**PYTHON**

# Object is Base Languaage

#Cell

# #**Operators**

# 1. Airthmatic Operators = [+, -, /, **%**, **//**, **\*\***] ( **% Modules, // floor division, \*\* exponetation)**

# 2. Assignment Operators ( =, +=, -=, \*=, /=, %=, //=, \*\*=)

# 3. Logical Operators = (and, or, not)

# 4. Compression Operators (>, >, >=,<=)

# 5. Identity Operators ( is, is not)

# 6. Membership Operator

# 7. Bitwise Operator

a="Nitin" **# value has a particular Data Type # Str # input statement**

print (a) **# output print**

print(id(a)) **# id variable its own id**

print(type(a)) **# Data Type**

**Output:**

Nitin

2297945026992

<class 'str'>

**# variable's in values & data type**

x= 10 #int

xx = -10 # int

ss = 302 # float

ed= True # Bool

de = 3+3j # complex

name = "nick" # str

print(x)

print(xx)

print(ss)

print(ed)

print(de)

print(name)

**Output**

10

-10

302

True

(3+3j)

nick

# Variable

# A variable can have a short name (like x and y) or a more descriptive name (age, carname)

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

# A variable name cannot satart with a nmumber.

# A variable name can only contain alpha-numeric charactiers and underscores (A-z.0-9,and \_)

# A variable names are case-sensitive (age, Age and AGE are three different variables)

# A variable name cannot be any of the Pythonn keywords.

age =13

Age = 34

AGE = 43

print(age)

print(Age)

print(AGE)

print(id(age)) # Id varible

print(id(Age))

print(id(AGE))

**Output**

13

34

43

140722445325480

140722445326152

140722445326440

# Many values in multi variable

x,y,z = "nitin", 29, "Jan"

print (x,y,z)

print (x,"+",y,"+",z)

# Multi variable to single value

m=n=c = 10

print (m)

print(n)

print(c)

**Output**

nitin 29 Jan

nitin + 29 + Jan

10

10

10

x= 10 #int

xx = -10 # int

ss = 302 # float

ed= True # Bool

de = 3+3j # complex

name = "nick" # str

ls=["nick","vishal","ravi"] # List - Square bracket

tup=("nick","vishal","ravi") # Tuple - round bracket

ran=('0,10') # To assign range we apply 0 to 100

print(x)

print(type(x))

print(xx)

print(type(xx))

print(ss)

print(type(ss))

print(ed)

print(type(ed))

print(de)

print(type(de))

print(name)

print(type(name))

print(ls)

print(type(ls))

print(tup)

print(type(tup))

print(ran)

print(type(ran))

Output:

10

<class 'int'>

-10

<class 'int'>

302

<class 'int'>

True

<class 'bool'>

(3+3j)

<class 'complex'>

nick

<class 'str'>

['nick', 'vishal', 'ravi']

<class 'list'>

('nick', 'vishal', 'ravi')

<class 'tuple'>

0,10

<class 'str'>

**Python Casting:**

There may be times when you want to specify a type on to a variable. This can be done with casting. Python is an object-orientated language, and as such it uses classes to define data types, including its primitive types.

Casting in python is therefore done using constructor functions:

* int() - constructs an integer number from an integer literal, a float literal (by removing all decimals), or a string literal (providing the string represents a whole number)
* float() - constructs a float number from an integer literal, a float literal or a string literal (providing the string represents a float or an integer)
* str() - constructs a string from a wide variety of data types, including strings, integer literals and float literals

**Command:**

x = int(1) # x will be 1

y = int(2.8) # y will be 2

z = int("3") # z will be 3

print (x)

print (y)

print (z)

Output:

1

2

3

**Data Type :**

String:

Strings in python are surrounded by either single quotation marks, or double quotation marks.

'hello' is the same as "hello".

You can display a string literal with the print() function:

Slicing strings:

You can return a range of characters by using the slice syntax.

Specify the start index and the end index, separated by a colon, to return a part of the string.

Get the characters from position 2 to position 5 (not included):

**Note:**The first character has index 0.

# Slicing String

ds="Today is sunday"

print(ds)

print(ds[6:8]) **#(move left to right) , end index value**

print(ds[-9:-7]) **# (move right to left), start index value**

# character which starts from 0

# & indexing from end its with -1

Output:

Today is sunday

Is

is

String Concatenation

To concatenate, or combine, two strings you can use the + operator.

a = "Hello"  
b = "World"  
c = a + b  
print(c)

output

Hello World

**23rd Dec 2023**

# Addition

add = x+y+z

print(add)

60

In [12]:

x,y,z **=** 12,45,78

add **=** x**+**y**+**z

print(add)

135

In [14]:

*# Modules : its show the reminder*

x **=** 19

y **=** 5

print(x**%**y)

4

In [20]:

*# floor division its round up the output with minimum value*

x**=**27

y**=**4

d **=** x**/**y

print(d)

6.75

In [21]:

x**//**y *# its remove the value after the decimal*

Out[21]:

6

# Exponets

x = 4

y = 6

z = x**\*\*y**

**print(z)**

In [26]:

4**\***4**\***4**\***4**\***4**\***4

Out[26]:

4096

# Data Type

1. Numerical = integar (int), float (always in decimal), complex ( end with J)

2. Text type = string (str)

3. Sequence - List [], tuple (), range()

4. set = {}

5. Mapping = Dictionary {key + value}

6. Boolen = True , False

7. None = None

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

x = "Hello world"

print(x)

print(type(x)) # **show data type**

print(len(x)) # **to identify length of string**

Hello world

<class 'str'>

11

x = """There may be times when you want to specify a type on to a variable. This can be done with casting. Python is an object-orientated language, and as such it uses classes to define data types, including its primitive types.'''

"""

print(x)

There may be times when you want to specify a type on to a variable. This can be done with casting. Python is an object-orientated language, and as such it uses classes to define data types, including its primitive types.'''

# use """ to print multiple string at start & end of the sentences

x **=** 400

print(type(x))

<class 'int'>

In [40]:

x **=** 100.0

print(type(x))

<class 'float'>

In [41]:

x**=** 21j

print(type(x))

<class 'complex'>

# Forcasting

#we cannot check length for integar, for that we need to change variable in string format

x = 10

print (type(x))

print(x)

<class 'int'>

10

y=str(x) # convert to string

print(y)

print(type(y))

print(len(y))

10

<class 'str'>

2

# str() It's convert in String

# Float convert in Float

# int() convert in Integer

# Complex () convert in complex

**#convert the 789 to float and string and check the length and data type**

x = 789

y = float(x)

print(type(y))

<class 'float'>

x = 789

y = str(x)

print(len(y))

3

Without convert into integar

x **=** "45"

y **=** "10"

print(x**+**y)

​

4510

In [59]:

with convert into integar

x **=**"45"

y **=** "10"

x**=**int(x)

y**=**int(y)

print(x**+**y)

55

**## comeprision operators**

> Greater than

< less than

== equal

!= Not equal

>= Greater than equal

<= Less than equal

x = 10

y = 20

print(x<y)

print(x>y)

print(x==y)

True

False

False

x= 10

y = 20

if x >= 20:

print("x is greater than y")

else:

print("X is less than y")

X is less than y

x = 200

if x != 200:

print("Right")

else :

print("Wrong")

Wrong

24th Dec :

# Write a python program to check the NUmber is Odd or Even ?

x = 5

y = 4

if x%2==0:

print("Even")

else:

print("Odd")

Output = Odd

# Write a python program to check the NUmber is Odd or Even ?

x = int(input("Enter the Number :-"))

if x%2==0:

print("Even")

else:

print("Odd")

Enter the Number :-75

Odd

Enter the Number :-74

Even

#write a python program to check the NUmber is Greater than 50 or Not ?

x= int(input("Enter the Number"))

if x>50:

print("Greater than 50")

else:

print("Less than 50")

Enter the Number45

Less than 50

# write a python program to check the Number is Divisible of 5 or Not ?

x=int(input("Enter the Number"))

if x%5==0:

print("Divisible")

else:

print("Not Divisible")

Enter the Number12

Not Divisible

Enter the Number25

Divisible

# Write a python program to check who is eligible for vote ?

age=int(input("Enter the age"))

if age>=18:

print("Eleigible for vote")

else:

print("Not Elegible for vote")

Enter the age15

Not Elegible for vote

Enter the age45

Eleigible for vote

# if per is greater than 60 ==== First div

# if per is greater than 45 and less than 60 ==== second div

# if per is greater than 33 and Less than 45 ===== third div

# if per is less than 33 ==== fail

Per = int(input("enter the percentage of student :-"))

if Per >=60:

print("First Divistion")

elif Per>=45:

print("Second Division")

elif Per>=33:

print("Third Division")

elif Per<33:

print("Fail")

**enter the percentage of student :-45**

Second Division

**enter the percentage of student :-75**

First Divistion

**enter the percentage of student :-25**

Fail

**enter the percentage of student :-35**

Third Division

**Python Assignment Operators**

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

Assignment operators are used to assign values to variables:

Python assigment operators

= assign

+= add and assign

-= substract and assign

\*= multiplication and assign

/= divide and assign

%= modules and assign

\*\*= exponentation and assign

//= floor divison and assign

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

--------------example--------------------

= x = 5 x = 5

+= x += 3 x = x + 3

-= x -= 3 x = x - 3

\*= x \*= 3 x = x \* 3

/= x /= 3 x = x / 3

%= x %= 3 x = x % 3

//= x //= 3 x = x // 3

\*= x \*= 3 x = x \*\* 3

**# And Operator**

x = int(input("enter number"))

if x>30 and x<50:

print("True")

else:

Print(False)

**enter number60**

False

**enter number35**

True

**# Or Operator**

x= int(input("enter number"))

if x<=50 or x<=75:

print("True")

else:

print("False")

**enter number80**

False

**enter number55**

True

30th Dec 2023

**# Write a python program to compare the three number which is the highest number**

**#take input with help of user input.**

x=int(input("enter the first number"))

y=int(input("enter the scond number"))

z=int(input("enter the third number"))

if x>y and x>z:

print(x,"highest number")

elif y>x and y>z:

print(y,"highest number")

elif z>x and z>y:

print(z,"highest number")

**enter the first number45**

**enter the scond number75**

**enter the third number50**

75 highest number

## Write a python program to whether a number is divisible by 2 and 3 both

x=int(input("enter the number"))

if x%2==0 and x%3==0:

print("Divisible")

else:

print("not divisible")

**Ouput - enter the number 6**

**Divisible**

**enter the number 20**

**not divisible**

# Write a python program to wether a number is divisible by 2 and 3 both

# show the number which is divisible

x=int(input("enter the number"))

if x%2==0 or x%3==0:

print(x," is Divisible")

if x%2==0:

print(x,"is divisible with 2")

else:

print(x,"is divisible with 3")

**enter the number10**

10 is Divisible

10 is divisible with 2

**enter the number15**

15 is Divisible

15 is divisible with 3

**# identity Operator**

# is

# is not

x= 10

y = 10

print (x is y)

**True**

x=10

y =15

print (x is y)

**False**

**# Membership Operator**

**# IN , Not In**

x = "Nitin Singh"

if "t"in x:

print("present")

else:

print("not present")

Present

x = "Nitin Singh"

if "z"in x:

print("present")

else:

print("not present")

not present

**## Bitwise operator**

**# & and**

**# | or**

x=15

if x==15 & x>10:

print("Correct")

Correct

# Casting

x= 10 # int

y = str(x)

type(y)

str

**# int() # str() # float() #list() #tupe() #dict() #set()**

**x="12"**

**y="10"**

**print(type(x))**

**print(type(y))**

<class 'str'>

<class 'str'>

**a=int(x)**

**b=int(y)**

**z = a+b**

**print(type(z))**

**print (z)**

<class 'int'>

22

**# Swapping**

x,y = 10,12

print("x :", x)

print("y :", y)

x : 10

y : 12

x,y = y,x # Swapping

print("x :", x)

print("y :", y)

x : 12

y : 10

**# Loop**

**# 1. while Loop**

**# 2. For Loop**

**x= 10**

**y=0**

**while x>=y:**

**print("Number",y)**

**y = y+1**

**Output:**

Number 0

Number 1

Number 2

Number 3

Number 4

Number 5

Number 6

Number 7

Number 8

Number 9

Number 10

**# print 1 to 10 with the help of while loop**

**x=10**

**y = 1**

**while x>=y:**

**print(y)**

**y=y+1**

**Output:**

1

2

3

4

5

6

7

8

9

10

**# print 10 to 1 with the help of while loop**

**x=10**

**y = 1**

**while x>=y:**

**print(x)**

**x=x-1**

**Output:**

10

9

8

7

6

5

4

3

2

1

**# print all evern number from 1 to 20**

**x=20**

**y=1**

**while x>=y:**

**if y%2==0:**

**print("Even Number :", y)**

**y+=1**

**Output:**

Even Number : 2

Even Number : 4

Even Number : 6

Even Number : 8

Even Number : 10

Even Number : 12

Even Number : 14

Even Number : 16

Even Number : 18

Even Number : 20

**# Print odd number from 1 to 25 ?**

**x=25**

**y=1**

**while x>=y:**

**if y%2!=0:**

**print("odd number :", y)**

**y+=1**

**Output :**

odd number : 1

odd number : 3

odd number : 5

odd number : 7

odd number : 9

odd number : 11

odd number : 13

odd number : 15

odd number : 17

odd number : 19

odd number : 21

odd number : 23

odd number : 25

**# write 5 table**

**x=50**

**y=5**

**while x>=y:**

**print(y)**

**y+=5**

**Output**

5

10

15

20

25

30

35

40

45

50

**# Enter any dynamic table**

**a=int(input("Enter the table name"))**

**x=10**

**y=1**

**while x>=y:**

**print(y\*a)**

**y= y+1**

**Output**

**Enter the table name 5**

5

10

15

20

25

30

35

40

45

50

**# # print like below**

**# 5 x 1 = 5**

**# 5 x 2 = 10**

**a=int(input("Enter the table name"))**

**x=10**

**y=1**

**while x>=y:**

**print(a,"X",y,"=", a\*y)**

**y= y+1**

**Output:**

Enter the table name5

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

**# # Break - it is used to stop the loop**

**# Continue - skip the text or number**

**# Print the number 1 to 10 but break the look at 5**

**x=10**

**y=1**

**while x>=y:**

**if y==5:**

**break**

**print(y)**

**y = y + 1**

**Output:**

1

2

3

4

**# # Print the number 1 to 10 but continue after 5**

**x=10**

**y=0**

**while x>=y:**

**y+=1**

**if y==5 or y==7:**

**continue**

**print(y)**

**Output:**

1

2

3

4

6

8

9

10

11

# to print Horizontalcounting

x=10

y=1

while x>=y:

print(y, end=",") # end is use for print value in Horizontal

Output

1,2,3,4,5,6,7,8,9,10,

**# Write a program to extract the all even number from 1 to 30**

x=30

y=1

while x>=y:

if y%2==0:

print(y)

y+=1

Output:

2

4

6

8

10

12

14

16

18

20

22

24

26

28

30

**O**

c=0

x=30

y=1

while x>=y:

if y%2==0:

c=c+1

y+=1

print("total even number is :", c)

Output

total even number is : 15

**# Write the program to add first 6 values with the help of while loop.**

**# 1,2,3,4,5,6 = 21**

**a=0**

**x=6**

**y=1**

**while x>=y:**

**a=a+y**

**y+=1**

**print("Addition :-", a)**

**Output:**

Addition :- 21

**# write a program to skip the 13,15 and 19 fromm 10 to 40 and print the number Horizontally**

**x=40**

**y=9**

**while x>y:**

**y+=1**

**if y==13 or y==15 or y==19:**

**Continue**

**Output:**

10 11 12 14 16 17 18 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40

**else:**

**print(y,end=" ")**

**FOR Loop**

**Sequence data type : list, tupple, range()**

**range (starting point, Ending Point, step size)**

**# write query to print 1 to 10 number**

**# range (starting point, ending point, step size)**

**for a in range(1,11):**

**print(a)**

**Output:**

1

2

3

4

5

6

7

8

9

10

**# write a program to print the back counting from 50 to 1**

**# with the help of loop**

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

print(a)

Output:

10

9

8

7

6

5

4

3

2

1

**# print even number with for Loop from 1 to 21**

c = 0

for a in range(1,21):

if a%2==0:

print(a)

c+=1

print("total even number :", c)

Output :

2

4

6

8

10

12

14

16

18

20

total even number : 10

**# Write a program to find the average of the first 10 number?**

**a=0**

**x=50**

**for i in range(1,x+1):**

**a = a+i**

**avg=a/x**

**print("average : -", avg)**

**Output :**

average : - 25.5

06th jan 2024:

# find factorial of any number with the help of user input

x=int(input("enter the number"))

fact = 1

for i in range (1,x+1):

fact = fact\*i

print("Factorial : -", fact)

Output:

enter the number5

Factorial : - 120

# Break & Continue Statement

for i in range (10):

if i==5 or i==7:

continue

else:

print(i)

Output:

0

1

2

3

4

6

8

9

# Break

for i in range(10):

if i==5:

break

print(i)

Output:

0

1

2

3

4

x="Data Science"

for i in x:

if i==" ":

break

else:

print(i,end="")

Output:

Data

# **Nested Loop**

**#Nested Loop : Loop inside the another Loop**

**for i in range(5):**

**print("Hello World")**

**for j in range(5):**

**print(j)**

**Output:**

Hello World

0

1

2

3

4

Hello World

0

1

2

3

4

Hello World

0

1

2

3

4

Hello World

0

1

2

3

4

Hello World

0

1

2

3

4

**# Print pattern**

**for i in range(10):**

**for j in range(0,i):**

**print("\*", end="")**

**print()**

**Output:**

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*

String

String - It always written in single quatation or double quatation

x= "data" o 'data'

1. **Upper**

x = "data science"

x = x.upper() # upper convert into capital letter

print(x)

**Output**:

DATA SCIENCE

1. **Lower**

x = "DATA SCIENCE"

x = x.lower() #Lower convert into Lower letter

print(x)

**Output**:

data science

1. **Title**

x = "data science"

x = x.title() # Title convert into 1st letter in capital letter

print(x)

**Output**:

Data Science

**# Capitalize() : - it convert only first caracter to upper case**

x= "data science"

x=x.capitalize()

print(x)

**Output**

Data science

Strip

#Strip - it's remove the extra space starting and ending of the text.

x = " Data Science "

print(x)

x=x.strip()

print(x)

Output

Data Science

Data Science

#Count - it count the number of alphabets or number or special caracter

x="data scince"

y=x.count("a")

print(y)

Output:

2

Find

# it shows the number of Index

x="Science"

y=x.find("i")

print("index of i :", y)

Output:

index of i : 2

# -7 -6 -5 -4 -3 -2 -1 Negative Indexing

# S C I E N C E

# 0 1 2 3 4 5 6 Positive Indexing

**Replace:**

**# Replace - it replace the text at place of old text**

**1.**

**x = "Data Science"**

**y = x.replace(" ","\*")**

**print(y)**

**Output:**

Data\*Science

2.

x= "Arunachal Pradesh"

x=x.lower()

y=x.replace("a","\*")

print(y)

**Output:**

\*run\*ch\*l pr\*desh

**# Format()**

#Formats specified values in a string

1.

x = "Hi my name is {}"

y = x.format("Nitin Singh")

print(y)

**Output**

Hi my name is Nitin Singh

2.

x= "Hi {} welcome to {}"

y = x.format("Nitin", "Delhi")

print(y)

**Output**

Hi Nitin welcome to Delhi

**Split**

#Split() : it convert the string into list

1.

x = "Python is a programing language"

y = x.split()

print(y)

**Output**

['Python', 'is', 'a', 'programing', 'language']

2.

x = "apple,mango,banana"

print(x)

y = x.split()

print(y)

**Output**

apple,mango,banana

['apple,mango,banana']

**Join**

# Join Function : it's join the multiple values from list to string

x=["hi","how","are","you"]

print(x)

print(type(x))

y = " ".join(x)

print(y)

print(type(y))

**Output**

['hi', 'how', 'are', 'you']

<class 'list'>

hi how are you

<class 'str'>

# **Swapcase**

x = "Data Science"

y=x.swapcase()

print(y)

**Output**:

dATA sCIENCE

**# isalpha () : its check if value is alphabetic**

**#isdigit () : its check if string is number**

**#isspace () : its check if string have any space**

**#islower () : check if have any lower letter value**

**#isuper () : cehck if have any capital letter value**

**#isalnum () : its check the string is mix character of text and strings value**

1. **#isalpha ()**

x="delhi"

x.isalpha()

**Output**

True

1. **#isalnum ()**

x="delhi110059"

x.isalnum**()**

**Output**

True

1. **#isdigit ()**

x="305001"

x.isdigit()

**Output**

True

1. **#isspace ()**

x=" "

x.isspace()

**Output**

True

**# 07 Jan 2024**

**Question : #x = "Himachal Pradesh"**

**#extract the text given Below**

**#1. chal**

**##2. desh**

**#3. ma**

**#4. him**

**#5. reverse the Text**

**#6. print the all text and step size should be 3**

**#7. join the first and last text**

**#8. count "t" how many times repeated**

**#9. find the index of the space**

**#10. convert into a list**

**Question : # x = "Himachal Pradesh"**

**# 1. count the total number of r**

**# 2. count the total number of a**

**# 3. show the index of space**

**# 4. replace the value "\*" at the place of "d"**

**# 5. convert the string in upper case**

**Find length**

x ="Himachal Pradesh"

print(len(x))

16

**#Question 4 : replace the value "\*" at the place of "d"**

**x="Himachal Pradesh"**

**y = x.replace("d","\*")**

**print(y)**

Himachal Pra\*esh

**# Indexing & Slicing # 13 Jan 2024**

**Indexing -** its extract the single number of the caracter from the text.

**Question** : Find c under Data Science

x = "Data Science"

x[6]

'c'

# "Arunachal P r a d e s h

# 0123456789 10 11 12 13 14 15 16"

**Question :**  **Extract - r, l, e, h, p**

#Positive Indexing

x = "Arunachal Pradesh"

print(x[1])

print(x[6])

print(x[8])

print(x[10])

print(x[11])

print(x[14])

print(x[16])

r

h

l

P

r

e

h

#Negative Indexing

x = "Arunachal Pradesh"

print(x[-1])

print(x[-3])

h

E

#Slicing

Slicing - it's extract the range of caracter from the text

Slicing = x[starting index : end index : step size]

**Question :** "Arunachal Pradesh" - extract starting 3 alphabet

X= "Arunachal Pradesh"

x[0:3]

'Aru'

**Question :** "Arunachal Pradesh" - extract last3 alphabet

X= "Arunachal Pradesh"

x[-3:]

'esh'

**Question :** x = "Arunachal Pradesh"

# extract these text from x

# Aruncachal

x = "Arunachal Pradesh"

x[0:9]

'Arunachal'

#Pradesh

x = "Arunachal Pradesh"

x[-7:]

'Pradesh'

#runa

x = "Arunachal Pradesh"

x[1:5]

'runa'

#desh

x = "Arunachal Pradesh"

x[-4:]

'desh'

# Auahl

x = "Arunachal Pradesh"

x[0:9:2] # [starting index (0) : end index (9) : step size (-2)]

'Auahl'

# last five alphabet from text

x = "Arunachal Pradesh"

x[-5:]

'adesh'

# **Reverse the string**

# Reverse string

x = "Arunachal Pradesh"

x[-1: :-1]

'hsedarP lahcanurA'

# ARUnacHAL PrAdeSH

**Question :** # count all a

# 1st we need to convert all test in lower format

x ="ARUnacHAL PrAdeSH"

x = x.lower()

print(x)

x.count("a")

arunachal pradesh

4

**Question -**  x = "prince1234324@$%%sharma"

# extract the Text from x

x = "prince1234324@$%%sharma"

for i in x:

if i.isalpha():

print(i,end="")

princesharma

**Question :**  extract the Number form x

x = "prince1234324@$%%sharma"

for i in x:

if i.isdigit():

print(i,end="")

1234324

**Question :** count the Total number in x

c = 0

x = "prince1234324@$%%sharma"

for i in x:

if i.isdigit():

c+=1

print(c,end="")

7

**Question :** extract special caracter

x = "prince1234324@$%%sharma"

for i in x:

if i.isdigit():

continue

elif i.isalpha():

continue

else:

print(i,end="")

@$%%

**Question:**  "datascienceclient" - print only C & T

x = "datascienceclient"

for i in x:

if i=="c" or i=="t":

print(i,end="")

Tccct

**Question:** x = "Arunachal Pradesh"

# find the positive and negative index of each alphabet

# positive index

x = "Arunachal Pradesh"

y = len(x) # 16

for i in range(y):

print(x[i],"-: positive index :-",i)

A -: positive index :- 0

r -: positive index :- 1

u -: positive index :- 2

n -: positive index :- 3

a -: positive index :- 4

c -: positive index :- 5

h -: positive index :- 6

a -: positive index :- 7

l -: positive index :- 8

-: positive index :- 9

P -: positive index :- 10

r -: positive index :- 11

a -: positive index :- 12

d -: positive index :- 13

e -: positive index :- 14

s -: positive index :- 15

h -: positive index :- 16

**# negative index**

x = "Arunachal Pradesh"

y = len(x) # 16

for i in range(y):

print(x[i],"-: negative index :-",i-y)

A -: negative index :- -17

r -: negative index :- -16

u -: negative index :- -15

n -: negative index :- -14

a -: negative index :- -13

c -: negative index :- -12

h -: negative index :- -11

a -: negative index :- -10

l -: negative index :- -9

-: negative index :- -8

P -: negative index :- -7

r -: negative index :- -6

a -: negative index :- -5

d -: negative index :- -4

e -: negative index :- -3

s -: negative index :- -2

h -: negative index :- -1

**LIST**

**Sequence Data Type : List, Tupple, Range()**

**List :**

**1. Always written in [] braket.**

**2. List is mutable or changable**

**3. List are orderded**

**4. List are indexed**

**5. List allow duplicate values**

**6. In List we can add multiple type of data**

**# List can pick any data type**

**x = [20, 45.50, 20j, True, None, "Data"]**

**print(x)**

[20, 45.5, 20j, True, None, 'Data']

# **List can pick duplicate value**

x = [20, 45.50, 20j, True, None, "Data", 20, 20, 20, "Data"]

print(x)

[20, 45.5, 20j, True, None, 'Data', 20, 20, 20, 'Data']

**# Indexing in List**

x=[12,45,45.45,"data",21j]

print(x)

print(type(x))

print(len(x))

[12, 45, 45.45, 'data', 21j]

<class 'list'>

5

x=[12,45,45.45,"data",21j]

print(x[0]) # always use [] braket for indexing , # 0 shows 1st index number which is 12

print(type(x))

12

<class 'list'>

**# find data type for all**

x=[12,45,45.45,"data",21j]

for i in x:

t=(type(i))

print(t,";",i)

<class 'int'> ; 12

<class 'int'> ; 45

<class 'float'> ; 45.45

<class 'str'> ; data

<class 'complex'> ; 21j

**# Indexing & Slicing in List**

#print Science through positive indexing

x= ["data", "Science", "nitin", "jatin", "prince"]

print(x[1])

Science

**# Indexing & Slicing in List**

#print Science through Negative indexing

x= ["data", "Science", "nitin", "jatin", "prince"]

print(x[-2])

prince

**#Slicing - print next to next text**

x= ["data", "Science", "nitin", "jatin", "prince"]

print(x[0: :2])

['data', 'nitin', 'prince']

**#Slicing - Reverse the list**

x= ["data", "Science", "nitin", "jatin", "prince"]

print(x[-1::-1])

['prince', 'jatin', 'nitin', 'Science', 'data']

FORMULA'S

1. Insert
2. Append
3. Extend

**# Insert - It add value under given index but can't add value at last**

x = ["Sunday", "Monday", "Tuesday", "Wednesday"]

x.insert(0,"Friday")

print(x)

['Friday', 'Sunday', 'Monday', 'Tuesday', 'Wednesday']

**# Append - it add value by default at last place**

x = ["Sunday", "Monday", "Tuesday", "Wednesday"]

x.append("Friday")

print(x)

['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Friday']

**# # Extend - it add Multiple value in the list by default at last place**

**# it merge 2 list in single list**

**# 1st Methord**

x = ["Sunday", "Monday", "Tuesday", "Wednesday"]

y = ["Data Science Class"]

x.extend(y)

print(x)

['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Data Science Class']

**#2nd Method: through addition x+y**

x = ["Sunday", "Monday", "Tuesday", "Wednesday"]

y = ["Data Science Class"]

print(x+y)

['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Data Science Class']

**#3rd Method: through loop**

x = ["Sunday", "Monday", "Tuesday", "Wednesday"]

y = ["Data Science Class"]

for i in y:

x.append(i)

print(x)

['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Data Science Class']

# **How to remove value from list.**

1. Pop
2. Remove
3. Del
4. Clear

#**Pop** : it's remove the value according indexing/ no. of position and by default remove the value of last index

x= ["One", "Two", "Three", "Four"]

x.pop() # by default delet last value

print(x)

['One', 'Two', 'Three']

**#Pop : by provide index**

x= ["One", "Two", "Three", "Four"]

x.pop(2)

print(x)

['One', 'Two', 'Four']

#**Remove**

**# Remove - it remove text through provided value**

x= ["One", "Two", "Three", "Four"]

x.remove("Two")

print(x)

['One', 'Three', 'Four']

**# Clear**

**# Clear - it remove complete value from list**

x= ["One", "Two", "Three", "Four"]

x.clear()

print(x)

[]

**# Del**

**# Del (Delete) - it deletes the vlues through indexing/slicing or complete delete**

**# del through index**

x= ["One", "Two", "Three", "Four"]

del x[0: : 2]

print(x)

['Two', 'Four']

**Complete list delete**

x = [1,2,3,4]

del x[0 :]

print(x)

[]

Useful Formula's

# **Count, copy, sort, reverse**

**# count**

x = [1,2,3,1,1,2,1]

x.count(1)

4

**#Copy**

x =[1,2,3,1,1,4,5,7,10]

y = x.copy()

x.clear()

print(y)

[1, 2, 3, 1, 1, 4, 5, 7, 10]

# **#Sort**

# Sort :

# Sort at assending order

x = [1,2,7,5,2,3,1,1,10]

x.sort()

print(x)

[1, 1, 1, 2, 2, 3, 5, 7, 10]

## Sort at Dessending order

x = [1,2,7,5,2,3,1,1,10]

x.sort(reverse=True) print(x)

[10, 7, 5, 3, 2, 2, 1, 1, 1]

**20th Jan 2024**

#x = [12,45,89,78,56.8,"data science",True]

# Remove data science from the list

# 2. Replace the value 500 at the place of 89

# 3. insert new value in this list given below

#y = [741,852,963]

#4. find the sum of all the integar and float value

#5. count the total number of Integar value in this list

# 6. delete all the value from the list

#x = [12,45,89,78,56.8,"data science",True]

1. **# Remove data science from the lis**t

x = [12,45,89,78,56.8,"data science",True]

x.remove("data science")

print(x)

**# 2. Replace the value 500 at the place of 89**

x = [12,45,89,78,56.8,"data science",True]

x[2] = 500

print(x)

**# 3. insert new value in this list given below**

#y = [741,852,963]

x = [12,45,89,78,56.8,"data science",True]

y = [741,852,963]

x.extend(y)

print(x)

**#4. find the sum of all the integar and float value**

x = [12,45,89,78,56.8,"data science",True]

y = sum(x[0:5])

print(y)

**#5. count the total number of Integar value in this list**

c = 0

x = [12,45,89,78,56.8,"data science",True]

for i in x:

if type (i)==int:

c = c+1

print("Total number of integar value : -", c)

#x = [[12,56,78,[96,85,[14,89,100],63,200],300],500]

# **indexing**

# Extract the Number

#14

#300

#56

#89

#200

#100

#500

#85

x = [[12,56,78,[96,85,[14,89,100],63,200],300],500]

print (x[0][3][2][0])

print(x[0][4])

print(x[0][1])

print(x[0][3][2][1])

print(x[0][3][4])

print(x[0][3][2][2])

print(x[1])

print(x[0][3][1])

Output:

14

300

56

89

200

100

500

85

**#x = [[12,56,78,[96,85,[14,89,100],63,200],300],500]**

**# slicing**

**# extract these text from the list**

**#12,78**

**#14,100**

**#63,200**

**#96,85,[14,89,100],63**

**#56,78,[96,85,[14,89,100],63**

**x = [[12,56,78,[96,85,[14,89,100],63,200],300],500]**

**# 12, 78**

**x[0][0:3:2]**

**Output :** [12, 78]

#14,100

x[0][3][2][0::2]

**Output**

[14, 100]

#56,78,[96,85,[14,89,100],63

a= x[0][1:3]

b = x[0][3][0:4]

a+b

[56, 78, 96, 85, [14, 89, 100], 63]

**#create a new list and add the all number that is devisible of 5 from 40,100**

**# create a new list and add the all odd number from 15 to 30.**

**# x=["jan","feb","march","april","june","july","aug","sep","oct","nov","dec"]**

**#conver the all text in capital letter in this list.**

x=["jan","feb","march","april","june","july","aug","sep","oct","nov","dec"]

for i in x:

i = i.upper()

print(i)

JAN,FEB,MARCH,APRIL,JUNE,JULY,AUG,SEP,OCT,NOV,DEC

**#print all the even number of the month**

**# add in new list all odd number of the month in given list.**

**#x = [12,45,7,9,56,23,25,45,78,89,56,23,25,24,26]**

**# print the NUmebr that is divisible by 9**

**# find the NUmber who is repeated more than one time**

**# show positive and negative index of the Number**

**# show the index Number of the even number elements**

**# show the Negative index of odd Elements**

**# print the NUmebr that is divisible by 9**

**x = [12,45,7,9,56,23,25,45,78,89,56,23,25,24,26]**

**for i in x:**

**if i%9==0:**

**print(i,end=",")**

45,9,45,

**# find the Number who is repeated more than one time**

**for i in x:**

**if x.count(i)>1:**

**print(i,end=" ")**

45 56 23 25 45 56 23 25

**# show positive and negative index of the Number**

**x = [12,45,7,9,56,23,25,45,78,89,56,23,25,24,26]**

**y=len(x)**

**for i in range(y):**

**print(x[i], "positive index is : ", i)**

12 positive index is : 0

45 positive index is : 1

7 positive index is : 2

9 positive index is : 3

56 positive index is : 4

23 positive index is : 5

25 positive index is : 6

45 positive index is : 7

78 positive index is : 8

89 positive index is : 9

56 positive index is : 10

23 positive index is : 11

25 positive index is : 12

24 positive index is : 13

26 positive index is : 14

# show the index Number of the even number elements

x = [12,45,7,9,56,23,25,45,78,89,56,23,25,24,26]

y=len(x)

for i in range(y):

if x[i]%2==0:

print(x[i],"index :", i)

12 index : 0

56 index : 4

78 index : 8

56 index : 10

24 index : 13

26 index : 14

# show the Negative index of odd Elements

x = [12,45,7,9,56,23,25,45,78,89,56,23,25,24,26]

y=len(x)

for i in range(y):

if x[i]%2!=0:

print(x[i],"index :", i-y) **# -y gave negative index**

45 index : -14

7 index : -13

9 index : -12

23 index : -10

25 index : -9

45 index : -8

89 index : -6

23 index : -4

25 index : -3

#create a new list and add the all number that is devisible of 5 from 40,100

# create a new list and add the all odd number from 15 to 30.

# x = ["data",45.8,9.1,21j,True,"science",96,85,74,41,52]

# 3. print the all the text that given in this list

# 4. print all integer value in list

# 5. print string and Boolen value in list

#create a new list and add the all number that is devisible of 5 from 40,100

x = []

for i in range(40,100):

if i%5==0:

x.append(i)

print(x)

[40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95]

# create a new list and add the all odd number from 15 to 30.

x=[]

for i in range(15,30):

if i%2!=0:

x.append(i)

print(x)

[15, 17, 19, 21, 23, 25, 27, 29]

# 3. print the all the text that given in this list

x = ["data",45.8,9.1,21j,True,"science",96,85,74,41,52]

for i in x:

if type(i)==str:

print(i)

data

Science

# 4. print all integer value in list

x = ["data",45.8,9.1,21j,True,"science",96,85,74,41,52]

for i in x:

if type(i)==int:

print(i, end=" ")

96 85 74 41 52

# 5. print string and Boolen value in list

x = ["data",45.8,9.1,21j,True,"science",96,85,74,41,52]

for i in x:

if type(i)==str or type(i) ==bool:

print(i, end=" ")

data True science

# remove all 12 from this list

x = [12,12,12,12,45,78,45,12,56,23,45,12,89,10,12,15,14,20]

while 12 in x:

x.remove(12)

print(x)

[45, 78, 45, 56, 23, 45, 89, 10, 15, 14, 20]

# remove all the even number from the list

x = [12,12,12,12,45,78,45,12,56,23,45,12,89,10,12,15,14,20]

# Copy, Clear , Append -

y = x.copy()

x.clear()

for i in y:

if i%2!=0:

x.append(i)

print(x)

[45, 45, 23, 45, 89, 15]

# **Tuple**

* 1. Written in round braket (...)

2. Tuple is unchanageble or immutable

3. tuple allow duplicate values

4. Tuple are ordered.

5. we can write multiple type of data in a tuple.

6. Tuple are indexed.

**#How to create blank tuple**

x = tuple()

print(x)

()

**#Find out length, type**

x=(12,45,78,5.1,2.2,"Apple","mango", True,21j)

print(x)

print(type(x))

print(len(x))

(12, 45, 78, 5.1, 2.2, 'Apple', 'mango', True, 21j)

<class 'tuple'>

9

**#Slicing**

x[0:4]

(12, 45, 78, 5.1)

#**Indexing**

x[5]

'Apple'

# how to change value in Tuple - need to change in list first

# replace 500 at the value of sun

x =("sun","mon",1,3)

x = list(x)

x[0]=500

print(x)

[500, 'mon', 1, 3]

# **Packing & Unpacking[¶](http://localhost:8888/notebooks/Python all class.ipynb" \l "Packing-&-Unpacking)**

# **Packing a Tuple**

#when we create a tuple, we normally assign values to it. This is called Packing a tuple.

# **Unpacking a Tuple**

#when we create a tuple, we normally assign values to multiple variable from a tuple.

# **Packing**

x = (12,45,78)

(a,b,c) = x

print (a)

12

print(b)

45

print(c)

78

# **Unpacking**:

x= (45,78,88,56,233)

(\*a,b,c) = x

print(a)

[45, 78, 88]

x= (45,78,88,56,233)

(a,\*b,c) = x

print(b)

[78, 88, 56]

x= (45,78,88,56,233)

(a,b,\*c) = x

print©

[88, 56, 233]

Difference in List & Tupple

# **# List**

1. 1. written in square braket
2. 2. List is changable or Mutable
3. 3. List allow duplicate values
4. 4. List are ordered

# **Tupple**

1. Written in round braket (...) 2 .Tuple is unchanageble or immutable
2. tuple allow duplicate values
3. Tuple are ordered.
4. we can write multiple type of data in a tuple.
5. Tuple are indexed.

In [ ]:

​

In [10]:

*#x=10,y=15,z=65*

*#x=65, y=10, z=15 - Exchange value*

x**=**10

y**=**15

z**=**65

x,y,z**=**z,x,y

​

print(x,y,z)

65 10 15

#x,y,z=12,45,78

#Y = 12

#y =12

#z = 45

x,y,z = 12,45,78

z=y

y=x

print(x,y,z)

12 12 45

# **Tuple formula and its use**

#x = (12,45,12,12,12,78,8,56,23,20)

Question:

# count the total number of 12

# show the index number of 78

# sort the Tuple in Descending order

# add the total values in tuple

**Formula’s in Tuple**

# 1. count()

# 2. find() or index()

# 3. sorted

# 4. min

# 5. max

# 6. sum

**# count the total number of 12**

x = (12,45,12,12,12,78,8,56,23,20)

x.count(12)

4

**# show the index number of 78**

x.index(78)

5

**# sort the Tuple in Descending order**

**# Sort = It works only in List**

**#Sorted = It works in List, Tuple, Dic. need to take new variable**

x = (12,45,12,12,12,78,8,56,23,20)

y=sorted(x,reverse=True)

print(y)

[78, 56, 45, 23, 20, 12, 12, 12, 12, 8]

**# add the total values in tuple**

x = (12,45,12,12,12,78,8,56,23,20)

a=sum(x)

print("Addition of tuple :",a)

Addition of tuple : 278

**# Minimum value**

x = (12,45,12,12,12,78,8,56,23,20)

min(x)

8

**# Max value**

x = (12,45,12,12,12,78,8,56,23,20)

max(x)

78

**# Print the name whose first alphabet start with vowel**

x=("India","Pakistan","England","Australia","Srilanka")

y=[]

for i in x:

if i[0].lower() in ("a","e","i","o","u"):

print(i)

India

England

Australia

**# Print the name whose first alphabet start with vowel in list.**

x=("India","Pakistan","England","Australia","Srilanka")

y=[]

for i in x:

if i[0].lower() in "aeiou":

y.append(i)

print(y)

['India', 'England', 'Australia']

**# replace the value 500 at the place of 56 and 23**

x=(45,78,89,56,23)

x=list(x) **# convert into list**

x[-2] = 500

x[-1] = 500

x

[45, 78, 89, 500, 500]

**# add the total numerical values from this string**

**# 1st Method**

y=[]

x ="pri234n5c45@e%$1e"

for i in x:

if i.isdigit():

y.append(int(i))

sum(y)

24

**# 2nd Method**

s=0

for i in x:

if i.isdigit():

s=s+int(i)

print("Sum of all number is :",s)

Sum of all number is : 24

**28Th Jan 2024 :**

**Python Assignment:**

#print the hello world 10 times with the help of While loop

x=0

y=11

while y>=x:

print("Hello World")

x=x+1

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

Hello World

#find the all Number from 50 to 100 that is divisibe of 7 and 9.

x=100

y=50

while x>=y:

if y%7==0 and y%9==0:

print(y)

y=y+1

63

# Count the all number form 30 to 80 that is divisible of 6 and 8

a=0

for i in range(30,81):

if i%6==0 and i%8==0:

a=a+i

print(i)

print("Total count :",a)

48

72

Total count : 120

# Write a python program to count the all even Number from 20 to 40.

a=0

for i in range(20,41):

if i%2==0:

a=a+1

print(i)

print("all even number :", a)

20

22

24

26

28

30

32

34

36

38

40

all even number : 11

# Write a python program to sum the total value from 10 to 20 ?

a=0

for i in range(10,21):

a=a+i

print("sum :", a)

sum : 165

# write a python program to count the all even and Odd Number from 10 to 35

a=0

b=0

for i in range(10,36):

if i%2==0:

a=a+1

else:

b=b+1

print("Even no.:",a)

print("odd no :",b)

Even no.: 13

odd no : 13

# Write a python Program to show the factorial of Any number according to the help of user input.

x=int(input("enter the number"))

fact = 1

for i in range (1,x+1):

fact = fact\*i

print("Factorial : -", fact)

enter the number5

Factorial : - 120

# . find the Average of the first 10 Number with the help of while loop

a=0

x=10

b=0

while x>a:

a = a+1

b=b+a

avg=b/x

print("average : -", avg)

average : - 5.5

# Reverse the Number from 80 to 50 with help of for and while Loop

c=0

for i in range(80,50,-1):

print(i)

# print your name of alphabet with the help of while loop

x="Nitin Singh"

y=len(x)

z=0

while y>z:

print(x[z],end=" ")

z=z+1

N i t i n S i n g h

# print the reverse name with the help of while loop.

x="Nitin Singh"

y=len(x)

z=0

while z<y:

y-=1

print(x[y],end=" ")

h g n i S n i t i N

# write a python program to show the number between 100 to 150 that is divisible of 5 and 9 and 3

x=0

for i in range(100,151):

if i%5==0 and i%9==0 and i%3==0:

print(i)

x=x+1

print("total no.:",x)

135

total no.: 1

# Write a python program to show the all leap year from 1947 to 2024 and also show how Many years is a leap year

x=0

for i in range(1947,2025):

if i%4==0:

print(i)

x=x+1

print("Total no of leap year :",x)

1948

1952

1956

1960

1964

1968

1972

1976

1980

1984

1988

1992

1996

2000

2004

2008

2012

2016

2020

2024

Total no of leap year : 20

# **# Sets :**

# **List : [] Changable, Orderded, Allow Duplicates**

# **Tuple : () Unchangable, Orderded, Allow Duplicates**

# **Sets : {} Unchangable, Un-Orderded, Don't Allow Duplicates**

x={"PRINCE","NITIN","JATIN"}

print(x)

{'PRINCE', 'JATIN', 'NITIN'}

print(x)

{'JATIN', 'NITIN', 'PRINCE'}

x={12,3,12,12,45,58,45}

X

{3, 12, 45, 58}

# **Sets : {} Unchangable, Un-Orderded, Don't Allow Duplicates**

Always written in {} bracket.

x={12,11,13,15,25,12,13,15,20,80}

print(type(x))

print(x) **# It don't show the duplicate value**

<class 'set'>

{80, 20, 25, 11, 12, 13, 15}

# Convert list to set to remove duplicate value

x=[12,11,13,15,25,12,13,15,20,80]

print(type(x))

x=set(x)

list(x)

<class 'list'>

[11, 12, 13, 15, 80, 20, 25]

# How to create blank Sets

x=set()

print(x)

set()

# this will create dictionary not a set.

x={}

print(type(x))

set()

# this will create dictionary not a set.

x={}

print(type(x))

set()

# How to create blank list

x=[]

print(type(x))

<class 'list'>

# How to create blank tuple

x=()

print(type(x))

<class 'tuple'>

**# convert into upper letter in sets.**

x ={"Jan","Feb","Mar"}

y=[]

for i in x:

i=i.upper()

y.append(i)

set(y)

{'FEB', 'JAN', 'MAR'}

**# How to add value in sets**

**# add() - with the help of this function you can add single value**

**# Update() - through this function we can add multiple value at a time.**

x ={"Jan","Feb","Mar"}

X

{'Feb', 'Jan', 'Mar'}

# **Add**

x.add("Jul")

X

{'Feb', 'Jan', 'Jul', 'Mar'}

x.update({"Aug","Sep"}) # always use{} to update multiple value with function

X

{'Aug', 'Feb', 'Jan', 'Jul', 'Mar', 'Sep'}

**# How to delete value from sets.**

# **Remove - delete any value**

# **Discard - delete any value**

# **Pop - It delete random value**

#**Remove**

x={"a","b","c","d"}

x.remove("c")

X

{'a', 'b', 'd'}

# **Discard**

x={"a","b","c","d"}

x.discard("b")

X

{'a', 'c', 'd'}

# **Pop**

x={"a","b","c","d"}

x.pop()

X

{'b', 'c', 'd'}

**# Union() : with the help of this functio we can join two sets**

x={1,2,3,4}

y={5,6,7}x.

z=x.union(y) # we need to create new variable to apply union function.

print(z)

{1, 2, 3, 4, 5, 6, 7}

**# Update vs Union**

x={1,2,3,4}

y={5,6,7}

x.update(y) # No need to create new variable to apply update function.

X

{1, 2, 3, 4, 5, 6, 7}

**# Intersection\_update() : it keeps the value that available in both sets**.

x={"apple","banana","lichi", "mango"}

y={"banana","mango"}

x.intersection\_update(y)

X

{'banana', 'mango'}

**# Intersection() : it give the value that available in both sets in new variable**

x={"apple","banana","lichi", "mango"}

y={"banana","mango"}

z= x.intersection(y)

Z

{'banana', 'mango'}

# **Find out value which does not exit in both sets**

**#Symmetrci\_difference\_update()** : - it show the value which not exist in both sets.

x={"One","Two","Three"}

y={"Four","Five","One","Three"}

x.symmetric\_difference\_update(y)

x

{'Five', 'Four', 'Two'}

**10 Feb 2024**

# delete all the value who is less than 50

x = [12,45,78,89,56,23,15,48,89,10]

y=x.copy() # **copy all data in y**

Y

[12, 45, 78, 89, 56, 23, 15, 48, 89, 10]

x.clear() **# clear data from x**

X

[]

for i in y:

if i>=50:

x.append(i)

X

[78, 89, 56, 89, 78, 89, 56, 89] # Final Output

# add 10 in all the elements in this list

x = [12,45,78,89,56,23,15,48,89,10]

for i in x:

print(i+10)

22

55

88

99

66

33

25

58

99

20

# find the average of the Number who is greater than or equal to 50

x = [12,45,78,89,56,23,15,48,89,10]

s=0

c=0

for i in x:

if i>50:

s+=i

c+=1

avg = s/c

print("Average :", avg)

Average : 78.0

# **# Dictionary**

# # List [] : Ordered, indexed, , changable, Duplicated value allowed, multiple value add,

# # Tuple () : ordered, indexed, Unchangable, allow duplicate, multiple value add,

# # Set {}: Unorderded, Unindexed, Unchangable, Don't allow duplicates

# # Dictionary {}: Ordered, Changeable, Don't allow duplicates, Written in pair & Key

# **Intruduction of Dictionary**

1. **Dictionary are written in {} braket.**
2. **It always written in pair of key and values**
3. **Orderded**
4. **Not allow duplicate value**
5. **Changeable**

**x={1:"One", 2:"Two", 3:"Three", 4:"Four"}**

print(x)

print(len(x))

print(type(x))

{1: 'One', 2: 'Two', 3: 'Three', 4: 'Four'}

4

<class 'dict'>

**car={"Brand":"BMW","Color":"Red","Year":2021,"Model":"XE"}**

print(car)

{'Brand': 'BMW', 'Color': 'Red', 'Year': 2021, 'Model': 'XE'}

print(car["Brand"])

BMW

print(car["Color"])

Red

print(car["Year"])

2021

print(car["Model"])

XE

Dd

**#Formula's under Dictionary**

**x={"Name":"Nitin","Age": 32,"City":"Delhi"}**

print(x)

print(type(x))

print(len(x))

{'Name': 'Nitin', 'Age': 32, 'City': 'Delhi'}

<class 'dict'>

3

**# How to check total key in dictionary**

**# dic.key() - It shows the total key**

x**=**{"Name":"Nitin","Age": 32,"City":"Delhi"}

y**=**x.keys()

print(y)

dict\_keys(['Name', 'Age', 'City'])

**# how to check total values in dictionary**

**# dic.values()**

x**=**{"Name":"Nitin","Age": 32,"City":"Delhi"}

y**=**x.values()

print(y)

dict\_values(['Nitin', 32, 'Delhi'])

**# How to show the key & value in tuple under list**

**# items()**

x={"Name":"Nitin","Age": 32,"City":"Delhi"}

y=x.items()

print(y)

dict\_items([('Name', 'Nitin'), ('Age', 32), ('City', 'Delhi')])

**# How to show the value of the key**

x={"Name":"Nitin","Age": 32,"City":"Delhi"}

x["Name"]

'Nitin'

x["Age"]

32

x["City"]

'Delhi'

**x={"Name":"Nitin","Age": 32,"City":"Delhi"}**

y=str(input("Enter the Key"))

print(x[y])

Enter the KeyCity

Delhi

# **Add & Remove in Dictionary**

**# How to add a new key and values in the dictionary**

**# Without Formula**

x={"Name":"Nitin","Age": 36,"City":"Delhi"}

x["Company"]= "BOA"

print(x)

{'Name': 'Nitin', 'Age': 36, 'City': 'Delhi', 'Company': 'BOA'}

**# With Formula - Update**

x={'Name': 'Nitin', 'Age': 36, 'City': 'Delhi', 'Company': 'BOA'}

x.update({"Designation":"Manager"})

print(x)

{'Name': 'Nitin', 'Age': 36, 'City': 'Delhi', 'Company': 'BOA', 'Designation': 'Manager'}

# How to delete values from the Dictionary

1. Pop

2. Popitems

3. Clear

4. del

#**Pop :**

x={1:"One",2:"Two",3:"Three"}

x.pop(2)

print(x)

{1: 'One', 3: 'Three'}

**#Popitem # : It delete the last value**

x={1:"One",2:"Two",3:"Three"}

x.popitem() # By default last values

X

{1: 'One', 2: 'Two'}

**# Clear : It delete the complete value**

x={1:"One",2:"Two",3:"Three"}

x.clear()

X

{}

ddd

# **11 Feb 2024**

**# State and Capital**

x={"Bihar":"Patna","Assam":"Dispur","Goa":"Panji","Haryana":"Chandigarh","Manipur":"Imphal","Tamil Nadu":"Chennai"}

print("|"\*100)

y=str(input("Enter the state name"))

z=x[y]

print("The Capital of",y,"is",z)

print("|"\*100)

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Enter the state nameBihar

The Capital of Bihar is Patna

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

**#Find out the Length of Dictionary**

x={"Bihar":"Patna","Assam":"Dispur","Goa":"Panji","Haryana":"Chandigarh","Manipur":"Imphal","Tamil Nadu":"Chennai"}

print("Total no. of State is :",len(x))

Total no. of State is : 6

**# keys() - It show the total keys of dictionary**

x={"Bihar":"Patna","Assam":"Dispur","Goa":"Panji","Haryana":"Chandigarh","Manipur":"Imphal","Tamil Nadu":"Chennai"}

x.keys()

dict\_keys(['Bihar', 'Assam', 'Goa', 'Haryana', 'Manipur', 'Tamil Nadu'])

**# values() - It show the total values of dictionary**

x={"Bihar":"Patna","Assam":"Dispur","Goa":"Panji","Haryana":"Chandigarh","Manipur":"Imphal","Tamil Nadu":"Chennai"}

x.values()

dict\_values(['Patna', 'Dispur', 'Panji', 'Chandigarh', 'Imphal', 'Chennai'])

**# items() - It show the total items of dictionary**

x={"Bihar":"Patna","Assam":"Dispur","Goa":"Panji","Haryana":"Chandigarh","Manipur":"Imphal","Tamil Nadu":"Chennai"}

x.items()

dict\_items([('Bihar', 'Patna'), ('Assam', 'Dispur'), ('Goa', 'Panji'), ('Haryana', 'Chandigarh'), ('Manipur', 'Imphal'), ('Tamil Nadu', 'Chennai')])

**# How to add the new keys in this dictionary.**

x={"Bihar":"Patna","Assam":"Dispur","Goa":"Panji","Haryana":"Chandigarh","Manipur":"Imphal","Tamil Nadu":"Chennai"}

x.update({"Delhi":"New Delhi"})

print(x)

{'Bihar': 'Patna', 'Assam': 'Dispur', 'Goa': 'Panji', 'Haryana': 'Chandigarh', 'Manipur': 'Imphal', 'Tamil Nadu': 'Chennai', 'Delhi': 'New Delhi'}

**# How to delete the new keys in this dictionary.**

x={"Bihar":"Patna","Assam":"Dispur","Goa":"Panji","Haryana":"Chandigarh","Manipur":"Imphal","Tamil Nadu":"Chennai"}

x.pop("Bihar")

print(x)

{'Assam': 'Dispur', 'Goa': 'Panji', 'Haryana': 'Chandigarh', 'Manipur': 'Imphal', 'Tamil Nadu': 'Chennai'}

**# how to sort the keys in dictionary**

x={"Bihar":"Patna","Assam":"Dispur","Goa":"Panji","Haryana":"Chandigarh","Manipur":"Imphal","Tamil Nadu":"Chennai"}

sorted(x.keys())

['Assam', 'Bihar', 'Goa', 'Haryana', 'Manipur', 'Tamil Nadu']

**# how to sort the values in dictionary**

x={"Bihar":"Patna","Assam":"Dispur","Goa":"Panji","Haryana":"Chandigarh","Manipur":"Imphal","Tamil Nadu":"Chennai"}

sorted(x.values())

['Chandigarh', 'Chennai', 'Dispur', 'Imphal', 'Panji', 'Patna']

**# print all values**

x={"Bihar":"Patna","Assam":"Dispur","Goa":"Panji","Haryana":"Chandigarh","Manipur":"Imphal","Tamil Nadu":"Chennai"}

for i,j in x.items():

print(i," : ",j)

Bihar : Patna

Assam : Dispur

Goa : Panji

Haryana : Chandigarh

Manipur : Imphal

Tamil Nadu : Chennai

**# Show the all state and capital whose state name have letter "i"**

x={"Bihar":"Patna","Assam":"Dispur","Goa":"Panji","Haryana":"Chandigarh","Manipur":"Imphal","Tamil Nadu":"Chennai"}

for i,j in x.items():

if "i"in i:

print(i," : ",j)

Bihar : Patna

Manipur : Imphal

Tamil Nadu : Chennai

**# Exchange all the key to values and values to key**

x={"Bihar":"Patna","Assam":"Dispur","Goa":"Panji","Haryana":"Chandigarh","Manipur":"Imphal","Tamil Nadu":"Chennai"}

y={}

for i,j in x.items():

y[j]=I

print(y)

{'Patna': 'Bihar', 'Dispur': 'Assam', 'Panji': 'Goa', 'Chandigarh': 'Haryana', 'Imphal': 'Manipur', 'Chennai': 'Tamil Nadu'}

# **Nested Dictionary**

# **17 Feb 2024**

**Example : 1**

dic={"Name":["Nitin","Prince","Ravi"],"Age":[25,36,34]}

dic["Name"]

['Nitin', 'Prince', 'Ravi']

**Example 2:**

a={"hnd":45,"eng":59,"maths":75,"sc":80,"sst":65}

b={"hnd":65,"eng":89,"maths":79,"sc":82,"sst":80}

c={"hnd":55,"eng":79,"maths":80,"sc":75,"sst":74}

d={"hnd":95,"eng":69,"maths":90,"sc":92,"sst":92}

Marks={"Nitin":a,"Prince":b,"Jatin":c,"Ravi":d}

len(Marks)

4

**Marks.keys()**

dict\_keys(['Nitin', 'Prince', 'Jatin', 'Ravi'])

**# check all the items**

**Marks.items()**

dict\_items([('Nitin', {'hnd': 45, 'eng': 59, 'maths': 75, 'sc': 80, 'sst': 65}), ('Prince', {'hnd': 65, 'eng': 89, 'maths': 79, 'sc': 82, 'sst': 80}), ('Jatin', {'hnd': 55, 'eng': 79, 'maths': 80, 'sc': 75, 'sst': 74}), ('Ravi', {'hnd': 95, 'eng': 69, 'maths': 90, 'sc': 92, 'sst': 92})])

**# check all the values**

**Marks.values()**

dict\_values([{'hnd': 45, 'eng': 59, 'maths': 75, 'sc': 80, 'sst': 65}, {'hnd': 65, 'eng': 89, 'maths': 79, 'sc': 82, 'sst': 80}, {'hnd': 55, 'eng': 79, 'maths': 80, 'sc': 75, 'sst': 74}, {'hnd': 95, 'eng': 69, 'maths': 90, 'sc': 92, 'sst': 92}])

**# How to check Nitin's marks**

Marks["Nitin"]

{'hnd': 45, 'eng': 59, 'maths': 75, 'sc': 80, 'sst': 65}

**# How to check marks in one subject**

Marks["Nitin"]["eng"]

59

**# Find the total marks of Nitin**

a={"hnd":45,"eng":59,"maths":75,"sc":80,"sst":65}

b={"hnd":65,"eng":89,"maths":79,"sc":82,"sst":80}

c={"hnd":55,"eng":79,"maths":80,"sc":75,"sst":74}

d={"hnd":95,"eng":69,"maths":90,"sc":92,"sst":92}

Marks={"Nitin":a,"Prince":b,"Jatin":c,"Ravi":d}

x=Marks["Nitin"].values()

sum(x)

324

**# Find out Marks with input the user name**

user=str(input("enter the name :-"))

x=Marks[user].values()

Total\_Marks = sum(x)

print("Total Marks of",user,"is",Total\_Marks)

enter the name :-Prince

Total Marks of Prince is 395

**# # Find out Marks with % input the user name**

user=str(input("enter the name :-"))

x=Marks[user].values()

Total\_Marks = sum(x)

t = len(Marks[user].keys())

per=Total\_Marks/t

print("Total number : 500")

print("Percentage :-",per)

print("Total number obtained is :-",Total\_Marks)

enter the name :-Nitin

Total number : 500

Percentage :- 64.8

Total number obtained is :- 324

**# # Find out Division with % input the user name**

user=str(input("enter the name :-"))

x=Marks[user].values()

Total\_Marks = sum(x)

t = len(Marks[user].keys())

per=Total\_Marks/t

if per>=60:

div="First Division"

elif per>=45:

div="Second Division"

elif per>=33:

div="Third Division"

else:

div="Fail"

print("Total number : 500")

print("Percentage :-",per)

print("Total number obtained is :-",Total\_Marks)

print("Division :-",div)

enter the name :-Nitin

Total number : 500

Percentage :- 64.8

Total number obtained is :- 324

Division :- First Division

**# Find the results through Roll. no.**

a={"hnd":45,"eng":59,"maths":75,"sc":80,"sst":65}

b={"hnd":65,"eng":89,"maths":79,"sc":82,"sst":80}

c={"hnd":55,"eng":79,"maths":80,"sc":75,"sst":74}

d={"hnd":95,"eng":69,"maths":90,"sc":92,"sst":92}

Marks={"Nitin":a,"Prince":b,"Jatin":c,"Ravi":d}

Roll\_no={1:"Nitin",2:"Prince",3:"Jatin",4:"Ravi"}

z=int(input("enter the Roll.no :-"))

name=Roll\_no[z]

x=Marks[name].values()

Total\_Marks = sum(x)

t = len(Marks[user].keys())

per=Total\_Marks/t

if per>=60:

div="First Division"

elif per>=45:

div="Second Division"

elif per>=33:

div="Third Division"

else:

div="Fail"

print("Student Name :-",name)

print("Total number : 500")

print("Percentage :-",per)

print("Total number obtained is :-",Total\_Marks)

print("Division :-",div)

print(Roll\_no)

enter the Roll.no :-2

Student Name :- Prince

Total number : 500

Percentage :- 79.0

Total number obtained is :- 395

Division :- First Division

{1: 'Nitin', 2: 'Prince', 3: 'Jatin', 4: 'Ravi'}

**# Write a programe to create a dict with the help of user input**

**#x={"Name":"Nitin Sing","Age":35,"city":"Ajmer"}**

dic={}

L=int(input("enter the length of dictionary :"))

for i in range(L):

key=(input("enter the key :-"))

value=(input("enter the value :-"))

dic[key] =value

print(dic)

enter the length of dictionary :3

enter the key :-name

enter the value :-Nitin Singh

enter the key :-Age

enter the value :-35

enter the key :-City

enter the value :-Ajmer

{'name': 'Nitin Singh', 'Age': '35', 'City': 'Ajmer'}

**# convert the dictionary of values from key to values and values to key**

x={"Name":"Prince sharma","Age":35,"city":"Delhi"}

x1=x.copy()

x.clear()

for i,j in x1.items():

x[j]=i

print(x)

{'Prince sharma': 'Name', 35: 'Age', 'Delhi': 'city'}

# **18th Feb 2024**

Function :

## **chapter:- function in python**

What is function? A function is a back of code which only runs when it is called. You can pass data , known as perameter , into a function. a functioncan return data a result.

creating a function in pythoon Function is defined using the def keyboard:

There are mainly two fuctions.

1. **user defined function** : the user defined function user perform the specific task
2. **Built in function** : the Built in function are those function that are predifinied in python

# **# Built in Function**

x=[15,30,45,75]

sum(x)

165

x=[15,30,45,75]

min(x)

15

x=[15,30,45,75]

max(x)

75

# **User Define Function**

**# Creating the Function**

def table(x):

for i in range(1,11):

print(i\*x)

**# Calling the Function**

table(5)

5

10

15

20

25

30

35

40

45

50

1.

def nitin():  **# Blank Parameter**

print("Hello World")

nitin()

Hello World

2.

def nitin(x,y): # with parameter

print(x+y)

nitin(12,13)

25

3.

def nitin():

print("Ajmer is a clean city")

nitin("My name is nitin") # cannot assign another parameter

**---------------------------------------------------------------------------TypeError** Traceback (most recent call last)

Cell **In[34], line 1----> 1** nitin("My name is nitin")

**TypeError**: nitin() takes 0 positional arguments but 1 was given

3.

def nitin():

print("Ajmer is a clean city")

nitin()

Ajmer is a clean city

4.

def nitin(x="Hello World"): # Give parameter as x="Hello World"

print(x)

nitin()

Hello World

nitin("Ajmer is a clean city")

Ajmer is a clean city

5.

def nitin(x,y,z): # Function for addition

a = x + y + z

print("Addition :-",a)

nitin(45,75,25)

Addition :- 145

sum([45,75,25]) # 2nd Way for Addition

145

6.

def nitin(x):

s=0

for i in x:

s+=i

print(s)

x=[12,35,45,70]

X

[12, 35, 45, 70]

sum(x) # Through Built in function

162

nitin(x) # through user define function

162

**# Create a function to show the number is odd or even**

x=45 **# Through Built in Function**

if x%2==0:

print("Even")

else:

print("odd")

odd

def check(x): **# Through user function**

if x%2==0:

print("Even")

else:

print("odd")

check(45)

check(72)

odd

Even

**# Create a function to print to find out wheater text is odd or even**

def nitin(x):

L=(len(x))

if L%2==0:

print("even")

else:

print(odd)

nitin("Himachal")

even

**# Create a function to print the all even text from the string**

def even(x):

for i in range (len(x)):

if i%2==0:

print(x[i],end=" ")

x= "Himanchal"

even(x)

H m n h l

**# Create a function to count all the odd number**

def odd(x):

c=0

for i in x:

if i%2!=0:

c+=1

print("Total odd number :", c)

x=[1,4,5,7,8,9,6,5,4,2,1,5,98,7,5,6]

odd(x)

Total odd number : 9

**24 Feb 2024**

**#default parameter**

**def fun(a,b,c=0):**  ### Create the function

**d=a+b+c**

**print("addition :",d)**

fun(45, 78) ### Call the Function

addition : 123

fun(45,55,65)

addition : 165

# create a function to extract the first 3 alphabet from the text

# and last three alphabet from the text

def ex(a):

print("First 3 letter is :", a[0:3])

print("Last 3 letter is :", a[-3:])

ex("India") ### call the function

First 3 letter is : Ind

Last 3 letter is : dia

**# create a function to reverse the text**

def rev(a):

print("Reverse Text :", a[-1: :-1])

rev("India")

Reverse Text : aidnI

**# Create a function to count the total number of alphabet "i" & "l"**

def co(x):

x = x.lower()

a=x.count("i")

b=x.count("l")

print("i :-",a)

print("l :-",b)

m="Hiiiiiimachall Pradesh"

n="arunachalllll Pradesh"

co(m)

co(n)

i :- 6

l :- 2

i :- 0

l :- 5

**# Number of arguments**

By default , a function must be called with the correct number of arguments. Meaning that if your functione expects two argument, you have to call he function with argument,no more ,and not less

**Arbitatry arguments, \*args**

If you do not know how many arguments that will be passed into your function, add astrick \* before the perameter name in the function definition. This way the function will receive a tuple of arguments, and can access the item in the index:

def abc(\*x):

print(x[0])

print(x[1])

print(x[2])

print(x[3])

abc(45,55,32,"Brillica")

45

55

32

Brillica

def xy(\*a):

print(a[0:])

xy(45,4564,454,321,)

(45, 4564, 454, 321)

Keyword Arbitatry Arguments

In [44]:

**# Keyword Argument**

# You can also send arguments with the key = value syntax

# this way the order of the arguments does not matter

**#without keyword parameter**

def x(Name,Age,Class):

print("My name is ", Name)

print("I am ", Class,"with Maths honors")

print("I am", Age," year old")

x(Name="Nitin Singh", Age=22, Class="Graduate")

My name is Nitin Singh

I am 12 with Maths honors

I am 22 year old

**# with Keyword parameter**

def x(\*\*a):

print("My name is ",a ['name'])

print("I am ", a['Class'],"with Maths honors")

print("I am", a['Age'],"year old")

x(name="Nitin Singh", Age=35, Class="Graduate")

My name is Nitin Singh

I am Graduate with Maths honors

I am 35 year old

# **Lambda Function :[¶](http://localhost:8888/notebooks/Python all class.ipynb" \l "Lambda-Function-:)**

#\* A Lambda function is a small anoynymous function

#\* A Lambda function ca take any number of arguments, but can only have one expression

x = lambda a,b:a+b

x(45,45)

90

x=lambda a,b,c,d: a+b\*c/d

x(45,55,10,2)

320.0

**25 Feb 2024**

# Exception Handlling

* When an error occurs, or exception as we call it ,python will normally stop
* and generate an error
* message. these exceptions can be handled using the try statement:
* the try block lets you test a block of code for errors.
* the except block lets you handle the error.
* the else block lets you execute code when there is no error.
* **TRY -** the try block lets you test a block of code for errors. We create logic in Try.
* **EXCEPT -** the except block lets you handle the error. Here, we write Threads or handle the Error.
* **ELSE -** the else block lets you execute code when there is no error.
* **FINALLY -** the finally block lets you execute code , regardless of the result of the try -and except block.

**Example: 1**

**try:**

**print(x)** # when we don't enter value it will show error.

**Cell In[5], line 3 ^SyntaxError:** incomplete input

**Example: 2**

try:

print(z) # we don't give any value but instead off error, system will print ("enter the value")

except:

print("please enter value")

please enter value

**Example: 3**

try:

Z=10

print(z) # Now it will give z value in output .

except:

print("please enter value")

10

**Example: 4**

**# Else :**

**when we don't place value.**

**try:**

**print(a)**

**except:**

**print("enter the value")**

**else:**

**("value is correct")**

enter the value

**Example: 5**

**# Else :**

**# when we place value.**

**try:**

**a = 25**

**print(a+10)**

**except:**

**print("enter the value")**

**else:**

**("value is correct")**

35

**# Try & Except**

**x=5000**

**if x>3000**

**print("True")**

**Cell In[9], line 3 if x>3000 ^SyntaxError:** expected ':'

**If we don’t want that error to be printed.**

**x=5000**

**if x>3000**

**print("True")**

**02 March 2024**

x =[1,2,3,4]

y=["One","Two","Three","Four"]

for i in (x,y):

for j in i:

print(j)

1

2

3

4

One

Two

Three

Four

# **Zip Function**

# **Zip - it is use to join two or more than two container in list, tuple and dictionary.**

# Print "Four" if we i = 4 from **List**

x =[1,2,3,4]

y=["One","Two","Three","Four"]

z=zip(x,y)

for i,j in z:

if i==4:

print(j)

Four

# Print "Four" if we i = 4 from **Dictionary**

dic={1:"One",2:"Two",3:"Three"}

for i,j in dic.items():

if i==3:

print(j)

Three

# Convert List into Dictionary

x =[1,2,3,4]

y=["One","Two","Three","Four"]

dict(zip(x,y))

{1: 'One', 2: 'Two', 3: 'Three', 4: 'Four'}

# how to zip 3 lists.

x =[1,2,3,4]

y=["One","Two","Three","Four"]

z=["i","ii","iii","iv"]

a= list(zip(x,y,z))

print(a)

[(1, 'One', 'i'), (2, 'Two', 'ii'), (3, 'Three', 'iii'), (4, 'Four', 'iv')]

#

x =[1,2,3,4]

y=["One","Two","Three","Four"]

z=["i","ii","iii","iv"]

a= list(zip(x,y,z))

for i,\*j in a: # \*j (it assign all remaining value in J and 1st value place under i )

print(i,j)

1 ['One', 'i']

2 ['Two', 'ii']

3 ['Three', 'iii']

4 ['Four', 'iv']

# **Oops**

oops :- oops object oriented programmings

* A class is an objective oriented programming , almost everything is python is an objective with its properties and methods.
* A class like an object constructor or a "blue print" for creating an object
* objects are related to real entities such as phone , mobile car extract
* Major principles of objects oriented programming

1. Class
2. objects
3. methods
4. inheritance
5. polymorphism
6. Data abstraction
7. Encapsulation.m

class A():

print("Hello World")

Hello World

Example :

class A(): #Function

x=10

y=20

z=x+y

b=A() # call the function

print(b.z)

30

## Two Def function in one class function

class B:

def fun(self,x,y): # Always use self function for 2 def function.

self.x = x

self.y = y

self.z = x+y

def fun2(self,a,b):

self.a = a

self.b = b

self.c = a+b

c = B() # First call the class

c.fun(45,75) # call the function with parameter

print(c.z) # print the value of function with class object

120

# 3 Function in 1 class

class parent():

def add (nitin,a,b):

nitin.a = a

nitin.b = b

nitin.c = a+b

def sub (nitin,d,e):

nitin.d = d

nitin.e = e

nitin.f = d-e

def exp (nitin,g,h):

nitin.g = g

nitin.h = h

nitin.i = g\*\*h

p = parent() # call the class

p.add(12,25) # entering parameter

p.sub(89,78) # entering parameter

p.exp(5,3) # entering parameter

print(p.c)

37

print(p.i)

125

print(p.f)

11

**03 March 2024**

**# OOPS - Object Oriented Programing**

# **Inheritance**

1. Single Inheritance
2. Multiple Inheritance
3. Hybrid Inheritance
4. Multilevel Inheritance

# 1 class with multiple function

class A(): # Program

def rev(self,x):

self.x = x

self.y = x[-1::-1]

def cap(self,z):

self.z = z

self.m = z.upper()

c = A() # call the class

c.rev("akansha") # enter the parameter

c.cap("nitin") # enter the parameter

print("Reverse the text :",c.y)

print("Capital letter text :", c.m)

Reverse the text : ahsnaka

Capital letter text : NITIN

**# Single Inheritance - one class inherited in 2nd class**

class A():

def table(self,x):

self.x = x

for i in range(1,11):

print(f"{i} X {x} = {i\*x}") # f "" - This function use jf need to type multiple "".

class B(A):

def even(self,y):

self.y = y

i in range(y):

if i%2==0:

print("Even Number :",i)

c= B() # call the 2nd class

c.table(5) # enter the parameter

c.even(20) # enter the parameter

1 X 5 = 5

2 X 5 = 10

3 X 5 = 15

4 X 5 = 20

5 X 5 = 25

6 X 5 = 30

7 X 5 = 35

8 X 5 = 40

9 X 5 = 45

10 X 5 = 50

Even Number : 0

Even Number : 2

Even Number : 4

Even Number : 6

Even Number : 8

Even Number : 10

Even Number : 12

Even Number : 14

Even Number : 16

Even Number : 18

**\_\_init\_\_ Function :** its call the function without the function name.

**# \_\_init\_\_ Function:**

class A():

def \_\_init\_\_(self,x,y):

self.x = x

self.y = y

self.z = x % y

c = A(89,15)

print ("Modules : ",c.z)

Modules : 14

**# Single Inheritance with multiple function**

class A():

def \_\_init\_\_(self,name):

self.name = name

class B():

def \_\_init\_\_(self,name,age):

A.\_\_init\_\_(self,name) # need to add 1st class in 2nd class

self.name = name

self.age = age

print("My name is :", name)

print("I am", age, "years old")

c = B("Nitin Singh", 24) call the 2nd class

My name is : Nitin Singh

I am 24 years old

# **Multiple Inheritance[¶](http://localhost:8888/notebooks/Python all class.ipynb" \l "Multiple-Inheritance)**

**# 1st class inherit in 2nd class then 2nd class inherit in 3rd class.**

class A():

def first(self,x):

self.x = x

print("Name : ",x)

class B(A):

def second(self,y):

self.y = y

print("Age : ",y)

class C(B):

def third(self,z):

self.z = z

print("City : ",z)

c = C() # call the class

c.first("Viart kohli") # give parameter

c.second(35) # give parameter

c.third("Delhi") # give parameter

Name : Viart kohli

Age : 35

City : Delhi

# **Hybrid Inheritance[¶](http://localhost:8888/notebooks/Python all class.ipynb" \l "Hybrid-Inheritance)**

**# 1st & 2nd class put in class 3rd**

class A():

def first(self,x):

self.x = x

print("Name : ",x)

class B():

def second(self,y):

self.y = y

print("Age : ",y)

class C(A,B):

def third(self,z):

self.z = z

print("City : ",z)

c = C() # call the final class

c.first("Viart kohli") # give parameter

c.second(35) # give parameter

c.third("Delhi") # give parameter

Name : Viart kohli

Age : 35

City : Delhi

**# Hybrid Inheritance with \_\_init\_\_**

class First():

def \_\_init\_\_(self,a):

self.a = a

class Second():

def \_\_init\_\_(self,b):

self.b = b

class third(First,Second):

def \_\_init\_\_(self,a,b,c):

First.\_\_init\_\_(self,a)

Second.\_\_init\_\_(self,b)

self.c = c

print("name :",a)

print("age :",b)

print("City :",c)

k = third("Rohit sharma",38,"Mumbai") # call the final class & enter parameter

name : Rohit sharma

age : 38

City : Mumbai

**09th March 2024**

# **Numpy - Numerical Python**

what is numpy?

* Numpy stand for numerical python. Numpy is python library used for working with array.

1. NumPy is a Python library.
2. NumPy is used for working with arrays.
3. NumPy is short for "Numerical Python".

**# How to install Python Library numpy**

**!pip install numpy**

**-==================================================**

**import numpy**

**x = numpy.array([15,23,40])**

**X**

array([15, 23, 40])

**# what is the key difference between list and array**

# 1. array is faster than List

# 2. Here is 100+ formula for mathematics

# 3. Here i can add or change the dimention of the array

# 4. In array we can create matrix easly

# Creating a numpy array

# In numpy most powerful data type is string.

x = numpy.array(["data",45.56,50,True]) #

print(x)

['data' '45.56' '50' 'True']

x = numpy.array([["data",45.56,50,True]]) #

print(x)

print(x.dtype)

print(type(x))

print(x.ndim)

[['data' '45.56' '50' 'True']]

<U32

<class 'numpy.ndarray'>

2

**#nd** = Number of Dimention

# **Dimentions** :- Dimention is the total NUmber of List in a Array

# **Dimention** :- it is a combination of row and column

# **dtype** :- its show the data type of array

# **shape** :- is show the shape total number of row and column

# **len** :- its show the length of the array

# **ndim** :- its show the number of Dimention of the array

# **type** :- its show the type of the array

# 2nd most powerful data type is Complex

x = numpy.array([45.56,50,True, 21j])

x

x.dtype

dtype('complex128')

# 3rd most powerful data type is float

import numpy as np

x = np.array([45.56,50,True])

print(x)

x.dtype

[45.56 50. 1. ]

Out[24]:

dtype('float64')

# 1. Strings : U64 or U32

# 2. Complex

# 3. Float

# 4. Integar : - int32 or int64

# 5. Boolen

import numpy as np

x = np.array([[[4,5,6],[7,8,4],[1,2,3]]])

print(x)

print("Total number of Row & Column :- ", x.shape)

print(x.ndim) # its show the total number of list

[[[4 5 6]

[7 8 4]

[1 2 3]]]

Total number of Row & Column :- (1, 3, 3)

3

x.dtype

dtype('int32')

**# arrange Function : its create a array with range of sequence number**

# create a array from 1 to 20

x= np.arange(1,21)

X

array([ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,

18, 19, 20])

**# range(starting point, ending point, step size)**

**# arange(starting point, ending point, step size)**

x = np.arange(1,21,2)

print(x)

[ 1 3 5 7 9 11 13 15 17 19]

**# How to create a metrix**

# 3 x 3

[[1 2 3]

[4 5 6]

[7 8 9]]

x = np.array([[1,2,3],[4,5,6],[7,8,9]])

x.shape **# Shape - provide rows & col. Count**

numpy.ndarray

x.ndim

2

type(x)

numpy.ndarray

len(x)

3

x.size

9

x = [[1,2,3],[4,5,6],[7,8,9]]

print(x)

len(x)

[[1, 2, 3], [4, 5, 6], [7, 8, 9]]

Out[51]:

3

**# create a dimention of 10 array**

x = np.array([1,2,3],ndmin = 10)

print(x)

[[[[[[[[[[1 2 3]]]]]]]]]]

**# create a dimention of 20 array**

x = np.array([1,2,3],ndmin = 20)

print(x)

[[[[[[[[[[[[[[[[[[[[1 2 3]]]]]]]]]]]]]]]]]]]]

**# ndmin : - parameter of array**

# its create dimention is array according to user

x = np.arange(10)

print(x)

[0 1 2 3 4 5 6 7 8 9]

**# reshape function : its provide the shape of any array in matrix**

# 10

# 2x 5

# 5 x 2

# 1 x 10

# 10 x 1

x = np.arange(10)

X

array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

x.reshape(5,2)

array([[0, 1],

[2, 3],

[4, 5],

[6, 7],

[8, 9]])

x.reshape(2,5)

array([[0, 1, 2, 3, 4],

[5, 6, 7, 8, 9]])

x.reshape(1,10)

array([[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]])

x.reshape(10,1)

array([[0],

[1],

[2],

[3],

[4],

[5],

[6],

[7],

[8],

[9]])

x = np.arange(20,60)

X

array([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])

x.size

40

# 4 x 5

# 5 x 4

x.reshape(2,4,5) **# reshape (no\_metrics,rows,columns)**

array([[[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]]])

**10 March 2024**

#numpy

* array
* size
* len
* dtype
* ndim
* arange
* Reshape
* ndmin

**# Create 3D array**

**# 2D array**

import numpy as np

x = np.array([[12,45,78,89,56]])

print(x)

[[12 45 78 89 56]]

x.ndim

2

x.size

5

len(x)

1

# Arange

x=np.arange(1,11)

x

array([ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

**# Indexing & Slicing in array**

**# Indexing - extract the one text or number at a time.**

**# Slicing - extract the range of character or number at a time.**

**# Indxing**

x = np.array([1,4,7,8,5,2])

x[1]

4

x = np.array([[[[45,78,89]]]])

X

array([[[[45, 78, 89]]]])

**# extract 45 from this array - 1st Methord**

x = np.array([[[[45,78,89]]]])

x[0][0][0][0]

45

**# extract 78 from this array - 2nd Methord**

x = np.array([[[[45,78,89]]]])

x[0,0,0,1]

78

**# extract 45,78 from this array**

x = np.array([[[45,78,89,56,23]]])

x[0,0,0:2]

45,78

**# Extract 45,89,23 with setp size 2**

x = np.array([[[45,78,89,56,23]]])

x[0,0,0:5:2] # step size 2

array([45, 89, 23])

**# Extract the these Number [56,89]**

x = np.array([[[[12,45,89,56]]]])

x[0,0,0,-1:-3:-1]

array([56, 89])

Syntax :

# np.ones((no\_of\_matrix,no\_of\_rows,No\_of\_Col))

# np.zeros((no\_of\_matrix,no\_of\_rows,No\_of\_Col))

# np.full((no\_of\_matrix,no\_of\_rows,No\_of\_Col),matrix\_num)

# **Ones** : - it create a metrix with combination of row & col. with value as 1.

# **zeros** : - it create a metrix with combination of row & col. with value as 0.

# **full** : - it create a metrix with combination of row & col. with user provided value.

x = np.ones((3,4,5)) #3 = no of matrics

#4 = no of rows

#5 = no of columns

X

array([[[1., 1., 1., 1., 1.],

[1., 1., 1., 1., 1.],

[1., 1., 1., 1., 1.],

[1., 1., 1., 1., 1.]],

[[1., 1., 1., 1., 1.],

[1., 1., 1., 1., 1.],

[1., 1., 1., 1., 1.],

[1., 1., 1., 1., 1.]],

[[1., 1., 1., 1., 1.],

[1., 1., 1., 1., 1.],

[1., 1., 1., 1., 1.],

[1., 1., 1., 1., 1.]]])

**# Ones :**

x = np.ones((4,5), dtype = int)

X

array([[1, 1, 1, 1, 1],

[1, 1, 1, 1, 1],

[1, 1, 1, 1, 1],

[1, 1, 1, 1, 1]])

# zeros

x = np.zeros((5,10))

x

array([[0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],

[0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],

[0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],

[0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],

[0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]])

**# full**

x = np.full((3,4,5),200) # 3 = no. of metrix

#4 = no of rows

#5 = no of columns

# 200 - user metrix number

X

array([[[200, 200, 200, 200, 200],

[200, 200, 200, 200, 200],

[200, 200, 200, 200, 200],

[200, 200, 200, 200, 200]],

[[200, 200, 200, 200, 200],

[200, 200, 200, 200, 200],

[200, 200, 200, 200, 200],

[200, 200, 200, 200, 200]],

[[200, 200, 200, 200, 200],

[200, 200, 200, 200, 200],

[200, 200, 200, 200, 200],

[200, 200, 200, 200, 200]]])

**# change data type**

x = np.full((4,5),200, dtype=str)

X

array([['2', '2', '2', '2', '2'],

['2', '2', '2', '2', '2'],

['2', '2', '2', '2', '2'],

['2', '2', '2', '2', '2']], dtype='<U1')

**# linspace : its distribute between the two number in same length by user**

x = np.linspace(1,10,20)

X

array([ 1. , 1.47368421, 1.94736842, 2.42105263, 2.89473684,

3.36842105, 3.84210526, 4.31578947, 4.78947368, 5.26315789,

5.73684211, 6.21052632, 6.68421053, 7.15789474, 7.63157895,

8.10526316, 8.57894737, 9.05263158, 9.52631579, 10. ])

x.size

20

x = np.linspace(1,2,10)

X

array([1. , 1.11111111, 1.22222222, 1.33333333, 1.44444444,

1.55555556, 1.66666667, 1.77777778, 1.88888889, 2. ])

**# Flattern** : It convert multiple dimention metrix in single dimentionn array .

x = np.array([[[1,2],[7,8],[9,6]]])

x = x.flatten()

print(x)

[1 2 7 8 9 6]

x = [[1,2,3],[2,3,4],[9,8,6]]

print(x)

[[1, 2, 3], [2, 3, 4], [9, 8, 6]]

y = np.array(x)

z = list(y.flatten())

print(z)

[1, 2, 3, 2, 3, 4, 9, 8, 6]

**# joins : its combine the two or more than two array**

**# concatenate - its combine the two or more than two array**

x = np.arange(1,10)

y = np.linspace(20,30,10)

X

array([1, 2, 3, 4, 5, 6, 7, 8, 9])

x.size

9

Y

array([20. , 21.11111111, 22.22222222, 23.33333333, 24.44444444,

25.55555556, 26.66666667, 27.77777778, 28.88888889, 30. ])

y.size

10

x = np.array([[45,56,89]])

y = np.array([[78,89,56]])

x,y

(array([[45, 56, 89]]), array([[78, 89, 56]]))

np.**concatenate**((x,y))

array([[45, 56, 89],

[78, 89, 56]])

np.**concatenate**((x,y),**axis** = 1)

array([[45, 56, 89, 78, 89, 56]])

np.**concatenate**((x,y),axis = 0)

array([[45, 56, 89],

[78, 89, 56]])

**# transpose : it convert the metrics from row to col. and col to row**

x = np.array([[1,2,3,4,5],[6,7,8,9,10]])

X

array([[ 1, 2, 3, 4, 5],

[ 6, 7, 8, 9, 10]])

x.**shape**

(2, 5)

x = x.**transpose**()

X

array([[ 1, 6],

[ 2, 7],

[ 3, 8],

[ 4, 9],

[ 5, 10]])

**16 March 2024**

# mean :- It show the average value of the array

# median :- its show the mid value of the array.

# std : - Its show the standard deviation of the array

# var :- it show the variance of the array

# sqrt :- its show the base value of any number

# axis = provide sum of rows & column

# diag :- its extract diagonal value from the metrics

# diagflat :

# argmax : - it show the index of the maxium value

# argmin : - it show the index of the min value

# max : - it show the maximum value

# min : it show the minimum value

# cumsum - it will provide running total of value

# fliplr() : - it reverse the value of each array

**# find out avg. Through loop.**

x = [23,45,56,64,45,67,65]

l = len(x)

v = 0

for i in x():

v+=1

avg = v/l

print(avg)

# **Mean** : It show the average value of the array.

# Find out avg through mean function

**import** numpy as np

x = np.array([12,45,78,89,56,23])

avg = np.**mean**(x)

print(avg)

50.5

# **Median** : its show the mid value of the array.

import numpy as np

x = np.array([12,45,78,89,56,23])

mid = np.**median**(x)

mid

50.5

**# mean & median with different value**

x = np.array([4,7,5,10,25,200])

mid = np.**median**(x)

mid

8.5

avg = np.**mean**(x)

Avg

41.833333333333336

# **std** :- Its show the standard deviation of the array

np.**std**(x)

71.08074438427204

# **var** :- it show the variance of the array

np.**var**(x)

5052.472222222222

**# create a numpy array with the help of user input**

x = []

l = int(input("please enter the length of array :"))

for i in range(l):

a = int(input("enter values :-"))

x.append(a)

ar = np.array(x)

print(ar)

please enter the length of array :2

enter values :-45

enter values :-78

[45 78]

**# linspace() : it give user defined numbers between two provided number**

x = np.linspace(10,20,5) # 2 no. are 10 & 20 . 5 is how many numbers we want between 10 to 20.

print(x)

[10. 12.5 15. 17.5 20. ]

# np.random.randint - its give random number between any range

x = np.random.randint(45,90,10)

X

array([84, 59, 85, 74, 80, 87, 62, 80, 64, 70])

**# # sqrt :- its show the base value of any number**

x = np.array([49,169,144,100])

np.sqrt(x)

array([ 7., 13., 12., 10.])

x = 7\*\*2

X

49

**# create matrix of 5 x 6 with the help of arrange and reshape function**

x = np.arange(1,31)

y = x.reshape(5,6)

Y

array([[ 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]])

x = np.full((5,6),10)

X

array([[10, 10, 10, 10, 10, 10],

[10, 10, 10, 10, 10, 10],

[10, 10, 10, 10, 10, 10],

[10, 10, 10, 10, 10, 10],

[10, 10, 10, 10, 10, 10]])

x = np.random.randint(1,20,20)

x = x.reshape(5,4)

X

array([[ 1, 17, 3, 17],

[18, 17, 15, 7],

[18, 6, 4, 7],

[10, 2, 3, 10],

[12, 9, 13, 10]])

x.sum()

199

np.sum(x)

199

# **axis** = provide sum of rows

x = np.random.randint(1,20,20)

x = x.reshape(5,4)

print(x)

np.sum(x,**axis = 1**)

[[ 2 15 13 8]

[ 9 6 11 15]

[ 9 2 5 10]

[14 5 7 13]

[ 7 10 15 14]]

Out[112]:

array([38, 41, 26, 39, 46])

np.sum(x,**axis = 0**) # it add value by column

array([41, 38, 51, 60])

# Diag :

np.diag(x) # its convert the diagonal value from the metrix

array([ 2, 6, 5, 13])

# diagflat -

x = np.array([[12,45,78],[86,56,23],[25,36,14]])

X

array([[12, 45, 78],

[86, 56, 23],

[25, 36, 14]])

x.size

9

y = np.diagflat(x)

Y

array([[12, 0, 0, 0, 0, 0, 0, 0, 0],

[ 0, 45, 0, 0, 0, 0, 0, 0, 0],

[ 0, 0, 78, 0, 0, 0, 0, 0, 0],

[ 0, 0, 0, 86, 0, 0, 0, 0, 0],

[ 0, 0, 0, 0, 56, 0, 0, 0, 0],

[ 0, 0, 0, 0, 0, 23, 0, 0, 0],

[ 0, 0, 0, 0, 0, 0, 25, 0, 0],

[ 0, 0, 0, 0, 0, 0, 0, 36, 0],

[ 0, 0, 0, 0, 0, 0, 0, 0, 14]])

y.shape

(9, 9)

# **cumsum - it will provide running total of value**

# 1

# 1+2

# 1+2+3

# 1+2+3+4

# 1+2+3+4+5

# 1+2+3+4+5+6

x = np.array([1,2,3,4,5,6])

np.cumsum(x)

array([ 1, 3, 6, 10, 15, 21])

**# Comperision Operator**

x = np.array ([45,78,89,56,23,20])

x [x>50] # compare & extracting the value

array([78, 89, 56])

x [x<50]

array([45, 23, 20])

**# fliplr() : - it reverse the value of each array**

x = np.array([[12,45,78],[12,15,42]])

np.fliplr(x)

array([[78, 45, 12],

[42, 15, 12]])

**17 March 2024**

**Pandas**

1. what is pandas?

* Pandas is python library use for working with data sets.
* It has function for analyze cleaning, exploring and manipulating the data.
* pandas instruction
* pandas is a python library used for working with data sets.
* it has fuctions for analyzing, cleaning,cleaning exploring,and manuplating data.
* the name pandas has a reference to both "panel data", and "python data analysis"and it was created by wes Mckinney in 2008.

1. why use pandas?

* pandas allow us to anayze big data and make conclusion based on statistical theoies.

#### pandas series?

1. what is a series?

* A pandas series like a column in a table.
* it is a one -dimentional array holding off any type .
* series is a one dimentional labled array capable of holding any data type

**# How to install Pandas in Python**

!pip install pandas

import pandas

**# creating a Series**

import pandas as pd

x = pd.**Series**([12,45,78,89,56,23,56,89,45])

print(x)

print(len(x))

print(type(x))

print(x.shape) # shape - show the column & ros

print(x.ndim) # ndim - no. of demension

0 12

1 45

2 78

3 89

4 56

5 23

6 56

7 89

8 45

dtype: int64

9

<class 'pandas.core.series.Series'>

(9,)

1

# string data type refer as object under Pandas

x = pd.Series(["Nitin","Jatin","Prince"], index=[1,2,3])

X

1 Nitin

2 Jatin

3 Prince

dtype: object

x[1]

Nitin

x[2]

Jatin

# **Pandas DataFrame[¶](http://localhost:8888/notebooks/Python all class.ipynb" \l "Pandas-Data-Frame)**

* A Data Frame is a 2 dimentional labled data structure, like a 2 dimention array
* or pandas with Row & Column

**# Create table through Dictionary**

x = {

"Name":["Nitin","Jatin","Prince","Ravi","Kunal"],

"Age":[35,26,30,32,34],

"City":["Ajmer","Delhi","Jaipur","Kota","Jodhpur"],

"Pincode" :np.random.randint(5000,8000,5)

}

x

{'Name': ['Nitin', 'Jatin', 'Prince', 'Ravi', 'Kunal'],

'Age': [35, 26, 30, 32, 34],

'City': ['Ajmer', 'Delhi', 'Jaipur', 'Kota', 'Jodhpur'],

'Pincode': array([5342, 7921, 7805, 6105, 7896])}

**# Create table through Data Frame**

df = pd.DataFrame(x)

Df

**Name Age City Pincode**

0 Nitin 35 Ajmer 5342

1 Jatin 26 Delhi 7921

2 Prince 30 Jaipur 7805

3 Ravi 32 Kota 6105

4 Kunal 34 Jodhpur 7896

print(f"Total number of rows is: {df.shape[0]} and total no of rows is:{df.shape[1]}")

Total number of rows is: 5 and total no of rows is:4

df.shape

(5, 4)

df.columns # To check column details.

Index(['Name', 'Age', 'City', 'Pincode'], dtype='object')

df.info() # To check complete information of table.

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 5 entries, 0 to 4

Data columns (total 4 columns):

# Column Non-Null Count Dtype

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

0 Name 5 non-null object

1 Age 5 non-null int64

2 City 5 non-null object

3 Pincode 5 non-null int32

dtypes: int32(1), int64(1), object(2)

memory usage: 272.0+ bytes

**# Create table direct under DataFrame**

df = pd.DataFrame({"Name":["A","B","C","D","E","F","G","H","I","J"],

"Sales1": np.random.randint(500,1000,10),

"Sales2" : np.random.randint(500,1000,10),

"Sales3" : np.random.randint(500,1000,10)},

index = [1,2,3,4,5,6,7,8,9,10]

)

print(df)

Name Sales1 Sales2 Sales3

1 A 775 583 577

2 B 655 687 833

3 C 851 757 841

4 D 882 825 998

5 E 891 556 556

6 F 736 851 770

7 G 854 814 674

8 H 911 707 550

9 I 564 698 602

10 J 860 726 625

df.info()

<class 'pandas.core.frame.DataFrame'>

Index: 10 entries, 1 to 10

Data columns (total 4 columns):

# Column Non-Null Count Dtype

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

0 Name **10 non-null** object

1 Sales1 **10 non-null** int32

2 Sales2 **10 non-null** int32

3 Sales3 **10 non-null** int32

dtypes: int32(3), object(1)

memory usage: 280.0+ bytes

df.columns

Index(['Name', 'Sales1', 'Sales2', 'Sales3'], dtype='object')

type(df)

pandas.core.frame.DataFrame

df.ndim

2

**# with None value (nan)**

df = pd.DataFrame({"Name":[np.nan,"B","C","D","E","F","G","H","I","J"],

"Sales1": np.random.randint(500,1000,10),

"Sales2" : np.random.randint(500,1000,10),

"Sales3" : np.random.randint(500,1000,10)},

index = [1,2,3,4,5,6,7,8,9,10]

)

print(df)

**Name Sales1 Sales2 Sales3**

1 NaN 947 637 692

2 B 512 921 770

3 C 660 855 766

4 D 998 919 954

5 E 978 879 795

6 F 746 924 719

7 G 714 628 796

8 H 745 912 907

9 I 808 948 722

10 J 911 721 794

**# duplicated() - To check the duplicate value from the table.**

df.duplicated().sum()

0

df.info()

<class 'pandas.core.frame.DataFrame'>

Index: 10 entries, 1 to 10

Data columns (total 4 columns):

# Column Non-Null Count Dtype

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

0 Name **9 non-null** object

1 Sales1 10 non-null int32

2 Sales2 10 non-null int32

3 Sales3 10 non-null int32

dtypes: int32(3), object(1)

memory usage: 280.0+ bytes

**# How to check Null value in datasets**

df.isnull().sum()

Name 1

Sales1 0

Sales2 0

Sales3 0

dtype: int64

df.isna().sum()

Name 1

Sales1 0

Sales2 0

Sales3 0

dtype: int64

Print(df)

|  | **Name** | **Sales1** | **Sales2** | **Sales3** |
| --- | --- | --- | --- | --- |
| **1** | NaN | 947 | 637 | 692 |
| **2** | B | 512 | 921 | 770 |
| **3** | C | 660 | 855 | 766 |
| **4** | D | 998 | 919 | 954 |
| **5** | E | 978 | 879 | 795 |
| **6** | F | 746 | 924 | 719 |
| **7** | G | 714 | 628 | 796 |
| **8** | H | 745 | 912 | 907 |
| **9** | I | 808 | 948 | 722 |
| **10** | J | 911 | 721 | 794 |

**# head() :- its show 1st five rows by default**

**# tail() :- its show last five rows by default**

df.head()

|  | Name | Sales1 | Sales2 | Sales3 |
| --- | --- | --- | --- | --- |
| 1 | NaN | 947 | 637 | 692 |
| 2 | B | 512 | 921 | 770 |
| 3 | C | 660 | 855 | 766 |
| 4 | D | 998 | 919 | 954 |
| 5 | E | 978 | 879 | 795 |

df.tail()

|  | **Name** | **Sales1** | **Sales2** | **Sales3** |
| --- | --- | --- | --- | --- |
| **6** | F | 746 | 924 | 719 |
| **7** | G | 714 | 628 | 796 |
| **8** | H | 745 | 912 | 907 |
| **9** | I | 808 | 948 | 722 |
| **10** | J | 911 | 721 | 794 |

31st March 2024

# **Groupby**

# sum : it is used to add the total number of values by the same cretaria

# Average or mean() : it show the mid values of numbers.

# min : it show the minnimum value from the dataset.

# max :L it show the maximum value from the dataset.

# describe : it show the all posible values of statistics

# count : It is used to count the total number of Rows in each category.

# std : - Standard deviation : it is square root of varience

# var : - varience : it show the spread of the dataset

**df.head()**

