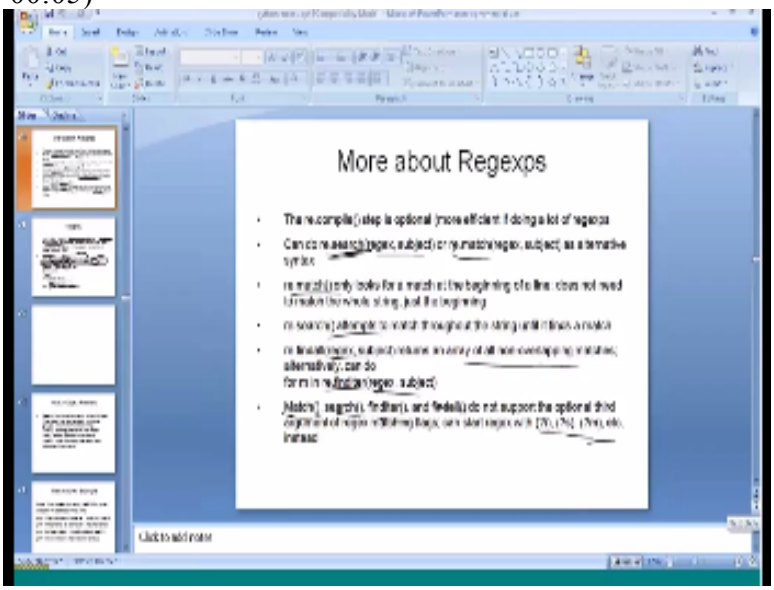
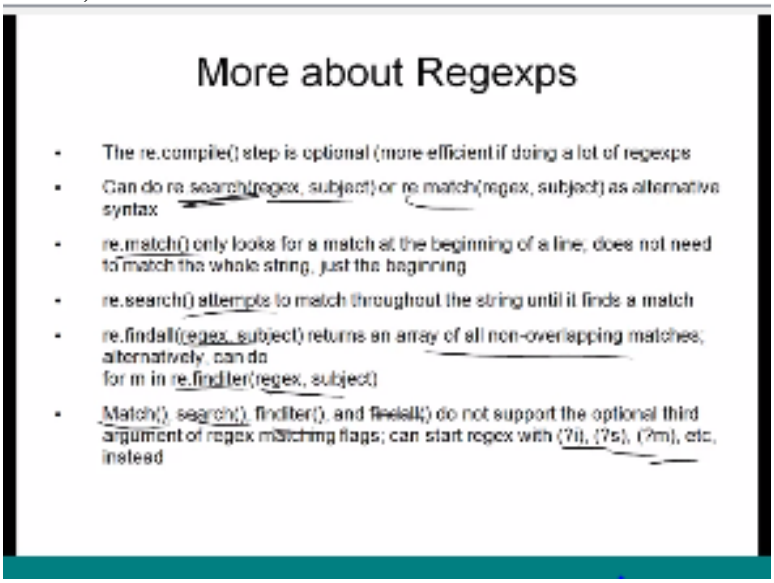
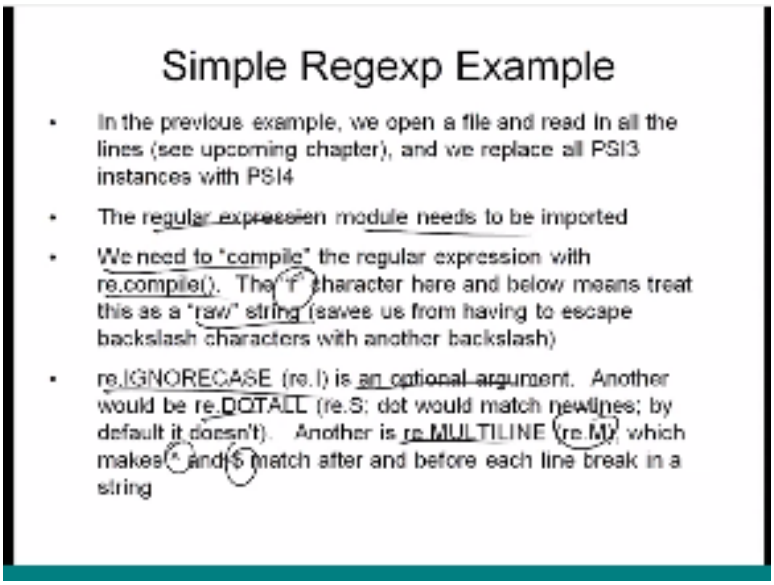
(Refer Slide Time: 00:05)



(Refer Slide Time: 00:10)

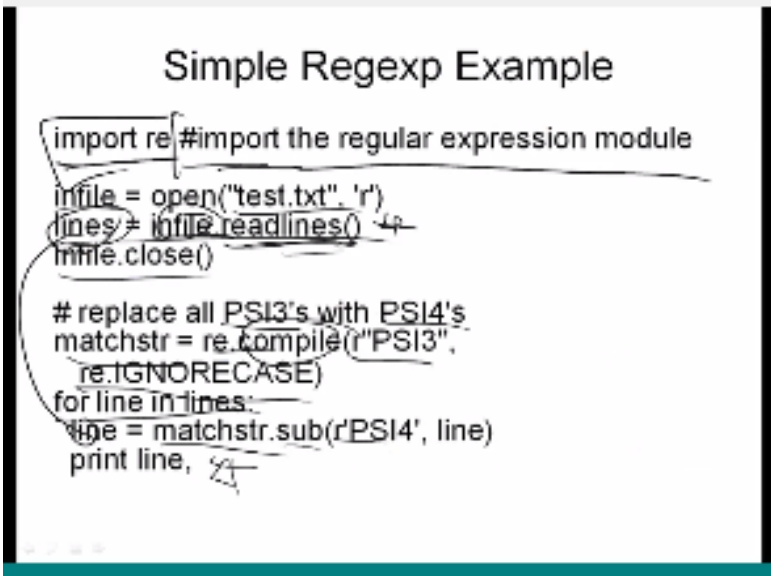


(Refer Slide Time: 00:16)



Hi everyone again welcome to the LPS class we are continuing our discussion on Python in the last lecture we saw like the regular expression pre much like we covered how to do the similar expression.

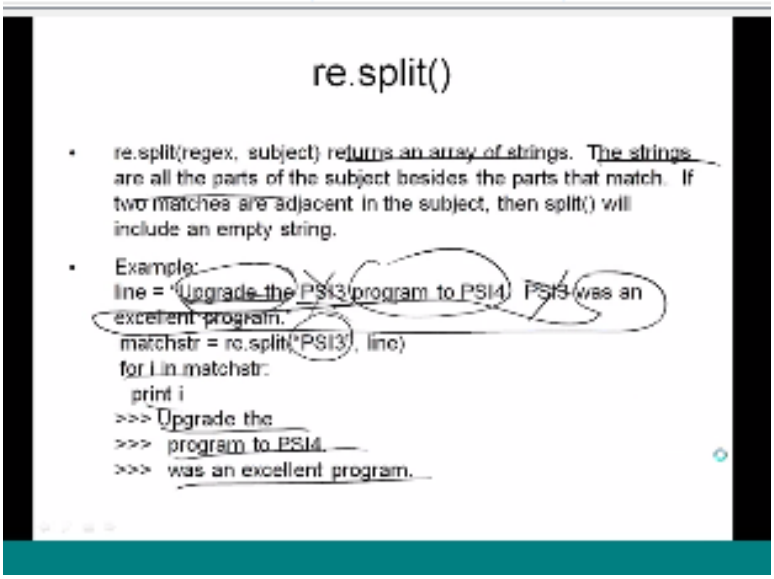
(Refer Slide Time: 00:51)



We saw this particular thing basically where we kind of match the string using the R which is the raw string basically like and then. Once we have that they won the match string then, we have substituted psi3 psi4. And then we explained this concept of this basically. So we went to this section this section we also talked about some of the additional stuff. So we can use basic we can do the search regards X with the subject or match the gigs with the subject as alternative syntax. And the match basically it only looks for the match at the beginning of the line it does not match the for string just the search item to match throughout. The string until it is fine for public match

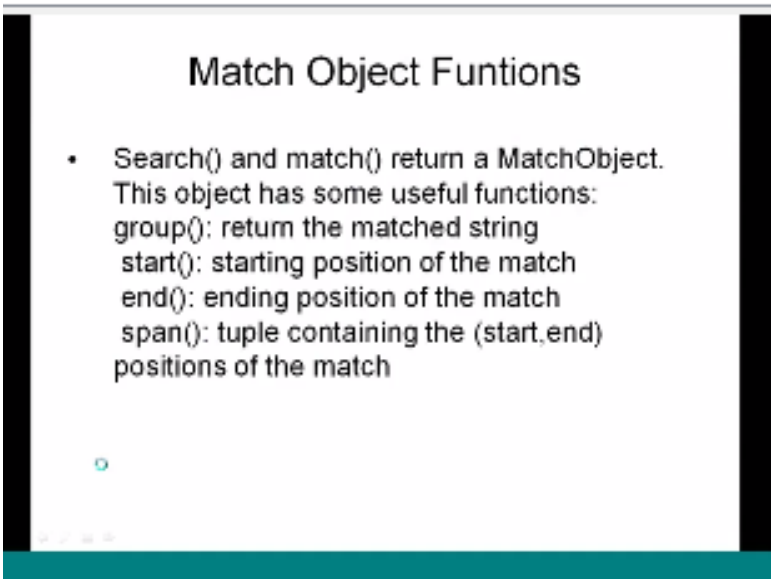
so that the difference between the match and the search. We also have this find all that is another method inside the class. And that actually returns an array of all the non overlapping matches we can also do like using a for loop here basically which is or M in more define better method. Makes A and then you can find each one and then not be image so basically until find all of them so this same at the kind of and then the matching itself is basically bottom the roof up like. We can start the regards with the person mark is smoke one injury.

(Refer Slide Time: 02:53)



So this is something that we saw and then we also saw the split the city which is another method in the regular expression class. And here when we do the split essentially sticks on that particular string so it basically if the replaces the psi3 with si4. So again in this one basically it just clicks and then we remove so your output is going to be just update the program to PS f4 was an excellent pub that is all. So today we will continue some more so the discussions on the regular expression finish off the regular expression and then move on to the next chapter.

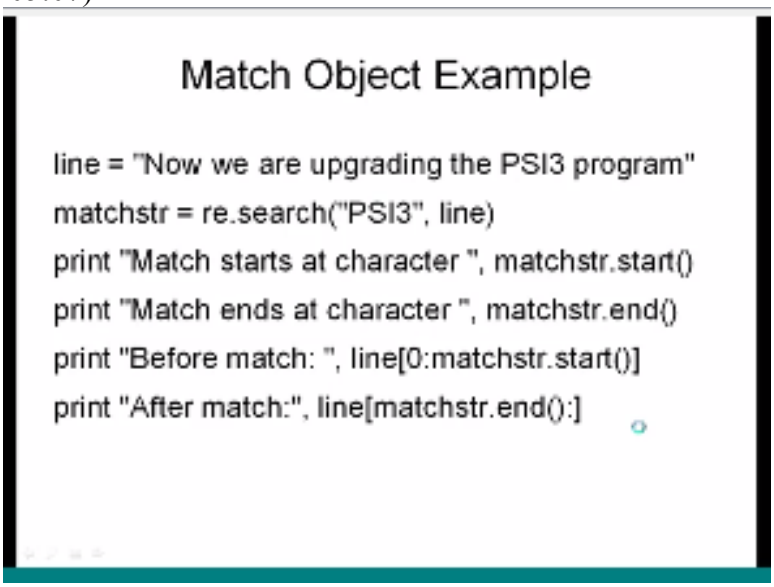
(Refer Slide Time: 03:44)



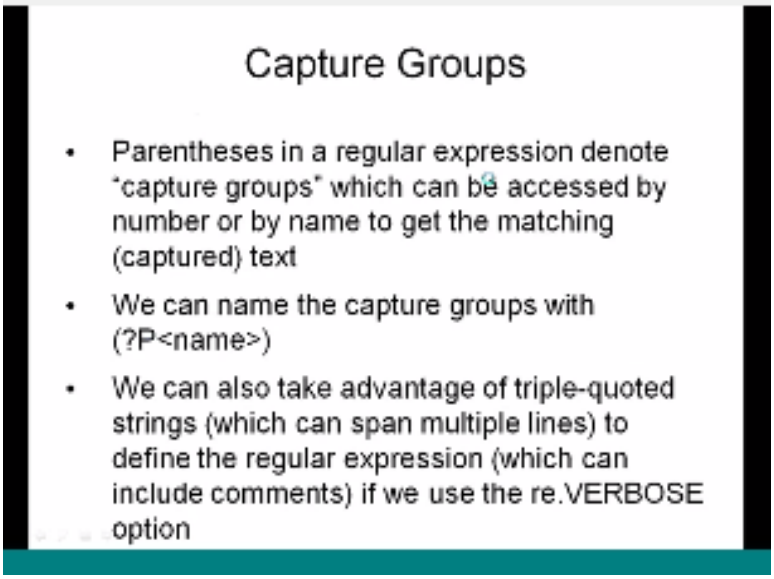
So basically the match object function essentially like, so search and match return. A Match object the object has some useful functions basically in the group that is to return the matched string and then start. Basically that is the starting position of match and then end is the ending position match and then we can also like the proof. And which is basically the tuple containing the start in positions of the match.

So just to understand it more thoroughly like them in another Python offers two different primitive operations based. On the match text for the matter only the beginning that is A and L the search basically searches for porting and both of them returns a match object. So here are some math object examples.

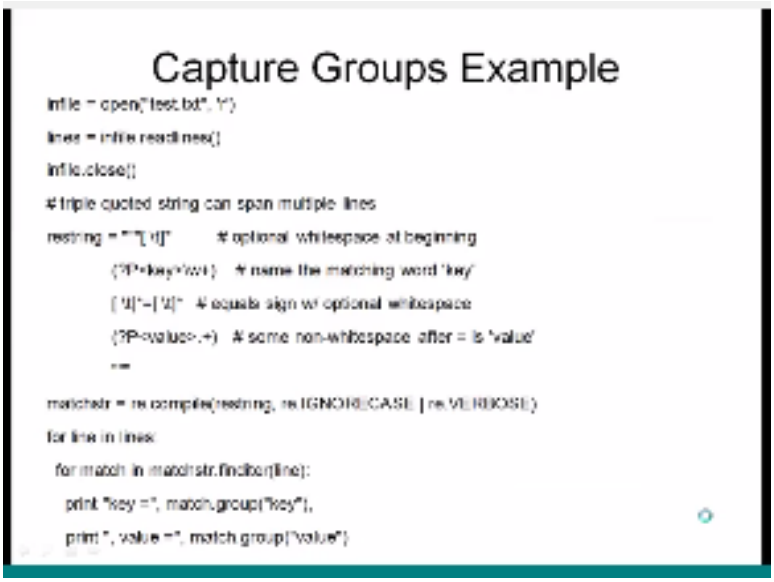
(Refer Slide Time: 05:07)



So we declare or we define one of the variables lines of the string. And then we say Mack string search for psi 3 in line. And then so basically, the match starts at character we can say match string start master this particular variable start dot start and that gives basically like a minimum where it starts the match. This is the basic three and then where it ends and then before the match before match and match this agrees that is with you can specify the index on the start.
And then all the way up to the end we now start at so this will give you like on this section so the first one will check cell b3 and then. Essentially like them from 3 next all the way ups the end and then here it will start from your M. So you so now let us talk about the capture group.
(Refer Slide Time: 06:24)



So this is useful to actually apply certain after obtaining a match take that match and then use it in then as a next variable to do some processing. So we did this for both pearl and table so the parentheses inside the regular expression denote a capsule group which can be accessed by the number or by name to get the matching attention of the Quiche text. So we can name the catch-up groups with this kind of syntax. There we can custom question mark P in the name. And then we can also take advantage of the triple quoted strings.
Which are multiple lines you remember we talked about that to define the regular expression and that can include comments as well? And if you use this particular method which is verbose method that option and you can use that so let us look at some examples here.
(Refer Slide Time: 07:48)



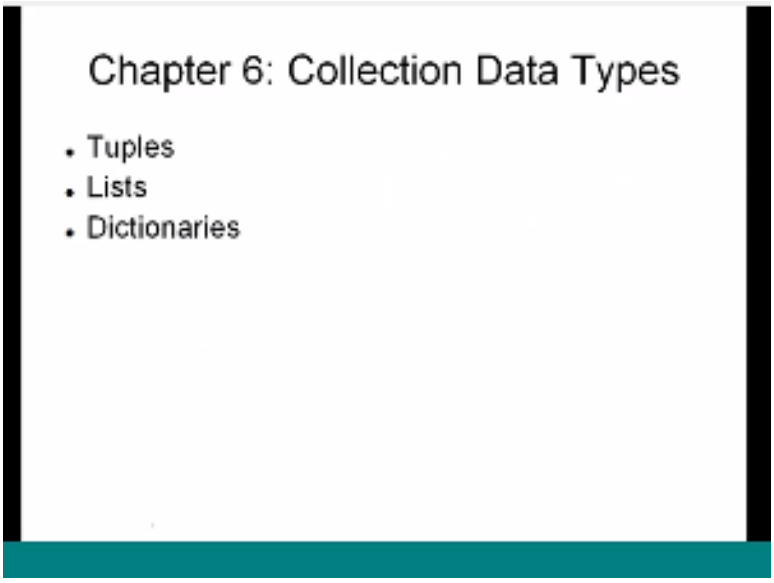
So again we define an in file then open the test for text with a read only and then we define all the lines. Basically like when we read the lines form and this is the method for the reading the lines or the inside and then we close that particular file. Now we use the triple quoted string essentially ligament so the one movie screen. So we start with the triple quoted string basically like some optional whitespace in the beginning and then we say basically. So okay now facility key and then followed by any other thing.

So any word that begins with T we match that here and then we again have like equal sign with some optional white spaces. So there are very stomach sign there and then finally like I am in the non space after the equals two is the best on this thing so we match the key and we match the value and they are stored in the key and then the band so here as we saw here basically the pin name. So assume that you have like a XY = 3, B = 4, C = 5.Something like that in this particular test or text file.

So now we have the thing before the equal to is a II and then in after the equal to is the value so this becomes an array .That you can store as a key value pair then we do the restring ignore case essentially like I mean then pipe it to the burbles this way and then that for their campaign and then we do like the four line in the lines. So prints key equal to the match group key and then value is smash book value so this again prints basically XY and then 3. And then it continued from b4c by so that is what we want to get.

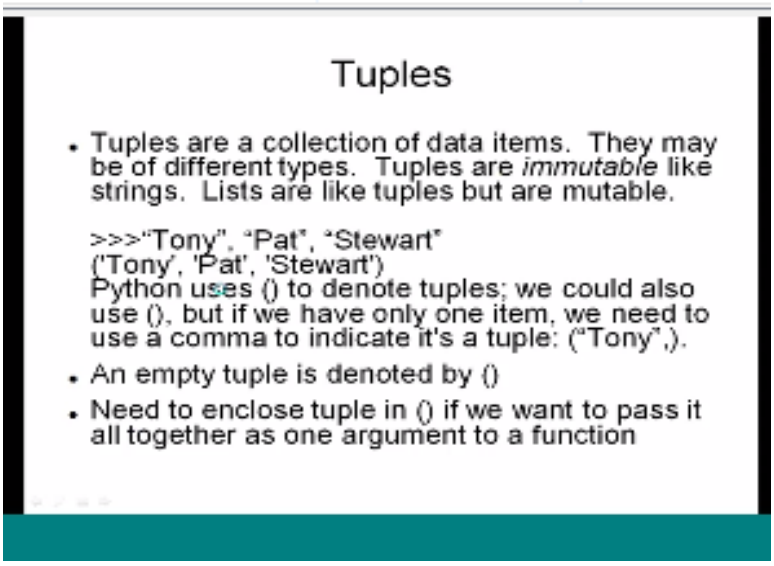
So in other words actually like now peaceful for as an ugly basically you can think of this as restraining x y = 3. This is what we install is not it so now let us move on to the next topic with collection data types.

(Refer Slide Time: 11:30)



So here we will talk about tuples the lists and dictionaries decision so a tuples our collections of collection of data items.

(Refer Slide Time: 12:18)

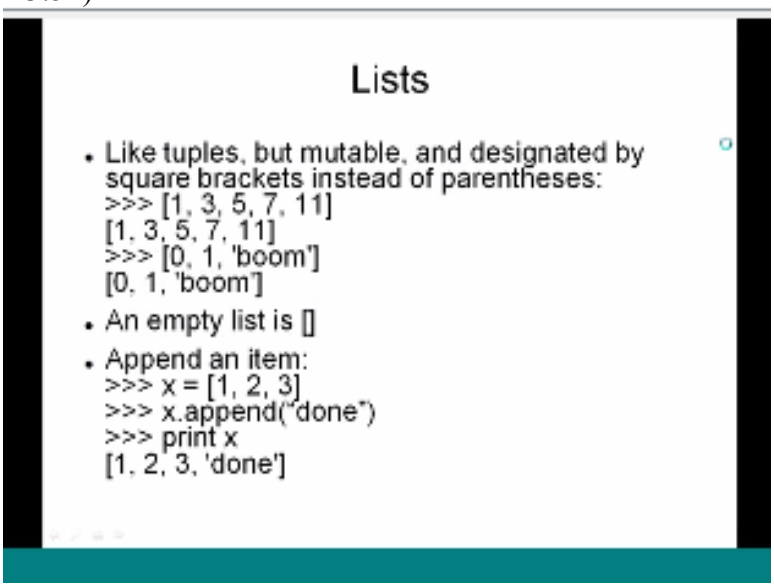


They may be of different types and usually tuples are immutable just like strings. That we saw busy like a number the lips are like tuples but they are mutable. So there is a difference between a distended tuple. So Python uses these parentheses to denote a tuple you could also use the parentheses. If you have this one you know only one item but if you have one I am sorry if you have only one item then we need to use a , to indicated to put.

So here, we have like three names so we just put , and then put it and put it into letters if you have just one name we just put the name then followed the , and then put it inside the parentheses and then the empty tuple is denoted by just empty parentheses we need to enclose a tuple in a

parenthesis if you want to pass it all together as one argument to a function so this is you can think of in typical. We had this basically like notation there we pass the interesting as this one object.

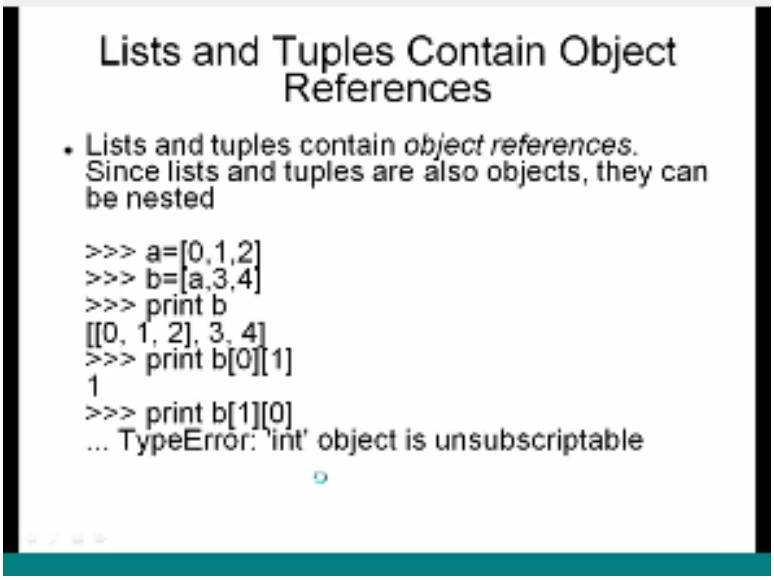
(Refer Slide Time: 13:51)



So some of the other things that you want to know about tuples are tuples are also like enterable meaning like you can trade one by one. And then you anything likes that that is pretty much it basically on the tuple now the list essentially list are like tuples. The only difference is that they are mutable that means that you can actually change that data structure. And they are denoted by [ ] Instead of the parenthesis.

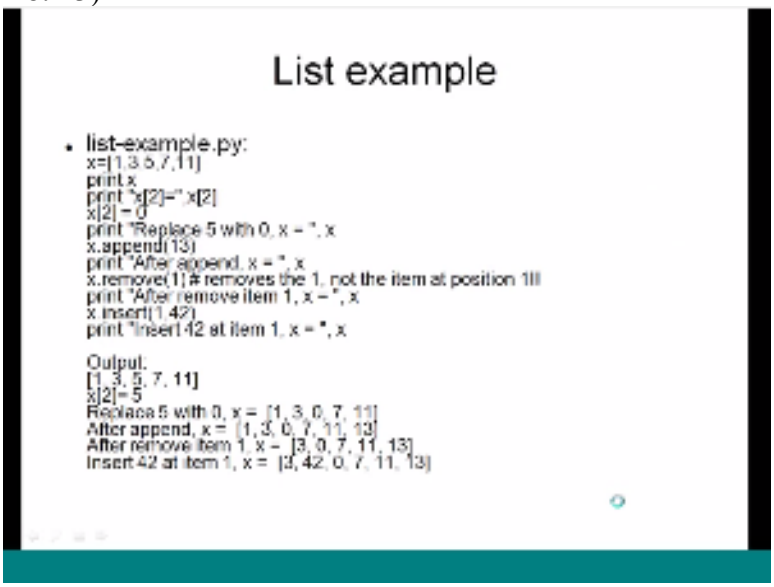
So here we have like 1, 3, 5,7,11 and that is the that is been ordered as a list and then you can also have different values between 0. One boom this also lists and empty list is just square brackets for tuck together. And so the mutable property described here we can add elements to the list or disagree you can append an item here. We define a particular list 1, 2, 3 and if you want to append done to it we use a method except and done. And then that prints out with that.

(Refer Slide Time: 16:31)



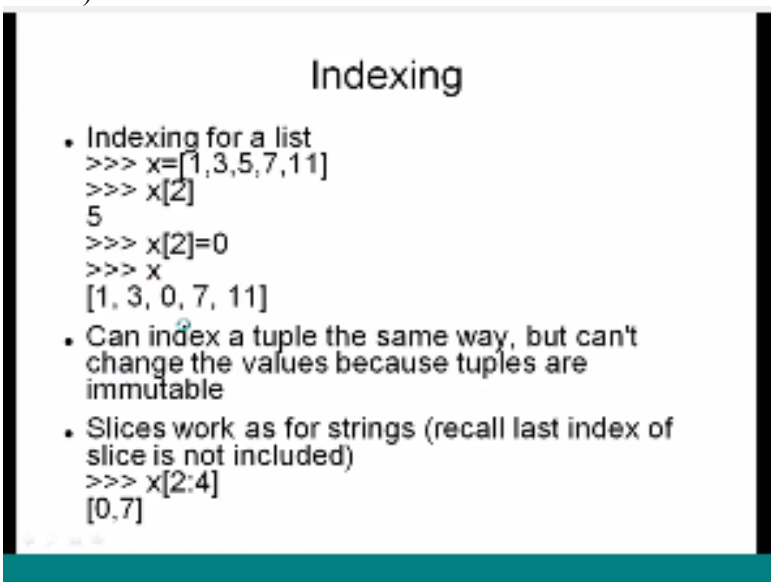
Both the lists and tuples contain object references, so the listing tables are also object they can be mixed it. So you can actually describe a tuple within a list and a list within a tuple and tuple within a tuple and this will be most. So here, we can say basically so on A = 0, 1,2 these A 3, 4 and when we print a it basically a nested list is getting printed. If you print the 0 1 it prints 1.So if you print b10 basically that results in a net. So the 0 one is the subject so be 0 if you just print B 0 that is 0 1 2 and B 1 is 3 so B 0 1. Exactly this subject is one that B 0 0 it is just you okay.

(Refer Slide Time: 18: 23)



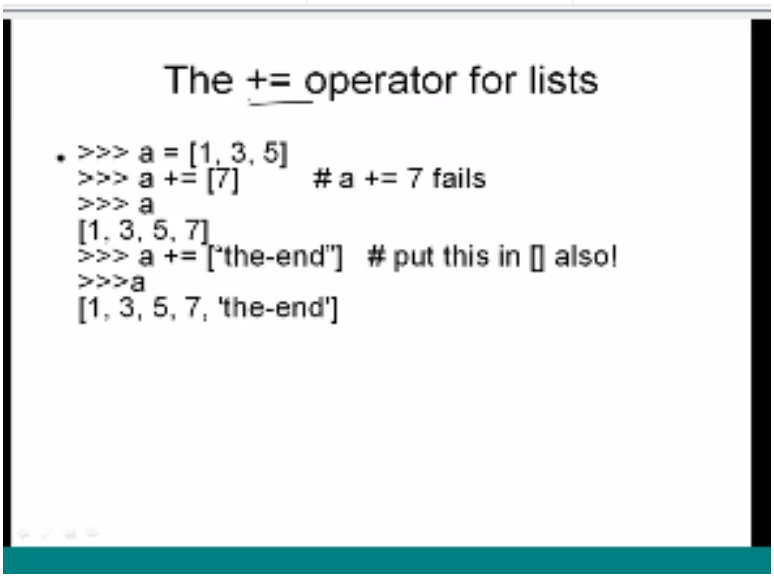
So some more examples of Lists so this is like a list example I we define one list which is just the odd number except or numbers except 9 and we print X. Now when we print X2 and anyone tell me like what is this one this is actually like now it is 5 X 2 is fine. Now we do this operation which is here doing an assignment X2 = 0 no what do you think the mill this is a replacement

happen it happens in lists because lives are mutable over then you can actually change them so we will replace 5 with 0 with the menu I send this one and then if you print out X if it prints like 1, 3, 0,7, 11 and then we can apply X append the method append to it with number 13 which just add 13 event when everything in [ ] okay.
And then if you remove one this removes one and it is not the item at position 1 it just removes one item. So the remove one basically maybe going to a stable the position one is actually number 3. So this is the first item this it is remote and then we after that we insert at the first position we insert number 42. So this now the insert actually takes this as the position thing so for that it starts from 0, 1, and 2. So the 1 position is going to insert for it.
So where overall list is going to be like 3, 42 then 0, 7 so the 0 is pushed and then it is inserted the I want you to remember this thing the remove one does not mean that the first position gets moved but the insert one means that the first insertion and the insertion happens at the first position okay.
(Refer Slide Time: 21:24)

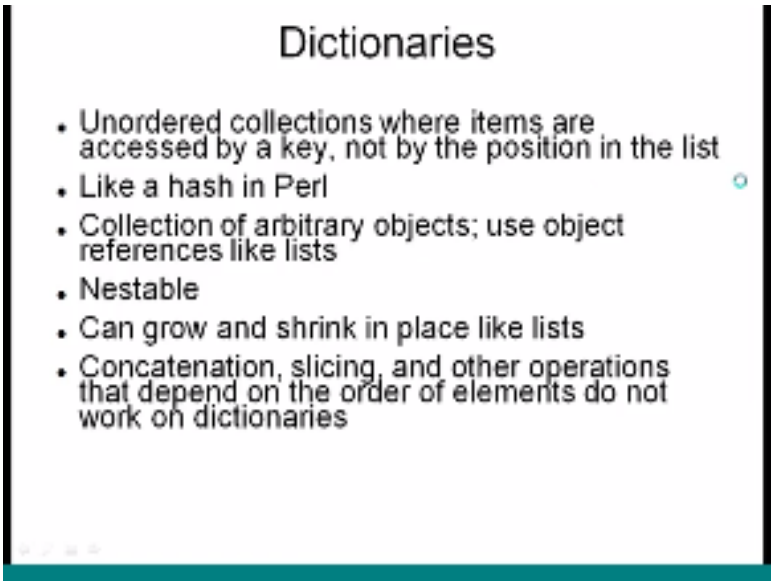


Now let us look at the some of the indexing so the indexing we know that actually it starts from 0, 1, 2, 3 and 4. So in this case if the X2 = 5 that is because it is a counting from Mr. Right it is 0 1and 2. And then when we do an assignment the X2 = 0 since it is immutable it basically. Now changes that and then so what the 5 is replaced with 0 ha so the difference between gain the list and the tuple that we saw is while the index indexing of a tipple also works exactly the same way we start from 0 1, 2, 3 or you cannot do this operation. Because, the tuple is an immutable data structure.

So you want to do that and then a slice it is another one slices workers for Strings because you later remember a string like I mean string. We can specify a slice of RA n which is like starting from two three four to Colin for it is the string those are still applied with lists so here we can specify like a menu to Colin for and then that is like 0, 1, 2, 4 so 2 and 3 will be printed out here it is PCP our aim is to combine.
(Refer Slide Time: 23:51)

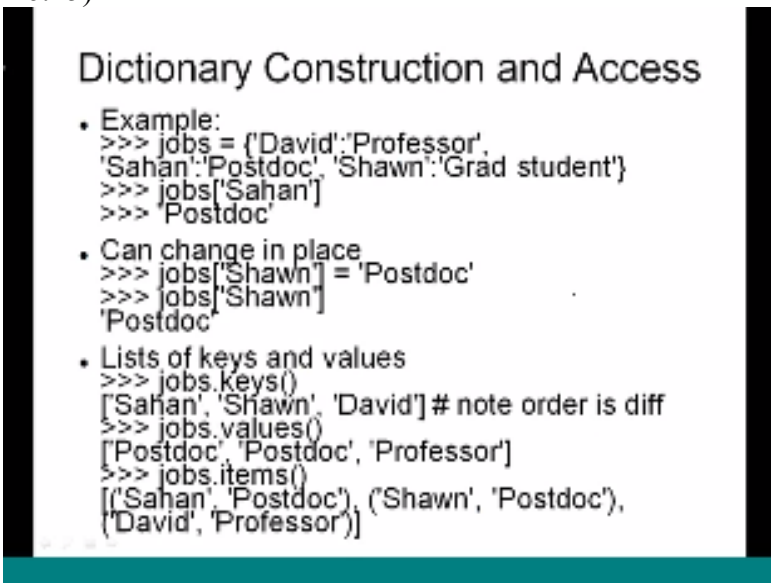


And now we cannot this additional operator which is basically the incremental apparently service. So here we have a list 1, 3, 5 so we can use the method A. upon this is one way to do it and then. We say 7 and then that add it at the end we can also do just simply a equal to plus and then the square brackets 7 and then this will add that element into that list if you try just A+ = this wiggle of 7 this is a fail failed case.
Because you need to specify the 7th inside both the list file for it to be added to the existing list so when we do this the result is 1 ,3,5 and 7 and then we can also add anything basically formulate the end add the end into the list. So again we use the same thing and then now we see printer A it is 1,3,5,7 then they soon the - M so this is another shortcut operator that you can use it instead of the append function.
(Refer Slide Time: 25:26)



Now let us look at the dictionaries, the dictionaries are nothing but items are it is an unordered collection where items are accessed by a key and not by the position in the list. So a dictionary is just like a hash in full the collections of arbitrary objects. And we can use object references like this and then dictionaries are also nest able we can grow and shrink in places like this and it allows concatenation slicing and other operations that depend on order of the elements. So they depend on the order of the elements do not work on this mobile for all these concatenation slicing in operations. Which depend on the order elements they do not work in dictionaries?

(Refer Slide Time: 26:45)



So let us look at some examples of so here in this example create like jobs basically David professor Sahan postdoc Shawn grad student so these are the three elements let me define inside the dictionary. And you notice that actually each of them is basically two ever followed by a

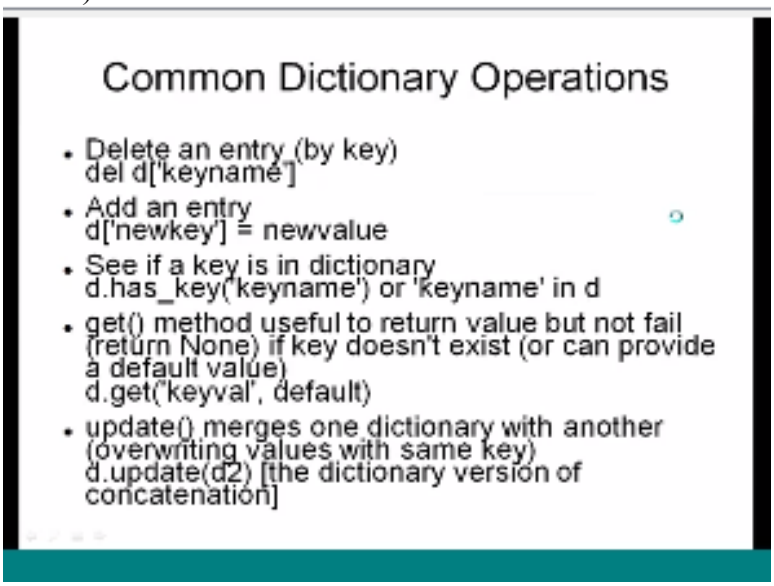
colon and then another keyword and then A , so anything before pollen is the easier key and then the other side is the value. So if you just say jobs Sahan mean if you go down and then this is also a mutable database. So essentially in the salic jobs Shan = to post up and then you can change the job or form Shan also to. And then if you want the list of ease and values we use this basically like the method piece.

So for jobs but is basically it prints the three keys and if you notice basically like the order actually it is changed. Originally our order was David Shahan and Shawn now the new order is a Shawn. On and then values generate again the corresponding values in the thing. So this is another method that you want to remember and then once you specify the values you get because postdoc out and purposes which has professors are professional for are the jobs of these people.

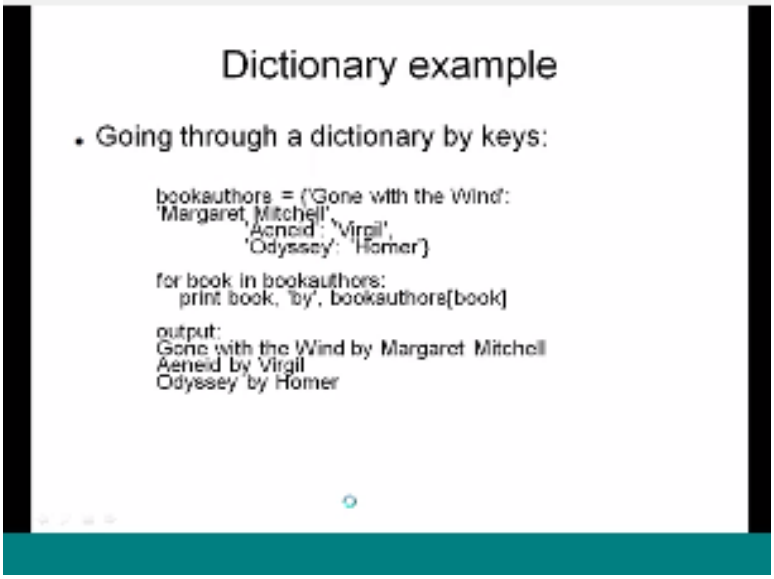
To query like a every element which is like a en the value together use the items function so the jobs are items function returns Shahan postdoc Shawn postdocs tuple it professor. So that is how you can access those values on in the individual. So the key thing that you want to remember is it starts with the curly braces ends with a curly brace that is the dictionary and then the items the each key is separated from value by this column.

And then the each items are separated by , is essentially so forth here and the whole thing is enclosed within the third basis. So the empty dictionary you can create it by just putting empty places so here only thing is even though. We said that basically like this is a mutable type of database the keys are of immutable data base in data type. So these are immutable they should be one of either string number or a triple these are the only data types allowed as. The key the value can be anything else so because the value itself is not immutable exactly immutable.

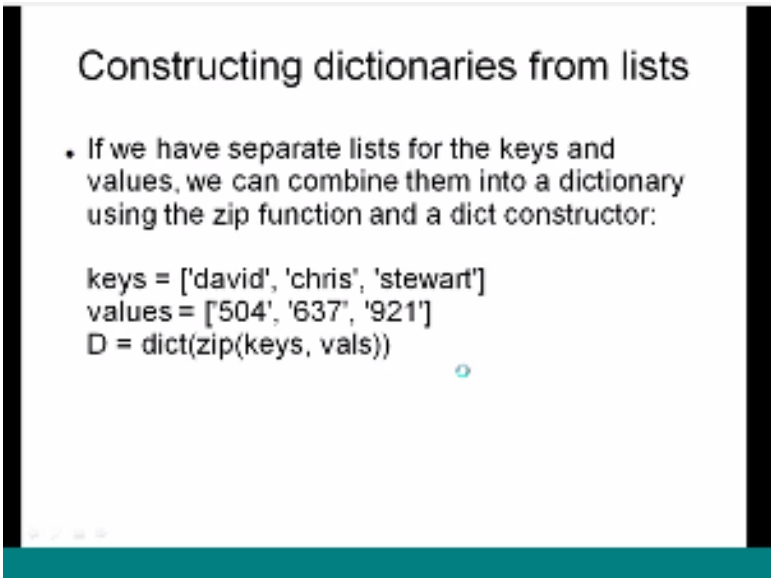
(Refer Slide Time: 31:30)



So now let us look at some of the common operations that we can do in a dictionary. So we can delete an entry by just using DEL followed by D and then the key name. Then if you want to add an entry then we push it in the new key is the devalue if you want to find out if particular key is in the dictionary and we can use the s key function DEL key name or a name in D this will this is another thing and see the gate is a method that is useful to beacon value. But we know it does not turn fail this way.
So it returns none if the key does not exist so you can use the get to get the one but if it would not fail this even exists. So that is only target so the way to use it is we get then the D value then default. Update merges one dictionary with another and then this overrides the values with the same key so the usages like the update d2 and then we give the diction version around definition.
(Refer Slide Time: 33:53)



So now if you want to iterate the dictionary through the keys we can use for commands. So here, actually like I mean we can specify like gone with the wind and Margaret Mitchell we need the Virgil and Odyssey by Homer and then we do the book in book authors. The key is actually you are going to be the book and then we basically got the book basically. So it gets each one and then it add the by and then fills the whole thing so wherever like I mean when you specify variable that always iterates on the T so as you can see like a key is the most important item in dictionary.
(Refer Slide Time: 35:16)

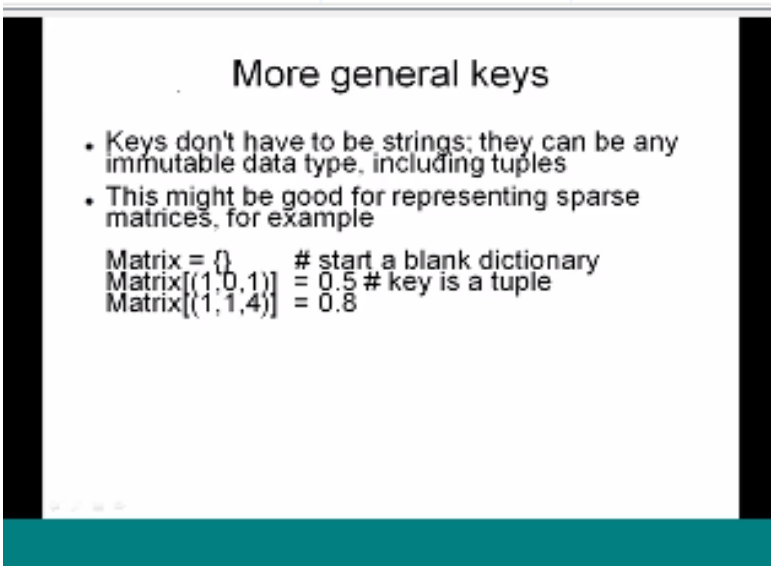


So the properties for the dictionary Chiefs essentially the value centers have no restrictions they can be any arbitrary Python of it either standard objects or use of different objects. But there are two important things that you want to remember one is more than one entry or key is allowed. So if you have so when duplicate keys encounter during an assignment the last assignment always win. So the simple example will be defining a dictionary it equals again the three things that you want to define in inclusion in spelling braces.

And then put the key say name and then put the : kitchen and then voila and then , act so this forms one element. And then H I mean pollen and then sale and then another , finally again we call it name Collin none this was not lesson we need to close the parentheses. So now in this particular dictionary if you want to print it [ ] name and this will always be many these are name is done only with its mean because that is what it is last updated values.

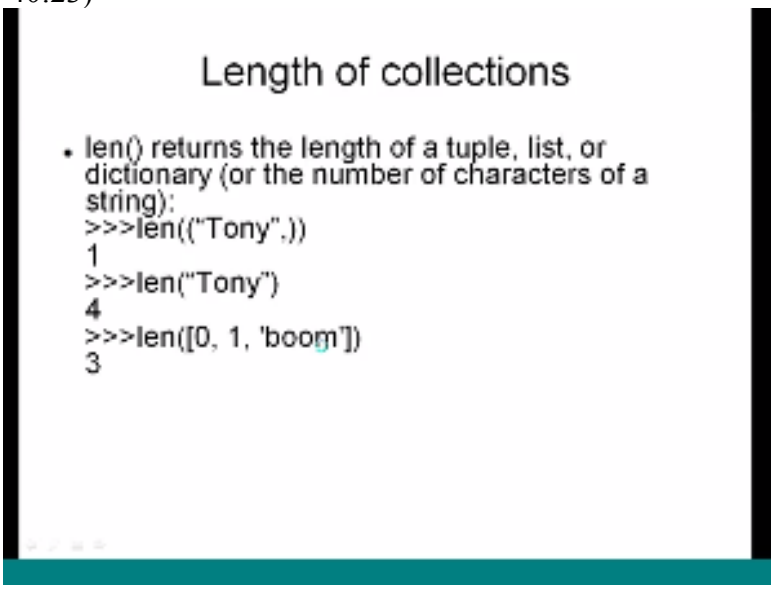
So if it cannot be hashed it just overrides it over X the previous value or with the new value now constructing dictionaries from lists. So for this we can use the zip function, so if there are two separate lists for keys and values we can combine them into a dictionary in single zip function and the big constructor. So that we have a dict constructor and a zip function. So the way to use is basically we define the P will become the values and then the dictionary variable equal to dick zip keys valuable.

(Refer Slide Time: 39:07)



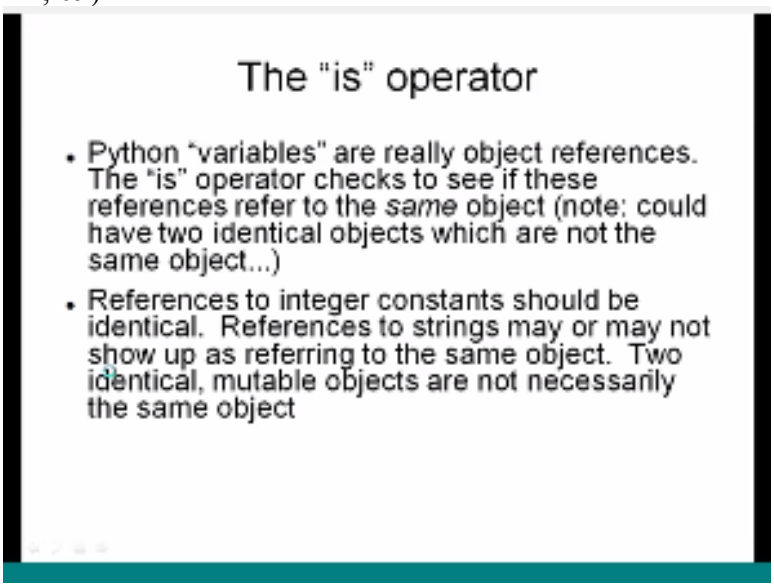
So some generalities instruction basically like so these do not have to be strings it can be devalue that inaudible. So it is only it is immutable developing including tuples this particular property is good for representing as washing dishes. So here there is a matrix which is just a blank a dictionary and then we populate essentially 1 value1, 0 ,1, 0 . 5. So the key is a tuple you can see that basically it is involved in the analysis and then 1,1,4 is another 1 which is .8. So to represent this basically. So it is it is a good that you do not have a string or any other visitor.

(Refer Slide Time: 40:23)



Now collection length essentially is computed using ball Len variable that returns the length of a tuple list or a dictionary that is like how many number of characters of a string. So Eli and Tony with these parentheses because this is a tuple and we are looking for that the length of the tuple it

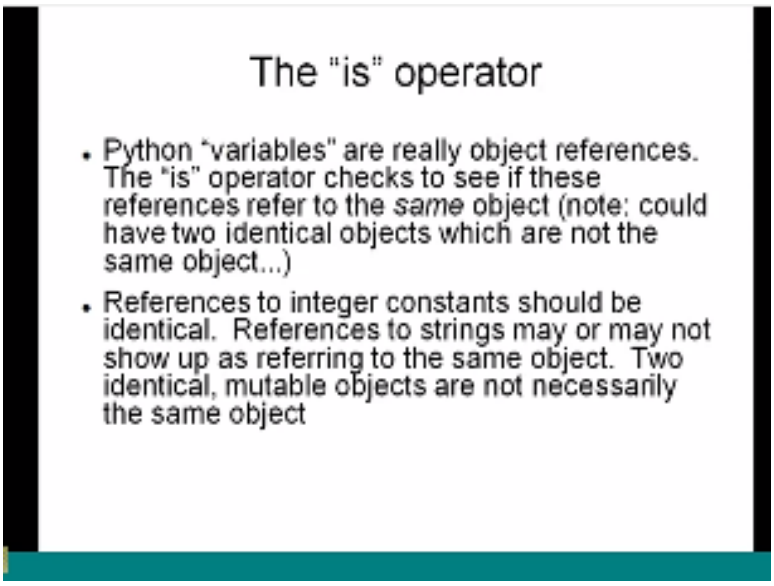
ends one but if you do a length of just tony without the triple now it is a tense for and length of a list essentially with this 01 is 3.
(Refer Slide Time: 41; 09)



So let me talk about some of the built-in operators dictionary functions and a method within the thing tool in LEM is one of them the other one is CMP basically the usage is picked one and it too. So this method actually uses CMP basically compares these two dictionaries and then sees whether they are the same or different. We can also have a STR dict which is a conversion from the dictionary to string. And then this is also a type variable this returns the type of the past available.

If the path variable is a dictionary with a literal dictionary defined or it will return a dictionary type. Now Python also supports a lather method one is a dict clear this clears the all the elements in Pictionary essentially picked. And then copy is another one which makes a shallow copy of the dictionary .So and then the dictionary from keys Integra that creates a new dictionary with the keys from the old dictionary and then it sets the values for 2-0 no sorry deceptive values to the valuable so the usage is this so but not from.

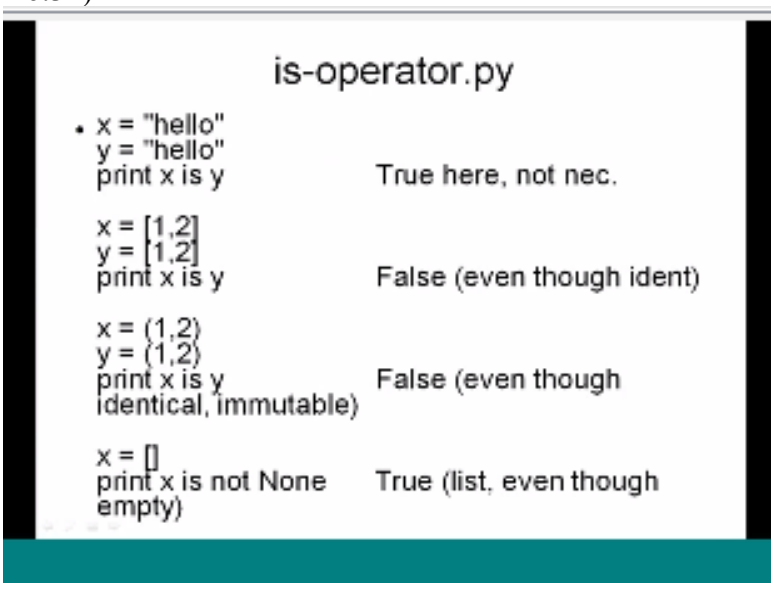
And then I am against the band of successful and then there are also a dict E with the default nun and now this returns the key and then it returns the value or default if the key is not in the dictionary. And then picked has key occurrence true if the key in the dictionary is in the dictionary dict is formed. And dict items essentially students a collection of dicks k-value tuple the standard faulty is another one which is similar to the dict but it.
(Refer Slide Time: 44:27)



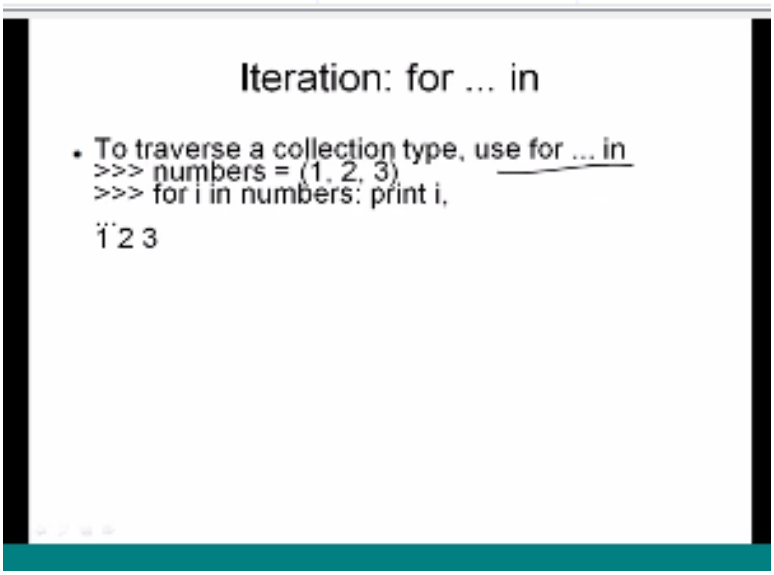
So the dick has key returns true if the key is in the dictionary dict or returns false. And then we have items update picked values that are several methods that are available now let us start up about the ease operator. The Python variables are really like objects reference is something that we call the e sub real checks to see if the references referred to by the same by the object the same object are they like actually like reference by the name of the comic you here we can have two identical objects.

We turn on the same object actually try to access the other location and then the references to the in detail Constance should be identical. So the reference to strings may or may not show up in the dough as referring to the same object two identical immutable objects or more necessary signal.

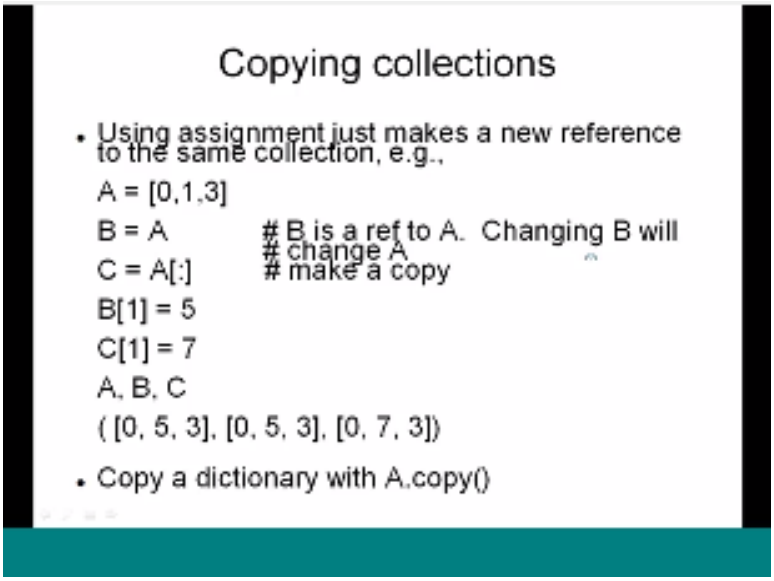
(Refer Slide Time: 46:34)



So let us look at an example some example for each operator. So here we define two of them and then print X is y then it is true here but it is not interested 1 , 2 this one is false. The littlest and then in a tuple again if you ask print X is why it is also false even though it is identical and immovable. And then it is physical to the side bracket basically where X is none true. Now we talk about an in operator in operator is for toxin detect game the in operator determines whether something is a member of economic.
So this is what we used for like some value whether that is part of the program mama to be used that. So here an example basically like team we define as three people and then we say like other Howell is in the team and that it is it is false and then Stewart not in team. So keywords are here then it becomes.
(Refer Slide Time: 48:31)



To travel the collection data type here to use the you we can use the iron essentially for something in something. So there is something in something is different here which numbers are designers one two three and then we can tell it for I numbers so basically expose things and then print the time so here we get 1, 2 or 3.
(Refer Slide Time: 49:19)



So now how do we copy collection So continue volleyball we can actually if you if you are Makin if you use an assignment it just makes and you can new opens simple. So here basically let me see the list B = A, C = A column A [ ] and then this column. We save be equal to one and C = 1 and then we say like then ask to pin the A, B and C so then now essentially like it means vilify 0, 5, 3 in the and we can copy the dictionary with the copy function. So essentially it is a column copy.

Now there are a lot of other methods eventually for this variable essentially so some of them are one is clear. Which removes all elements of the dictionary or external which other big Bible Institute I am the copy that we saw basically its returns a shallow copy of the dictionary? From keys with already like oh that is set and then get key people none we get all the keys and then hash key returns the value basically of the P returns true if the key is in the dictionary else very false.

And then the item that is a keys we saw and then we can also do a set default there we can which is really similar to the get or DB copy sorry did we get of the get parentheses function. And update this essentially we saw that that these are updates the value pair and then values actually do so the dictionary of the dictionary status. So let us stop at this point today and then we will go into the advanced function in the next class.