Python—Class board notes

Python got popular in 2008 when Data Science came..

Analysis of data..Python and R two lang .basic used used for dscience..

python feature

Simple

Intereprested

Easy

Handy

Network

Platform independent

Many modules

Machine Learning

Web Enabled

------------------

Python is Simple no complex rule..

No semi colon

No {}

No data type

No header file

---------------------

print('Hello Python')

print("Welcome Here")

print('''A

B

C

D

E''')

----------------------

Comments are either by # Single line

''' or """ multi line comments

-------------

Single line print ---

print(message/expression,end='any delimiter ')

print('Hello ',end=' ')

print('Hi ',end='#')

print('Welcome ')

--------------------

Keywords of python-- (reserved) 32 keywords

assert as in not and or if else elif while

def del import return yield....

-----------------------------------

Variable--

Data Types in python

-----------------------

Python is loosly typed .. we assign any value with variable with out specifying data type. It automatic process..

Some main types are

number- int,float octal,hexadecimal

boolean -True/False

String -

List

Tuple

Dictionary

Set

........................

Number Type.....

#integer

x=10

y=5

z=x+y

print(z)

#float

x=10.5

y=5.3

z=x+y

print(z)

-----------

Octal Number-- (base of 8)

To octal we 0o (zero o) as prefix....

#octal

x=0o123

print(x) #decimal value

-------------------------

Hexadecimal - base of 16

we use 0x (zero x) as prefix..

#Hexa decimal

x=0x123AB

print(x)

-------------------

Arithmetic Operators

----------------

+

-

\*

/

%

// floor division

\*\* power

x=12.4

y=2.7

print(x+y)

print(x-y)

print(x\*y)

print(x/y)

#floor division

print(x//y)

# power

print(x\*\*y)

#mod

print(x%y)

--------------

In print we use , to concatenate message with value.

a=12

b=8

print('Sum is..',(a+b))

---------------------

Boolean - True/False...

a=12

b=9

c=a>b

print(c)

d=False

print(d)

-----

String - immutable object.. which is sequence of letter...we can use single/double/triple quote.

Each letter of string has position (index) starts from 0.

a='Hello Dear'

b="Welcome Python"

c='''this is

multi line string'''

print(a,b,c)

x='Hello'

#immutable

x+'python'

print(x)

----------------

We can use \n and \t as well.

a='Hello \n Hi \t Bye'

print(a)

------------------

Indexing and slicing in String

--------------------------------

a='This is python code'

print(a[0]) #T

print(a[0:5]) #0-4

print(a[3:]) #3 index onwards

#reverse string

print(a[::-1])

#multi time

print(a\*3)

-------------------

Some imp function...

a='This is python code'

#length of string

print(len(a))

print(max(a)) #y

print(min(a)) #space

-------------------------

a='This is python code code'

print(a.count('code')) # 2

-------------------------

a='hello'

print(a.islower())

print(a.isupper())

w='a1234'

print(w.isalnum())

r='with'

print(r.isalpha())

d='123'

print(d.isnumeric())

-----------

a='hello'

print(a.center(40,'\*'))

print(a.find('e')) #1

--------------------

p='\*'

s=["w","t","p","s"]

print(p.join(s))

--------------

d='this is is my code'

print(d.split())

----------

s='this \t python \t java'

#tab size change to 20 char

print(s.expandtabs(20))

s='this python'

print(s.startswith('t')) #true

print(s.endswith('t')) #false

--------------------

a='this is My coding Python'

print(a.capitalize())

print(a.title())

print(a.upper())

print(a.lower())

print(a.swapcase())

---------------

.................................................

Multiple variable declaration using ,

--------------------------------------

type(var) returns the datatype associated..

a,b,c,d=32,'abc',True,90.3

print(a)

print(b)

print(c)

print(d)

print(type(a))

print(type(b))

print(type(c))

print(type(d))

----------------

Coversion of data

----------------------

int() for int casting

str() for string

float() for float casting

-------------------

a='123'

b='54'

print(a+b)

print(int(a)+int(b))

-----------------

x=89

y=9

print(x+y)

print(str(x)+str(y))

----------------

'del' keyword is used to deallocate/delete a variable in python..

x=12

y=5

z=x+y

print('sum..',z)

del z

print('sum..',z)

----------------

List-- is spl type to store multiple value (diff types), we use [] to create a list , each value is identified by unique index that starts from 0. Index is used to get/update or delete element as well.

x=[32,54.8,'abc',True,91,False]

print(x)

print(x[2])

print(x[1:4]) # 1 -3

#no of element

print(len(x))

------------------

x=[32,54.8,'abc',True,91,False]

print(x)

#update

x[0]=90000

print(x)

#delete

del x[1]

print(x)

---------------

#nested list

#list inside list

x=[[32,54,54],['a','b','c'],

[32.54,32,54]]

print(x)

-----------

x=[54,12,65,89,90,32]

print(max(x))

print(min(x))

print(len(x))

------------------

x=['a','b','p','d','c','m']

#append at end

x.append('test')

print(x)

---------------

x=['a','b','p','d','c','m']

#insert

x.insert(3,'xyx')

print(x)

--------------------

x=['a','b','p','d','c','m']

#insert

x.insert(3,'xyx')

print(x)

---------------------

s=['as','ps','as','kr','rr']

#count the elements

print(s.count('as'))

-------------

s=['as','ps','as','kr','rr']

r=['x','y','z']

#append r into s

s.extend(r)

print(s)

------------------

s=['as','ps','as','kr','rr']

print(s)

# 1

print(s.index('ps'))

----------------

s=['as','ps','as','kr','rr']

# get last element and remove

print(s.pop())

print(s)

------------------

s=['as','ps','as','kr','rr']

#order change

s.reverse()

print(s)

------------------

s=['as','ps','as','kr','rr']

#order change ascending

s.sort()

s.reverse()

print(s)

------------------

sort() or reverse() change the list itself.

-------------------------------------------

sorted() --return the new sort list. orginal no effect.

s=[90,43,6,1,49,9]

t=sorted(s)

# if reverse

#t=sorted(s,reverse=True)

#still same

print(s)

#new list sorted

print(t)

---------------------

Tuple-- similar to list but it is readonly (fixed) we cannot change/delete values. it trying so it will rasie error. we use () to make tuple.

s=(21,54,'abc',43.6,True)

#error

s[1]=3200

#error

del s[2]

Dictionary-- is another spl type which store the data in key value pair concept...each value belongs to a key, key is used to get/update/delete elements.

We use {} to define it. Key and value are separated by :

...............................

empdata={'empno':101,'name':'Ravi','salary':90000}

print(empdata)

empdata={'empno':101,

'name':'Ravi',

'salary':90000}

print(empdata)

print(empdata['name'])#Ravi

empdata['salary']=130000

print(empdata)

del empdata['name']

print(empdata)

-----------------------

If there is duplicate key , the new will be replaced by older one..

empdata={'empno':101,

'name':'Ravi',

'salary':90000,

'name':'Anuj'}

print(empdata)

------------------

Complexity can be add like - we can have multiple values as a list for single key..

empdata={'empno':[101,102,103],

'name':['Ravi','anuj',

'Raj'],

'salary':[90000,120000,

32000]

}

print(empdata)

print(empdata['salary'])

----------------

s={'empno':101,'name':'ravi',

'salary':78000,

'city':'pune'}

print(s.keys())

print(s.values())

--------------

s={'empno':101,'name':'ravi',

'salary':78000,

'city':'pune'}

#None

print(s.get('grade'))

# N/A in place of None

print(s.get('grade','N/A'))

---------------------

s={'empno':101,'name':'ravi',

'salary':78000,

'city':'pune'}

a={'grade':'a','leaves':40}

# after s .. a will be appended

s.update(a)

print(s)

-------------------------

s={'empno':101,'name':'ravi',

'salary':78000,

'city':'pune'}

a=s.copy()

print(a)

-----------------

a=['name','city','age']

#keys from a value- None

d=dict.fromkeys(a)

print(d)

#keys from a value- 10

r=dict.fromkeys(a,10)

print(r)

---------------

d={'empno':101,'name':'abr'}

#clear blank dictionary

d.clear()

print(d)

------------------

Set-- is a type which only allow unique value (non duplicates) we use {} to define a set. If we pass any duplicate it will get replaced by old value.

Note- Set is unordered ..

add(val) to add value

remove(val) to remove value

f={'apple','mango','grapes',

'apple'}

print(f)

print(len(f))

f.add('banana')

print(f)

f.remove('mango')

print(f)

-------------------

Math in python

-------------------

math is spl module for various function of math. we need to import math module.

import math

print(abs(-90)) #90

print(math.ceil(89.6)) #90

print(math.floor(89.6)) #89

print(math.pow(5,5))

print(math.sqrt(25)) #5

print(max(54,23,654,76))

print(min(54,23,654,76))

print(math.sin(60))

print(math.cos(60))

print(math.tan(60))

#pi constant

print(math.pi)

-----------------

input-- input() is used to dynamic input from user. It allow to input string values...

n=input('Enter name pls.')

print('Thanks..',n)

----------------------

x=int(input('Enter no..'))

y=int(input('Enter no..'))

print(x+y)

------------

Relational Operator..

>

<

>=

<=

==

!=

Logical Operator

and

or

not

Spl. operator... in,as

------------------------------

Conditional coding (selection/ if condition)

indentation (tab key/ 4 spaces)

-------------------------------

if condition:

tab to do

to do

else:

tab to do

spyder- will take care for most indentation.

---------------------------------------------

x=float(input('Enter no..'))

if x>=0:

print('positive')

else:

print('negative')

-------------

x=int(input('Enter number'))

y=int(input('Enter number'))

if x>y:

print(x,'is big..')

else:

print(y,'is big')

-------------------

Q. Input sale of shop give discount of 15% (sale>10000) else disc. of 10%. Print final price..

Input 20000

Output now bill.. 17000

-------------------

x=float(input('Enter Sale'))

if x>=20000:

dis=x\*0.15

else:

dis=x\*0.10

f=x-dis

print(f)

----------------------

Q. Input a name and check it is palindrome or not..

ex. ABCBA palindrome

ABC not palindrome

n=input('Enter name.')

if n==n[::-1]:

print('palindrome')

else:

print('not palindrome')

------------------------

Single line if clause

------------------------

x=int(input('Number..'))

s='Valid' if x>10 else 'Invalid'

print(s)

----------------

AND

---------

x=int(input('Number..'))

y=int(input('Number..'))

if x>10 and y>10:

print('OK')

else:

print('Not Ok')

--------------------

x=int(input('Number..'))

if x%2==0 or x%5==0:

print('Valid')

else:

print('Invalid')

--------------------

x=int(input('Number..'))

#13

if not x==10:

print('OK')

else:

print('Not Ok')

--------------------

in - spl. operator that can be used in if or loop as well.

In if condition is used to check existance of a value in sequence string/list..

s='this is my code'

d=input('Enter letter..')

if d in s:

print('letter found')

else:

print('not found..')

------

Q.. We have a list of cities input a city from user and check it is present or not if not it will append to the list...

c=['delhi','chennai','pune']

a=input('City..')

if a in c:

print('Error')

else:

c.append(a)

print(c)

------------------

if ... elif.... else

---------------------

x=int(input('Enter number.'))

if x>0:

print('+ ve')

elif x<0:

print('- ve')

else:

print('zero..')

---------------

Nested if --- if inside if...

----------------

c=input('Country..')

if c=='india':

x=int(input('Age..'))

if x>=18:

print('Vote')

else:

print('Cannot vote')

else:

print('not indian')

------------

pass- null operation keyword...

# if x >10 nothing will happen

x=int(input('Number..'))

if x>10:

pass #null operation

else:

print('not ok')

-----------------

Looping-- in python

------------------------

For loop

----------------

1.)

#end -1

for variablename in range(start,end,step):

------------------

2.)

automatic step 1

for variablename in range(start,end):

--------------------

3.)

# 0 to end -1

for variablename in range(end):

--------------------

for i in range(1,11,2):

print(i)

for i in range(10,0,-1):

print(i)

for i in range(1,11):

print(i)

for i in range(11):

print(i)

0-10

---------------

else - can be used with loop

------------------------

for i in range(11):

print(i)

else:

print('end for')

----------------------

in -- with for.. used to iterate values one by one from any squence such as list / string..

a=['ard','pop','rtt','qww']

for x in a:

print('value..',x)

while loop in python

-------------------------

i=1

while i<=10:

print(i)

i=i+1

else:

print('End while')

-----------------------

Function in python

---------------------

Function is group of statement(block of statement), which is reusable entity, that helps to reduce the code size. helps to achieve modular programming approach.

Function in python declared by 'def' keyword..

Types of functions in python

-----------------------------

no value pass no return

value pass no return

value pass and return

function with default arguments

function with variable arguments

function with change arguments sequence

call be reference

def Hello():

print('Hi')

print('Hello')

print('Ends')

print('First')

Hello()

print('Second')

Hello()

#Value pass

def calc(a,b):

print(a+b)

print(a-b)

print(a\*b)

print(a/b)

calc(90,4)

calc(40,4)

calc(30,4)

# a & b are formal

def calc(a,b):

print(a+b)

print(a-b)

print(a\*b)

print(a/b)

x=int(input('Enter no.'))

y=int(input('Enter no.'))

#x & y are actual

calc(x,y)

--------------

Return function

--------------------

return keyword is used to return a value from function.. if a function is returning value it must call by another variable or inside print..

def fact(x):

f=1

while x>=1:

f=f\*x

x=x-1

return f

r=int(input('Enter no.'))

d=fact(r)

print('Facto..',d)

-----------------

Q. Create a function in which i will pass a list of empno it return a dictionary containing all empno as values with key 'empno'.

e=[101,102,103,104]

r=dt(e)

{'empno':[101,...........]}

----------------------------------

def ch(w,t):

d={} # blank dictionary

d[w]=t

return d

a=[101,102,103,104]

s=ch('empno',a)

print(s)

------------

function with change argument sequence

----------------------------------------

def show(empno,name,city):

print('No..',empno)

print('Name..',name)

print('City..',city)

show(101,'raj','pune')

show(city='delhi',empno=102,name='amit')

show(name='ravi',city='pune',empno=103)

-----------------

Function with default parameters/arguments.

def details(name='test',age=0,salary=0):

print('Name..',name)

print('Age..',age)

print('Salary..',salary)

details('raj',28,89000)

details()

details('amit',29)

details('anuj')

details(salary=39000)

details(salary=41000,age=24)

---------------------

Function with variable arguments... in python it is possible to create a function that can accept n number of values. For this we prefix \* with paramater name. TO get the value we use for in loop.

def showdata(\*x):

print('show data')

for m in x:

print('Value..',m)

showdata(10)

showdata(32,43)

showdata(32,43,43)

showdata(12,43,43,54)

showdata('ay',43,'xyx',80)

-----------------------

Q. create function that accept n no. of value(int) return sum of them.

def sum(\*x):

d=0

for m in x:

d=d+m

return d

print(sum(32,32,32,9))

Call by reference

----------------------

#call by reference

def change(d):

d.append(1000)

r=[21,43,65,675,76]

change(r)

print(r)

-------------------------

global - to reference global variable

def Hi():

global a

print(a)

a='Hello'

print(a)

a='My Python'

Hi()

-------------

lambda function..

lambda keyword is used to define lambda functions.

Hi=lambda x:x+5

print(Hi(90))

--------------

Generator function-

it is a like a normal but it use yield keyword instead of return , this help to return a sequence of values to calling area.

def basicgen():

yield 'a'

yield 'b'

yield 'c'

x=basicgen()

print(next(x))

print(next(x))

print(next(x))

------------------

def basicgenone(x):

for a in range(x):

yield a

f=basicgenone(10)

print(next(f))

print(next(f))

print(next(f))

----------------------

Comprehensions-- this is very interesting and cocise way to put logic against sequence type (list).

For example

List of numbers

create new list of even number

for loop

get element check if condition

then append to other list.

--------------------------

x=[12,4,5,7,8,12,14,5,7]

y=[var for var in x if var%2==0]

print(y)

-------------