>>PROFESSOR: Why is is very useful? Why is this useful? Data that has the format of a dictionary. Is data (AWAY FROM MIC)

That is here to manipulate as a database. These terms hear that you have in your data, connecting data with different types and they are accepting certain values and that is part of the external role, but when you want to bring this into the database, imagine that you have 10,000 sensors and you have all of these with different values, you would like to be able to do analysis with these different sensors so now you have a whole bunch of triples or values and just by doing this, now you can treat this now as a dictionary in which the key for the dictionary is precisely the name of those sensor variables and the value associated with that key is the same value, but now you can apply all of the dictionary operations that are available to you. When you have lots of perimeters, then one thing to use is to separate and one it depends on what you're doing, but you can separate the different pieces into something that is the 1st 1, because many times the 1st 1 is something more about the rest, for example at this is data about students, maybe the 1st parameter is the student name and then after that and so in many cases the first parameter has special semantics because it tells you something about the origins of information as associated with our perimeter. So, in many cases it is good to separate that so you can define a linear function in which the first is the first parameter coming from the data, whatever it is. And then this is the rest of those parameters and remember we have seen this before in the first is the first and this is you and the rest is what the rest is. So, something like this really has no differentiation between the first parameter and the rest. Just by doing this and now this is being seen as this - this is separating this into two pieces. The first and the rest. So, this list is going to be divided into two.

One with the first parameter and then the rest of the parameters which is also a list so this can be manipulated now by whatever method you are doing. This is relatively not difficult, but you need to be aware of when to use that and how to use it. Again, this is an example of a list of perimeters that is being transformed into the dictionary. And it is important when you look at this thing, it is important to you to realize this is a dictionary. Well, here you have something and then a value like okay, this is a dictionary, if the data does not look like this, that is fine by using something like this.

And you can look at all of the examples of this kind of manipulation and you want to isolate the last value and you can do the same thing and so for example you have one, two, three, four, five, six and you can isolate the first and the middle and the last one. So, all of this is manipulation or perimeters that may need to be passed and then you have lots of perimeters and they have some meaning in the process in that you are isolating or separating the perimeters in this way before you pass them through a function which is usually very, very useful.

Also, I would like to mention that there is this syntactical symbol that you have lots of perimeters and you're going to use that to indicate that there is something different before that and something different after that. The difference is before that, all of these

perimeters are taken as positional parameters. Positional parameters means that the first is the first and the second is the second. They are treated differently. And then sometimes you start building these very large programs and it happens be that you are not documenting them properly and that is a problem. And then this provides you something called DOC assist. Now, this is very useful, because it is very simple to use, supposed to hear you are defining a certain function and you have in your mind that you want to define a function. So, you want to define a function and you do not know what you're doing, but you want to write that right now. I'm going to do this later in the way you do this is that you specify the name and then somewhere at the end you put pass and then just by this it is not going to complain. So, they are going to define some function that that is something that will be for later. And so we (INDISCERNIBLE)
Even if nothing is defined here. Okay? Answer now you will be asking yourself, what is the function, right? I want to function, but I don't know exactly what it is. So, at the end you can type in AGLP, but you don't have the function - this is preparing templates of code that run okay in the interpreter so the interpreter is going to tell you what? At the end of the day you have to do this function at some point and so there is some computation that will be included.
Okay, these are syntactical tricks that when people are trying to hire people as programmers, usually they give you some code like that and they will ask you what it is doing. And so you need to understand what is this doing. And these are very easy mechanisms that interviewers use to detect the type of skills the interviewee has.
Okay, question. How many of you know how to open a file? Just raise your hand, I'm not testing, I just want to know. Okay. How many of you know how to classify? Okay. Can you explain to me what is that? Opening a file? Yes?
>>STUDENT: (AWAY FROM MIC) allows you to read a text (INDISCERNIBLE).
>>PROFESSOR: And what happens if that file is something that you don't have?
>>STUDENT: It shows you something that there is an error and the program kills itself (INDISCERNIBLE).
>>PROFESSOR: I understand your reasoning, but that is not a lie. I support that now and I say look at it , I want you to write a piece of code that takes the names of all of the students in this class and look at the GPA of each student in the class and spits out the top 10 students with the GPA. Suppose that is the question. And I'm not giving you the data yet, I'm asking you to write code. But, you do not have the data. And so what is the first thing that you do? Where is the data? I can tell you the data is in a file, but I am not giving it to you? So, what can you do? CS students, you create a file. That is the purpose. Opening the file to create some name for something that I will use as a file and in that file I am going to do the data stuff later on. Is that clear? Yes? Okay. Next question.
So, what is a file? Is it a potato or a tomato, is it a car or a video, what is a file?
>>STUDENT: (AWAY FROM MIC).
>>PROFESSOR: That is a very good answer, but it does not help much. I have a very good answer, but this is a collection of what?
>>STUDENT: A collection of bites?

>>PROFESSOR: No, lines. A file is a collection of lines. Now, moreover these lines have an order. There is a first line and a second line and 1/3 line. And the last line. That is what the file is. It is true what you said, that this has to be stored in the computer in a certain way, but you don't need to worry about how is this installed in the computer. You as a programmer just need to think that a file is a collection of lines that have an order. Another way to express that, a file is a order collection of lines. So, you have an order, why having an order is important? For you as a Python programmer? In whatever you are doing? What is the use of having an order? Excellent, having an order in whatever you're doing, a collection of tomatoes - there is a first tomato, second tomato and 1/3 tomato, collection of people. There will always be a first, second, third and fourth, etc. and still having data in order is going to allow you to iterate. Iterate over the data. And when you have something that you can enter it, what is the Python statement that allows you to iterate over a order collection like a Java order? What is the Python statement that you want to use to iterate? Four loops or five loops , this is done and you have whatever data and the data has an order and you want to iterate over the data and so you will use a loop and you are able to enter it, because there is a order in it.

There is an example of that and there is an order collection of lines and therefore you should be able to iterate on each line. So, when you say open the file, you are just saying that I have something here and it is a collection of lines, but you don't know what is inside. And you have to iterate over the file line by line. Okay, so -

The open function is - a way to start this process. Now, in test processes, these methods that take a (NAME) and do something to them are very useful. And a test processing has become the whole industry, why? Because of the language processing. You can use things like chat GPT and there is a lot of language processing behind this processing. But, at the end of the day, it is a whole bunch of words glued together. That is how we speak. So, test processing inherently leads to methods that do something to them. Now, there is something that is called a strip and that is something that we already covered, but I want you to recall that when you take a string and you tie a method to a string, you do not put any parameters here (INDISCERNIBLE).

These files are produced by many different processes. And you don't know in the middle that there is a whole bunch of blank characters. And then for you to be able to do any useful naturalness processing, one of the first things you do is get rid of those characters. Okay? So, this is a very first useful manipulation that will get rid of these blank characters. But, notice that the name of this is the same as the name of that. We take characters from the string. But, here there is something as a parameter and here there is nothing as a parameter. What this is saying is that this is a special case of this in which the parameter is blank. That is why there is a copy of this thing which is a very special case of that. More interesting is suppose now that here you would like to specify a collection of characters. A set of characters. Well, I want to get rid of not only the blank, but maybe and a whole bunch of things that you do not want. You do not want these things to influence or processing.

And so after removing all of the characters, especially right here. Very powerful method for

text processing. And so the character is (INDISCERNIBLE). Okay, let's just do one example of this so you get the feeling of this. So, here is a file and this is the name of the file (INDISCERNIBLE) and this is saying that I had a file cluster and the type of file is text. Why am I pausing here? Are all of the files contained in text? And you think of some files that do not contain text, that they contain something else. Okay, for example a image file contains different things. Give me another example. If I could be what is called a binary file. If I could use a whole bunch of zeros and ones - when in fact, those are the files that the computer works with. So, there is a business of files containing text (INDISCERNIBLE) if you type a document with a whole bunch of text (INDISCERNIBLE) and eventually each of those characters has to be transformed. Transformed into a collection of bits and each character has a collection of bits that is encoded into the character. The array may be 00 11, but those bits have a length that is usually eight bits and then all of this will transform and the computer (INDISCERNIBLE) and you don't need to - but, this is important dealing with files in important for you to understand the type of file that you are dealing with in your processing and when you write this like this, this is (INDISCERNIBLE).

But, this is just a name. This has no existence until you open the file and so this is Python that is staying open that file. And in that file it contains stuff and do not forget what we just mentioned, the file is an ordered list of lines. So, we are going to do a follow-up of lines in that file. And I want to make sure that you understand that there is nothing especially different here to the four groups that we used before. This is for use with something in something, do something. Here, in this particular case within the case, in this case, each of something is a line. The most atomic collection of files in a file is a line in that file. And so by saying this we are saying okay, a data file, open the file and the lines are in that file and you're going to read these one at a time. Now you're going to start seeing how we are going to be using everything that we have discussed before. And so over here we are going to be referring to one line in that file and so now that one line is a string of characters, no matter how complicated it is.

And then we would like to do something without line and because there is a string of characters, the methods that we have for the strings can be used. So, a particular method that is used is the split method. And the split is a method that takes a string and is going to take some specifications here and that specification is a parameter and that parameter in this case is saying I want to take that line and I want to look for the separator color and I want to split the line into two pieces. This is one piece and after that separator, that is another piece.

Are you a color, yes you are a color and anything before him and anything after, that is a split. So, after you do that, you have to divide this into two pieces. So, now you are going to assign those two pieces of the line and remember this is shopping like this into two pieces and you assign those to two variables, one called movie and the other one the year. And before that, movie. And after that, the year.

Well, it could be that the data is contingent, but this is what this is doing. This is somehow assuming that what you have before that separator and this is a string and what is the year? It is a number. Okay, this is what this is doing. Splitting and creating these two

variables and assigning these two variables the name of the movie and the year from this. Now, look at the next line. Can anybody tell me what is this? It may be simple, what is this? We have already discussed this method and so movie at this point is a string. It is this thing that came from here. And so we are taking the string and we are stripping what? The blank characters, right? This is what this is doing. So, stripping out the blank characters and now it is assigning that to something and whatever that value is I'm going to assigned in that to what is this? Again, the year is a string and we are stripping that from blank characters and you are assigning this to this. And then you can print whatever you are storing here. So, what this gives you is this - the year and then the name of the movie. Comma, year in the name of the movie and comma, year and name of the movie. Etc. Can you tell me what is this? Yes? And so here you start with some text file and you take each line of the file and broke it into two pieces and for each of the pieces you took out the characters and then you print and that gives you the dictionary. So, this is like taking data and creating a dictionary out of it. You have issues and we give you the picture and so at the beginning you store in a variable called line and so let's say the first line of the file and look at this as some blanket stuff here and there are some blanks here etc. after you have the split divided into two pieces meaning this piece is here and in the second piece is here. And so afterwards there is some stuff there and you would like to clean that somehow and at this point this is the name of the movie with some of the most strange stuff and this is the year which is some strange stuff and you would like to clean this out by removing these strange extra characters that you have. And then that is when you take this string and this string and you strip them from the blank characters and you take the year and you take the name of the movie and you take out the bad characters and now you store that in these value clusters. And what this is doing, this is updating a keyvalue in the dictionary called Oscars. I want to pause here. Because, the problem with this syntactically which is that you will be looking at the symbols and we are really understanding what is happening behind this. This instruction here, this instruction is an assignment and there is something on the right-hand side and I assign this to what is on the left-hand side . But, this kind of assignment is associated to this object and this object is creating an association and so this is creating a new keyvalue pair where they key in this case is the year and the name of the dictionary is Oscars and the value is the name of the movie. And so we already covered the dictionaries and so at this point what is it that you need to recall? When we have a dictionary how I have a keyvalue pair to a dictionary and how do you do it? By signing a value to the key and if this is a dictionary, this is what you do. And so this is updating the dictionary and in this case by pairing 2014 with that and then that will give you this.

Again, this was a string that we will obtain after this and will we apply this same method to it with all of this extra stuff and that is now cool and we take a movie and we strip it from the extra stuff and this is cool and now we make this assignment and this assignment is literally doing the following. It is taking this dictionary that has two entities and remember this is dictionary and it is storing it right here, the year - and this is a reference to the year value and this is a reference to the movie value. This is exactly that. The semantics of this is This here

One more, this is an assignment statement and it is taking something from the right-hand side and it is assigning something on the left-hand side. This assignment, you have a dictionary that is looking at the component year which is this and it is pairing that with the second value and so this is creating a pair, 2014 column 20 years (SP?) this is what the computer is doing and this is what the language is asking the computer to do in the interpreter is doing exactly this. Raise your hand if you got this.
I guess if this is something that you want to take home out of this class is that programming is a collection of templates. People that have program for many years have a whole bunch of templates in their head. This is a template. What is a template? It is some visual representation in your head for how to do something. And in this particular case, what is this something that we are doing? We are creating a dictionary that is going to have as a first component the year and as a second component a movie. Associated with that year.
Now, suppose we are not dealing with movies, suppose we are dealing with videos. There is no difference. There is really no difference. They are the same thing. We are creating a dictionary with videos and years. Or maybe you are not dealing with that, maybe you like American football. Well, you can also associate this with the champion. You see, this template are really independent of data and that is what you should take from this class. It is not (INDISCERNIBLE) yes, at some point you have to write this down, but this is more important and so the template here is to create a dictionary that associates years with something and this is an important instruction. And so where do you get the data from? Where did you get the data? From the file. If it is American football, you open the file and whatever the line, you do this association. With a corresponding naming convention. Okay, I think that that should be enough.
Okay, and after you have that dictionary creation, then you can do many things. For example, you can solve by the year and remember this is a file and the whole point of this, it could be gigantic, after you did this very nice simple piece of Python code, now you have something very nice, it is very nicely organized like this. And so you have a whole bunch of stuff like this and so now, do we want to sort it? I'm going to pause here and there are something called Oscars because this is the name we were talking about. Oscars. Text was a file. Full of a whole bunch of stuff. Without txt, it is the name of a dictionary that we created from the file. Which is a collection of (NAME). Now, we would like to sort - do not forget, when we have dictionaries we have three things. One is a key and another one is a value and what is the third thing?
>>STUDENT: (AWAY FROM MIC).
>>PROFESSOR: Let's look at the movies example. We have a year and we have the name of the movie. The year is the key. The name of the corresponding movie is a value, right? Two things. What is the third thing? Back there - the dictionary has keys, it has values, what is the third thing? What? Let's see, you can help us. This scenario has keyvalue, what is the third thing?
>>STUDENT: (AWAY FROM MIC).
>>PROFESSOR: No.

>>STUDENT: (AWAY FROM MIC).
>>PROFESSOR: What? Z value as a pair. Yes. It is (INDISCERNIBLE) you see, when you say keys in a dictionary, what is the first component? Like what is the years? If I tell you give me the list of keys, you will bring me the list of all the years, right? And now the names of the movies, if I tell you to give me the list of the values, you are going to give me the list of the movies. On so this is divided into keys and values, but there is something else which is exactly may be more important. If I gave you here the list of years by themselves and a list of movies by themselves, that does not mean the dictionary. If I gave you here a list of years by themselves, 2015, 2020 - and I give it to you here by themselves like a list of movies by themselves, it is a list of years , but this is not a dictionary, yes?
Again, I have here the list of years and here the list of movies. If I give you those two things independently, what do you need to do to make it a dictionary? You? You seem to know this. If I give you a list of years by themselves and a list of movies by themselves, that does not make it a dictionary. What makes it a dictionary?
>>STUDENT: (AWAY FROM MIC).
>>PROFESSOR: Excellent. You need to specify the mapping between the year and the names of the movies. That is what makes it a dictionary. It is the mapping. And what is the mapping? 1984, the king of whatever continent does the mapping, what is that? That is called an item. In a dictionary. So, a dictionary has three things and one is the list of keys. Another one is the list of values and the other one is the mapping between the two wishes the list of pairs or the list of items. Each item is a pair. Why are we digging into this? Well, because if you do not have clarity on that, then you do not understand what it means. And so Oscar was the name of that dictionary and so when we put on those items, what is this referring to? It does not say the Oscar does keys and it does not say Oscar does values, it says Oscar does items and what does this refer to? I just told you what are the items in the dictionary? The mapping, right? This is referring to that mapping. This is saying that we are going to be dealing with the pairs in the dictionary. And so each element is this which is a key value in the pair. It is not just one year or one movie name. It is the pairing of the two that the element in this - raise your hand if you got that. Right there - no? Yes? Okay.
So, you understand this, then? These are the dictionary items (INDISCERNIBLE) and then you can look at the details, okay. Now we were assuming that we have a file with all of this information and that that file can be read by opening the file, line by line and doing something to each line and creating a dictionary out of it. And yes, this is the reverse. Somebody has to have done that. And so (INDISCERNIBLE) is another operation for file. And yes, you know this. You have heard of this. But when you go through language like Python, you need to know how to write this down. Okay, so for example suppose that you have something like this - what is this? Grades are equal to that, what is not? This is a what? This is a potato, this is a tomato, what is this? Come on, come on, come on, come on. It is a dictionary. It is a dictionary. It is a dictionary. It is a dictionary. It is a dictionary. This is a dictionary. This is a dictionary. This is a dictionary and I hope from now on you know what a dictionary is. The way to identify a dictionary is these two brackets and in the middle you have comma and each keyvalue pair is specified in this way with the column. This is a

dictionary. This is a dictionary. This is a dictionary. Okay.
Now you want to put that into a file. Now you would like to put that into a file and by that I mean this. (INDISCERNIBLE) Whatever it is called, this pair you would like to put into a file. And so you open a file and give it some name, whatever the purposes and here is the second parameter of this function. This is a function that will open this file and here you specify (INDISCERNIBLE) and the W which is saying that I want to open this file in the right month. That means you're going to put the stuff into that file. This is doing nothing yet, this is just preparing the file to write something into it. Now, we have a dictionary and at the dictionary is going to be something that you can traverse by looking at each keyvalue pair. And so for key values, for this pair in this dictionary, what we are going to do is something. So, open the file and for each pair in this dictionary notice here - the dictionary is grades. We are going to iterate on that dictionary based on the value pairs and at the keyvalue pairs can be accessed by specifying that out of this dictionary I'm going to look at this dictionary by looking at the items i.e. this is item 1 and this is item 2 and this is item 3. okay. So, I'm going to do something to the key values and what is it that we are going to do? Okay, here let me ask you, can anybody tell me how to interpret that? I have one person -
>>STUDENT: (AWAY FROM MIC).
>>PROFESSOR: Very good, I just wish the description instead of the value and calculate, okay? And so what he was explaining to us, what this is doing in this parentheses is taking a key and so this key is that key, whatever that is, we are comparing this with a string that is a column plus this value, and what is not? Well, the value is coming from here. And so in order to (NAME) we need to treat the value as a string and so we are staying STR for the value and then we calculate that with this special character and this special character is the way to specify the end of a line. The end of the line.
This is just an agreement that whenever you see in the file (NAME), okay, this is the end of the line. Remember, you are creating this file by doing this. You are creating a dictionary and you are pushing that into a file in certain formats. This is what this is doing. So, this is creating this string which is the combination of these pieces into something and now it is going to - this is a method that takes this file and remember this is a file variable and apply this method to that file and it is going to write that whole string into a file.
And at the end, you close it. And to close the file, do not leave it open, that creates some problems. Now, to make your life safer, always create an exception. Because, there is nothing new except for the loop (INDISCERNIBLE) and so if there are any issues, because there are always issues. Then you will be notified by the interpreter. And you want to take whatever it happens, at the end of the day, finally and remember this goes through Python, finally I want you to close the file. And over here, if things go wrong, then you may not get to hear and so the file may not be closed. And so by using this, whatever happens, whatever happens, close the file.
And so here there are some more examples and here are some examples parsing the population of the file, that is question three for the quiz and so understand this, this is doing nothing different to what we have done before. Except that there is some special

methods. One is a special method called replace that replaces something by something. And so understand that for your quiz. And so (NAME), do you want to show them this map? Okay. Yes. Okay, this is the map of the live performance and we ask these questions in class life and each of these presents a group and so these all represent a group and there is some very little stuff inside and that means that you have very little credits and the idea of this is to fill in the Circle in the internal circle represents the percentage of what you could achieve and so a circle like this with a little tiny circle inside (INDISCERNIBLE) and so you like that can turn the circle in as close as possible and is anybody here from group 13? No? Is anybody here from group 53? Interesting. 53? You guys are doing very good. Group 57? You guys are doing very well. And so how did I choose those three groups? Because, those three groups have a big circle inside and out color. The color is green. And so the goal here is that we would like to see not only a big circle inside, but also a colorful one. And so if your group gets 90%, then that means the color here will be read. So, this is the goal and the goal is to get color from the circles and you can check your group number for this. This is to entice you. This is a very simple mechanism to accumulate bonus points. And you entice her group to participate and you can multiply that bonus point by three. Because, each one will be answering the questions, right? And so the first part of the semester is over and remember, the grades at the end are put in the standard grade and the bonus will be used after the grades are computed to push your grade up. But, there are some groups that are very (INDISCERNIBLE). However, it is your choice. Okay, see you on Thursday for the quiz! R