Today Topics

- Python Basics

variable

- Is a container that stores the value/data
- let x=10 : 10 is storing in a var called x

In [4]:

```
value=10
print('value=',value)
```

```
value= 10
```

variable assignment

In [7]:

```
x=10
y=32
z=90
print(x,y,z,sep="o")
```

```
10o32o90
```

In [8]:

```
x,y,z=10,56,7
print("x:",x)
print("y:",y)
print("z:",z)
```

```
x: 10
y: 56
z: 7
```

In [9]:

```
a=65
b=65
c=65
print(a,b,c)
```

```
65 65 65
```

In [10]:

```
a=b=c=90
print(a,b,c,sep=",")
```

```
90,90,90
```

keywords in python

- keyword is the reserved word that contains special meaning and purpose
- 36 keywords

In [12]:

```
import keyword
keyword.kwlist #displays the list of keywords
```

...

In [14]:

```
import keyword as stu
li=stu.kwlist
print(li)
```

...

Errors

- an invalid statement said to be error
- 3 types of errors
  1. Syntax Error
     - Error related to syntax
  2. Value Error and
     - type error,value error,filenot found error,exceptions etc.
  3. Indentation Error
     - Error related to space

In [17]:

```
# syntax error
print()
```

In [18]:

```
print("") # empty str
```

In [19]:

```
print("started with double and ended with single')
```

...

In [20]:

```
help(print)
```

...

Highlighting the sub string in o/p

In [22]:

```
print("I am from 'LBRCE' located at Mylavaram")
```

...

In [24]:

```
print('I am from "LBRCE" located at Mylavaram')
```

...

In [25]:

```
student='Ramu'
student+90
```

...

In [26]:

```
x+90
```

Out[26]:

```
100
```

Data types

- int,float,complex,str,binary

input()

- predefined function in python
- reads the string data from user dynamically be default
  - var=input()

In [32]:

```
clg=input()
print(clg)
```

```
python
python
```

In [33]:

```
clg

```

Out[33]:

```
'python'
```

In [38]:

```
name=input("enter your name:")
name
```

...

In [39]:

```
name
```

Out[39]:

```
'elisha'
```

In [42]:

```
type(name) # checks the datatype of the variable
```

Out[42]:

```
str
```

Type Conversion

- convesion of the datatype of the var
- from int to float
- from float to int
- int to str and vice verca
- 2types
    1. Implicit Conversion
        - Default by interpreter
    2. Explicit Conversion
        - User conversion

In [43]:

```
1 90+5.6
```

Out[43]:

```
95.6
```

In [44]:

```
1 99+45
```

Out[44]:

```
144
```

In [49]:

```
1 9+6.7
```

Out[49]:

```
15.7
```

In [47]:

```
1 abs(3.4766)
```

Out[47]:

```
3.4766
```

In [48]:

```
1 int(3.45)
```

Out[48]:

```
3
```

In [51]:

```
1 "ruthu"+"90"
```

Out[51]:

```
'ruthu90'
```

In [54]:

```
1 # add two values read from user
2 num=int(input("number:"))
3 num
```

...

In [58]:

```
gpa=float(input("enter your cgapa:"))
gpa
```

...

In [59]:

```
x
```

Out[59]:

```
10
```

In [60]:

```
bn=bin(x)
bn
```

Out[60]:

```
'0b1010'
```

In [63]:

```
val=int(input("interger value:"))
# 20
print("binary format=",bin(val))
```

```
interger value:20
binary format= 0b10100
```

In [64]:

```
cm=5+6j
cm
```

Out[64]:

```
(5+6j)
```

In [65]:

```
type(cm)
```

Out[65]:

```
complex
```

operators in python

- anything that performs an operations b/w operands
  - a+b: + is the op'r performing addition op'n on a&b
- Types of Operators
  - arithmetic
    - +,-,/,,//,*
  - assignment

    - =,+=,-=,/=,*=
- boolean
    - None,True,False
- bitwise
    - ^,|,&
- comparisional/relational
    - ==,!=,>,>=,<,<=
- identity operators
    - is, is not
- logical operators
    - and,or
- membership
    - in,not in
- operators used in print()
    - sep,end

In [66]:

```
# read 2 integers from user and
# perform all the arithmetic op'ns
```

In [68]:

```
fst=int(input("first value="))
sec=int(input('second value='))
print("sum=",fst+sec)
print("difference=",fst-sec)
print("product=",fst*sec)
print("divisor=",fst/sec)
print("exponential=",fst**sec)
print("int divisor=",fst//sec)
```

. . .

In [69]:

```
9>4
```

Out[69]:

```
True
```

In [70]:

```
st,sp=int(input()),int(input())
st=sp #
```

```
90
56
```

In [72]:

```
st==sp # comparision
```

Out[72]:

```
True
```

In [73]:

```
st is sp
```

Out[73]:

```
True
```

In [74]:

```
x=10
y=10

```

In [75]:

```
x==y
```

Out[75]:

```
True
```

In [76]:

```
x is y
```

Out[76]:

```
True
```

In [77]:

```
id(x)
```

Out[77]:

```
966750005840
```

In [78]:

```
id(y)
```

Out[78]:

```
966750005840
```

In [79]:

```
count=9
sim=8
id(sim)
```

Out[79]:

```
966750005776
```

In [80]:

```
id(count)
```

Out[80]:

```
966750005808
```

In [81]:

```
count!=sim
```

Out[81]:

```
True
```

In [82]:

```
count is 90
```

```
<>:1: SyntaxWarning: "is" with a literal. Did you mean "=="?
<>:1: SyntaxWarning: "is" with a literal. Did you mean "=="?
C:\Users\admin\AppData\Local\Temp\ipykernel_5212\2546295911.py:1: SyntaxWarn
ing: "is" with a literal. Did you mean "=="?
  count is 90
```

Out[82]:

```
False
```

In [96]:

```
val
```

Out[96]:

```
2.0
```

In [83]:

```
count
```

Out[83]:

```
9
```

In [84]:

```
x=10
```

In [85]:

```
1 x is 10
```

...

In [87]:

```
1 print(x is not 10)
```

...

In [88]:

```
1 sp
```

Out[88]:

```
56
```

In [89]:

```
1 st=67
2 sp=100
3
```

In [92]:

```
1 st=st+9 # st:85
2 sp=st-56 # 85-56
3 print(st,sp)
```

```
85 29
```

In [97]:

```
1 val=50
2 val/=10 # val=val/10
3 # val/10
4 new=val*5
5 hi=new+5
6 print(hi)
7
```

...

In [95]:

```
1 20/2
```

Out[95]:

```
10.0
```

In [99]:

```
1 hi>100
```

Out[99]:

```
False
```

In [100]:

```
1 print(val,new,hi,sep=",")
```

```
5.0,25.0,30.0
```

In [101]:

```
1 print(val,new,hi,end=",")
```

```
5.0 25.0 30.0,
```

In [102]:

```
1 5*'hi'
```

Out[102]:

```
'hihihihihi'
```

In [106]:

```
1 msg='department'
2 print(msg)
```

```
department
```

In [109]:

```
1 print(*msg,sep=",")
```

```
d,e,p,a,r,t,m,e,n,t
```

In [110]:

```
1 msg
```

. . .

In [111]:

```
1 "a" in msg
```

. . .

In [112]:

```
1 type('a')
```

. . .

In [113]:

```
"part" in msg
```

...

In [114]:

```
'part' in 'partial'
```

...

In [115]:

```
'r' not in msg
```

...

conditional statements

- which specifies the condition in our program
- 3 statements
  1. if
  2. else
  3. elif
- **syntax**
  - if (condition):
    - statements goes here

In [116]:

```
val
```

Out[116]:

```
5.0
```

In [117]:

```
# read a number from user and
# print I am good if it is greater than 50
```

In [119]:

```
num=int(input()) # number
if num>50:
    print("I am good")
```

```
1105
I am good
```

In [120]:

```
# read the input from user and print okay
# if it is in your name
char=input()
if char in "Nandini":
    print("okay")
```

```
a
okay
```

In [121]:

```
char=input("character:")
name=input("your name:")
if char in name:
    print("okay")
else:
    print('not okay')
```

...

In [ ]:

```
# read an integer from user and
# print your friend name if it is even
# print your enemy name if not

```

In [122]:

```
num=int(input())
if num%2==0:
    print("I am your friend")
else:
    print("I am your fighter")
```

```
9
I am your fighter
```

In [127]:

```
# read a value from user
# print hi if it is +ve integer
# print hello if it is -ve integer
value=int(input("enter the value:"))
if value<0:
    print("hi")
else:
    print("hello")
```

...

In [129]:

```
# read a value from user and print
# good it is in b/w 20 and 100
# okay it is greater than 100
# bad it is in b/w 1 and 20
n=int(input())
if n in range(20,100):
    print("good")
else:
    print("bad")
```

```
556
bad
```

In [130]:

```
range(10,100)
```

Out[130]:

```
range(10, 100)
```

In [133]:

```
#elif
n=int(input()) #
if n in range(20,101): #
    print("good")
elif n in range(1,21):
    print("okay..!")
else:
    print("bad")
```

. . .

In [136]:

```
# read 3 values from user and
# print the biggest one
n=int(input("first number"))
a=int(input("second number"))
g=int(input("third number"))
# 56,90,78
if n>a and n>g:
    print(n,'is the biggest value')
elif a>n and a>g:
    print(a,'is the biggest value')
else:print(g,"is the biggest value")
```

. . .

read an year from user and check whether it is leap or not

In [139]:

```
year=int(input())
if year%400==0 or (year%4==0 and year%100!=0):
    print("leap year")
else:print("non leap year")
```

...

In [140]:

```
# read a number from user and
# if it is the muliple of 3&7 print common
# if it is the factor of 93 print okay
# if not print bye
```

In [145]:

```
x=int(input('x value:'))
if x%3==0 and x%7==0:
    print("common")
elif 93%x==0:
    print("okay")
else:print('bye')
```

```
x value:3
okay
```

In [142]:

```
93%98
```

Out[142]:

```
93
```

In [ ]:

```

```

In [144]:

```
93%21
```

Out[144]:

```
9
```

In [146]:

```
98)93(1
      ====-5
```

...

In [148]:

```
14%19
```

Out[148]:

14

In [153]:

```
# read a char from user and print
# vowel,
    # if it is lower/upper
# consonant
    # lower/upper
char=input()
if char=='a' or char=='e' or char=='i' or char=='o' or char=='u':
    print("vowel")
else:print("consonant")
```

k
consonant

In [154]:

```
ch=input("Enter the char:")
if ch in "aeiouAEIOU":
    print("vowel")
else:
    print('consonant')
```

j
consonant

In [155]:

```
# bitwise operators
# Truth Table
OR -->1,0,AND:0,1,
    XOR,X AND
```

. . .

In [157]:

```
X=20
Y=15
X&Y #X and
```

Out[157]:

4

In [158]:

```
X|Y # XOR
```

Out[158]:

```
31
```

In [ ]:

```

```

In [159]:

```
x=10100
y=01111
========
00100===4
========
11111===31
```

. . .

In [160]:

```
X^Y
```

Out[160]:

```
27
```

In [161]:

```
10100
01111
======
11011
===27
```

. . .

In [163]:

```
#read 2 values from user and perform all the bitwise
# operations
lw,up=int(input("first:")),int(input("second:"))
print(lw&up)
print(lw|up)
print(lw^up)
```

. . .

In [165]:

```
# login credentials
# set username and pwd manually
# your user credentials should match with .....
# otherwise login failed
user='LBRC908'
pwd='ABC@lbrc'
username=input("user name:")
if username==user:
    pswd=input("enter password:")
    if pswd==pwd:
        print("login successful..!")
    else:
        print("wrong password")
else:
    print("invalid user")
```

. . .

Looping Statements

- Iteration of statements/iterable for multiple times
  - 2 types:
    1. for loop
    2. while loop

For Loop

- **syntax**
  - for iterator in iterable:
    - statements
  - for ix in range(len(iterable)):
    - statements
- there is default incrementation of 1 at every iteration
- we cannot interupt the iteration

In [166]:

```
# for the iteration of statements
#printing your name for multiple times
name="vanitha" #
# 10 times
print(name)
print(name)
print(name)
print(name)
print(name)
print(name)
print(name)
```

. . .

In [169]:

```
for num in range(1,6): # upper bound is exclusive
    print(num,end=" ")
```

```
1 2 3 4 5
```

In [168]:

```
range(10,14)
```

Out[168]:

```
range(10, 14)
```

In [170]:

```
for num in range(10,16):
    print(num)
```

...

In [171]:

```
for num in range(10,16): # 6 values
    print(name)
```

```
vanitha
vanitha
vanitha
vanitha
vanitha
vanitha
```

In [172]:

```
10,11,12,13,14,15,
```

Out[172]:

```
(10, 11, 12, 13, 14, 15)
```

In [173]:

```
name
```

Out[173]:

```
'vanitha'
```

In [174]:

```
print(*name,sep=",")
```

```
v,a,n,i,t,h,a
```

In [180]:

```
for ch in name:
    print(ch)
```

. . .

In [182]:

```
van=90 #
nums=range(10,9000)
for dig in nums:
    print(dig)
```

. . .

range()

- range() is the method that can store number of values at a time
- range(stop)
  - by default starts from 0
- range(start,stop):
  - it will start from start_Value and upto stop-1
- range(start,stop,step_count):
  - skipping positions represented by step_Count

In [183]:

```
for num in range(11):
    print(num,end=" ")
```

```
0 1 2 3 4 5 6 7 8 9 10
```

In [184]:

```
for val in range(10,40):
    print(val,end=" ")
```

```
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 3
5 36 37 38 39
```

In [186]:

```
for dig in range(100,400,6):
    print(dig,end=" ")
```

```
100 106 112 118 124 130 136 142 148 154 160 166 172 178 184 190 196 202 208
214 220 226 232 238 244 250 256 262 268 274 280 286 292 298 304 310 316 322
328 334 340 346 352 358 364 370 376 382 388 394
```

In [ ]:

```

```