Python (d-1) basic knowledge

 # Basic roadmap for Data analyst and DS

1. python core-->to build the logics-->loops(for,while),conditional statements
2. python Adv.-->Numpy,Pandas,Matplotlib and Seaborn
3. SQL
4. Adv.Excel
5. Tableau
6. Power BI
7. ML
8. AI

Note:- *Python*

 # interpreted language
 # Case Sensitive
 # 20 feb 1991
 # Guido Van Rossum

 # Data Types

1. Integers=2134,-234,234,-3245
2. String='wdfegrt3456.45',"dsf45.wd","""dfvgbh45.45""" [inverted comma k andar sab string hota hai]
3. Float=2345.345,-456.345
4. Boolean=True,False

String

```
Eg 1:- x="""dwefrgt                output
        Wfergthy          ---------------------------->  'dwefrgt\nwfergthy\nwfert'
        wfert"""
         print( x)
```

```
Eg 2:- x="don't"        output
       print( x)   ---------------------->  "don't"  # ("don't", can't ,doesn't ) aise words k liye double inverted comma(“  ”)  use karte hai
```

Variables

-variable is like a container (container jaise box ya bag jo hmare element hold krta hai)

```
a=234                          output
print(a)              ---------------------------->   234      (here x is a container)
```

```
Eg 2:- x='43567.546'              output
       print(type(x))  ---------------------------->  <class 'str'>     (str=string)
```

```
Eg 3:-  x=456.67                  output
        print(type(x))  ----------------------------> <class 'float'>
```

```
Eg 4:-  x=True                    output
        print(type(x))  ---------------------------><class 'bool'>  (bool= boolean)
```

Note :-

A variable name must start with a letter or the underscore character

- ***A variable name cannot start with a number***
- ***A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9,x10 and _)(for example=s10,_d,-__etc)***
- ***Variable names are case-sensitive (age, Age and AGE are three different variables)***

Eg 1:- _x_=2345 ***output***

_x_ ----------> 2345

Eg 2:-_=3456 ***output***

_=3456 ----------> 3456

Eg 3:- ____=23456 ***output***

____ ----------> 23456

Mathematical operators

1. +(Addition)

2. -(subtraction)

3. *(multiplication)

4. /(division)

5. %(Modulus)

6. //(floor division)

7. **Exponent

Eg addition :- 3+2 ----------> 5

Eg subtraction :- 3-2 ----------> 1

Eg multiplication :- 3*2 ----------> 6

Eg division :- 3/2 ----------> 1.5

Eg modulus :- 3%2 ----------> 1 modules reminder deta hai or kisi bhi number ko 10 se modules karne k baad last element ko access kar sakte hai eg 1234%10 —->4

Eg floor :-10//3 —----------> 3 floor quesient deta hai division ki tarah par isme point(.) k baad k no. ni show karta hai or ye kisi bhi no.(1234) ko 10 se floor karne k baad element ko hatane ya delete karne k kaam aata hai jaise 1234//10 —->123

*Eg Exponent :- 2**8 #2*2*2*2*2*2 —----------> 256*

comparision Operators

1. >(greater then)

2. <(less then)

3. >=(greater then equals)

4. <=(less thern equals)

5. ==(Equals equals)

6. !=(not equals)

Eg 1:- print(7 > 6) —----------> True

Eg 2:- print (7 < 6) —----------> False

Eg 3:- print(7 >= 6) —---------->True

Eg4 :- print(7 <= 6) —----------> False

Eg 5:- print(7 != 8) —----------> True

Eg 6:- print (7 != 7) —----------->False

Eg 7:- a=1234 #b=4 c=3 d=2

b=a%10 #b=1234%10=4

a=a//10 #a=1234//10=123

c=a%10 #c=123%10=3 —----------> 4 3 2

a=a//10 #a=123//10=12

d=a%10 #d=12%10=2

print(b,c,d)

logical operators

1. and 3. not

2. Or

OPERATOR	*DESCRIPTION*	*SYNTAX*
and	*Logical AND: agar dono condition true hongi* *Tabhi overall condition true hogi*	*x and y*
or	*Logical OR: dono condition me se koi bhi ek condition true hui toh true hoga*	*x or y*
not	*Logical NOT: True if operand is false*	*not x*

The truth table for all combinations of values of X and Y.

X	Y	X and Y	X or Y	not(X)	not(Y)
T	T	T	T	F	F
T	F	F	T	F	T
F	T	F	T	T	F
F	F	F	F	T	T

* *If -else*

String and its functions

#slicing [start : stop : step]

Eg:- x='python' **#expected output : pyt**

Output

print(x[0:3]) ----------> pyt #yaha o se start hua or 2 indexing pe stop hogaya or yaha slicing me stoping ka format hota hai vo hameha -1 pe hota hai. Jaise agar me 3 dalta hu stoping position k liye toh vo 2 ki indexing lega

Or

Output

print(x[:3]) ----------> pyt

Eg Output

print(x[: :]) ----------> python # Isme na stating positin mention hai na stoping toh ye full string print karega.

Eg:- x='python' **#expected output : hon**

Output

print(x[3 :]) ----------> hon

Eg:- x='python' **#expected output : pyon**

Output

print(x[:2] + x[4 :]) ----------> pyon

#steping in slicing # 1 ka step 0 hoga or 2 ka step 1 hoga or 3=2 or 4=3

Eg:- x='python' **#expected output : pto**

Output

print(x[: :2]) ----------> pto # yaha pe 2 hai is liye ye 1 ka step lega PYTHON

#EG WAP to print all the even indexing elements from the given substring

x='python is a dynamic language'

Output

print(x[: :2]) ***—----------> pto sadnmclnug***

 # python is a dynamic language ***yaha pe 1 indexing ko skip karega or ye skiping k time pe space ko bhi count karta hai***

#EG WAP to print all the odd indexing elements from the given substring

x='python is a dynamic language'

Output

print(x[1 : :2]) ***—----------> yhni yai agae***

#negative indexing

Eg:- x='python' ***#expected output : on***

Output

print(x[-2 :]) ***—----------> on***

Eg:- x='python' ***#expected output : ytho***

Output

print(x[–5 : -1]) ***—----------> ytho***

Eg:- x='python' ***#expected output : nohtyp***

Output

print(x[–1 : :]) ***—----------> nohtyp*** ***# ye string ko reverse k liye use hota hai***

Functions of String

1. len()

2. upper()

3. lower()

4. title()

5.capitalize()

6. swapcase()

7. abs()

8. isalhpa()

9.isdigit()

10. isnumeric()

11. isalnum()

12. startwith()

13. endwith()

14. replace()

15. join()

16. split()

17. center()

18. del()

19. count()

20. index()

21. rindex()

22. find()

23. rfind()

#note :- funtions values ko modify karte hai par original variable ki values ko change ni karte kyu ki string immutable hoti hai

upper() #it takes no argument

x='python'

print(x.upper()) —----------> PYTHON

#len() --> To show the how many elements are there in string. and its take 1 arguments.

syntex:-
x="python"
print(len(x)) —----------> 5

{ len ('kis variable ki length chiye') }

replace() → it takes 3 arguments

Eg 1 :- x= 'initial interest'
print(x.replace('i','_')) —----------> _n_t_al _nterest

Eg 2:- x= 'initial interest'
print(x.replace('i','_','2')) —----------> _n_tial interest

{ replace ('KIS CHIJ KO REPLACE KARNA HAI', 'KISSE REPLACE KARNA HAI','kitne element tak replace karna hai sure k') }

lower() → it takes no arguments

x='PYTHON'

print(x.lower()) —----------> python

del() → it takes 1 arguments

x='python is a dynamic '

del(x) —----------->

#delete function humari values k sath pure variable ko delete kar deta hai

casefold() → it takes no arguments

x='PYTHON'

print(x.casefold()) —----------> python

Note:- *lower() function ki tarah same hai par casefold() par bohot agressive hota hai or fast hota hai as compare to lower() function*

find() → it takes 3 arguments

#find() function indexing deta hai

Eg 1 :- x= 'python is a dynamic language'
print(x.find('a')) —----------> 10

Eg 2 :- x= 'python is a dynamic language'
print(x.find('a',11,18)) —----------> 15

{ find ('kis element ki indexing chiye', 'kon si indexing se chalna hai','kon si indexing tak chalna hai') }

index() → it takes 3 arguments

#index() function bhi indexing deta hai

Eg 1 :- x= 'python is a dynamic language'
print(x.index('a')) —----------> 10

Eg 2 :- x= 'python is a dynamic language'
print(x.index('a',11,18)) —----------> 15

{ index ('kis element ki indexing chiye', 'kon si indexing se chalna hai','kon si indexing tak chalna hai') }

#note:- agar koi aisa element ki indexing chiye jo exist nahi karta ho toh ye error deta hai(substring not found) . Par find() function error nahi deta vo (-1) deta hai.

rfind() → it takes 3 arguments

#rfind() function indexing deta hai par piche se

Eg 1 :- x= 'python is a dynamic language'
print(x.rfind('a')) —----------> 25

rindex() → it takes 3 arguments

#index() function bhi indexing deta hai par picche se.

Eg 1 :- x= 'python is a dynamic language'
print(x.rindex('a')) ——————————> 25

capitalize() → it takes no arguments

#ye string k pehle word ko capital kar deta hai BAKI KO SMALL.

Eg 1 :- x= 'python is a dynamic language'
print(x.capitalize()) ——————————> Python is a dynamic language

Eg 1 :- x= 'PYTHON IS A DYNAMIC LANGUAGE'
print(x.capitalize()) ——————————> Python is a dynamic language

title() → it takes no arguments

ye space k baad jo pehla letter hoga usko capital kar dega oe baki sabko small kar dega.

Eg 1 :- x= 'python is a dynamic language'
print(x.title()) ——————————> Python Is A Dynamic Language

swapecase() → it takes no arguments

small ko capital or capital ko small kar deta hai.

Eg 1 :- x= 'python is a dynamic language'
print(x.swapecase()) ——————————> PYTHON IS A DYNAMIC LANGUAGE

isalpha() → it takes no arguments

ye bool me output deta hai or check karta hai is sab element alphabet hain ki nahi

Eg 1 :- x= 'python is a dynamic language'
print(x.isalpha()) ——————————> false [beech me space ki vajah se false dera]

Eg 2 :- x= 'python'
print(x.isalpha()) ——————————> true

isdigit() → it takes no arguments

ye bool me output deta hai or check karta hai is sab element string me digit hai ki nahi.

Eg 1 :- x= '1234.56'
print(x.isdigit()) ——————————> False #decimal ki vajah se false hai ye

abs() → it takes 1 arguments

ye negative ko positive me badalta hai

Eg 1 :- x= '-1234'
print(abs(x)) ----------> 1234

split() → it takes 2 arguments

 # ye chijo ko todta/partition karta hai or unko list me convert kar deta hai. Or jis element ko todenge vo list me nahi hoga
Eg 1 :- x= 'python is a dynamic language'
print(x.split()) ----------> ['python','is','a','dynamic','language']

Eg 2 :- x= 'python is a dynamic language'
print(x.split('a',2)) ----------> ['python is', 'dyn','mic language']

{ split ('kaha se chijo ki todna hai ya kon se element se todna hai', 'suru k kitne selected element ka partition karna hai') }

join() → it takes 1 arguments

 # ye chijo koek sath join karta hai or list ko string me badalta hai .
Eg 1 :- x= ['a','b','c','d']
print('', .join(x)) ----------> abcd

count() → it takes 3 arguments

#count() kisi m=bhi element ki counting deta hai

Eg 1 :- x= 'python is a dynamic language'
print(x.count('a')) ----------> 4

Eg 2 :- x= 'python is a dynamic language'
print(x.index('a',12,28)) ----------> 3

{ index ('kis element ki counting chiye', 'kon si indexing se chalna hai','kon si indexing tak chalna hai') }

startswith() → it takes 1 arguments

 # ye bool me output deta hai or bata hai ki element jo hai vo given element se start ho raha hai ki nahi.
Eg 1 :- x= 'python'
print(x.startswith('p')) ----------> True

endswith() → it takes 1 arguments

 # ye bool me output deta hai or bata hai ki element jo hai vo given element se endt ho raha hai ki nahi.

Eg 1 :- x= 'python'

print(x.startswith('n')) —----------> True