUTH TABLE:  True True True False
not TRUTH TABLE:  False True
```

```
#Practising LOOPS. Python has two types of loops i.e. while, for.
# WHILE LOOP
i=1
while i<4:
  print(i)
  i=i+1 #No, there is no ++ operator in Python

# USING BREAK STATEMENT WITH WHILE LOOP
#With the while loop we can execute a set of statements as long as a condition is true or the loop execution reaches a break statement.
while True:
  print('Please type your name.')
  name = input()
  if name == 'your name':
    print("NOT LITERALLY")
    break
  print('Thank you',name,"!")
print('Thank you')

# NOTE: input() in the above example is a built in Python function to take input from user
```

```
1
2
3
Please type your name.
your name
NOT LITERALLY
Thank you
```

```
#Practising While Loop Example with continue Statement.
#When the program reaches a continue statement, the program execution immediately jumps back to the start of the loop.
while True:
  print('Who are you?')
  name = input()
  if name != 'Joe':
    continue
  print('Hello, Joe. What is the password? (It is a fish.)')
  password = input()
  if password == 'swordfish':
    break
print('Access granted.')
```

```
Who are you?
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-66-49f1929898a1> in <module>
      3 while True:
      4   print('Who are you?')
----> 5   name = input()
      6   if name != 'Joe':
      7     continue

 1 frames
/usr/local/lib/python3.8/dist-packages/ipykernel/kernelbase.py in
_input_request(self, prompt, ident, parent, password)
    902             except KeyboardInterrupt:
    903                 # re-raise KeyboardInterrupt, to truncate traceback
--> 904                 raise KeyboardInterrupt("Interrupted by user") from None
    905             except Exception as e:
```

```
# Practising for Loop Example with range()
print('My name is')
for i in range(5):
  print('Jimmy Five Times ({})'.format(str(i)))

#for Loop Example with range() arguments.
#The range() function can also be called with three arguments. The first two arguments will
#be the start and stop values, and the third will be the step argument. The step is the amount that the variable is increased by after each i
for j in range(0,8,2):
  print(j)
```

```
My name is
Jimmy Five Times (0)
Jimmy Five Times (1)
Jimmy Five Times (2)
Jimmy Five Times (3)
Jimmy Five Times (4)
0
2
4
6
```

```
#PRACTSING FUNCTIONS
#Simple Function Example

#A function in Python starts with def keyword followed by the function name with round brackets.
#Function parameters can be passed depending on the requirement.

"""
#E.g 1
def fun1(var):
  print(var)

fun1(90)

#E.g 2
def hello(name):
  print('Hello {}'.format(name))

hello("ar")
hello('ab')
"""

#Function Example with Return Statement

import random #Syntax to import Python libraries
def getAnswer(answerNumber):
  if answerNumber == 1:
    return 'It is certain'
  elif answerNumber == 2:
    return 'It is decidedly so'
  elif answerNumber == 3:
    return 'Yes'
  elif answerNumber == 4:
    return 'Reply hazy try again'
  elif answerNumber == 5:
    return 'Ask again later'
  elif answerNumber == 6:
    return 'Concentrate and ask again'
  elif answerNumber == 7:
    return 'My reply is no'
  elif answerNumber == 8:
    return 'Outlook not so good'
  elif answerNumber == 9:
    return 'Very doubtful'

r = random.randint(1, 9)
print("r is: ",r)
fortune = getAnswer(r)
print(fortune)
```

```
    r is:  1
    It is certain
```

```
  # Practising BUILT - IN FUNCTIONS

  # abs integer number
  num = -8
  print('Absolute value of 8 is:', abs(num))
  # Notice print here, it is also a built in function

  # abs floating number
  fnum = -1.45
  print('Absolute value of 1.45 is:', abs(fnum))

  # input function
  x = input('Enter your name:')
  print('Hello, ' +x)

  # max function
  number = [3, 2, 8, 5, 80, 6]
  largest_number = max(number);
  print("The largest number is:", largest_number)


  # print usage
  print('Hands','on','python','programming','lab',sep='~')

  # NOTE: THE USAGE OF SEP= function enters a whatever you wish in seperators
  # sum function
  my_list = [1,1,1,1,4]
  print ("The sum of my_list is", sum(my_list))
```

```
    Absolute value of 8 is: 8
    Absolute value of 1.45 is: 1.45
    Enter your name:Aisha
    Hello, Aisha
    The largest number is: 80
    Hands~on~python~programming~lab
    The sum of my_list is 8
```

```
  #PRACTISING EXCERSICE QUESTIONS

  # Q1: 9.1 Reverse String (5 Marks)
  #Write a function that reverses a string.
  #The input string is given as an array of characters.
  #You can use Python list to create the array of characters.
  #Do not allocate extra space for another array, you must do this reversal by modifying the input array in place with O(1) extra memory.

  #Example 1: Input: ["h","e","l","l","o"] Output: ["o","l","l","e","h"]
  #Example 2: Input: ["H","a","n","n","a","h"] Output: ["h","a","n","n","a","H"]

  #SOLUTION :

  #function that reverses a string. (Using the concept of slicing [::-1])
  def reverseIt(theList):
    theList[::1]=theList[::-1]
    return theList


  mylist=[]# Creating the array of characters.
  n=int(input("Please enter number of characters in your list: "))
  print("NOW PLEASE ENTER the characters ONE BY ONE")

  for i in range (0,n):
    inp=input("Enter element: ")
    mylist.append(inp)

  print("Input: ",list(mylist))
  reverseIt(mylist)
  print("Output:",mylist)
```

```
    Please enter number of characters in your list: 3
    NOW PLEASE ENTER the characters ONE BY ONE
    Enter element: a
    Enter element: g
    Enter element: h
    Input:  ['a', 'g', 'h']
    Output: ['h', 'g', 'a']
```

```
#9.2 Valid Palindrome (10 Marks) Given a string s, determine if it is a palindrome,
#considering only alphanumeric characters and ignoring cases.
#Refer to this link to learn about Python string functions which might come handy in this task.


# TAKING STRING AS INPUT FROM USER
orgS=input("Please enter a string: ")
s=orgS # TO RETAIN ORGINAL FORM
print("YOU ENTERED:",s)

#Removing special characters
s=s.replace(" ","")
s=s.replace(",","")
s=s.replace(".","")
s=s.replace(":","")
s=s.replace("'","")

print("Without Special Characters: ",s)
s=s.lower()
print("In lowerCase: ",s)


# FUNCTION TO CHECK WHETHER IT IS PALINDROME. (USES SLICING CONCEPT)
def isItPal(aStr):
  if aStr[::1]==aStr[::-1]:
    return True
  else:
    return False

print("\n")

print("Input: s=\"",orgS,"\"",sep="")
if(isItPal(s)):
  print("Output: true")
  print("Explanation: \"",s,"\" is a palindrome.",sep="")
else:
  print("Output: false")
  print("Explanation: \"",s,"\" is not a palindrome.",sep="")
```

```
    Please enter a string: A man, a plan, a canal: Panama
    YOU ENTERED: A man, a plan, a canal: Panama
    Without Special Characters:  AmanaplanacanalPanama
    In lowerCase:  amanaplanacanalpanama


    Input: s="A man, a plan, a canal: Panama"
    Output: true
    Explanation: "amanaplanacanalpanama" is a palindrome.
```

```
#QUESTION 3: Sqrt(x) without any built-in methods (10 Marks)
#Given a non negative integer x, compute and return the square root of x.
#Since the return type is an integer, the decimal digits are truncated, and only the integer part of the result is returned.

#Example 1: Input: x = 4 Output: 2
#Example 2: Input: x = 8 Output: 2
#Explanation: The square root of 8 is 2.82842…, and since the decimal part is truncated, 2 is returned.

def getSqrt(x):
  if (int(x)<0):
    return "ERROR COMPUTNG SQUARE ROOT! INPUT NOT A NON NEGATIVE INTEGER."
  else:
      return int(int(x)**(1/2))

inpt=input("PLEASE ENTER  A NON NEGATIVE INTEGER: ")
print("Input: x =",inpt)
print("Output: ",getSqrt(inpt))
```

```
PLEASE ENTER  A NON NEGATIVE INTEGER: 4
Input: x = 4
Output:  2
```