x = (Hello, '4', 5') -> Hello not det 3 chord automobile

vice automobile proposition of personal automobile automobile

vice automobile proposition of personal automobile

vice automobile proposition of personal automobile

vice automobile proposition of personal automobile automobile

vice automobile proposition of personal automobile automobile automobile automobile

vice automobile proposition of personal automobile automobil Set: unordered collection of items. @ doesnot have duplicate values! colors = g'red', blue, blace 4 Characteristics of collection like lists, tuples storing mixed data sets are a type of collection like lists, tuples -) enclosed within curly brackets with comma seperated -) sets are unordered Does not allow duplicates. -> if I want to count the unique Dedaring a Set: elements then it is going to be Sett = 21, 2, 3, 44 converting the list into set. Adding element to set. Ex: A = [1, 1, 2, 4, 3, 3, 2] set1. add ("India") A = Set(A) 10 leulo prior (A) Set1= {1, 2,3,4, "India"} Print(A) = {1,2,3,44 To remove you simply do Sett remove ("India")

Set operations trib a de must de 18 pristesses &
A = 50, 2, 4, 6,02
81,2,3,4,53
Onion operation: OR operation is called Union operation (1) ntersection operation: AND operation is called Intersection operation.
ntersection operation: AND operation is called intersection operation.
CALB) on (A. union(B))
Inion: can be done by CAIB) on (A. union(B))
ntersection: Drint (A & B) (or) (A. intersection (B))
or of (A: Not terence (D))
Difference . Prince of the common elements.
Symmetric difference: Removes all the common elements. Symmetric difference: Removes all the common elements. Print (A^B) ichonary: (2northere 1/2 collection of Data.
Print a Mary Grant a Mary Grant and All and Al
chonary: unordered collection of Data.
ictionary: unordered collection of Data. Data in dictionary is stored as a key: value pair key should not be mutable and value can be of any type.
> key should not be murable and value be vertically the
Dict = {"name": "Joho", 'age': 109
Key is till an make,
> values are objects that contain throwns on some objects that contain their keys. > values are accessed using their keys. > values are accessed using their keys. > can key is followed by a value seperated by colon > Each key is followed by a value seperated by colon > values can be immutable, mutable & duplicates > values can be immutable, mutable & duplicates
I carb law is followed by a value seperated by colon
a univer can be immutable, mutable & duplicates
Declaring a Dictionary:
Declaring a Dictionary: "India": [NR, "USA": USD, "Hong Kong": "HED"}
D Accessing volume
di ["Inda"]
A LINE OF THE PARTY OF THE PART
Deplacing value for a key in dictionary
CHANGE CHANGE TO THE STATE OF T

d1 = {"India": 'INE', "USA": 'USD', "Wong kong": "HK\$"}

1 Inserting at New Key-value pair granoities o 1000 stoots (8) difi Japani J = VEN " "Heard"] : ["preek!" " NAY = ptob - toubute 191")

dia (1): Shident data: → d1= {"India": INP, "USA": USD, "Hong Kong": HK\$, Japan: YEN'Y Deleting a key value pair. and de di ["Japan"] wellow of the sound of the & Sorbing a dictionary moison comes our The stagent gates & L. L. L. 1. 15. 5: L. by Shi 18.3 d. Sorted(d1) =) ['Hong rong', 'mdia', 'USA'] : Dansti shob trobute at 1 rol (value method di values () => ['HR', 'USD', 'HK\$'] d1: Keys () => ['India', 'USA', 'Hongkong'] @ get() method -> Returns the value => dl.get('USA') @ update() method -> d1. update (& "India" & Rupee" }) d1["India"] = "Rs" and file handling Python Loops, Functions iterable 9919 Loops: Used for repetition- iterating over a > iterable: object that can be iterated over. Eg: Lists, Tuples, Dictionaries, Strings, Sels. > iterator: variable that goes through each element in the iterable -> 2 Types of Loops: For loops & while Loops. iterating through a list biterating through String String"Krishna" cisti = [1, 24, 6]

for i in cisti:

print(i)

olp: K

classification

ological print(i) olp: 1(1)aus

```
1 iterate over a dichonary. The and the charge of Charges and sent to
IP: Student - data = {1: ["KK", 25], 2: ["pree Hi", 23] }
     for i in student-data:
    Olo: Print (i) | 111 : " paod profil" , 020 : "A20" (0/1) : "Bibril" } : 15 &
                                     1 Deleting was least value pair
 I But to print the records or the values in a dictionary, we can
                                                 (Keys, values)
    use items() function.
 11P: Student_data = & 1: ["KK", 25], 2:[" preetti, 23]}
                                           - Hora Rorg " Hidia, "USA
       for i in Student_data. items():
              Print(1)
                                                     badlan (seulov (9)
     olp: (1, ["kk', 25])
          (2, ['preethi',23])
  > if i just need, values i can simply use a new iteratorifiand

Print it.
 IP: for i, i in student data items ():
            print (j)
     olp: [' et', 25]
                           Pethon Leaps I twickers
          ['preethi', 23]
                           Loops I used More repetitions iterations (ava a
( Comprehensions
      List 1 = ["KK", "Kanth", "Mark", "Preethi"]

List 2 = [Lenci) for i in list]

As same as

List 2 = [ J

for i in list]
   OlP => [2,5,4,7]
                         squal stiller is agood to last 2. append ( lenci))
  =) create a dickonary.
        di = {i: len(i) for in list 3 / splig de le la devoit
       OP = { 22:2, 'kanth':5, 'Mark!:4, 'Preethi!:73
                                     -> As same as Hall all 190
                ping (1) tring
                                     or in lister
                                               di[i] = lenci)
```

> Named sequence of statements that performs some operations. > User defined functions are created using a keyword called get. def func_1 (name, age) def - function name (arg 1, arg 2, func-1 ("KK", 25) Obj = arg1 + arg 2 (2) default valu return obj def fuc'l (name, a=30) function name (val1, val2) -> calling it. func_1 (1KK1) -> lambda function: Small anonymous function to make developers life → can take any no of arguments, but only one expression. Syntax: lambda argument: Expression Example X = lambda a: a+10 Example ! print even (or) odd x = lambda a : even if a 1.2=0 print (x(5)) else 1 odd' olb: 12 Print (x (100)) Example: Adding two numbers c = lambda a,b: a+b Print (c(4,5)) tripe and the tripe of the property of the state of the s OIP: 9