BASIC PYTHON PROGRAMMING STRYCTURES

In [27]:

```
x = 1
y = 35.656222554887711
z = -325.e100

print(type(x))
print(type(y))
print(type(z))
```

```
<class 'int'>
<class 'float'>
<class 'float'>
```

In [31]:

```
x1=float(x)
y1=int(y)
print(x1)
print(y1)
```

```
1.0
35
```

In [33]:

```
name="Hi"
print(type(name))
```

```
<class 'str'>
```

1. Program to accept two numbers from the user to add, subtract, multiply and divide it and print the results.

In [1]:

```
n1=int(input('Enter a number: ')) #BASIC MATHEMATICAL OPERATIONS
n2=int(input('Enter a number: '))
a=n1+n2;
s=n1-n2;
p=n1*n2;
d=n1/n2;
print('The sum of the two numbers = ',a)
print('The difference of the two numbers = ',s)
print('The product of the two numbers = ',p)
print('The quotient of the two numbers = ',d)
```

```
Enter a number: 3
Enter a number: 4
The sum of the two numbers =  7
The difference of the two numbers =  -1
The product of the two numbers =  12
The quotient of the two numbers =  0.75
```

2. Program to check the biggest of three numbers entered by the user.

In [4]:

```
a1=int(input('Enter a number: '))  # RELTIONAL OPERATORS, CONDITIONAL STATEMENTS
a2=int(input('Enter a number: '))
a3=int(input('Enter a number: '))
if(a1>=a2) and (a1>=a3):
    largest=a1
elif(a2>=a1) and (a2>=a3):
    largest=a2
else:
    largest=a3
print("")
print('The largest of the three numbers is',largest)
```

```
Enter a number: 9
Enter a number: 0
Enter a number: 9

The largest of the three numbers is 9
```

3. Write a program to compute the factorial of a number.

In [5]:

```
fact=1;                          # LOOPING STRUCTURES
f=int(input('Enter the number: '))
while(f>0):
    fact=fact*f
    f=f-1
    print('The factorial is ',fact)
```

```
Enter the number: 2
The factorial is  2
The factorial is  2
```

4. Program to generate the multiplication table of a number (from 1 to 10)

In [6]:

```
x=int(input('Enter the number: '))
for i in range(1,11):
    print(x,' x ',i,' = ',x*i)
```

```
Enter the number: 5
5  x  1  =  5
5  x  2  =  10
5  x  3  =  15
5  x  4  =  20
5  x  5  =  25
5  x  6  =  30
5  x  7  =  35
5  x  8  =  40
5  x  9  =  45
5  x  10  =  50
```

Lists

In [8]:

```
groceries=['milk','bread','eggs','curd','sugar'] # defining a list
groceries
```

Out[8]:

```
['milk', 'bread', 'eggs', 'curd', 'sugar']
```

In [9]:

```
groceries[2] #index of list

```

Out[9]:

```
['milk', 'bread']
```

In [14]:

```
groceries[-1]#negative indexing means begining from the end

```

Out[14]:

```
'tea powder'
```

In [15]:

```
groceries[0:2]#specify a range of indexes by specifying where to starts and where to er
```

Out[15]:

```
['milk', 'bread']
```

In [13]:

```
groceries.append('tea powder')
print(groceries)
```

```
['milk', 'bread', 'eggs', 'curd', 'sugar', 'tea powder', 'tea powder', 'tea
powder', 'tea powder']
```

In [17]:

```
groceries[2]='wheat'
print(groceries)
```

```
['milk', 'bread', 'wheat', 'curd', 'sugar', 'tea powder', 'tea powder', 'tea
powder', 'tea powder']
```

In [18]:

```
del groceries[0]
print(groceries)
```

```
['bread', 'wheat', 'curd', 'sugar', 'tea powder', 'tea powder', 'tea powde
r', 'tea powder']
```

6. Write a program to define a list and check whether a particular item belongs to the list

In [19]:

```

list1=[10,20,30,40,50,60]
if 30 in list1:
    print("present")
else:
    print("absent")
```

```
present
```

7. Python code to demonstrate the working of sum() in list

In [20]:

```

numbers = [1,2,3,4,5,1,4,5]

# start parameter is not provided
Sum = sum(numbers)
print(Sum)

# start parameter is provided
Sum = sum(numbers,10)
print(Sum)
```

```
25
35
```

Tuple

In [21]:

```
name=('Arushi','Kiran') #declaring a tuple
name
```

Out[21]:

```
('Arushi', 'Kiran')
```

In [22]:

```
name[0]='Harry Potter' # tuples are immutable
name
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-22-62da7bf58880> in <module>
----> 1 name[0]='Harry Potter'
      2 name

TypeError: 'tuple' object does not support item assignment
```

In [24]:

```
tuple1 = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)

x = tuple1.index(8) #Search for the first occurrence of the value 8, and return its pos

print(x)

y = tuple1.count(7)  #Return the number of times the value 5 appears in the tuple

print(y)
```

```
3
2
```

7. Python code to demonstrate the working of sum() in tuples

In [25]:

```

numbers = tuple((1,2,3,4,5,1,4,5))
# start parameter is not provided
Sum = sum(numbers)
print(Sum)

# start parameter is not provided
Sum = sum(numbers,10)
print(Sum)
```

```
25
35
```

Exercise problem:

1. Compute greatest common divisor using Euclidean Algorithm.

2. Find roots of a quadratic equation and also determine their nature.

3. Find if a given number is perfect number or not.

4. Check if a given input is a palindrome.

In [ ]:

```
1
```