

```
In [2]: ▶ #simple interest
p=int(input("enter the amount:"))
t=int(input("enter the time period:"))
r=float(input("enter the rate of interest:"))
si=p*t*r/100
print("Simple interset is:",si)
```

```
enter the amount:1000
enter the time period:5
enter the rate of interest:5
Simple interset is: 250.0
```

```
In [4]: ▶ print('This sentence is output to the screen')
a=5
print("The value of a is:",a)
print('x:',1,2,3,4)
x = 10 ; y = 20
print('The value of x is {} and y is {}'.format(x,y))
print('I love {} and {}'.format('bread','butter'))
print('I love {} and {}'.format('bread','butter'))
```

```
This sentence is output to the screen
The value of a is: 5
x: 1 2 3 4
The value of x is 10 and y is 20
I love bread and butter
I love butter and bread
```

```
In [5]: ▶ print('Hello {name}, {greeting}'.format(greeting = 'Good Morning!!',\
                                                    name = 'John'))
```

```
Hello John, Good Morning!!
```

```
In [6]: ▶ x = 15.3456789
print('The value of x is %3.2f' %x)
print('The value of x is %3.4f' %x)
```

The value of x is 15.35  
The value of x is 15.3457

```
In [7]: ▶ for x in range(1, 11):
        print('{0:2d} {1:3d} {2:4d}'.format(x, x*x, x*x*x))
```

```
1  1  1
2  4  8
3  9 27
4 16 64
5 25 125
6 36 216
7 49 343
8 64 512
9 81 729
10 100 1000
```

```
In [8]: ▶ table = {'Raju': 9481567890, 'Ravi': 9481234567, 'Rahul': 9481987655}
for name, phone in table.items():
    print('{0:10} ==> {1:10d}'.format(name, phone))
```

```
Raju      ==> 9481567890
Ravi      ==> 9481234567
Rahul     ==> 9481987655
```

```
In [14]: ▶ import math
print('The value of PI is approximately %6.3f.' % math.pi)
```

The value of PI is approximately 3.142.

```
In [16]: x = input("Enter a string: ")
print("The entered string is :{0}".format(x))
y = int(input("Enter a integer: "))
print("The entered integer is :",y)
z = float(input("Enter a floating point number:"))
print("The entered real number is :",z)
```

```
Enter a string: str
The entered string is :str
Enter a integer: 5
The entered integer is : 5
Enter a floating point number:5.55
The entered real number is : 5.55
```

```
In [20]: x = ('1' + '2' + '3' + '4')
y = '1' + '2' + \
'11' + '12'
weekdays = ['Monday', 'Tuesday', 'Wednesday',
'Thursday', 'Friday']
weekday = {'one':
'Monday'}
print ('x has a value of', x)
print ('y has a value of', y)
print(weekdays)
print(weekday)
```

```
x has a value of 1234
y has a value of 121112
['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday']
{'one': 'Monday'}
```

```
In [21]: import os;
x = 'Hello'; print(x)
```

```
Hello
```

```
In [22]: ▶ var = -1
if var < 0:
    print(var)
    print("the value of var is negative")

if ( var == -1 ) :
    print("the value of var is negative")
```

```
-1
the value of var is negative
the value of var is negative
```

```
In [23]: ▶ #else if
var = 1
if var < 0:
    print("the value of var is negative")
    print(var)
else:
    print("the value of var is positive")
    print(var)
```

```
the value of var is positive
1
```

```
In [24]: ▶ # nested if else
score = 95
if score >= 99:
    print('A')
elif score >= 75:
    print('B')
elif score >= 60:
    print('C')
elif score >= 35:
    print('D')
else:
    print('F')
3.0
```

B

Out[24]: 3.0

```
In [25]: ▶ print("First Example")
for item in [1,2,3,4,5]:
    print('item :', item)
```

First Example  
item : 1  
item : 2  
item : 3  
item : 4  
item : 5

```
In [26]: ▶ print("Second Example")
letters = ['A', 'B', 'C']
for index in range(len(letters)):
    print('First loop letter :', letters[index])
```

Second Example  
First loop letter : A  
First loop letter : B  
First loop letter : C

```
In [27]: ▶ # while loop
count = 0
while (count <5):
    print('The count is:', count)
    count = count + 1
```

```
The count is: 0
The count is: 1
The count is: 2
The count is: 3
The count is: 4
```

```
In [29]: ▶ # Create lists
list_1 = ['Stat', 'Programming', 2016, 2017, 2018]
list_2 = ['a', 'b', 1, 2, 3, 4, 5, 6, 7 ]
# Accessing values in lists
print("list_1[0]: ", list_1[0])
print("list2_[1:5]: ", list_2[1:5])
```

```
list_1[0]: Stat
list2_[1:5]: ['b', 1, 2, 3]
```

```
In [30]: ▶ print("list_1 values: ", list_1)
list_1.append(2019)
print("list_1 values post append: ", list_1)
```

```
list_1 values: ['Stat', 'Programming', 2016, 2017, 2018]
list_1 values post append: ['Stat', 'Programming', 2016, 2017, 2018, 2019]
```

```
In [31]: ▶ print("Values of list_1: ", list_1)
print("Index 2 value : ", list_1[2])
list_1[2] = 2015;
print("Index 2's new value : ", list_1[2])
```

```
Values of list_1: ['Stat', 'Programming', 2016, 2017, 2018, 2019]
Index 2 value : 2016
Index 2's new value : 2015
```

```
In [35]: ► print("list_1 values: ", list_1)
          del list_1[3];
          print("After deleting value at index 2 :", list_1)
```

list\_1 values: ['Stat', 'Programming', 2015, 2017, 2018]

After deleting value at index 2 : ['Stat', 'Programming', 2015, 2018]

```
In [41]: ► import string
import operator
# basic operations on lists

print("Length: ", len(list_1))

print("Concatenation: ", [1,2,3] + [4, 5, 6])

print("Repetition :", ['Hello'] * 4)

print("Membership :", 3 in [1,2,3])

print("Iteration :")
for x in [1,2,3]: print(x)

# Negative sign will count from the right
print("slicing :", list_1[-2])

# If you dont specify the end explicitly, all elements from the specified
#start index will be printed
print("slicing range: ", list_1[1:])

print("Max of list: ", max([1,2,3,4,5]))

print("Min of list: ", min([1,2,3,4,5]))

print("Count number of 1 in list: ", [1,1,2,3,4,5,].count(1))

list_1.extend(list_2)

print("Extended :", list_1)
print("Index for Programming:",list_1.index(5))
print(list_1)
print("pop last item in list: ", list_1.pop())
print("pop the item with index 2: ", list_1.pop(2))
list_1.remove('b')
print("removed b from list: ", list_1)
list_1.reverse()
print("Reverse: ", list_1)
list_1 = ['a', 'c', 'b']
list_1.sort()
```



```
print("Sort ascending: ", list_1)
list_1.sort(reverse = True)
print("Sort descending: ", list_1)
```

Length: 39

Concatenation: [1, 2, 3, 4, 5, 6]

Repetition : ['Hello', 'Hello', 'Hello', 'Hello']

Membership : True

Iteration :

1

2

3

slicing : 6

slicing range: ['b', 'a', 'a', 'b', 1, 2, 3, 4, 5, 6, 7, 'a', 'b', 1, 2, 3, 4, 5, 6, 7, 'a', 'b', 1, 2, 3, 4, 5, 6, 7, 'a', 'b', 1, 2, 3, 4, 5, 6, 7]

Max of list: 5

Min of list: 1

Count number of 1 in list: 2

Extended : ['c', 'b', 'a', 'a', 'b', 1, 2, 3, 4, 5, 6, 7, 'a', 'b', 1, 2, 3, 4, 5, 6, 7, 'a', 'b', 1, 2, 3, 4, 5, 6, 7, 'a', 'b', 1, 2, 3, 4, 5, 6, 7]

Index for Programming: 9

['c', 'b', 'a', 'a', 'b', 1, 2, 3, 4, 5, 6, 7, 'a', 'b', 1, 2, 3, 4, 5, 6, 7, 'a', 'b', 1, 2, 3, 4, 5, 6, 7, 'a', 'b', 1, 2, 3, 4, 5, 6, 7]

pop last item in list: 7

pop the item with index 2: a

removed b from list: ['c', 'a', 'b', 1, 2, 3, 4, 5, 6, 7, 'a', 'b', 1, 2, 3, 4, 5, 6, 7, 'a', 'b', 1, 2, 3, 4, 5, 6, 7, 'a', 'b', 1, 2, 3, 4, 5, 6]

Reverse: [6, 5, 4, 3, 2, 1, 'b', 'a', 7, 6, 5, 4, 3, 2, 1, 'b', 'a', 7, 6, 5, 4, 3, 2, 1, 'b', 'a', 7, 6, 5, 4, 3, 2, 1, 'b', 'a', 7, 6, 5, 4, 3, 2, 1, 'b', 'a', 'c']

Sort ascending: ['a', 'b', 'c']

Sort descending: ['c', 'b', 'a']

```
In [42]: ▶ # Creating a tuple
Tuple = ()
print("Empty Tuple: ", Tuple)
Tuple = (1,)
print("Tuple with single item: ", Tuple)
Tuple = ('a', 'b', 'c', 'd', 1, 2, 3)
print("Sample Tuple :", Tuple)

Empty Tuple: ()
Tuple with single item: (1,)
Sample Tuple : ('a', 'b', 'c', 'd', 1, 2, 3)
```

```
In [43]: ▶ # Accessing items in tuple
Tuple = ('a', 'b', 'c', 'd', 1, 2, 3)
print("3rd item of Tuple:", Tuple[2])
print("First 3 items of Tuple", Tuple[0:2])

3rd item of Tuple: c
First 3 items of Tuple ('a', 'b')
```

```
In [52]: ▶ # Deleting tuple
Tuple = ('a', 'b', 'c', 'd', 1, 2, 3)
print("Sample Tuple: ", Tuple)
del Tuple
print(Tuple)

Sample Tuple: ('a', 'b', 'c', 'd', 1, 2, 3)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-52-ae37d0c6fab7> in <module>
      3 print("Sample Tuple: ", Tuple)
      4 del Tuple
----> 5 print(Tuple)

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

```
In [53]: ▶ # Basic Tuple operations
Tuple = ('a','b','c','d',1,2,3)
print("Length of Tuple: ", len(Tuple))
Tuple_Concat = Tuple + (7,8,9)
print("Concatinated Tuple: ", Tuple_Concat)

print("Repetition: ", (1,'a',2, 'b') * 3)
print("Membership check: ", 3 in (1,2,3))
# Iteration
for x in (1, 2, 3): print(x)
print("Negative sign will retrieve item from right: ", Tuple_Concat[-2])
print("Sliced Tuple [2:] ", Tuple_Concat[2:])
# Find max
print("Max of the Tuple (1,2,3,4,5,6,7,8,9,10): ",
max((1,2,3,4,5,6,7,8,9,10)))
print("Min of the Tuple (1,2,3,4,5,6,7,8,9,10): ",
min((1,2,3,4,5,6,7,8,9,10)))
print("List [1,2,3,4] converted to tuple: ", type(tuple([1,2,3,4])))
```

```
Length of Tuple: 7
Concatinated Tuple: ('a', 'b', 'c', 'd', 1, 2, 3, 7, 8, 9)
Repetition: (1, 'a', 2, 'b', 1, 'a', 2, 'b', 1, 'a', 2, 'b')
Membership check: True
1
2
3
Negative sign will retrieve item from right: 8
Sliced Tuple [2:] ('c', 'd', 1, 2, 3, 7, 8, 9)
Max of the Tuple (1,2,3,4,5,6,7,8,9,10): 10
Min of the Tuple (1,2,3,4,5,6,7,8,9,10): 1
List [1,2,3,4] converted to tuple: <class 'tuple'>
```

```
In [54]: ▶ # Creating dictionary
dict = {'Name': 'Jivin', 'Age': 6, 'Class': 'First'}
print("Sample dictionary: ", dict)
```

```
Sample dictionary: {'Name': 'Jivin', 'Age': 6, 'Class': 'First'}
```

```
In [55]: ▶ # Accessing items in dictionary
print("Value of key Name, from sample dictionary:", dict['Name'])
```

Value of key Name, from sample dictionary: Jivin

```
In [56]: ▶ dict0 = {'Name': 'Jivin', 'Age': 6, 'Class': 'First'}
print("Sample dictionary: ", dict0)
k=1
for i in dict0:
    print(k,i,dict0[i])
    k=k+1
del (dict0['Name']) # Delete specific item

print("Sample dictionary post deletion of item Name:", dict0)

dict0 = {'Name': 'Jivin', 'Age': 6, 'Class': 'First'}
dict0.clear() # Clear all the contents of dictionary
print("dict post dict.clear():", dict0)

dict = {'Name': 'Jivin', 'Age': 6, 'Class': 'First'}
del (dict0)
```

Sample dictionary: {'Name': 'Jivin', 'Age': 6, 'Class': 'First'}

1 Name Jivin

2 Age 6

3 Class First

Sample dictionary post deletion of item Name: {'Age': 6, 'Class': 'First'}

dict post dict.clear(): {}

```
In [57]: ▶ dict = {'Name': 'Jivin', 'Age': 6, 'Class': 'First'}
print("Sample dictionary: ", dict)
dict['Age'] = 6.5
print("Dictionary post age value update: ", dict)
```

Sample dictionary: {'Name': 'Jivin', 'Age': 6, 'Class': 'First'}

Dictionary post age value update: {'Name': 'Jivin', 'Age': 6.5, 'Class': 'First'}

```
In [58]: dict = {'Name': 'Jivin', 'Age': 6, 'Class': 'First'}
print("Length of dict: ", len(dict))

# Copy the dict
dict1 = dict.copy()
print("Copy:\n",dict1)

# Retrieve value for a given key
print("Value for Age: ", dict.get('Age'))

# Return items of dictionary
print("dict items: ", dict.items())

# Return items of keys
print("dict keys: ", dict.keys())

# return values of dict
print("Value of dict: ", dict.values())

# Concatenate dicts
dict1 = {'Name': 'Jivin', 'Age': 6}
dict2 = {'Sex': 'male' }
dict1.update(dict2)
print("dict1.update(dict2) = ", dict1)
```

Length of dict: 3

Copy:

{'Name': 'Jivin', 'Age': 6, 'Class': 'First'}

Value for Age: 6

dict items: dict\_items([('Name', 'Jivin'), ('Age', 6), ('Class', 'First')])

dict keys: dict\_keys(['Name', 'Age', 'Class'])

Value of dict: dict\_values(['Jivin', 6, 'First'])

dict1.update(dict2) = {'Name': 'Jivin', 'Age': 6, 'Sex': 'male'}

```
In [59]: ▶ def someFunction():  
          print("Hello World")  
  
          someFunction()  
  
Hello World
```

```
In [60]: ▶ def sum_two_numbers(x, y):  
          return x + y  
          print(sum_two_numbers(1,2))  
  
3
```

```
In [61]: ▶ x = 10  
          def sum_two_numbers(y):  
              return x + y  
          print(sum_two_numbers(10))  
  
20
```

```
In [62]: ▶ def sample_function(*args):  
          for a in args:  
              print(a)  
          sample_function(1,2,3)  
  
1  
2  
3
```

```
In [63]: ▶ def sample_function(**args):  
          for a in args:  
              print(a, args[a])  
          sample_function(name='John', age=21)  
  
name John  
age 21
```

```
In [65]: ► def add(x, y):  
           return x + y  
           print("FUNCTION ADD:", add(3, 2))  
  
           add = lambda x, y : x + y  
           print("LAMBDA ADD :", add(3, 2))
```

FUNCTION ADD: 5

LAMBDA ADD : 5

```
In [ ]: ►
```