# DAY 1

Language Fundamentals:

1. manipulate using a list

In [8]:
```
# 1.1 to add new elements to the end of the list
list = [11, 22,76,57]
```

In [9]:
```
list.append(44)
list.append(69)
print(list)
```

```
[11, 22, 76, 57, 44, 69]
```

In [10]:
```
#1.2 to reverse elemets in the list
list[::-1]
```

Out[10]:
```
[69, 44, 57, 76, 22, 11]
```

In [11]:
```
# 1.3 to display the same list of elements multiple times
# list*n where n is the number of times you want the list to multiply
list * 2
```

Out[11]:
```
[11, 22, 76, 57, 44, 69, 11, 22, 76, 57, 44, 69]
```

In [12]:
```
# 1.4 to concatenate two lists
list1 = [11, 22,76,57]
list2 = [44,69]
newlist = list1+list2
newlist
```

Out[12]:
```
[11, 22, 76, 57, 44, 69]
```

In [13]:
```
#1.5 to sort the elements in the list in ascending ord
newlist.sort()
newlist
```

Out[13]:
```
[11, 22, 44, 57, 69, 76]
```

1. Write a Python program to do in the Tuples 2.1 manipulating using tuples

In [15]:
```
#2.2 to add new elements to the end of the tuple
tuple1 = ('bangalore','chennai','mumbai')
tuple2= tuple1 + ('delhi',)
tuple1
```

Out[15]:
```
('bangalore', 'chennai', 'mumbai', 'delhi')
```

In [17]:
```
#2.3 reverse the elements in the list
tuple1[::-1]
```

Out[17]:
```
('delhi', 'mumbai', 'chennai', 'bangalore')
```

In [20]:
```
#2.4 to display the same list of elements multiple times
n=int(input('how many times to be multiplied\t'))
t=tuple2 * n
t
```

Out[20]:
```
how many times to be multiplied 2
('bangalore',
 'chennai',
 'mumbai',
 'delhi',
 'bangalore',
 'chennai',
 'mumbai',
 'delhi')
```

In [30]:
```
#2.5 to concatenate two lists
tuple1 = ('bangalore','chennai','mumbai')
tuple2= ('delhi',)
concat = tuple1 + tuple2
concat
```

Out[30]:
```
('bangalore', 'chennai', 'mumbai', 'delhi')
```

In [32]:
```
# 2.6 to sort in asc order
tuple1 = ('bangalore', 'chennai', 'mumbai')
tuple2 = ('delhi',)
concat = tuple1 + tuple2
sorted_tuple = tuple(sorted(concat))
print(sorted_tuple)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[32], line 5
      3 tuple2 = ('delhi',)
      4 concat = tuple1 + tuple2
----> 5 sorted_tuple = tuple(sorted(concat))
      6 print(sorted_tuple)

TypeError: 'tuple' object is not callable
```

1. Write a python program to implement the following using list.

In [34]:
```
#3.1 Create a list with integers (min 10 numbers)
list = [11, 12, 33, 34, 55, 69, 17, 98, 29, 10]
list
```

Out[34]:
```
[11, 12, 33, 34, 55, 69, 17, 98, 29, 10]
```

In [35]:
```
#3.2 how to Display the last number in the list
list[-1]
```

Out[35]:
```
10
```

In [36]:
```
#3.3 command to Display the values from the list [0:4]
list[0:4]
```

Out[36]:
```
[11, 12, 33, 34]
```

In [37]:
```
#3.4 command to display the values from the list [2:0]
list[2:0]
```

Out[37]:
```
[]
```

In [38]:
```
#3.5 command to Display the values from the list [:6]
list[:6]
```

Out[38]:
```
[11, 12, 33, 34, 55, 69]
```

1. write a python program : tuple1=(10,50,20,40,30)

In [39]:
```
# 4.1 to display the elements from 10 and 50 from tuple1
tuple1 = (10, 50, 20, 40, 30)
tuple1[0:2]
```

Out[39]:
```
(10, 50)
```

In [40]:
```
#4.2 to Display the length of tuple1
len(tuple1)
```

Out[40]:
```
5
```

In [41]:
```
#4.3 To find the min element from tuple1
min(tuple1)
```

Out[41]:
```
10
```

In [42]:
```
#4.4 to Add all elements from tuple1
sum(tuple1)
```

Out[42]:
```
150
```

In [43]:
```
#4.5 To display the same tuple1 multiple times
tuple1 * 3
```

Out[43]:
```
(10, 50, 20, 40, 30, 10, 50, 20, 40, 30, 10, 50, 20, 40, 30)
```

1. write a python program :

In [44]:
```
#5.1  Calculate the length of a string
string1 = "Bangalore is the best city"
len(string1)
```

Out[44]:
```
26
```

In [46]:
```
#5.2 To reverse the words in a string
string1[::-1]
```

Out[46]:
```
'ytic tseb eht si erolagnaB'
```

In [50]:
```
rev_string1 = ' '.join(reversed(string1.split()))
rev_string1
```

Out[50]:
```
'city best the is Bangalore'
```

In [51]:
```
#5.3 Display the same string multiple times
string1 * 3
```

Out[51]:
```
'Bangalore is the best cityBangalore is the best cityBangalore is the best city'
```

In [53]:
```
#5.4 Concatenate two strings
string2 = ' in INDIA'
concat= string1 + string2
concat
```

Out[53]:
```
'Bangalore is the best city in INDIA'
```

In [54]:
```
#5.5 Using string slicing to display "india" from "south india"
str1 = "south india"
str1[6:]
```

Out[54]:
```
'india'
```

1. Perform the follwing

In [55]:
```
# 6.1 Create the dictionary
dict = {'name': 'suhas', 'age': 23, 'sex': 'male', 'country':'india'}
```

In [58]:
```
#6.2Accessing values and keys in the dictionary
print(dict['name'])
print(dict['age'])
```

```
suhas
23
```

In [60]:
```
# 6.3 Update the dictionary using a function
def update_dict(dict, key, value):
    dict[key] = value
update_dict(dict, 'age', 22)
dict['age']
```

Out[60]:
```
22
```

In [62]:
```
# 6.4 Clear and delete dictionary values
dict.clear()
dict
```

Out[62]:
```
{}
```

In [63]:
```
del dict
```

In [70]:
```
list = [1, 2, 3, 4, 5]
del list[2]
list
```

Out[70]:
```
[1, 2, 4, 5]
```

In [1]:
```
#9 program to display a number from 1 to 100
for i in range(1, 101):
    print(i)
```

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
```

```
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

In [67]:
```
#9 write a python program to find the sum of all items in a tuple
tuple = (10, 50, 20, 40, 30)
sum(tuple)
```

Out[67]:
```
150
```

In [69]:
```
list = ['hello', 'Dear', 'hOw', 'ARe', 'You']
for i in list:
    if i[1].isupper():
        print(i)
```

```
hOw
ARe
```

In [22]:
```
#13 weight on moon
Gfm = 1.622
Gfe = 9.81
WOE = {'John': 45, 'Shelly': 65, 'Marry': 35}
WOM = list(map(lambda x: (x[1] * Gfm) / Gfe, WOE.items()))

WeightOnMoonDict = dict(zip(WOE.keys(), WOM))
for name, weight in WeightOnMoonDict.items():
    print(f"{name}'s weight on the moon: {weight:.2f} kg")
```

```
John's weight on the moon: 7.44 kg
Shelly's weight on the moon: 10.75 kg
Marry's weight on the moon: 5.79 kg
```

# CONTROL STRUCTURES

In [5]:
```python
# 1 first N prime numbers
def prime(num):
    if num <= 1:
        return False
    elif num <= 3:
        return True
    elif num % 2 == 0 or num % 3 == 0:
        return False
    i = 5
    while i * i <= num:
        if num % i == 0 or num % (i + 2) == 0:
            return False
        i += 6
    return True

def first_n_primes(N):
    primes = []
    num = 2
    while len(primes) < N:
        if prime(num):
            primes.append(num)
        num += 1
    return primes


N = int(input("Enter the value of N: "))

prime_numbers = first_n_primes(N)
print(f"The first {N} prime numbers are: {prime_numbers}")
```

```
Enter the value of N: 10
The first 10 prime numbers are: [2, 3, 5, 7, 11, 13, 17, 19, 23, 29]
```

In [25]:
```python
#2 to find net salary
bs= float(input("Enter Basic Salary: "))
hra = float(input("Enter House Rent Allowance: "))
ta = float(input("Enter Travel Allowance: "))
da = float(input("Enter Dearness Allowance:"))

gross_sal = bs + hra + ta + da
tax = 0.10 * gross_sal
net = gross_sal - tax

print(f"Gross Salary: {gross_sal:.2f}")
print(f"Tax: {tax:.2f}")
print(f"Net Salary: {net:.2f}")
```

```
Enter Basic Salary: 13500
Enter House Rent Allowance: 8500
Enter Travel Allowance: 4500
Enter Dearness Allowance:4500
Gross Salary: 31000.00
Tax: 3100.00
Net Salary: 27900.00
```

In [24]:
```python
# 3 Write a python program to search for a string in the givenlist
str = ['suhas', 'mphasis', 74, 'mango', 'hp', 'messi']
f = 'messi'

# check if string is present in list
```

```
if str.count(f) > 0:
    print(f'{f} is present in the list')
else:
    print(f'{f} is not present in the list')
```

```
messi is present in the list
```

In [35]:
```
# 4 Write a Python function that accepts a string and calculates the number of uppe
def count_upper_lower(str1):
    upper= 0
    lower= 0
    for i in str1:
        if i.isupper():
            upper += 1
        elif i.islower():
            lower += 1
    print("No. of Upper case : ", upper)
    print("No. of Lower case : ", lower)

str1 = "Mphasis,NeXt AppliED"
count_upper_lower(str1)
```

```
No. of Upper case :  6
No. of Lower case :  12
```

In [37]:
```
#5 Write a program to display the sum of odd numbers and even numbers that fall bet
odd = 0
even = 0
for i in range(12, 37):
    if i % 2 == 0:
        even += i
    else:
        odd += i

print("Sum of odd numbers:", odd)
print("Sum of even numbers:",even)
```

```
Sum of odd numbers: 288
Sum of even numbers: 312
```

In [42]:
```
#6 tables of n number
def tables(n):
    for i in range(1,11):
        print(n, 'x', i, '=', n*i)

n = int(input("Enter the number to print the table of: "))
tables(n)
```

```
Enter the number to print the table of: 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
```

In [44]:
```
# 7 first 10 prime numbers
def prime(num):
    if num <= 1:
        return False
    elif num <= 3:
```

```
        return True
    elif num % 2 == 0 or num % 3 == 0:
        return False
    i = 5
    while i * i <= num:
        if num % i == 0 or num % (i + 2) == 0:
            return False
        i += 6
    return True

def first_n_primes(N):
    primes = []
    num = 2
    while len(primes) < N:
        if prime(num):
            primes.append(num)
        num += 1
    return primes


N = 10 # int(input("Enter the value of N: "))

prime_numbers = first_n_primes(N)
print(f"The first {N} prime numbers are: {prime_numbers}")
```

```
The first 10 prime numbers are: [2, 3, 5, 7, 11, 13, 17, 19, 23, 29]
```

In [45]:

```
#8 airthmetic operation
def calculator(n1, n2,sign):
    if operator == "+":
        result = n1 + n2
        print(n1, "+", n2, "=", result)
    elif operator == "-":
        result = n1 - n2
        print(n1, "-", n2, "=", result)
    elif operator == "*":
        result = n1 * n2
        print(n1, "*", n2, "=", result)
    elif operator == "/":
        result = n1 / n2
        print(n1, "/", n2, "=", result)
    else:
        print("Invalid operator")

n1 = int(input("Enter the first number: "))
n2 = int(input("Enter the second number: "))
operator = input("Enter the operator (+, -, *, /): ")

calculator(n1, n2, operator)
```

```
Enter the first number: 1
Enter the second number: 4
Enter the operator (+, -, *, /): *
1 * 4 = 4
```

In [8]:

```
#9 Function to convert Celsius to Fahrenheit
def cel(c):
    farh = (c * 9/5) + 32
    return farh
c = float(input("Enter temperature in Celsius: "))
farh = cel(c)
print(f"{c} degrees Celsius is equal to {farh} degrees Fahrenheit.")
```

```
Enter temperature in Celsius: 45
45.0 degrees Celsius is equal to 113.0 degrees Fahrenheit.
```

In [47]:
```
# 10 Write a python program to find a maximum and minimum number in a
def maxmin(list1):
    max= list1[0]
    min = list1[0]
    for i in list1:
        if i > max:
            max= i
        if i < min:
            min = i
    print("Maximum number in the list is:", max)
    print("Minimum number in the list is:", min)

list1 = [75, 33, 69, 420, 55, 16, 77, 0, 9, 100]
maxmin(list1)
```

```
Maximum number in the list is: 420
Minimum number in the list is: 0
```

In [50]:
```
# 11 & 12 Seconds in a month and year
seconds_in_a_day = 24 * 60 * 60
seconds_in_a_month = seconds_in_a_day * 30
print("Number of seconds in a 30-day month:", seconds_in_a_month)

siy = 365 * 24* 60 * 60
print("Number of seconds in a year:", siy)
```

```
Number of seconds in a 30-day month: 2592000
Number of seconds in a year: 31536000
```

In [11]:
```
# 13 A high-speedtrain can travel at an average speed of 150 mph,
# how long will it take a train travelling at this speed to travel from Londo
d= 414
s= 150

t = d/ s
remt = (d % s) * 60 /s

print(f"Time: {t} hours {remt} minutes")
```

```
Time: 2.76 hours 45.6 minutes
```

In [51]:
```
#15 age of ram sam and khan
ram = int(input("Enter Ram's age: "))
sam = int(input("Enter Sam's age: "))
khan= int(input("Enter Khan's age: "))

eldest = max(ram, sam, khan)
youngest = min(ram, sam, khan)

print("Eldest:", eldest)
print("Youngest:", youngest)
```

```
Enter Ram's age: 12
Enter Sam's age: 14
Enter Khan's age: 11
Eldest: 14
Youngest: 11
```

In [57]:
```
#16 Write a python program to rotate a list by right n times with and without slic
def rotate(list1, n):
    for i in range(n):
        temp = list1[-1]
        for j in range(len(list1) - 1, 0, -1):
            list1[j] = list1[j - 1]
```

```
        list1[0] = temp
    print("Rotated list without slicing:", list1)

list1 = [5, 10,15, 25]
n = 3
rotate(list1, n)
```

```
Rotated list without slicing: [10, 15, 25, 5]
```

In [18]:
```
# 17 pattern

for i in range(5):
        for j in range(i + 1):
            print("*", end=" ")
        print()
```

```
*
* *
* * *
* * * *
* * * * *
```

In [19]:
```
n = 4
for i in range(n):

    for j in range(i + 1):
        print("*", end=" ")
    print()

for i in range(n - 1, 0, -1):
    for j in range(i):
        print("*", end=" ")
    print()
```

```
*
* *
* * *
* * * *
* * *
* *
*
```

In [20]:
```
str = "Python"
for i in range(len(str)):
    substring = str[:i+1]
    print(substring, end="\t")
```

```
P	Py	Pyt	Pyth	Pytho	Python
```

In [21]:
```
str = "Python"
for i in range(len(str)):
    substring = str[:i+1]
    print(substring)
```

```
P
Py
Pyt
Pyth
Pytho
Python
```

In [ ]: