python is a general purpose,interpreter,high level languages.it was created by guido van rossum in 1991.

use of python-- Machine learning IOT python serinum deep learning Python is a dynamically typed language,object oriented programming language.(data typ is internally defined by interpreter) Python is platform independent.with python interactive interpreter,it iseasy to check python command

Write a program to write a poem twinkle twinkle little ....

In [1]:
```
print(print('''Twinkle,twinkle,little star,
How i wonder what u r!
Up above the  world so high,
Like a diamond in the sky.'''))
#this problem is a solution of problem multiline comment
```

```
Twinkle,twinkle,little star,
How i wonder what u r!
Up above the  world so high,
Like a diamond in the sky.
None
```

#agr ...akela hai to comment hoga otherwise vo print m hai to vo string literal hai.

In [2]:
```
#write a python program to print the content of a directory using os module .sear
import os
print(os.listdir())
```

```
['.ipynb_checkpoints', 'ASSIGNMENT.ipynb', 'Book3.xlsx', 'codewithharry 1.ipyn
b', 'complete python.ipynb', 'COMPLETE.ipynb', 'complete.py', 'EBOOK', 'friend
s.csv', 'mysirg python.ipynb', 'mysirg-sequence.ipynb', 'Numpy.ipynb', 'Pandas
- Copy.html', 'Pandas.ipynb', 'paras', 'phishing.txt', 'practice Q.ipynb', 'pyt
hon basics.ipynb', 'Untitled.ipynb', 'Untitled1.ipynb']
```

variable is the name given to a memory location in a program.

variable contains to store a value, k/w are reserved words in python.there are 33 k/w in which tree are True,False,None identifier --class/function/variable name datatype-int,float,string ,boolean,none variable names cab contain alphabets,digits and underscore and it can only start with an alphabet and underscore. a variable name cannot start with a digit ,nowhitspace is allowed

to be used inside a variable name The first character of the variable must be an alphabet or underscore (_). Identifier name must not contain any white-space, or special character (!, @, #, %, ^, &, *).

In [3]:
```
a="harry"
b=345
c=45.32
d='harry'
e='''harry is a good boy'''#string
print(a)
print(b)
print(c)
print(d)
print(e)
print(type(a))
print(type(b))
print(type(c))
print(type(d))
```

```
harry
345
45.32
harry
harry is a good boy
<class 'str'>
<class 'int'>
<class 'float'>
<class 'str'>
```

In [4]:
```
name = "A"
Name = "B"
naMe = "C"
NAME = "D"
n_a_m_e = "E"
_name = "F"
name_ = "G"
_name_ = "H"
na56me = "I"

print(name,Name,naMe,NAME,n_a_m_e, NAME, n_a_m_e, _name, name_,_name, na56me)
```

```
A B C D E D E F G F I
```

In [5]:
```
#ASSIGNING SINGLE VALUES TO MULTIPLE VARIABLES
x=y=z=50
print(x)
print(y)
print(z)
#ASSIGNING MULTIPLES VALUES TO THE MULTIPLE VARIABLE
a=10
b=30
c=90
print("a")
print("b")
print("c")

print(a)
print(b)
print(c)
```

```
50
50
50
a
b
c
10
30
90
```

In [6]:
```
w=10000000000000000000000000000000000008999999999999
w=w+1
print(w)
print(type(w))
#PRINT SINGLE VARIABLE
a=10
print(a)
print(type(a))
#printing multiple variables
m=17
b=34
print(m,b)
```

```
100000000000000000000000000000000000009000000000000
<class 'int'>
10
<class 'int'>
17 34
```

In [7]:
```
print(None)
```

```
None
```

python data types

NUMERIC-INTEGER,COMPLEX NUMBER,FLOAT dictionry boolean sequence type-string,list tuples set

In [8]:
```
a = 5
print("The type of a", type(a))

b = 40.5
print("The type of b", type(b))

c = 1+3j
print("The type of c", type(c))
print(" c is a complex number", isinstance(1+3j,complex))
```

```
The type of a <class 'int'>
The type of b <class 'float'>
The type of c <class 'complex'>
 c is a complex number True
```

In [9]:
```
# arithmetic operators(/,*,//,**,%,+,-)
a=4
b=6
print("the value of 3+4 is",3+4)
#assignment operator(=,+=,-=)
a=78
a+=2
print(a)
```

```
the value of 3+4 is 7
80
```

comparsion operator(==,>=,<=,>,<,!=)

In [10]:
```
b=(14>=7)
print(b)
c=(14<=7)
print(c)
d=(12>6)
print(d)
o=(12<6)
print(o)
```

```
True
False
True
False
```

logical operator(and,or,not)

In [11]:
```
bool1=True
bool2=False
print("The value of bool1 and bool2 is",(bool1 and bool2))
print("The value of bool1 and bool2 is",(bool1 or bool2))
print("The value of bool1 and bool2 is",(not bool2))
```

```
The value of bool1 and bool2 is False
The value of bool1 and bool2 is True
The value of bool1 and bool2 is True
```

typecasting

In [12]:
```
a="3534"#string
a=int(a)
print(type(a))
print(a)#integer
```

```
<class 'int'>
3534
```

In [13]:
```
int("32")#string to int conversion
```

Out[13]:
```
32
```

In [14]:
```
#string to int conversion
str(32)
```

Out[14]:
```
'32'
```

In [15]:
```
float(32.8)
```

Out[15]:
```
32.8
```

In [16]:
```
name=input("enter your name:")
```

```
enter your name:sonu
```

In [17]:
```
a=input("enter your name:")
print(type(a))
```

```
enter your name:sonu
<class 'str'>
```

In [18]:
```
a=90
b=8
print("the sum of a and b is ",a+b)
```

```
the sum of a and b is  98
```

how to calculate average of number

In [19]:
```
#how to calculate average of to number?
a=int(input("enter first number:"))
b=int(input("enter second number:"))
avg=(a+b)/2
print("the average of a and b is ",avg)
```

```
enter first number:34
enter second number:34
the average of a and b is  34.0
```

In [20]:
```
b="harry"
print(b)
print(type(b))
```

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

In [21]:
```
greeting ="good morning,"   #adding to string(concatating to string)
name="harry"
print(type(name))
c=greeting+name
print(c)
print(name[0])
print(name[1:3])
print(name[1:-1])
print(name[1:])#is same as print(name[1:5])
print(name[-4:-1])
len("harry")
len("good morning")
print(greeting[1:6])
print(greeting[1:-1])
print(greeting[1:])
```

```
<class 'str'>
good morning,harry
h
ar
arr
arry
arr
ood m
ood morning
ood morning,
```

Slicing

In [22]:
```
word="amaging"
print(word[:7])
print(word[0:7])
print(word[0:])
print(word[0:7])
```

```
amaging
amaging
amaging
amaging
```

string is a data type in python.string is a sequene of characters in quotes.we can primarily write a string in two ways.

1.single quoted string 'sonu' 2.double quoted string "sonu" 3.triple quoted string """sonu"""

In [23]:
```
#string function...
story="once upon a time there was a you tuber named harry who uploaded python cou
print(len(story))

print(story.count("o"))
print(story.count("a"))
print(story.find("a"))
```

```
88
8
6
10
```

In [ ]:
```

```

In [24]:
```
#write a program to display a user entered name followed by goodnoon using input
name=input("enter the name")
print("GOODNOON-"+name)
```

```
enter the nametina
GOODNOON-tina
```

In [25]:
```
#escape sequencing ....\n,\t,\,\\(newline,tab,single qote,backslash etc)
story="Harry is good.\nHe\t is verygood"
print(story)
```

```
Harry is good.
He	 is verygood
```

list and tuples

The list has the following characteristics: The lists are ordered. The element of the list can access

by index. The lists are the mutable type. The lists are mutable types. A list can store the number of various elements.

In [26]:
```
#create a list using []
a=[1,46,89,98,78]
print(a)
print(a[0])
print(a[3])
print(a[2])
#change the vaue of list using
a[0]=12
print(a)
```

```
[1, 46, 89, 98, 78]
1
98
89
[12, 46, 89, 98, 78]
```

In [27]:
```
#List slicing
friends =["shaily","sonu","divya","tom",89]
print (friends[0:4])
print(friends[:-4])
```

```
['shaily', 'sonu', 'divya', 'tom']
['shaily']
```

In [28]:
```
thislist = ["apple", "banana", "cherry"]
[print(x) for x in thislist]
```

```
apple
banana
cherry
```

Out[28]:
```
[None, None, None]
```

In [29]:
```
l=[1,23,45,67,89,90,23,4,5]
print(l)
print(len(l))
print(l.sort())
l.sort()
print(l)
```

```
[1, 23, 45, 67, 89, 90, 23, 4, 5]
9
None
[1, 4, 5, 23, 23, 45, 67, 89, 90]
```

In [30]:
```
l.reverse()
print(l)
```

```
[90, 89, 67, 45, 23, 23, 5, 4, 1]
```

In [31]:
```
l
l.append(67)#insert at the last
print(l)
```

```
[90, 89, 67, 45, 23, 23, 5, 4, 1, 67]
```

In [32]:
```
l
l.insert(1,4567)
print(l)
```

```
[90, 4567, 89, 67, 45, 23, 23, 5, 4, 1, 67]
```

In [33]:
```
l
```

Out[33]:
```
[90, 4567, 89, 67, 45, 23, 23, 5, 4, 1, 67]
```

In [34]:
```
t=(1,2,3,4,5,67,8,6)
print(t.count(1))
```

```
1
```

In [35]:
```
print(t.count(2))
print(t.index(4))
```

```
1
3
```

In [37]:
```
f1=input("enter fruit number1")
f2=input("enter fruit number2")
f3=input("enter fruit number3")
f4=input("enter fruit number4")
myfruitlist:list
myfruitlist=[f1,f2,f3,f4]
print(myfruitlist)
```

```
enter fruit number1apple
enter fruit number2lichi
enter fruit number3pineapple
enter fruit number4peech
['apple', 'lichi', 'pineapple', 'peech']
```

In [38]:
```
a=[2,3,4,5,690]
print(a[0]+a[1]+a[2]+a[3]+a[4])
```

```
704
```

In [39]:
```
list = [1,2,3,4,5,6,7]
print(list[0])
print(list[1])
print(list[2])
print(list[3])
# Slicing the elements
print(list[0:6])
# By default the index value is 0 so its starts from the 0th element and go for i
print(list[:])
print(list[2:5])
print(list[1:6:2])
```

```
1
2
3
4
[1, 2, 3, 4, 5, 6]
[1, 2, 3, 4, 5, 6, 7]
[3, 4, 5]
[2, 4, 6]
```

In [40]:
```
list = [1, 2, 3, 4, 5, 6]
print(list)
# It will assign value to the value to the second index
list[2] = 10
print(list)
# Adding multiple-element
list[1:3] = [89, 78]
print(list)
# It will add value at the end of the list
list[-1] = 25
print(list)
```

```
[1, 2, 3, 4, 5, 6]
[1, 2, 10, 4, 5, 6]
[1, 89, 78, 4, 5, 6]
[1, 89, 78, 4, 5, 25]
```

In [41]:
```
list = ["John", "David", "James", "Jonathan"]
for i in list:
    # The i variable will iterate over the elements of the List and contains each
    print(i)
```

```
John
David
James
Jonathan
```

In [42]:
```
#removing element to te list
list = [0,1,2,3,4]
print("printing original list: ");
for i in list:
    print(i,end=" ")
list.remove(2)
print("\nprinting the list after the removal of first element...")
for i in list:
    print(i,end=" ")
```

```
printing original list:
0 1 2 3 4
printing the list after the removal of first element...
0 1 3 4
```

Example: 1- Write the program to remove the duplicate element of the list.

In [43]:
```
list1 = [1,2,2,3,55,98,65,65,13,29]
# Declare an empty list that will store unique values
list2 = []
for i in list1:
    if i not in list2:
        list2.append(i)
print(list2)
```

```
[1, 2, 3, 55, 98, 65, 13, 29]
```

In [ ]:
```
#Declaring the empty list
l =[]
#Number of elements will be entered by the user
n = int(input("Enter the number of elements in the list:"))
# for loop to take the input
for i in range(0,n):
    # The input is taken from the user and added to the list as the item
    l.append(input("Enter the item:"))
print("printing the list items..")
# traversal loop to print the list items
for i in l:
    print(i, end = "  ")
```

In [ ]:

In [ ]: