

StreamBox

>> INSTRUCTOR: In the meantime we will go ☐ ☐ if I choose one through eight, one thing I would like you to pay attention to is the last item in lecture eight. We will not cover in midterm lecture nine. You do some questions we have review which we will do Monday. Monday we will have an in class midterm practice. So that way you will be able to face similar questions to the midterm. And that will help you to polish your preparation for the midterm. We will put an announcement, maybe. I am going to go to lecture eight. Because these will be important portions of the midterm. The way to use these lectures is to look at the index and understand that. I understand the most important one. Here for example, you can see these are not bulleted, so it means we will not be dealing with these topics now. So for the midterm you just look at these once. The same thing for the assets. So these we will finish up in the first 20 minutes of Monday. But this is very important material in computer science. We try to get you as much as we can today. Remember again, there is a way to browse. ☐ ☐ But this is wrong wrong wrong. You should say I want to polish this from a and this will take you to the corresponding pieces for the material. Okay. The scenarios are in my view may be the most important contribution of a language like Python. These were not popular before. With all programmers at all levels. Now, this is probably the most popular type of data software. Especially for massive data sets. So these methods allow you to do certain things with dictionaries. So the way for you to parse this material is to look at everything that is bolted and that means there is a particular instance of what we are covering that is important. First of all iteration. Remember that iteration is not being able to ☐ ☐.[INDISCERNIBLE] It is just a way that ☐ how do you reiterate over a dictionary. The way to understand this in general is that a dictionary really can be viewed in different ways. So this is a typical dictionary. A dictionary called grades. Remember the dictionary is a collection of pairs. This is one item in the dictionary, this is another item in the dictionary, this is another item in addition, so this dictionary has three items. So this dictionary also has keys. So you have items which are pairs. You have these ☐ ☐ you have keys in each of these pairs has a key. They have value. So the question is do we take the view of a dictionary, we could try to iterate on the dictionary using the keys or we can try to traverse the dictionary using the values. Or we can try to traverse the dictionary using the pairs. All of them are possible. So a dictionary can be viewed again as a collection of pairs. A collections of items, inside that is a pair, or we can view a collection of keys by themselves or as a collection of values by themselves. And the programming with dictionaries these three views are useful. Very useful. So for example, we are going to traverse this using a key. In the dictionary of grades to do whatever I want. We will iterate over the keys. Because he, ☐ ☐ because keys will do something. According to the keys so for example here, we are just printing. So for every key it will bring the corresponding pair. I'm doing this slowly because it happens to be if you are not paying attention to this, when you write code, you will be thinking you are doing something, and you are doing something different. You are iterating using keys. And when you are getting out ☐ ☐ and what you are getting out, whatever that key, the corresponding pair. It is not difficult, but it is quite

different to what comes next. Again, even a dictionary is using a dictionary, we can access the dictionary using keys which is looking at the dictionary based on the keys or another method called values which means looking at the dictionary based on the values or another one is items which is looking at the dictionary by looking at the pairs of the dictionary. So it teaches us a dictionary is a collection of items. This green box. Each pair is an item. That one view. Another view for a dictionary, the keys are the corresponding first elements of those pairs and the values of the second elements of those pairs. You want to access the items, access the value it is the value. Access to keys, use the keys. So again one more. You can iterate using the keys of the dictionary. Or the value. [INDISCERNIBLE] And what you get out is this. Is that clear? What is the important thing? The important thing is to access the values. But what do you get out? You get a list of values. Important. He started the dictionary now you get a list. A list of what? Of values. What is it that you are getting. You're getting the pairs of the dictionary. Each of these things is one item. Look at the difference. Okay. So with this now you can start looking at the power of the language so you will have a keyvalue pairs. For example this is the dictionary again. You can now have a full loop and here in the full loop you are saying you know what, I would like to look at all the pairs in this dictionary. So for that key and value in this dictionary because these are pairs. So for these pairs I would like to represent it. Why? A very simple example, but that this could be a lot more complex. You like to do very complex operations on all the pairs. For every pair, maybe you want to bring the key two hours, or for a repair you take the key and take the first letter of the name, for example. The important thing is that you can iterate over all the pairs and after that you can do whatever you want here. One power of all these methods and computer sciences that you can use is when we say a list of integers, that's kind of a very simple list. It's a collection of units. So you can have a list of very complicated objects, but each element of the list is a list itself. And these can be done that so you can have a list of lists or you can have a list in which each element of the list is a dictionary. So you can have a list of dictionaries. This is the power of algorithmic thinking. So let's look up a simple example. Keys cannot any kind of value and values could be complex. A very simple example. This could be a dictionary, keyvalue pairs, and the keys could be things or the values could be lists of things. The value associated with this key is this whole list. For example you are trying to do something with prerequisites of classes in computer science. So the key could be a particular class code, and the value associated with that key could be the list containing all the prerequisites for that class. So for the first one CS 112, you have to pass CS 111. For the second one CS 226, two prerequisites. The point of this is that you can model very complicated relationships between data by using precaution when you need to. So in this case values can be lists themselves. Questions? How many items in this dictionary? Come on guys. How many items in that dictionary? How many items in that dictionary Emma come on, come on. 2. One row is one item. The second row is the second item. Okay. So how do you build such a dictionary? This is a dictionary. And then you put the pairs. The first pair is this one. CS 112, this is the pair. The second pair is CS 226. In the value is this list of these guys. Raise your hand if you got this? What happened here? Wake up. Did you get this? No? You are in a different space. Come

here.

>> STUDENT:I have a question.

>> INSTRUCTOR:Loud.

>> STUDENT:If you print , is it automatically assigned ?

>> INSTRUCTOR:The print is not really what we are trying to explain here. The print is just to show you what is it. That is here. This is saying what is TR ER ER EQ as . What is this dictionary we will defined here. In that dictionary, what is it if you put brackets CCS 112, it is taking this dictionary, and accessing these particular in this dictionary. So in this dictionary take the tears here, CS 112, access to, and print what you get. Well, what is it that you get? You get CS 11 which is this. Yeah. Another example of a dictionary , can you tell me anything? How many items in this dictionary? In that example. 2. The first item, what does it consist of? What is the key of the first item? Ajay. Which I guess you think is just ashy. Excuse me? Suppose? Suppose. Suppose you made the dictionary. You try to do it the proper way. I'm trying to appreciate this idea that you have to know what are the items and they can be complicated. Good. So Ajay , and what happened when you see that arrow. That is pointing to the . What is the example we have in our heads. It can be taken to be a key. The second component could be the value associated with IT. And the value associated with IT is that whole rectangle. The value associated with this key and is valued here so this is encoding you are dealing with something about the United States you can create a representation of the data if you wish. They keys are the names of the states and the second component is what? [INDISCERNIBLE] Countries. So here is a example of a representation of data, states of the US, and the corresponding counties you are going to process all of the data, you can represent that as a dictionary where the first entry is a key which is the name of a dictator, and the second entry are they counties . But they counties themselves , what can you tell me about the counties. What can you tell me about this rectangle? In this discussion. If I give you this table of information. I will you represent that ? Only one person yet? I meant this whole rectangle. Is a dictionary dictionary is just a collection of pairs. So these by themselves is a dictionary. Is that clear? So what is not. This is saying this dictionary has two levels. The first level consists of a state estate names and the second level is a reference to another dictionary and that other dictionary is in this case they counties associated with a key. And this one has another dictionary which are they counties of California. Here you have a two level dictionary . Is that clear ? So you have a three level dictionary meeting you could hear, instead of this, this is may be the area code. But in this particular case, the value of the data is this number, but here you could have another one bonded to a dictionary. For example, pointing here to another dictionary which is the names of all the scientists that lead in Essex County. That would be a three level dictionary. Raise your hand if you got this . What happened back there? Yeah? Okay. How do you build this? You can have something like counties, like the name we are given here. You have these brackets. Inside the bracket , this is a dictionary, you will have keys, this is one key. Color. This is another key. That is calm. But you have to pass these keys in the proper way. The key is associated with this, the second component of that is this whole thing and this holding by itself is a dictionary. So this is the way that you write this invite on

that `pop` in Python. First component has the dictionary associated with the second component has a dictionary associated with this. That is the picture. And then `pop` okay. Now one of the things you want to do when you manipulate data, you would like to be able to read this stuff. And don't take this lightly. This is also a common mistake. There is this particular operator in Python `DEL`. [INDISCERNIBLE] It's important to remember that there are semantics. In this case, the semantics is when you use `DEL`, and you remove a keyvalue pair, okay, but this is delete. Is something is going to be applied, to grades, and great is what? It's a dictionary. In here you say grades, in brackets, a and I as, how should I interpret this. Grades are the dictionary, a key in the dictionary, this one. So what this is saying is, identify the key that matches this one, and delete the corresponding pair to delete this whole thing. Not just to this. You specify the deletion with a key if you are using `DEL`. The semantics is whatever is that key, the corresponding pair is removed. And that's what you get after that. It could be that this particular key is not in the dictionary. So you are trying to do something that is not there. Then I will give you an error. And you can actually go put exceptions and all of that. So that is one way of deletion. There is also something else `pop` and I want to emphasize, that the way this is written, this is saying `pop` meeting this is a method you will apply to this which happens to be a dictionary. The other one there is `no`.here. An operator is going to take whatever is here and do something. This one is a method that can be applied to a dictionary. Oh but when do you use this or that? You have to decide. You can use one or the other depending on what you want to do with your code. It is important to understand the semantics. `POP` which is in operation that we just specify with a key, and they keys in the dictionary, you remove the corresponding keyvalue pair. But what you get back is the value of that key. By the way it is good to put the full value. If you don't specify that an error that `pop` so you want to look at the semantics of this. We create these pictures for you. They keys in the decree, the semantics of that is you go access the dictionary with that key, and get the value associated with that key from the dictionary and now you delete the corresponding key with the previous method we had before. So look at that `pop` used in the semantics, the previous operator called deletion. Okay. You are giving a collection of data and you are asked to do something. So one of the questions is, from the data, what is it that I am going to build in order to develop this application? So that our `pop` so there are different ways to build the stuff. Okay. If you have something you can iterate on, you can create a dictionary using that iteration like we are going to exemplify here. They key here is this. Another method that can be used in dictionaries is called `fromkeys`. Now you have some peace. Maybe names of professors. Or names of your favorite players. Or names of books. And somehow in order to start whatever you are doing with this pieces of data, you will need to initialize something. This is a way to initialize that without you having to go set up the initial value of this do that, set up the initial value of this do this. You can do it in one shot. If all you have is the names of you guys, somewhere, and you would like to initialize that because you would like to create a dictionary with those names, you can just say `fromkeys`, dictionary.from keys and specify a value. So it's over you guys, the names that I want to initialize to evaluate the grades of each student so you will get a dictionary with each of you as a value add the grades of the

students. Is that clear? You want to associate to each of you a great, so the name and then the grade . In this particular case, we are associating to everybody the same value. These are the purpose of the front keys is to initialize. So you have two students and you would like to initialize them to value 10, you can say I want to create a dictionary, a dictionary that I created, I will use these guys as keys, and initially I will give to each of them the value 10, what this does after you apply this, what this does is create a dictionary with this many pairs. How many items in this picture? 2. As many as the elements here. The value of each of them has been initialized to 10. So that initialization by using the front keys dictionary. This is more interesting now. I'm going to use the same method from the keys, and apply that to the string, ABCD E . Now you know you can apply that, well, we are discussing this that here, as a parameter to this, we can put anything that you can iterate on. Again as a parameter here, you can put anything where you can iterate on. What if this is a strength. And we know that for a string you can iterate because a string is a sequence of characters . So you can iterate over each character on this string. And then we just say this , there is no particular value specified here. So we are going to take it to be done. And just by doing this, now you have created a dictionary. The first element in this iteration is A. That a is going to be associated with the value and B with a value from a C with a value etc. This is forming the string into a dictionary by doing this. And when you do that, now you can use all the dictionary methods that you have. Very useful. So for example, dictionary from keys, you can create a dictionary from keys. I think the next one may also be interesting. It can also create a dictionary out of the use of topics. That is a list of topics. This is very useful. Remember that we said that one of the most important views of data are tables and we are going to be working with tables a lot in the second part of the glass. And remember tables are basically a collection of label columns when each column corresponds to an attribute, and here you have values. And now each of these things is a record of information. And that record can be viewed as a record when you have a table like this you, you have this in the data, you as a programmer say, oh, okay, okay, I can use this record as a record and I can operate with that data in one way to do it is, to create a dictionary out of the collection. What exactly is a topo. There is a way to create dictionaries from lists of topo's. For example, you have something like this. Remember, why is this a list? Because here, the pairs get assigned something that is with these brackets. The collection of pairs . But wait a minute? I don't like this business of a list of pairs for whatever reason or for the processing you are doing, you need something that gives you more access to the radar. According to they keys and according to the values even though here there are keys and values that you look at with your eyes. This is a list, there are no keys and values here. But you would like to treat that information as something that has keys and values. This is going to be treated as a list. This is a list with this item on this item. It's a list with two items. The first element of that list is this. And the next element of that list is this. If you would like to treat this as a dictionary, you have to do something so that instead of these brackets you have this. The difference between this line and this line is here you are using these brackets, and here you are using this coded bracket. This might sound trivial, it is not. Why. The first line allows you to use list methods to complete a list and a list of matters. The second one allows you to create

that as a dictionary so that means you can access keys and values , you can access pairs. The power that you have to manipulate the same data by being treated as a dictionary is a lot. You gain a lot of power. Can anybody tell me what happened when you apply the previous method from keys dictionary to this list here. Of these bears. So this is saying I want you to take the list and I want you to associate with each item in this list . You can do this in a dumb way. What is the dumb way. You can create this and go put column and type, oh that's a good code. That's what you want to associate with this. There isn't one way to do it. Again, we take this list, of these pairs, apply this method, we take this list , associate with each other, whatever you put here. Or whatever you want. Whatever you want. And then by doing that, now this list has been transformed into a dictionary. Because here is a dictionary that contains each of the elements of the list , and they keys and a value associating the special value would like to associate. Now this is a dictionary. Another way to think about this, this is a way to take a list to annotate each of the elements of the list with something. And then use this method and now you have a dictionary. What is the power of this? The power of this is access to list . Because list have first element, psychedelic, all the way to six elements. When you have a dictionary, you have random options. In a dictionary you can give me any key. And the system doesn't go get the information, that despite from the programmer point of view, that's one of the advantages of using a dictionary. You can look at this example. This is a tricky one. Creating dictionaries by using this okay . You are building these things, it could happen you associate to a key some value. The name is a key, I am so sorry. Oh my gosh. So I can put his name again and put K now. The language itself associate with a key. The latest value entered in the data. Oh I'm sorry, not a A, a V so I enter it again. Okay. Another important aspect of manipulating dictionaries is being able to copy them. Now what does coffee mean? This type of copy is a shallow copy. Shallow. How many of you have heard of a shallow copy back this site is beating that site so far. Shallow copy. What is a shallow copy? Should she said that she knows.

>> STUDENT:[INDISCERNIBLE]

>> INSTRUCTOR:Loud loud. Can you say it louder? Volume . We call it a shallow copy. You really don't copy. That's why it's called shallow. What you do is you copy a reference to an object. And the sounds like a whole bunch of words in English. Suppose each of you have a number. It happens to be is John here? James is here? Peter? Give me a name . Xavier. Xavier, you are the student I want to make a shallow copy of Xavier . I don't want to take him and copy him. I know he is a student. So whatever I am doing, I will use students 20. So I can copy student 20. Student 20. And whatever I do to student 20, is going to revert to Xavier. This is extremely powerful. And in computer science, this is the result of something called pointers. You need to think about pointers . You think that is reference to an object and let me give you the picture and I am sure the picture can explain this. This picture looks very complicated. All right. Come on guys. Tell me what is that picture. In one word. In one word. What is that picture? I have one person. What is that picture in one word? Two people? Wow! What is that picture in one word ? If you want to. What is that picture in one word? Use something we are discussing in the class .

>> STUDENT:[INDISCERNIBLE]

>> INSTRUCTOR:A bunch of references, that is true. He said pointers, a whole bunch of references. It's a dictionary. What is a dictionary . How did you discover that ? Because it has `{ }` .[INDISCERNIBLE] That has a narrow that means that is the value of a variable there is something in that rectangle that you have on the left-hand side is that rectangle you have has two columns. The first column is ABC and the next column has a whole bunch of other arrows. That's how you did it, right? So that's a dictionary. That's all. Come on guys. Come on guys you should know this. Here is something pointed to it object that has two columns in the first column there is something. Pointed to something, referring to something. So what is this, where my mouse is? What is this, anybody different than Nathan, tell me what is this? Come on guys. Come on guys. What is this?

>> STUDENT:[INDISCERNIBLE]

>> INSTRUCTOR:What? Too complicated. We need an object . This is the object being appointed to like this. Whatever that object is, `{ }` now it is obvious now, the way to view this is top-down. The top part of this object has two things. One thing. Another thing. Yes. The first thing is pointed to another object, whatever that is. In the second thing is pointed to a second object, whatever this is. Don't get lost in the details. At the end of the day , you have keys and references to objects, and those references can be crazy. If you want to understand what they mean, then you go to that object and again you apply everything that you know about computer science about this class or whatever, and now you try to express to your self what the heck is this object here . This is very complicated. There is nothing different here than this one migrate here. That just this one line right here. This one line is encoding all of this. You should be able to go back and forth. This one-liner added this. Now, dictionary, this is a dictionary, besides this dictionary, you have these two brackets. So you have brackets on this. You have a bracket and another bracket, that means what? Come on guys, without this, you will not make it on the midterm. If you don't understand what this means and you don't understand what this means, and you don't understand what this means, we are doomed to fail. The screens are quite different. So the first thing you need to understand, are the two parentheses. What the heck did they mean? Would you have these two, what the heck that means? When you have these two, what does that mean? I don't want to repeat the whole thing again. Because the way to parse this is to look at the first bracket here you understand what that type of object is. Then you keep going inside. You are able to follow to see that this is precisely this. This part here is precisely this. The point about this is that this A is a key and the value which is this whole thing is precisely this. The value associated with IT is a list . The first pieces one, the other piece is composting of my mind. We need a dictionary. This is what you need to understand. Keys .2 objects and the objects are the values associated with a key. You take a picture of this and copy it. It has no shallow copy. You don't take a picture of this and copy it. You do not do that. Shallow copy is this. Can anybody tell me? What is it that is being done here? Okay, yeah. So the basic idea is the copy that you are doing is only at this level. You are really copying the keys and the values associated with those keys are exactly as the same thing before. They point to the same thing as before. So now this is a new dictionary .

You can assign it to another name. And this dictionary for all practical purposes, is a basic copy of this dictionary. At a shallow copy. Raise your hand if you got this. Now the site is beating that side. What happened here? This is the same picture as before. The same picture as before. But now we are going to this dictionary and we are associating with this PC another object which is different to what it was before. So you are changing this but you are not changing that. The copy allows you to maintain what you had before the action. And now do all this stuff. Without entering the initial one. That's the purpose of copying. I want to do something with you guys, I want to make shallow copy, and put it in the shallow copy and you guys will be affected. There is this business for comprehension. In this business of comprehension, this is very Python. This is a Python creation. In the old days there was a strength of creating short pieces of code that is a trend of creating short pieces of code. Some people became trendy with that. These guys want to do this and do that. And in the old days of units, that was a trademark. I can do this in only two lines of code. This has been completely debunked. It is the wrong thing to do. The purpose of code is to share with others so they understand. I thought allows you to do the things that you want to. And these things have a syntax. Look something like this. So you can create dictionaries with the one line of code. These two brackets, why? Because these brackets be dictionary. What you want to do is create keyvalue pairs. For what? Core elements. So you have something to iterate on. For example, here. Let's try to parse this out. It says I want you to create keyvalue pairs. Keyvalue pairs, keyvalue pairs well you have this column. And they key I want to be a square of a number. And the value I want to be the number that is that's what you want to do. Can you do this in normal Python. Sure. But I want to restrict those pairs from values of X in a particular range. In this case one through 10. Just by doing that you have created a dictionary. What are the first components of that dictionary? What are the keys of the dictionary? According to this? Try to get your own description of this. What are they keys of this dictionary? Yeah? Values of square, so they keys will be a whole bunch of numbers that are square. The first guy is a whole bunch of numbers. That's what this line here is doing. They keys. Now what are the values associated with those keys. Those numbers raised to the third power. How many items in this dictionary? Just with his life. How many icons have we created? 10. Why? Because we are applying that for every X from one through 10. Is that clear? That's it. The whole I made the whole thing. So if X is one, what is the first pair? One, one. If X is 10, what is that fair? If axis X, first one is? Second one is 10 cube. A thousand. So this is creating for you a whole bunch of pairs ordered by the pair and this is the effects is this one will have seven squared and here you have seven cubed. This creates all of this for you in a dictionary. People refer to this and Python as comprehension. So that statement creates this. You want to create 100 levels. Make a table. Will you change this parameter here. Yes? You can make this a little more complicated if you wish. By adding a condition. So the same thing I did before. You want to create keyvalue pairs for elements this is Python. This is Python. This is something that you call an element. It is something that can be iterated on. But you cannot do that at the end and if condition. The same thing as before, but this is saying, is if certain conditions are satisfied. For example, one clinician can be if X is prime. And you happen to have some

function called prime acts that the text the Texas plan, you can code it here, and it creates the same table as before but you're only applying that condition when X is prime. Let's give you a flavor of the sort of things if you want to impress your folks at Thanksgiving, you can create a very complicated table by using this. Look at this one. What is bracket, what is this code bracket? What is this whole thing? What is insight? What is this saying? You are creating a what? What do these two brackets mean? Okay. Now do you see what's inside? X column in this whole mess here. So by the way, this one matches this one. I'm sorry. So what is been extracted from here to here? It is one of those cryptic pieces of code. You would like to have inside, pairs XY such that when are you going to put a pair there? You are going to take a pair of numbers, that you will associate with the pair of numbers, the product of those two numbers. When Y is a certain range. It here only Y and you are adding a condition. A Y is equal to X. And then after that you have another condition on X. Is this something you have to become an expert on? My personal suggestion, not really. At some point, at the Thanksgiving table you can show this to your folks. And they will say destination you will say I could do a whole bunch of stuff in one line. You can write that and Python, there's equivalent code and Python. The semantics of this is precisely this. CS 210 Python code but there's no need to use this cryptic description, but this is

[INDISCERNIBLE] I hope you can look at it, and not, if you want to impress your folks, just this line creates that thing. Pair product. Pair product. Pair product. Pair product. For every pair, there is a product. You want to organize this in a certain way. Okay. And the organization if these conditions these are the ones that. Oh my gosh I don't understand I don't understand what a table does. Here we will give you line by line, line by line execution of that code. Okay. At this point we are not going to go into this in this lecture so you can jump over that. And orders dictionary, we will not go over that night now. And there is this funny thing called counter and we will not go over that now. And there is also a couple of other things now let's move to sets. So we were doing a whole bunch of things. Mapping these two dictionaries, but you can do similar things for sets. Let's look at this one. What is this? Moral of the story, anytime you have a string you can iterate over it. A string is something you can iterate over. You can apply this crazy thing before. This is Python. A which is a data string. Remember what this means. These are methods. Methods that tell you if a particular character is an English character or not. These are methods that take a character and transform it to lowercase if the character is not lowercase. Just with this statement, this is what happened. Now tell me English, what this is doing. In English. Let me help you a little bit. For every character in this string, leave that character, it's an English character. Do this if the condition is satisfied. Same people. What happened back there? You what? Text processing has become an industry in natural language. Solve this kind of stuff this kind of stuff will show up in your professional life. Okay set operations, you can do operations and sets. You can take the difference up to sets. Difference between set a and set the is whatever it will be. The intersection is whatever is in both sets. Whatever is in A will go and be and you can impart on this. Take the unit up to sets. The state the intersection of two sets. You can indicate if one set is a subset of another. But be careful if it's less than or equal, you are checking if that's a subset

of that. He may differentiate something as a subset . So this is very simple. It's important to learn the symbol that you can use in Python to do these operations. What about something like this? In many obligations we like to find unique words in the document. So let's go over this quickly so you have a particular string you would like to define a function or method you want they would like to get something out of that. That something out of that you are going to specify here what you want. I would like to point out this is very powerful. Very powerful Python construction. This string is the parameter . This is a method called a friend. So you can put whatever characters you want. So Monday we have a practice midterm.