Day 9 Python Deeper Dive

Select text to create highlights Learn more

Transcript

S1 Speaker 1 0 24

Class. And so guys, hello, happy Monday to everyone. How you feeling today? How you feeling to close Python, because today we actually, we're gonna close Python and tomorrow we're gonna start with some Python libraries. But today it's, it's a fun class again, why? Because today we're gonna see some more advanced functions in Python and we're gonna see some, some interesting exercise as well. In this class, I am going to help you or we are gonna work together in a lot of activities. We have some of the activities that you're gonna be doing by yourself.

So for the first part of the class, we're gonna be doing Python. And the second part of the class, we are gonna go into gi, okay? I am going to do some examples. You're gonna work with me in order to do the GI Parama. That is actually how to act, how to commit, how to push, how to pull, okay? And we're gonna have enough time for you to actually try and ask questions. Actually, in that part, what I'm gonna do is I'm gonna present like really quick how to do it, and then I'm going to create the breakout rooms. And, and we are gonna, we are gonna start bumping the whole class with questions, right? So I can put this file here, I can match my branches, whatever you want to do. We actually gonna leave that part for Kit specifically and for you to try out as well. Okay?

S1 Speaker 1 ▷ 2 12

So we're gonna start with the slides of today's class. Let me just start sharing my screen so we can jump right into that. And please, lemme know, can you see my screen? Thumbs up if you can. Yeah. Perfect. Okay. So let me slide. So this, and, okay, so a deep dive into Python. So today's day three in Python, and we are going to see how to create dictionaries. What is a dictionary? How to create a list, list comprehensions. That's a really, really powerful function in Python and it's really well just well used everywhere in the industry market. I, I am going to give you some real examples there, right? And reused python functions. Again, this is one thing that you are going to be doing every day if you're working with Python, use coding logic and the part of GitHub that is actually at commit and push code into GitHub.

S1 Speaker 1 ▷ 3 33

Okay? Welcome. Welcome to today's class. We're gonna start with a warmup activity that we are gonna be doing together. Okay? This activity is called serial cleaner. So in this activity we will have to create an application that reads a serial data from CSB file. And then we're gonna print only those cereals that contains five or more grams of fiber. So as you can see, this is instructions. We're gonna open this C S B in the first part. We are gonna, we are gonna start not skipping the header row, but we're gonna consider the head of row, okay? And we're gonna read through that reminding rows some hints here, how you're gonna, how we are gonna do it. We're gonna actually open the C S B, we're gonna read

it and then we're gonna create a a, a next function to read the header. And then with a for loop, we are going to actually read all of the rows. Eh, one thing that we're gonna be doing in two years are whole number and as such cannot contain decimal. So decimal numbers will have to be casted or converted as a float or double, okay? The bonus part is we're gonna do exactly the same thing, but in a CSB that does not include ahead. Okay? I am going to start sharing this part so you can open it with me.

S1 Speaker 1 ▷ 5 15

So if you go to GitLab, then you go to lesson plan, Python three, day three go to activities. And it's gonna be this one serial cleaner. You open this one, we have the instructions, the one that I just showed you, and we can open the soft one, right? Or we can open the soft one and just keep following, okay, with what I'm doing. Any questions guys, before we can start this activity, this warmup activity together? No questions. Okay, perfect. Okay, great. So let me start sharing. I am going to do, I'm going to work with Visual Studio Code again, you can work with any, any text editor.

S1 Speaker 1 ▷ 6 16

Okay? So the first part guys, can you see my screen? First off, thumb up. Perfect. Okay, in the first part, one of my recommendations is that when you're gonna start any activity that involves A C, S B involve data that you are going to read or, or, or a database or tables or anything, please go first and read the date. Okay? One of the important part of your code or the successful of your code is that you understand the date. So in this case, for example, eh, who can tell me, I have a CSV here, right? So who can tell me the row number one here?

S1 Speaker 1 ▷ 7 12

What is the suggestion that, is it a header or is just data? So what do you think that this can be a header. Header or not really? It's a header, it's header it like the suggestion that maybe that is a header, right? So we, we part from there and then we can say that each of these roles belong to one column. So this cell belong to one column in the header. I am guessing that name belongs to a hundred percent brand, a hundred

percent natural brand and all of it, right? So if I go to my instructions, it says that get only the ones that have equal or bigger than five of fever. So I look here and I can see fever here. So I am going to count, so I have this in my head, 0, 1, 2, 3, 4, 5, 6, 7, right? 0, 1, 2, 3, 4, 5, 6, 7. Okay? And this is one thing that you have to do sometimes manually or you can, I mean print the whole columns and then check the indexes. But I mean if you can look at the data and figure out that like in three or four seconds it will be, it will be faster, right? So now I know what is the C S B containing what kind of information. So let me go to the solve part. So remember these two I'm going to ask you here questions guys, what are these two things doing and what are they? These two part import os, import csb, who can help me with that?

S2 Speaker 2 ▷ 9 25

Those are Python modules that we use OS for kind of like, you know, define our path and CVS for anything we're gonna do for our CVS files, such as read and write.

S1 Speaker 1 ▷ 9 38

Perfect, yes, Maryam, thank you very much for that. And like Maryam said, yes, this actually we're gonna open the path, but remember that this OS means O operative systems okay? As well is gonna give us the key to use this pack, join in in Linux, in Mac, in Windows, in whatever operative system we're gonna work. Okay? Even in Android or iOS or whatever. Okay? So thank you very much for that. And who, tell me what the next, the line number four is going here. Okay, what about

S3 Speaker 3 ▷ 10 22

The gel CSV file? You're referencing that CSV file.

S1 Speaker 1 ▷ 10 28

Thank you. Yeah, yeah. Yes. It's actually giving us the path to that CSV file that we are going to use. As you can see here in my explorer, in my visual studio code, I have this folder, I have three folders, right? I have several folders, but in my folder zero one ft U serial cleaner, I have three folders. One folder is resources and I have two, two more folders, one that is solved and another that I can call. So in order for me to reach

there, what I have to do is actually I am going to close this one for example, okay? I am going to run this call and you are going to see because maybe you still having problems with that. So I run the code and goes directly to the part that I am working on, right? And it runs and everything.

S1 Speaker 1 ▷ 11 21

It's, it's good. So let me, but what happens if I go for example, CD I, I go one, one back and then I run it again. It's not finding that resources or this document. Okay? So what I have to do is I need to go to that folder, right? And then enter, and in this folder I can run it so it find another way to do it guys, that it's really quickly as well. If you write this, then you say open in an integrate terminal, it's gonna open the whole path in the integrate terminal and you're ready to just write it. Okay? Question too far? No. Perfect. Okay, so the next block of code from seven to eight, who can help me explain in this part? Anyone? Okay, what about Sundar? Sundar N? Yeah.

S4 Speaker 4 ▷ 12 54

So what we're doing here is we're getting ready to read the CSV file or the path that we've just directed to. And the second line is providing how to read the file. And in this case, each row is separated by commas. And so we're indicating that the delimiter, which is the separator in this case is the comma. So essentially, like you've commented there, we're opening and reading the C csv.

S1 Speaker 1 ▷ 13 22

Thank you. Yes, thank you very much for that. Yes, exactly, exactly. And one of the of the things that I want you to, to mention here as well is that look I am using with open functional Python, what I am going to open my file path, okay? And I am naming this as CSB file because I don't want to use this whole thing, okay? You're gonna see this kind of no scripting as well in sql, okay? But in this case, I want to do it like this, then I am defining a new variable CSB reader, it's equal to and I am using my module csb, what kind of function I am going to do with that module. A reader, like maybe tos, right? So what I am going to read my CSB file that

is de by a account. Perfect. So now the next part, the next part is actually reading the header.

S1 Speaker 1 ▷ 14 24

The header that is the roof first. So I am going to skip this part if there is no header, but in this case there is a header and I don't want to skip it. Why not? Because if not, I am going to bring this first column that doesn't contain values that I am looking in my, that I am looking for in my csb. That is why I'm going to use something, a new variable. I'm gonna create a new variable that it's called CSB header. This is equal to next that it's function that I can use with my, with my readers, with my width open, right? And with my width open my next indicate that go, go to the next line. Okay? So it's in indicating that the first line of my CSV file is Heather. That's why I can print print Heather of this variable. Okay? And I can print

S5 Speaker 5 ▷ 15 28

Google. I have a question on line seven. When you put S C S V file, is that a variable that you're setting? Because you don't have the S C S V file anywhere, so you just setting a new variable or that means that it actually needs to go reading by line?

S1 Speaker 1 ▷ 15 45

That means that I am renaming this whole part. Okay.

S5 Speaker 5 ▷ 15 51

Oh, okay.

S1 Speaker 1 ▷ 15 52

I don't want to use this whole part. Like for example, if I don't do this, I am removing it, right? Just like that it's gonna say, so what is CSV file? Okay, so I have to put this whole thing in here. In here.

S5 Speaker 5 ▷ 16 12

Oh, I see.

S1 Speaker 1 ▷ 16 14

Okay.

S5 Speaker 5 ▷ 16 15

Okay.

S1 Speaker 1 ▷ 16 15

And, and I have to clear out the errors and I have to work more. So, right. That's why I am renaming that part and I am using those. This you can say that as a new bio.

S5 Speaker 5 ▷ 16 28

Okay, thank you.

S1 Speaker 1 ▷ 16 30

No problem. So the last part is actually to read through each of the rope. Sorry, sorry, I

S6 Speaker 6 ▷ 16 38

Have a, I have a question.

S1 Speaker 1 ▷ 16 39

Yes, Mimi.

S6 Speaker 6 ▷ 16 40

Okay, so on line 11 its says CSB underscore header equal next file. So the next means go to the next row,

S1 Speaker 1 ▷ 16 54

The next means yes, go to the next row after you took the first one.

S6 Speaker 6 ▷ 17 01

Ok. Ok.

S1 Speaker 1 ▷ 17 02

So that's cause

S6 Speaker 6 ▷ 17 03

Next to me looks like skip the header and go to the one next to it after.

S1 Speaker 1 ▷ 17 07

Oh no, the next works for that specific reason. When you have any headers, you go to the next item of the iterator that you're doing in order to bring that as a header. Okay?

S6 Speaker 6 ▷ 17 22

Okay, thank you.

S1 Speaker 1 ▷ 17 24

No problem. Any other questions? Okay, so one thing that you have to remember is that when you want to bring the heather, you, you, you will use next. Okay, so now we're gonna look through my CSV for row in CSB reader, that is this variable. And then I am going to put the condition up, float, converting everything into a float because sometimes you can get some decimals row. That is my I, that is my K, that is my X index seven. I am accessing to the index seven. So in my csv, like I showed you before, I'm going to count from zero to seven to bring that number. And then I'm going to use a, a condition here. If this condition is met, then you're gonna print the rules and that's it. That's how we print the rules. Questions so far here,

S7 Speaker 7 ▷ 18 33

Can you just real quick show me how, how you run that so it doesn't throw in air one more time?

S1 Speaker 1 ▷ 18 39

Yes, no, no problem. So basically, well the safest way is that you go here where you have your activity, where you have your resources, where you have like all the document, all the documents that you're gonna use right click and then open in an integrate terminal and it's gonna open exactly where you want to run that part as you can see it here and then you use, you use run it and that's it. It's gonna, it's gonna take everything. Okay? The serial bonus solve basically is to read this one that does not have any headers. So in this case I don't care about the the the next, because I am not using any headers, I am going to do exactly the same thing, but without using my next block of code and that I can print it as well. Okay.

S4 Speaker 4 ▷ 19 37

Hey Hugo?

S1 Speaker 1 ▷ 19 39

Yes.

S4 Speaker 4 ▷ 19 40

For the CSV file that has a header and you have the command to print the header, can you scroll up in terminal to i'd? I'd like to see what you got.

S1 Speaker 1 ▷ 19 54

Oh, here.

S4 Speaker 4 ▷ 19 56

Okay, so maybe I did something wrong. Yeah, I get a weird printed line, but I can, I can sort that out later. I just wanted to see what it turned out to be. Thanks.

S1 Speaker 1 ▷ 20 06

Okay, no problem. Any other questions here guys? Okay, perfect. Perfect, perfect, perfect. So great. So we're gonna jump to our first topic of today's class and let me get, let me get the stop sharing, I'm gonna share my screen again. We're gonna talk about dictionaries guide. Can you see my screen? Thumbs up please? If you can. Perfect. Okay, great. Let me just put this one here and hide this part out. So repeat questions. All good dictionaries. This is something that we are gonna be using a lot in Python. It's a data type as you can see right here, and we can define it in different ways in two ways actually with this word that says, or with quarterly bracket. Okay? As you can see here, this blue quarterly bracket, that's how you define a dictionary and a dictionary in programing language. It's an object. Okay?

S1 Speaker 1 ▷ 21 32

When we talk about object programming, that means that one object can contain several data. Okay? So basically it, this object can store a collection of data. Each object that you create, it has to, has its own structure, okay? Dictionaries has its own structure that we're gonna talk about, okay? But the dictionary, it's an, it's an object programming

languages. And in Python specifically, it's an object that stores a collection of data. So it is like list and people like. The main difference here is that in dictionaries you will have two things, remember about the structure. Well, you can store key and value pairs, okay? The key is going to be always a string, okay? And that string refers to a specific value, as you can see here is this is my key, it's alive, true is my data type, this is my key. Everything is gonna be string points and I have an integer lives and integer. Okay? So PackMan, this PackMan is an object, it's gonna be equal to curly brackets. And then remember the structure, key value, key value, key value. Coma. Coma, no coma, the at your last element, this is important guys. And one thing as well here is that dictionaries can take any kind of data. Okay? So this is one example of a physical dictionary.

S1 Speaker 1 ▷ 23 45

You consider actually word equal a key definition, equal value, and for example, python, none, constriction, snake, any of variable large constriction, snakes, especially any of the large por snake. And well, the whole definition of that snake. Okay? So now to initialize a dictionary, to initialize a dictionary, we have two ways of doing it. One, you define the name of your variable and then equal two, and then you put click curly brackets and you are telling Python that you are going to do a dictionary inside those curly brackets. The other way is to use the word okay, function as well in Python. So you define the name of the dictionary equal to book, and here inside you can define your dictionary as well, okay? You can add any, any values to your dictionary, like I mentioned with a column. And then placing the desired barrier, well after the column, and then you're gonna refer a value within a dictionary.

S1 Speaker 1 ▷ 25 10

We, I mean, when you want to refer it, just call it, okay? For example here, actors equal and then they open quarterly brackets here, this is my key. That is the name. This is my value, Tom Cruise. And then if I want to print, print my name, I just gonna put print F and everything here. I don't have to put the, the currently brackets for a specific variable or anything like that. I will, I'm gonna put everything in in, in quote quotation, calling this dictionary and then my square bracket calling my name. So here I

am going to print some cruise, okay? Because I am calling a key of my diction.

S1 Speaker 1 ▷ 25 58

So basically you can add things into your dictionary as well. When you actually put actors, then you refer to the key and then you add a string. Important part, you can add different items to that key, as long as makes sense, okay? Because a name that is an integer, it's weird, okay? And for this part, let me open my code because it's more about taking the code. Yes, this code and we can ask questions because it's better to see it in, okay? This one, this one, this one I am going to open. Can you see my screen? Yes. Perfect.

S1 Speaker 1 ▷ 27 04

Like I mentioned, we can create dictionaries to call actor's name. This is the first part, right? I'm going to run it, you can see it. Remove this, I cannot, another element like this, I am, I am calling my dictionary name and I am calling the key that I, I want to add something and then I add my value, okay? I can print the actors. So in this case, I am going to print to Cruise and Denzel Washington because I add Denzel Washington. Again, the, I can print only the name of the actors. In this case I only have the name, I don't have any other keys. So it's gonna print exactly the same thing. Okay? Then for example, I have a list actors here, right? I have a list as you can hear. As you can see here are square brackets. So my list of actors contain one with a coma, two, another coma, and three another coma and four actors. Okay? If I print actors name equal my actor list, what I am going to put is actually I am going to add all these lists inside my key name. Okay? Is that clear guys? This is going to override the values that I have in Washington with the list of actors. Okay? So basically here I am going to print this name. Please leave if I wanna print. Yes, Denise,

S5 Speaker 5 ▷ 28 49

Go ahead. No, go. I was gonna ask you if that is going to rewrite your dictionary. Yes. How, despite the note, because you, you have an actor's list, it's different than the name that you give to actors. How does Python

know that it's going to replace Dental Washington for some Cruz and Angelina, Kristen and all of them?

S1 Speaker 1 ▷ 29 15

Good question. Actually. When you are working with Python and when you're working with objects, one thing to remember is that the cold is going to be reading always from left and right top to bottom. Okay? So whatever I put before one block, if for example, if I put this before my Denzel Washington is gonna read first this one and it's gonna populate my dictionary with the actual list after. If I put Denzel Washington is gonna replace it by Denzel Washington because I am reading it from the top to bottom, left to right. Okay. And another thing here that it's important is that actually I Python, well in dictionaries, Python knows what to do with the element that you are adding, right? How well did, that's why it's really intuitive because it's something that they did inside the code, embedded in the code, okay? I dunno if that answers your question, please.

S5 Speaker 5 ▷ 30 29

Yeah, sort of, or I was ask also wondering if you can do actually append on that like Tom Cruises and Angelina and Kristen and then append those to the list, the, the list and then, or if you can actually remove

S1 Speaker 1 ▷ 30 48

Yes, we we're gonna, we're gonna see that, we're gonna see how to deal with information that is inside dictionaries. But yes, we're gonna, we're gonna reach there Denise, no problem.

S5 Speaker 5 ▷ 30 59

Oh, thank you.

S1 Speaker 1 ▷ 31 01

No problem. Okay, so in that part I can actually print the first actor how I, I just add my, my dictionary because I am acting, I am accessing my actors dictionary. Then in order for me to access to that key, I add my key in square brackets and then actually I add the index because inside my dictionary I have indexes as well. So I add my index. In this case I'm

gonna print the actual index zero zero, that is in my dictionary. Okay, then this is what I was going, that I wanted to, to reach on a dictionary can contain multiple pairs of information. Okay? Remember here I just work with names, but I can create a new dictionary equal to currently brackets notation structure, key value. Key value, okay. With different keys. This key is name, this key is is Henry, this key is actually nationality. Okay? And I have different data in each of my keys. Okay, is this clear Guys, any questions about this so far? No. All good question. Yes.

S8 Speaker 8 ▷ 32 39

Do we have to use F strings when we're running out like values dictionary?

S1 Speaker 1 ▷ 32 48

Do you have to use what? Sorry?

S8 Speaker 8 ▷ 32 50

Do we have to use F string in particular?

S1 Speaker 1 ▷ 32 53

No, not really. No. Oh, okay. F string is something that we are gonna use quite often because makes our life easier instead of doing concatenations and that kind of things. But if you don't like it for some reason, that's okay.

S5 Speaker 5 ▷ 33 11

Hugo, I have a question early. You mentioned that it's two ways to create dictionaries, one with curly brackets and the other one with a deck name before the parentheses. So when you adding all those different objects on your, if you scroll down a little bit on your, like the multiple information, like ne name, genre, nationality you're using currently brackets. So what is the difference between creating a simple dictionary just with the names versus with the currently brackets, brackets versus the information, the three more items that you're adding to that dictionary from one actor? So just basically from one item.

S1 Speaker 1 ▷ 34 00

Okay, lemme check if I understood your question. Basically what's the difference of just using name for example, in the, in my first dictionary that I create that I just use name and then a value and the difference.

S5 Speaker 5 ▷ 34 15

So basically the, you said there's two ways to make greater dictionary. One with the curly brackets and line four and then in line in line five, and then in line eight you have another one that is actors equals dick and then the parentheses. So that's the two ways to create dictionaries. Now when you adding, when you're creating in the dictionary, for example the name, the gender, nationality, those three data points are for one after only, and you using currently brackets,

S1 Speaker 1 ▷ 34 53

Right? Yes, yes, yes.

S5 Speaker 5 ▷ 34 55

So, so coming back to line five, like creating a dictionary, actors with the currently brackets and a list of of different actors, what is the difference there? Because you're using basically the same actors and correctly brackets for both

S1 Speaker 1 ▷ 35 17

In this one and in this one you mean what's the difference between

S5 Speaker 5 ▷ 35 20

No, scroll down,

S1 Speaker 1 ▷ 35 22

Scroll down and

S5 Speaker 5 ▷ 35 22

Then I mean up, sorry,

S1 Speaker 1 ▷ 35 24

Up, up, up, up. Yeah.

S5 Speaker 5 ▷ 35 27

Yeah. When you have name JA and nationality.

S1 Speaker 1 ▷ 35 31

Yeah, what's the difference between this one and which one, which other

S5 Speaker 5 ▷ 35 36

And the dictionary that you're creating. So, okay, maybe let me rephrase this. So line 40 is a dictionary, is that right?

S1 Speaker 1 ▷ 35 45

That is correct.

S5 Speaker 5 ▷ 35 47

Okay. So you're creating that dictionary with three items for one from from one actor. So basically you can put in that dictionary, you can put three actors with the three definitions, name, genre, and nationality. Is

S1 Speaker 1 ▷ 36 02

That right? Yes. But you have to the final list, right? Because it's key value. If I want to put more actors here, I do have to put a list here. I mean like this.

S5 Speaker 5 ▷ 36 13

Okay, that's what I was, that's what I was trying to get.

S1 Speaker 1 ▷ 36 16

And then you put another actor here, right? Oh,

S5 Speaker 5 ▷ 36 21

I

S1 Speaker 1 ▷ 36 21

See. And you have to close the list. Got it. And then here is telling you, hey, what's going on here? I put that, I put a coma. Okay. And here, in order for them to relate, I have to put another list. Okay.

S5 Speaker 5 ▷ 36 36

And then you have to also define a different genre there.

S1 Speaker 1 ▷ 36 41

Exactly.

S5 Speaker 5 ▷ 36 43

Oh, I see. Okay, got it.

S1 Speaker 1 ▷ 36 45

That was your question. It doesn't

S9 Speaker 9 ▷ 36 47

Mean that Angelina Jolie is within the actress and within the United States there's like different sets.

S1 Speaker 1 ▷ 36 54

Yeah. Okay. Exactly. Thank you. Yeah, that's clear.

S5 Speaker 5 ▷ 37 00

Yep.

S1 Speaker 1 ▷ 37 01

Thank you. Okay, perfect.

S1 Speaker 10 ▷ 37 04

Quick question. When you're putting in the square brackets, Angelina Jolie, coma, Brad Pitt, the bottom line genre action and then horror like you just borrowed,

S1 Speaker 1 ▷ 37 16

Okay,

S1 Speaker 10 ▷ 37 18

You're saying that for Angelina Jo, the genre is action and for Brad Pitt the genre is horror?

S1 Speaker 1 ▷ 37 26

Yes. I can relate to that later one

S1 Speaker 10 ▷ 37 27

To another. So if you want to put for example, action for Brad Pitt, you would have to write action again.

S1 Speaker 1 ▷ 37 37

Yes, exactly. If I want to relate them, yes. Okay, thank you.

S3 Speaker 3 ▷ 37 45

I don't get that. But how are you relating it? I thought Angelina Jolie, the genre will both, will be both action and ho, isn't that right?

S1 Speaker 1 ▷ 37 56

Well, you can relate them. You can relate them. You can relate. The whole thing about the dictionary is you can hold pair values. Okay. You can hold key and you can have values. Just give me one second to turn on the light.

S1 Speaker 1 ▷ 38 24

Yeah, sorry, I just turned the light on. So the whole thing about here is that I can hold key values. Okay? Don't, don't get lost on this part, guys. Yes, I can relate. I can relate. You we're gonna see how to create a dictionary that actually relate Angelina with action, Brad pit with action or, or, but the, the point here, like the main point here is, and it's really good that you are thinking ahead of it because we are gonna see it and maybe you're gonna have those questions. But the main point here is that this name whiskey can contain multiple values. Values exactly. A list of values, strings, integer, bullions, doubles. It can contain as well timestamp, it can contain whatever data types we want to, it can contains even other dictionaries, right? Right. It was gonna reach there, but yes, like the, the main point. Yeah.

S3 Speaker 3 ▷ 39 28

Okay. So basically it can contain other dictionaries and at the same time, if you were to put multiple values for one key, then you have to put them in a list. Is that correct?

S1 Speaker 1 ▷ 39 41

Yes. For example, in this example, another actor and creating another actor, right? This alone, because this is another object, okay? And the, in this object, actually I put more keys. Yes. H Mary Best movie for example. Here

S3 Speaker 3 ▷ 39 57

You have multiple of them.

S1 Speaker 1 ▷ 39 59

Multiple, exactly. And I am putting a a list inside my key. Okay, got it. And I can access feed with my index with my key values. Okay? For example, the example that I told you that even you can put another dictionary in site, a dictionary. So basically I am defining this is my value inter inter seller. And then the revenues. In that revenues, I want to put three more keys. United States, China and United Kingdom with their integer values. I close my dictionary and in order for me to access to that revenues, I can put three, a film tied. Two, I am calling my inter made, then I am calling him again, my dictionary revenues and I am calling. Here I am. I can't call an index because what I have here is a dictionary. It's

S3 Speaker 3 ▷ 41 00

A dictionary.

S1 Speaker 1 ▷ 41 01

So I call a key, right in my dictionary. Yes. I can call my indexes zero. I know that I'm, I'm gonna bring this one Rocky. But here I want to bring this, this one I am going to call the key. Okay, let me run it. And you're gonna see the first one is Tom Cruise, but then my name Denzel Washington is replaced. Remember that? I told you. Then I print again, just remove this one I printed again from Silver Salon was in Rocky Inre made 350 million in the us Okay?

S5 Speaker 5 ▷ 41 52

So when you have multiple dictionaries in one dictionary, you only use quarterly brackets, is that right?

S1 Speaker 1 ▷ 41 59

That is correct. You only use quarterly brackets on the body. Okay.

S5 Speaker 5 ▷ 42 03

And when you print, you have the film. So if you put film is the name of the dictionary, is that right?

S1 Speaker 1 ▷ 42 15

That's correct.

S5 Speaker 5 ▷ 42 16

Do you have to call the film every time that you wanna print every like every item of the dictionary because Yes. Okay. Even if it's just one dictionary inside

S1 Speaker 1 ▷ 42 28

Number? That's correct. Yes, correct. Because we are here, we are finishing to print this one. Okay. But I want to print something else in my dictionary so I, I have to call my dictionary again.

S5 Speaker 5 ▷ 42 40

Okay. Every time that you, you always have to call it for whatever you wanna print.

S1 Speaker 1 ▷ 42 45

Exactly.

S5 Speaker 5 ▷ 42 46

Okay, got it. So if you have like film two, we'll do film two and that will print what is inside that dictionary,

S1 Speaker 1 ▷ 42 55

Inside that object, that dictionary. Yes. Got

S5 Speaker 5 ▷ 42 58

It, got it. Thank you.

S1 Speaker 1 ▷ 43 00

Okay, no problem. Any more questions guys?

S1 Speaker 12 ▷ 43 04

Yes. Hugo, once we can we apply the concept, concept of dictionaries to access the CSV file. Like the keys can be the header of the pile and values can be the cell values of the CSV file. Can we do that?

S1 Speaker 1 ▷ 43 22

Exactly? We can do that actually. We're gonna be doing that a lot.

S1 Speaker 12 ▷ 43 27

Okay.

S1 Speaker 1 ▷ 43 28

The keys are going to be like my column names and my values are going to be the roles that I have in the, in the CSP file. Correct? Okay.

S1 Speaker 12 ▷ 43 41

Yeah. Thank you.

S1 Speaker 1 ▷ 43 43

No problem. Any other questions guys? Is there

S1 Speaker 13 ▷ 43 46

A reason why you put revenues first and then United States or can you switch 'em?

S1 Speaker 1 ▷ 43 52

No, because you have to access in the dictionary as a structure. But you have to access like that you, if I put it, if you switch them, it's not going to find where to find United States because I have two elements. I have two keys in my dictionary. Makes sense. And revenue.

S1 Speaker 13 ▷ 44 10

It's like multiple pockets and you have to go through

S1 Speaker 1 ▷ 44 13

First,

S1 Speaker 13 ▷ 44 13

One second, then the third pocket. Okay, that

S1 Speaker 1 ▷ 44 15

Makes sense. You have to drill down, right? Yeah. Any other questions? Good questions guys. Really interesting questions. Thank you for that.

Okay, perfect. Perfect guys. So in that case, let's work in our first activity of today's class is this one s u Hobby book dictionaries. Let me put, read me. This one is gonna be again, thank you for opening the breakout rooms. We're gonna open four breakout rooms. I dunno, we don't have the 40 today, right? Hundreds. I dunno if it's, but we have a sub today, right? Let check

S1 Speaker 1 ▷ 45 13

Anyone is studying today. Ta maybe. No. So well we just have an, we have Andrew Bill and today, okay, so you go to GitLab three activities and activities, zero three, that's where we at right now. In this activity you will create an access dictionary that are based on your own hobbies. Create a dictionary to store the following information you see name is a key, A key, a list of few of your hobbies. It's a key dictionary that include includes a few days and the time you typically wake up on those days. Figure out this one, okay? And then you're gonna print all what is telling you for this activity. We have 10 minutes guys, no, 15 minutes guys. Sorry. So this no 10 minutes. 10 minutes for this one. This is a quick one, so please start working on this one. We have the record rooms open until 27 packs. We're gonna start checking the solution please, if you have questions, just let us know.

S1 Speaker 1 ▷ 55 41

Perfect guys, I'm going to call the people for the record rooms. They can join us. Okay, so let me share my screen. Who finished this activity guide? Yes, and up. You did perfect. Oh great, great. Most of you finishing the activities. Great guys, congratulations on this. Okay, so basically this activity was just for you to actually go and play around a little bit with dictionaries. So here you define your dictionary, my info equals, then you define the names of your dictionary, I mean the keys of your dictionary that you want to put the instructions were pretty clear on that. So you put the name of patient a hobbies in the hobbies, we define a list as you can see here. And then in wake up I define addit. Okay, in the ary you have to open quarter brackets, close quarter bracket. So as you can see here in my last dictionary, I open quarterly bracket and close quarterly brackets.

S1 Speaker 1 ▷ 57 05

That's why I can see the issue. Quarterly bracket one. Once again, remember Python is an indentation program. Sensitive. So indentation, moderately. Okay, I am going to print my results. Print Hello. I am accessing to my dictionary and getting my name. I am reckless and I am accessing to my dictionary occupation. Okay, then I will have to print, I have, and then I want to get an integer here. So I want to count all the elements that I have in my list with key name hobbies. That's why I put length, my info. Hobbies is gonna count 0, 1, 2, 3. Okay, have three hobbies on the weekends I get up at my dictionary name, wake up because I access to this key. And on Saturdays on the weekends I get up at 10.

S1 Speaker 1 ▷ 58 10

Okay? Questions about these guys? Okay, perfect, perfect, perfect guys. Okay, so let's jump to our next topic of list class. That is actually a comprehensive list. That's a cool one. And let me work here. Copy book. We actually read this comprehension list. Let me show you here, here my screen. Cause I think it's better when I, when you see the code. So yes, perfect comprehension. Okay. Okay. Please let me know when you can see my visual studio code. Guys, thumbs up if you can see maybe for the studio code. Perfect. So you can see my great, oh, what we're gonna cover. It's a really powerful function in Python or feature of Python that is called this comprehensions. Okay? One thing for you to remind is that we have useful loops to iterate through a list and perform actions for each element. Okay? For example, eh, we want to print individual users paper, right? So here as you can see we have a variable, it's called all fish, okay? This variable is equal to, as you can see here, and we have different letters, right? We are gonna loop through that through each of the letters with letters defining and list there. And then four letters in English in my variable letters.app, pen letters. So you can tell me what I am going to here

S8 Speaker 8 ▷ 1 00 51

Will be h A l iut.

S1 Speaker 1 ▷ 1 00 55

Perfect. Yes. H a l iut. Great, great, great. So the next piece of code that you see, there is a list comprehension that provides a syntex so powerful that it's gonna treat fish like an array first off. And it's gonna turn it into a list of, its constitute constitutes letters. Okay? Basically replacing three lines with just one line, okay? That's the power of comprehension list. Right now, for example, those because we have only have three lines, it doesn't, it don't look like really Whoa, but it is right? How you're gonna do it is the syntax. Basically you start with square bracket letter, okay? You're gonna find this variable letter two times letter four letter image and that's it. You don't have to attend, you don't have to create ant list, you don't have to define a loop or anything like that. So here I am going to print again H A L I P U. Okay?

S1 Speaker 1 ▷ 1 02 24

So we can then use this list to create a new list of capital capitalized letters using the comprehension list and calling upper on each of the letters. Okay? That's what we are gonna do first, how we do it without the comprehension list. Capital letter, define the mtra four lettering, fish capital append letter, this letter function or feature. And in the future in python upper, oh I am going to print my letters, my fish letters, eh, with capital, capital letters. So my comprehension list is gonna be the same, but I am going to call letter here again. First my, my square bracket letter upper, basically whatever I'm doing here, I'm going to put it here before my for loop. And then my for loop is gonna be exactly the same question so far. Perfect. Okay? Then I have another example. In this example I have a need a statement. Okay? I have a list with July temperatures, I have hot days with an empty list and I am going to do an an if statement that is gonna populate my hot day list. Okay? Four temperature in July temperatures, if temperature is bigger than 90, then hot days that happen. Temperature, I'm doing all this. Who can guess how it is going to be my comprehension list before I throw it? Who can take a while? Guess here

S5 Speaker 5 ▷ 1 04 29

You're gonna get only up to 90, the temperatures up to 90.

S1 Speaker 1 ▷ 1 04 35

Yes, correct. That's gonna be my hot case. But I want to write this code in a, in a list comprehension like this, I want to write exactly the same but in a list comprehension

S2 Speaker 2 ▷ 1 04 51

It will be temperature for temperature in July, temperature and then we're gonna add the if

S1 Speaker 1 ▷ 1 04 59

Yes, perfect. Good. So basically I'm gonna put temperature for temperature in July temperature and then I'm gonna add the, the conditions are going to be after my four. Okay? So this is a structure. You're gonna write your I first, your this one, okay? Then you're gonna put your loop and if it's a condition, if you have a statement, you then put your conditional state if it's a feature that it's gonna be you basically, if you're gonna do something with that like this, let her up, then you put it at the beginning. Okay? Question so far guys? All good? All good?

S7 Speaker 7 ▷ 1 05 55

Can you, can you repeat, would you mind repeating that last bit again? Just how like the, the sections of that last bit of code?

S1 Speaker 1 ▷ 1 06 03

Yes, no problem. Basically the sections are going to be like this. Like when you write flu, right In the flu you have these two eyes, these I and these eyes. So when your comprehension is, you will have those to tame eyes here or letter, right? Basically like this equal you go right? And then you want to look through that, right? Four i I am going to use I in name, okay? For I in name point I have to define my, my my list because I want to append something. Okay? New list, new list, pen. And then in a pen I'm going to put my, not finding my new list because I haven't put it it here but I'm gonna put it, it here equal to my MT list. Okay? And then my work that works. Okay, the comprehension list is gonna be like exactly the same comprehension list. Just gonna put it like that equals to I open way my square bracket and I'm gonna use my twice first. This I, oh sorry.

First is I four I in name. Okay and that's it. That's everything that I have to do. If this I for example contains this upper function, then I have to add that same here.

S1 Speaker 1 ▷ 1 08 07

Okay? If I have a condition here, I dunno why it's bigger than three, I have to two doubles and everything is not going to be working. Remember? Indentation matter. Okay then I will have to put this if in the last part as well. Okay? If I here as well be here. Okay, that that was clear Sam? Yep. Perfect. Any other questions? Wait,

S5 Speaker 5 ▷ 1 09 00

So if you print that, because it C list is not in the end list, right? The new list?

S1 Speaker 1 ▷ 1 09 11

No, I'm creating a new list.

S5 Speaker 5 ▷ 1 09 13

So the creating the new list is the one that if your eye values is bigger than three, that's gonna add to the different list. The C list.

S1 Speaker 1 ▷ 1 09 26

Exactly it is creating these list.

S5 Speaker 5 ▷ 1 09 28

Can we see how that Ashley looks Print?

S1 Speaker 1 ▷ 1 09 31

Yes. Let me just print the ones that are working actually. Cause that one I don't want to debug it. So if I click here, you're gonna see the first one is gonna be printing H L I B U T. The second one that is a comprehension list is gonna be printing as well. H A L I B ut this one is printing the upper letters, this one is printing the upper letters as well. And this one is giving us the temperatures ninety two, a hundred and six that are bigger than here.

S5 Speaker 5 ▷ 1 10 11

Also the July. July does the July temperatures is not included on the, actually in the temperatures at the end 92 is 1 0 6. I thought that was a variable that that will be print out.

S1 Speaker 1 ▷ 1 10 27

Yes, it's as you can see it here, I'm printing from here 92 and hundred.

S5 Speaker 5 ▷ 1 10 34

Oh yeah, okay. But I was saying that July temperature is not print and the terminal.

S1 Speaker 1 ▷ 1 10 41

Oh no because I'm not printing it right, I mean like a string. You like temperature. I'm

S5 Speaker 5 ▷ 1 10 46

Not Oh like a string. Okay, got it. No, nevermind, I got it.

S1 Speaker 1 ▷ 1 10 49

Okay, no

S5 Speaker 5 ▷ 1 10 50

Problem. Thank you.

S1 Speaker 1 ▷ 1 10 52

Any other questions? Okay, perfect. So in that case we have then 15 minutes to work in our next activity of this class that it's zero five comprehension list and they will give you some code here. Okay, in this code you have these four loops that you will have to put into a comprehension list. Okay? As you see it here, please read the instructions and we have 15 minutes work in this one. And then afterwards we can, we can review the,

S5 Speaker 5 ▷ 1 11 37

Can we have time on the breakout rooms? Because I kept on like messing the solutions

S1 Speaker 1 ▷ 1 11 44

Be Yes. If you want like one minute before I, I give the solution I close the like breakout rooms so you can come back. Okay? Okay, awesome.

S5 Speaker 5 ▷ 1 11 52

Thank you.

S1 Speaker 1 ▷ 1 11 52

No problem.

S3 Speaker 3 ▷ 1 11 58

Ju I have a question from the previous one, can I share my screen?

S1 Speaker 1 ▷ 1 12 02

Yes. Yeah, yeah, no problem.

S3 Speaker 3 ▷ 1 12 03

Thank you. I know for some reason it's the F state, the print statement that's giving me an error and I can't seem to understand what I've done wrong there. Do you see anything wrong with the print statement here? The line 15,

S1 Speaker 1 ▷ 1 12 24

Check line 15, print. Hello, my name is then I am referring to way of life then I'm getting mean way of line name I need telling you what's the error? Let me see the error. Let me remove this.

S3 Speaker 3 ▷ 1 12 42

It's saying sring unmatched the square brackets. I don't understand what I have done wrong there.

S1 Speaker 1 ▷ 1 12 56

Okay, wait, wait, wait. Let me check something really quick. Name, name is there name is group and patient inside. It's okay. Let me check the error. Can you copy and past your error please?

S3 Speaker 3 ▷ 1 13 42

Sure. It's given me a syntax error, syntax error for some reason, but I can't seem to understand what I'm doing wrong. Everything

S1 Speaker 1 ▷ 1 13 56

Looks, everything looks fine.

S3 Speaker 3 ▷ 1 13 59

Yeah, I,

S1 Speaker 1 ▷ 1 14 00

But let me

S3 Speaker 3 ▷ 1 14 01

Check, let me just stop sharing because I, I can't seem to find the chat.

S1 Speaker 1 ▷ 1 14 06

Okay.

S3 Speaker 3 ▷ 1 14 08

Okay.

S1 Speaker 1 ▷ 1 14 13

Okay. The syntax F string and manage and it says print F something wrong within and something something like really, really simple and I know that we find the answer here. Try to copy and place this, this part just like that. Just that part and change it with your way of life.

S3 Speaker 3 ▷ 1 15 17

Okay? Yeah, let me do that. And also you, I keep seeing this message pop up on my screen that says uncommitted changes and it doesn't let me go and type anything.

S1 Speaker 1 ▷ 1 15 32

This is on, this is on the studio call?

S3 Speaker 3 ▷ 1 15 35

Yeah.

S1 Speaker 1 ▷ 1 15 40

Oh, you will have to share to be honest to check.

S3 Speaker 3 ▷ 1 15 43

Okay, sorry.

S1 Speaker 1 ▷ 1 15 48

Try to copy and paste that one to see if that works. Yeah, yeah,

S3 Speaker 3 ▷ 1 15 52

I will do that. I'm not able to copy and paste, that's my problem. I don't understand. So I keep getting this line. I don't know if it is clear. It says uncom changes for

S1 Speaker 1 ▷ 1 16 03

You five minutes ago. Uncommitted changes working three. Oh well that's for some reason linked to your it you're working a repository already. You can, you can do any addition there.

S3 Speaker 3 ▷ 1 16 22

So now I'm here and when I do enter it is taking me back to the print statement that is not letting me go out. So another thing I don't know, it's not letting me edit.

S1 Speaker 1 ▷ 1 16 37

Okay, is that yes someone,

S1 Speaker 14 ▷ 1 16 44

Have you tried saving it, we'll let you save it.

S3 Speaker 3 ▷ 1 16 46

Yes, I have done the saving. It still says uncommitted changes. I've done command as to save it but for some reason and then I don't know what I do, it resolves automatically, but it's a pain for me to like, if I do that now it's like working. Okay. So now if I were to do what you said and just name this, you should change that, right? So another thing that I have noticed is it was complaining about the quotation marks. I don't know if that is the reason why it hasn't been working, but I can try executing this. Yeah, why is that? Because in the previous class I know that we've used double quotation marks and it worked pretty fine with the F statement. Is it always a single quotation mark? Is that the reason?

S1 Speaker 1 ▷ 1 17 42

To be honest, i I always use a single quotation mark, but I agree with you that sometimes it works with double corporation marks. I don't know the answer why it has to be with some configuration on the Python package that you are using. Oh, okay. Actually

S4 Speaker 4 ▷ 1 18 02

I actually why that's happening. It's because when you use a double, so if you decided to use the double quotations first to open your string and then you say you go with life square bracket and then you go quotation mark again the way the code reads is it's, it's assuming this string is within those two quotes, which is why we can, it doesn't look, go back and forth between a single to to, it's kind of like how brackets work we, we would use right. Complete bracket and inside a square bracket and and you can, you can invert it and put double quotes on the outside as long as you have single quotations on the inside.

S3 Speaker 3 ▷ 1 18 43

Okay, got it. So either of them have to be different in order for this to work.

S1 Speaker 1 ▷ 1 18 48

Yeah.

S4 Speaker 4 ▷ 1 18 49

That the code can read through and it doesn't, it doesn't think that you're trying to

S3 Speaker 3 ▷ 1 18 53

Right can Yeah. Understood. Thank you so much Avita. I, I don't know if any ta some can help me with this uncommitted changes because every time I type I'm getting stuck and I'm losing time because of this part. If anyone can tell me what I need to do that to fix that, that'll really help.

S1 Speaker 1 ▷ 1 19 15

I will try to close it and open it again. Okay. Just to see if, because you are working in, in a tree, you are working in the repository as you can

see, say it's a legend up there that it says you six seconds ago, two alters you and others working in that book tree.

S3 Speaker 3 ▷ 1 19 36

Yeah. I don't understand who's this other person that's working on it.

S1 Speaker 14 ▷ 1 19 40

You have another instance of visual studio running per chance with this file open because that would make it think there's two people in the file.

S3 Speaker 3 ▷ 1 19 49

Oh, okay. All right. No, I don't, I don't think so, but let me just quick vs code and then come back in again and see if that helps. Yeah. But yeah, let me do that. Thank you so much.

S1 Speaker 1 ▷ 1 20 02

But try to close all the folders, huh?

S3 Speaker 3 ▷ 1 20 04

I'll do that,

S1 Speaker 1 ▷ 1 20 05

Yes. Save it and try to all the Yes, sure, I'll do that. Thank you.

S1 Speaker 10 ▷ 1 20 13

I have a question about the the hobby book.

S1 Speaker 1 ▷ 1 20 17

Yes.

S1 Speaker 10 ▷ 1 20 20

I don't know why it's screening My ho my my hobbies with the square brackets.

S1 Speaker 1 ▷ 1 20 28

Maybe can you share your screen and we can have a look? Yeah, sure.

S1 Speaker 10 ▷ 1 20 39

Can you see it?

S1 Speaker 1 ▷ 1 20 40

Yes, I can see it.

S1 Speaker 10 ▷ 1 20 43

Here you go. So this part still printing with the, the square bracket

S1 Speaker 1 ▷ 1 20 54

And you close the FS statement because I can't see Yep, that is closed. Oh yes, it's closed. Ok, lemme check on that. Thank you. He's printing a list. So it would be in bracket? Yes. Cause it's printing a list will be in bracket. I just, let me pick something here.

S1 Speaker 10 ▷ 1 21 23

It's here. I try putting like a zero like this in printing and it comes like dense, no square brackets.

S1 Speaker 1 ▷ 1 21 36

Yes. That's you're just printing one element. Exactly. You're, you are accessing to just one hobby. Yeah. Yeah,

S1 Speaker 10 ▷ 1 21 42

But if I remove, it's when I print it's printing the hub. But with the square records

S1 Speaker 1 ▷ 1 21 48

That's okay because the, that object is a list so you have to print the list. Alright.

S1 Speaker 10 ▷ 1 21 55

Yeah.

S1 Speaker 1 ▷ 1 29 09

Perfect guys, I'm gonna start giving the solution right now and here is the solution. Can you see my screen? Thumbs up? Yes. Perfect. Okay, great. So let's see here. They ask us to create first the, so they ask us to open this file then to run the provide program. And you're gonna see that game instead of game and you, you will have to use the list comprehend to fix this. Okay. So basically we have this empty list where, where,

where we are going to store our input names here. This is a for loop. I am not using any comprehension list because in this for loop I am telling the machine to actually input selling the user. Sorry to actually input names. Input please enter the name of someone you know and name append my list name. Okay. The first part to fix my problems is to lower case everything that is here.

S1 Speaker 1 ▷ 1 30 29

Okay, so how I am going to do that, I'm going to call name lower or name in names, okay? Remember this is my I, this is my i my variable. So here I am creating now a new list that is not called names is called lower case. That is actually lowering every name that I have here. Okay? Then they tell you to go and inspect this tutorial point, Python string title html. Basically it's a tile case, Python function or feature that help us that whenever you put the title it's gonna take any, any letter that is in is in lowercase and it's gonna put the initial letter in capital in uppercase. Okay? That's the, that's what it does. So basically you put name but title for name in lowercase. Okay? If you have everything in upper letters it it is just gonna ignore it. Okay?

S1 Speaker 1 ▷ 1 31 43

But because we have everything in lowercase, in lowercase, then it's gonna take the initial letter and it's gonna open that one and it's gonna write the other ones in lower. Okay. And then you write your invitations in another list. In another list comprehension that is gonna say invitation and then you open the list F Dear name, it's my eye, please come to the wedding this Saturday for name in this new list, title case. And then you do a look, print your results. So basically you did something like this. I am going to put here random names. Well right, so what it's gonna do is gonna lower this then it's gonna fix the typing of the users. Here goes the raw their chain there. Will the please come to the waiting visa questions about these guys? Okay, perfect. Okay. If

S3 Speaker 3 ▷ 1 33 12

You have them as upper cased, it'll not do anything. Title will not change anything,

S1 Speaker 1 ▷ 1 33 18

It's

S3 Speaker 3 ▷ 1 33 18

Just, it'll not make the remaining letters after the first letter. Small. Okay.

S1 Speaker 14 ▷ 1 33 23

I think it actually does cause I made a mistake and it, I just commented out lower and it lowered the other letters.

S3 Speaker 3 ▷ 1 33 30

Oh it did? Okay.

S5 Speaker 5 ▷ 1 33 33

Wait, is that for the first letter, Jeff?

S1 Speaker 14 ▷ 1 33 38

It capitalized the first and then lowered every letter after that for each word. Like if I did capital letter B, capital O, capital bob, it did, it lowered the O and the B when I ran it, which

S5 Speaker 5 ▷ 1 33 52

Surprised me. Oh, I see.

S1 Speaker 1 ▷ 1 33 54

That's weird. Well the the definition, the actually HTML that it's in there, it initial

S3 Speaker 3 ▷ 1 34 00

Letter,

S1 Speaker 1 ▷ 1 34 01

The method ignores ignore it if the letter is already in upper case the metal ignores.

S5 Speaker 5 ▷ 1 34 07

I did that with the metal upper case in the middle of the name and that changed it too. So

S1 Speaker 1 ▷ 1 34 16

Maybe so they have their fix there as well. Yeah.

S5 Speaker 5 ▷ 1 34 23

Okay.

S1 Speaker 1 ▷ 1 34 23

Okay. Any other questions guys? Okay, perfect. So let's jump to the last topic of today's class that is about functions function, function functions. I share my screen. Can you see my screen up? Perfect. Okay, so I'm gonna start showing my screen. So functions, guide functions basically in software engineering we have something that is dry, do not repeat yourself. And it's a principle of software development that we can use functions and modules to avoid repeating code. So question for you guys are any disadvantages to writing code that does the same thing in three different places? What disadvantages? You can think of

S3 Speaker 3 ▷ 1 35 33

Human error, you might change something.

S1 Speaker 1 ▷ 1 35 36

Human error and execution time takes more time. Executing time takes more time. Yes. Well that help us here.

S1 Speaker 15 ▷ 1 35 47

It clutters the code, it makes it harder to read. And if there's an error with the code, that error kind of repeats throughout it.

S1 Speaker 1 ▷ 1 35 57

Exactly Terry. And you will have to fix it everywhere you have it, right? So that's good. Great answers guys. So here and here are your answers. Actually we'll add overhead of fucking duplicate code changes in several places and become un widely. So efficiency is the motivation for the dry matra. Okay? Do not repeat yourself matter. We'll stop this, not that we understood about that part. We're gonna see life examples, okay? And then after that we have quick activity. We, we run through the functions so it's not new for you. Basically a recap, basic definition, how you write

it, right name, then you go an argument, then you put some code here and then you return whatever is in that code here. Okay? Simple function with no parameters, the function, name of the function, and then some code bring high. Then when you print, you call your function with code, it's gonna print high. Okay? Then simple function with one parameter. You find your function, the name of the function, and then you put the parameter message, you're gonna print message. So whatever you put inside that message is inside that show. Whatever parameter you put is a message that is gonna print here. Okay?

S1 Speaker 1 ▷ 1 37 47

One thing that we haven't checked out before is that functions can have more than one parameter. For example, define your function. The name, name of the function is make illa. And then you put protein and you put the copy here. So you add your coat that it's adding a variable. Quesadilla, it's equal. Here is a protein quesadilla with, and then you're gonna print quesadilla. Then you're gonna supply the arguments, make a ca of beef and guacamole, make another cadia of chicken and salt. Okay, you're gonna put here is a beef ca with guacamole. Here is a a chicken cadia with house order is order is important here when supplying argument guys, so basically if you put make illa soar cream and beef, it's gonna say here is a soar cream quesadilla with beef. Okay? You cannot specify the default values for those parameters. Basically you define your function, then you put your protein.

S1 Speaker 1 ▷ 1 38 55

It doesn't contain any default values, but for the topping, contains always so cream and you define it here, quesadilla here is a protein that you're gonna choose quesadilla with topping. When you are gonna make a quesadilla, you use the default topping, you use past one argument, okay? You can have new topping, a new topping because this is just a default, it's not fixed, okay? So you actually can change it for omo again, here, functions can return a value. For example, the the this function, that square number, basically what it does it, it returns the, the square of a number numbers time number. So basically if you put square equal to square two, you can print square because I am calling this function and it's gonna print two things too. So then you can also print the return

value of a function where two and square three, and it's gonna do exactly the same thing with defining a variable and in that variable, defining a function. Okay, I just printed three questions about functions. Guys, let me run this.

S1 Speaker 1 ▷ 1 40 22

Basically we have hi, hello world. First we have the hi here show, then we have hello world. Then we have our first Illa beef illa with guacamole. The second one, chicken illa with alsa. Then we have the, here is a sore cream illa with wheat beef, the order. And then we actually add the chicken que with cream because that's the full value. Even though if I don't have that argument, then I changed that, the full value topping for guacamole. So you can see that it changed. Here is quesadilla with guacamole. Then I'm going to work with the square value. The first one I'm printing two times two, four. And in the, in the second part as well, I'm doing two times two and three times three. So I'm printing 4, 4, 9 questions.

S1 Speaker 1 ▷ 1 41 28

All good. Okay. Perfect guys, we can work in our next activity of today's class is this one. And in this activity, open it In this activity, basically we will write a function to compute the Ari mean average for a list of numbers. Write a function, collaborate that, accept a list of numbers. So basically your argument is gonna, is gonna contain you guess what? And the function average should return the arithmetic mean basically. You can check that in wiki and then you can test your function by calling it with different values and printing the results. Okay? If you go here to the unsolved version, you're gonna find just this. Okay, so actually we have 10 minutes only to work with this one. And after that we can take a break and we can come for the last part of the class. Okay, sounds good. As you want to take the break now? No, you can. Okay, let's, let's do the activities then

S1 Speaker 10 ▷ 1 42 47

We go. Can you, can you please share the website? You were seeing a, like a lot of Python instructions.

S1 Speaker 1 ▷ 1 42 56

Yes, of course. Let me share that

S1 Speaker 10 ▷ 1 42 58

Up here. My tab, I can go over it.

S1 Speaker 1 ▷ 1 43 01

The gi right? Kenny,

S1 Speaker 10 ▷ 1 43 05

You have one Python website open with a lot of,

S1 Speaker 1 ▷ 1 43 11

A lot of information about,

S1 Speaker 10 ▷ 1 43 14

About Python overall.

S1 Speaker 1 ▷ 1 43 16

Oh, the one that it's calling about the this one?

S1 Speaker 10 ▷ 1 43 21

Yes.

S1 Speaker 1 ▷ 1 43 22

Yeah, perfect. Oh, you want me to share that link to you Kevin? I'm gonna share it here.

S1 Speaker 10 ▷ 1 43 48

Thank you. Go.

S1 Speaker 1 ▷ 1 43 49

No problem. In report, I'm gonna put it and Slack. Slack in resources. Yeah. Awesome. Thank you. No problem. And actually I'm gonna open the breakout room story. I didn't open the breakout rooms. Open our rooms and you can join whatever room you want

S1 Speaker 16 ▷ 1 44 14

You go. Can you make me co-host? I'm not a cohost. No.

S1 Speaker 1 ▷ 1 44 19

Yes, we no hold. Let me see if I can, because I'm not, oh, I'm the hosts now. So yes, I can make you co-host Cohot Andrew. Andrew is a cohost now, right?

S3 Speaker 3 ▷ 1 49 49

You either you or if a TA can help me with that. Uncommitted changes on VS code. I finished running the program, I was able to finish the function program, but that uncommitted thing is constantly coming up and I don't know how to spell it.

S1 Speaker 1 ▷ 1 50 06

Yes, yes. Let me, because I think I will have to take time because Bsco connects to Git. We can actually disable that extension in order to make it like clean and you can it through GI Bash only. You are using Windows or Mac, sorry?

S3 Speaker 3 ▷ 1 50 24

Mac.

S1 Speaker 1 ▷ 1 50 25

Mac. That's okay. So you can commit only with terminal for example,

S3 Speaker 3 ▷ 1 50 30

Right? Yeah.

S1 Speaker 1 ▷ 1 50 31

Can unlink that part from Visual Studio code for the moment and maybe as soon as we under,

S3 Speaker 3 ▷ 1 50 38

It's basically unlinking the GI from VS code. Let me try to Google and see if I can find the solution.

S1 Speaker 1 ▷ 1 50 44

Okay. Yes, I, I mean you can. I'm pretty sure that you can disable the extension.

S3 Speaker 3 ▷ 1 50 48

Okay. All right. I'll try that. Thank you.

S1 Speaker 1 ▷ 1 50 53

Perfect guys, I think we have like yes. One minute left. Let me hold people from breakout rooms.

S1 Speaker 1 ▷ 1 52 22

Perfect guys. Perfect guys, let me show you the solution and then we can take a break. This activity, that's why they give you the week page because it was more thinking about the automatic part than the coding part. But sometimes you face that kind of situations when coding, to be really honest on my experience, I deal more with data problems than are medical problems. Okay? Data problems means like I have a bunch of data that I, I need to figure out how to handle, how to handle new values, how to handle like clean strings, concatenate, different columns, joints, that kind of things. As a data analyst, data scientist, okay? There are some arithmetic parts that you can deal with. Features that are in silo inside Python, in a statistic with non Python. We're gonna talk about that. But it is good to to understand this sometimes for technical interviews I face some of these problems, right?

S1 Speaker 1 ▷ 1 53 45

Basically they're asking you to do a function that returns are ame average for a list of numbers. So here my function name is average and is getting one a argument. In this case, my argument should be a list. Okay? So I am going to find two variables. One is my link that is gonna be counting the elements that I have in my list, okay? And the other one is gonna be my total. The total is a counter. Again, thats why I am doing a double. If the user needs to put a double, I am dealing with doubles here. That's why I am defining 0.0, not only zero as an user. And I am going to use that to count some every time I repeat my list every time I go through my list. Okay? And because I am handling a list, I need a for look.

S1 Speaker 1 ▷ 1 54 42

Hugo, can I use a comprehension, A list comprehension? Yes you can as well. So four number design in numbers in my list. Total equal number plus total. Okay, so my total is gonna change every time it looks. Remember this cost equal time is the same as doing total equal number plus total. And I'm going, what you are going to return is because I want to get them in my total divided by the length. So it doesn't matter how deep my list is, it doesn't matter if I'm using doubles intes, this should work. Okay, gonna do some tests. Nine and five questions about the, can we not use some? You can use some as well. Yes. Good. Any other questions? Perfect guys. So let's take a 15 minute break and then we can come back for the last part of the class. Okay? Take a quick bath bathroom break and coming back if you still have questions. Okay,

S1 Speaker 17 ▷ 1 56 29

Is there any TAs? I have a problem with the, you know, last activities.

S1 Speaker 16 ▷ 1 56 37

Yeah, go ahead. Yeah,

S1 Speaker 17 ▷ 1 56 41

Let me share my screen first. Yeah,

S1 Speaker 16 ▷ 1 56 43

Yeah,

S1 Speaker 17 ▷ 1 56 46

Yeah. Here's my code for the last things, but he got wrong. How do you say that? It's a wrong things here. Actually for the first I do in the total as the sum of the numbers and I want to use the totals with the, and the totals divide by land numbers, but I don't know why it's got wrong things here and I cannot, not bug, but I cannot run anything. I'll perform the terminal.

S1 Speaker 16 ▷ 1 57 21

Can you move those things out? Something's blocking my, my view of your, the top lines of your code.

S1 Speaker 18 ▷ 1 57 28

Also, can you hit play? You, you're in the debugger so it's kind of like stuck there. Oh yeah. I'm just wondering if if it's, has it just finished yet because it's not giving an error or anything.

S1 Speaker 17 ▷ 1 57 40

Yeah, he got nothing here.

S1 Speaker 18 ▷ 1 57 42

Yeah, but hit hit the play button. We can't really see that cuz there's something blocking, but like up at the top of the screen there should be a little play, like a little continue button in the middle. You can try that. So it, it does work. It's just, it just got stopped by the debugger

S1 Speaker 17 ▷ 1 58 02

But it's didn't have anything here with a debar. Cause like

S1 Speaker 16 ▷ 1 58 09

That orange, that orange triangle thing to the left of number five. Try clicking on that. No,

S1 Speaker 18 ▷ 1 58 20

Just, yeah,

S1 Speaker 17 ▷ 1 58 23

I don't know how, how do you see that Costa? I

S1 Speaker 18 ▷ 1 58 31

Yeah, remove the break point and why? I'm curious, why is it not shown the whole screen? It's kinda weird. It's kind of blocking out the little control panel for the debugger. I think that's his, but you just use the run the normal run button and you don't run it in a debugger, it should just run through and give you your five.

S1 Speaker 17 ▷ 1 58 57

Yeah, he gave the five, but it's kind of, I don't know why they have like a run since on this two two line

S1 Speaker 18 ▷ 1 59 06

There, there isn't necessarily anything wrong with that. You're just, you've clicked on a line where the total variable is, so it's referencing the total variable I believe like hit stop on the debugger, the little red square to the right on the, in middle of the screen and the top

S1 Speaker 17 ▷ 1 59 27

Where

S1 Speaker 18 ▷ 1 59 28

Middle of the screen top the little stop button, the red square

S1 Speaker 17 ▷ 1 59 34

Here,

S1 Speaker 18 ▷ 1 59 35

Right? Something keeps popping up. It's, it's kind of weird, but you see the red square, little red, red square and you, yeah, right there. Yeah.

S1 Speaker 17 ▷ 1 59 49

Oh, here.

S1 Speaker 18 ▷ 1 59 51

Okay. Float and range. Oh numbers. Okay. So yeah, so it is stopping right there. Numbers is a list.

S1 Speaker 17 ▷ 2 00 06

Yeah, I define numbers as,

S1 Speaker 18 ▷ 2 00 08

Oh, nevermind, nevermind, nevermind, nevermind. I just misread it. And what are you passing that into this? Down at the bottom. Just hit stop.

S1 Speaker 17 ▷ 2 00 25

Stop. Okay. Yeah,

S1 Speaker 18 ▷ 2 00 28

Just run it with the run button over to the right. Stop. Let's not use the debugger right now. I just wanna see if what if it runs? Not, not that one.

S1 Speaker 17 ▷ 2 00 38

Okay, then which one?

S1 Speaker 18 ▷ 2 00 40

No, no, it's fine. It's fine. It'd be, I can't really see the whole screen, so it's kind of hard to tell what you're passing into that

S1 Speaker 17 ▷ 2 00 49

Here. Like is that better?

S1 Speaker 18 ▷ 2 00 54

Does anyone else see those gray blocks? Is there, okay, yeah, I see them. So let's scroll down. It's, it's, it's, it's doing the first one. Okay. Okay, so scroll down. I, I see the problem on line 14. You're putting a range into a list. So range is a list on itself. So you, on line 14, if you take out those brackets

S1 Speaker 17 ▷ 2 01 22

Bracket

S1 Speaker 18 ▷ 2 01 24

Yeah, the, the brackets and just past the range and it should treat it as a a list. So try try running that it stop and try running that. Yeah, just, just run it however, however you want. So the problem was that you had a range inside a list. So it, it, it was going through a list and it was trying to add that list to a number. And they're not the same type. The range is a list on its own. You wanna go through the range and hit the numbers inside the range rather than having it in a list and then trying to add the range itself, the list to a, to a number. You can't, you can't do that. So,

S1 Speaker 17 ▷ 2 02 16

Okay. Got it. Got it. Thank you. Oh, thank you. I got it. Yeah,

S7 Speaker 7 ▷ 2 02 28

I got a, I got a potentially easy one if somebody's got just a minute.

S1 Speaker 18 ▷ 2 02 34

Yeah. What, what do you have?

S7 Speaker 7 ▷ 2 02 36

Lemme share my screen real quick. I can't get this to run, it won't, it keeps throwing me an error and it's right here. I, I got, I mean I don't understand.

S1 Speaker 18 ▷ 2 02 49

No ahead. Yeah, let's see, let's take a look. So hobby is the length of the hobby. Got the, that's fine. The F string have name curly bracket style I have. Okay, so you see where it says hobby y n

S7 Speaker 7 ▷ 2 03 14

Yeah,

S1 Speaker 18 ▷ 2 03 15

You've closed the quotes right there. Rather than using the curly braces, if you close your quotes, you have to concatenate with that little plus sign. So if you put that in curly braces it should work. Ok. Cause FST strings, you know they expect the variable is in curly braces. Yeah. And then when you closed it, there was no plus sign telling it that you wanted to add that new variable in. It also probably wasn't. Yeah. Now this is a different one UNH type list.

S3 Speaker 3 ▷ 2 03 55

No, it should be me of hobbies. The in the ly braces, the second cly braces. That should be you, you're calling you a dictionary, right?

S1 Speaker 18 ▷ 2 04 04

Well he already on line eight, he, he set the, he set it into a variable.

S3 Speaker 3 ▷ 2 04 09

Oh, okay, I didn't see that.

S1 Speaker 18 ▷ 2 04 11

Okay, so scroll up. Okay, just scroll. Okay, this is fine. Let's, let's take a look. So me of hobbies. So in that situation online, there's these early braces there online eight, he's getting the length, not the actual Yeah, but which is fine getting the length's fine, but the, the me of hobbies is

inside a curly braces putting it back into another dictionary. So those curly braces really aren't supposed to be there if it was like

S7 Speaker 7 ▷ 2 04 51

That makes sense. Love it. Thank you. I there you go.

S3 Speaker 3 ▷ 2 05 04

Can someone help me with the kit extension please? On my VS code I keep getting uncommitted changes and I don't understand how to navigate beyond that.

S1 Speaker 18 ▷ 2 05 17

Well I would suggest is, is is if it for the GitLab material, if it asks you to open it as a Git repository, you just say no, we're not gonna at this point really be doing too much with VS code and managing your git repositories until maybe down the road. So you probably clicked yes and now it's treating it like a repository and it's, you've made changes. It wants to know if you want to commit them and push them and all that stuff.

S3 Speaker 3 ▷ 2 05 47

Okay,

S1 Speaker 18 ▷ 2 05 48

We're gonna be looking at get here in a second. Okay. We might look out how it works in VS code, but I would just say no and not really open it as a repository until we learn how to u we really learn how to do that anyway. You're not gonna be pushing changes to the GitLab website. You're not gonna be committing, pushing, adding changes.

S3 Speaker 3 ▷ 2 06 09

I agree. How do I disable the extension though?

S1 Speaker 18 ▷ 2 06 12

Share your screen.

S3 Speaker 3 ▷ 2 06 14

Okay, just a second. Because once it's gives me the uncommitted changes, I can't seem to proceed beyond that. So I

S1 Speaker 18 ▷ 2 06 35

So go to the, you have this tab that's the little source control tab. It's the three three circles with the 57 on the left, right? Yeah. Of the yeah click there and then go up to the top or not the top. It's one of those, yeah, open up more actions and one second, let me look it up. Can I take control? Your screen real

S3 Speaker 3 ▷ 2 07 58

Should be, those are one in the top. I don't understand. Is this the one?

S1 Speaker 18 ▷ 2 08 12

I don't wanna do anything too, too messy right now. Give me a sec moment. I gotta look it up.

S3 Speaker 3 ▷ 2 08 21

Okay.

S1 Speaker 18 ▷ 2 08 22

I've never undone a repository in vs code. Mm because you, you we're not committing and changing things in the GitLab so there's really no reason to have those you, I mean you should still be able to run it right?

S3 Speaker 3 ▷ 2 08 40

I'm able to run it but sometimes it says uncommitted changes and it doesn't let me type.

S1 Speaker 18 ▷ 2 08 46

Ah yes

S3 Speaker 3 ▷ 2 08 48

That is my problem. So when I'm trying to write the code fast, I lose time trying to like figure out how do I get to typing again in my code in vias code. So that is the main issue.

S1 Speaker 1 ▷ 2 09 03

I was thinking Jaya, did you install this with this course or you had it before

S3 Speaker 3 ▷ 2 09 14

The extensions you mean?

S1 Speaker 1 ▷ 2 09 16

No, the whole visual studio code.

S3 Speaker 3 ▷ 2 09 21

So I did have a visual studio code before and I think Anaconda also installed another version of it. So that is also kind of messing with me because when I open vs code sometimes it either opens up, it opens up the one that I have on my system. But if I go to Anaconda and launch it, then this one opens up.

S1 Speaker 1 ▷ 2 09 46

And can you try to work with the one that you have in your system instead of this one?

S3 Speaker 3 ▷ 2 09 53

Okay, you want me to try doing that? All right, I can try that out.

S1 Speaker 1 ▷ 2 09 58

Yes. Just to see if that works.

S3 Speaker 3 ▷ 2 10 04

So it, when I googled it, it said on your project you right click it and then go to extensions and then disable it. But I didn't understand what the project name here is and where do I right click and what do I do? That was where I lost it and I didn't understand what to do beyond that. Ok, I was trying to see if there was a video as to how to disable that extension. Haven't come across it yet.

S1 Speaker 1 ▷ 2 10 36

Okay. When you go to the to the project name, just go there

S1 Speaker 11 ▷ 2 11 14

In

S1 Speaker 1 ▷ 2 11 18

The explorer of your files,

S3 Speaker 3 ▷ 2 11 22

Right?

S1 Speaker 1 ▷ 2 11 24

Go there then go to I guess three that is the top of the hierarchy of the folder and then right click there. Not there in three,

S3 Speaker 3 ▷ 2 11 42

No it's not giving me any option two

S1 Speaker 1 ▷ 2 11 45

But below one of the folders. Yes. Like for example, that one

S3 Speaker 3 ▷ 2 11 56

Didn't gimme an option here to do anything to the extension Joe.

S1 Speaker 1 ▷ 2 12 05

Yeah, let me try to find that as well.

S3 Speaker 3 ▷ 2 12 09

I don't understand this GI keep, I am not sure if I should be doing anything with it. It only gives me the same options where it's say either open it or delete it. I didn't wanna delete it and create further problems myself so I left it alone. This one is not

S1 Speaker 1 ▷ 2 12 29

Let's, I think we can get, let's know, I think we can check that in the office hours. Definitely. Yeah. Yeah, I can help you stations there.

S3 Speaker 3 ▷ 2 12 43

Yeah, it takes you, oh

S1 Speaker 1 ▷ 2 12 44

Yeah, you were there. You were there already.

S3 Speaker 3 ▷ 2 12 46

Yeah, I, I was trying to experiment and see what I need to do, but I didn't understand if, so I did, I think for Git lands I probably went and said disable it but that didn't seem to fix it. So what I could probably do is I realized that for the changes to take if I, I need to close and open vs code again. So I will try doing that.

S1 Speaker 1 ▷ 2 13 12

Yeah

S3 Speaker 3 ▷ 2 13 13

And let's hope that it takes effect and it doesn't give me that problem again, but I'll try that out.

S1 Speaker 1 ▷ 2 13 18

Okay. Yeah.

S3 Speaker 3 ▷ 2 14 05

Okay. How do I find my zoom thing again so that I can stop sharing my screen?

S1 Speaker 1 ▷ 2 14 21

No, go for it. Bottom

S1 Speaker 10 ▷ 2 14 25

Part or the top part If you

S1 Speaker 11 ▷ 2 14 33

Where

S1 Speaker 1 ▷ 2 14 35

What you wanna do.

S3 Speaker 3 ▷ 2 14 36

How do I stop sharing?

S1 Speaker 1 ▷ 2 14 38

Oh, click this.

S1 Speaker 1 ▷ 2 14 45

Okay, so we are gonna start the class now. Thank you guys. Thank you. Welcome back to the class. And for this part we have two things that we're gonna do. Okay? The first thing is that we are going to do a group activity. Basically I'm gonna create some breakout rooms here and in this group activity, lemme start sharing. I'm gonna create a random group so you can actually start working with your classmates. Basically we will, we will create a function that search, a list of a student can graduate by a state to determine state graduation rate for public, private, nonprofit, and private for profit.

S1 Speaker 1 ▷ 2 15 43

You're gonna analyze the code and CSV provided looking specifically for what needs to still be added to the application. Okay? You're gonna use a starter code and you're gonna create a function called printed percentage, which takes in a parameter code, it states data and does the following. Use the data storage within the state date to calculate the estimate graduation rate in each category of title that you're gonna get in the E S B. And you're gonna print out the graduation rate for each school type for the state to the term. Okay? The there you have a bonus, you have some references there. And I was going to actually put it here, run it. Oh, I need to fix this. Basically this is how the code looks. Yeah, I fix this part. What state do you want to look for? And that's it. It goes because it didn't find any texts. Okay. But lemme check if I can run one with the ion dates.

S1 Speaker 18 ▷ 2 17 57

It's looking for a uppercase Texas, it's not lowering it.

S1 Speaker 1 ▷ 2 18 00

Alright, perfect. So let me run it and then

S1 Speaker 18 ▷ 2 18 34

Well just, the first TE is first letter's upper cased.

S1 Speaker 1 ▷ 2 18 38

Thank you. And that need to give you, I need to give you all these, okay. For some reason

S5 Speaker 5 ▷ 2 18 52

Mine is not giving me that. Well the salt part, the one that is salt, but I'm not sure I put the estate and then it, it just doesn't respond.

S1 Speaker 1 ▷ 2 19 07

Okay. In that part, I mean you can check it out with your, with your group. I am actually going to send you to the group that you have the soft version there, you have the soft version, you can go through the call, you have 15 minutes for this activity and then we can go all together. Okay? Okay. Sounds good. So let me open the breakout rooms right now. I'm going to recreate randomly, I'm gonna put like four to five participants including TAs as well. You are co-host so you can move around if you want and if you have a question, just raise your hand and I can go to your rooms. Okay? Perfect. Please start working this activity guys and see you in the main room in 15 minutes.

S1 Speaker 19 ▷ 2 35 56

Would you like me to join any breakout room?

S1 Speaker 1 ▷ 2 36 04

No, don't worry. We actually gonna finish the activity now.

S1 Speaker 19 ▷ 2 36 07

Yeah, because the breakout room you assigned me there was already under there.

S1 Speaker 1 ▷ 2 36 14

Oh. Oh you are the TA right here? Yes I

S1 Speaker 19 ▷ 2 36 18

Am.

S1 Speaker 1 ▷ 2 36 19

Oh perfect. Here. Oh, are you a co-host or not? Hmm? Are you a co-host?

S1 Speaker 19 ▷ 2 36 26

Nope.

S1 Speaker 1 ▷ 2 36 27

Let me make you a co-host now. Yeah, perfect. Here. Yeah, sorry I didn't introduce you. I, I didn't know who was it.

S1 Speaker 19 ▷ 2 36 38

No worries.

S1 Speaker 1 ▷ 2 36 40

At the ending office hours we can, we can send you one of the report. Okay,

S1 Speaker 19 ▷ 2 36 45

Thank you.

S1 Speaker 1 ▷ 2 36 48

Okay, so I'm calling everyone back.

S1 Speaker 1 ▷ 2 37 24

Okay, perfect guys, we're back and I'm gonna start sharing the solution. Actually you have the solution already there so I'm just gonna discuss it really quick. If you have questions about this solution specifically, we can check it in in office hours because I'm gonna go through that in order for me to go through GI and you have time to go through gi, right? So basically he here we are gonna do exactly, can you see my screen first off, thumb off? Yeah. Okay, we're gonna have like the whole thing here. We're gonna read a file that, it's called graduation data sb. And we're gonna find this function that basically is taking different variables, right? State variable, public, student, public. So it's collecting all the information from this from any file, any CSB that I'm reading in this case the graduation date, csb, okay? Not only that, he is doing some totals like total students total graduated.

S1 Speaker 1 ▷ 2 38 27

It's getting this public grade rate that is division between public graduated divided by public students times a hundred in order to get the public grade rate and as well is getting me some conditionals. If nonprofits students equal, equal zero, then the nonprofit grade rate is

gonna be equal zero. Okay? This is to handle that else. If it's different, then do this ARI Medic eh, operation. Okay, again, if for profit student equal equals zero for profit grad, graduated, zero else, we're gonna do this operation. Everything is in the same function as you can see here. Invitation matters. So the next part is that we're gonna talk about the overall grade rate. That it's again another operation and I am going to handle that with another conditional. If overall grade is bigger than 50, then send this message else. Send this message, okay? And now I'm gonna do my printing within my op, within my functions, okay?

S1 Speaker 1 ▷ 2 39 37

Printing, state printing, public grade, nonprofit rate for-profit, grade rate, overall rate and rate and math. I'm going to read in the CSB file that has a graduation CSVs, I am going to split the data on the er. I am going to get Heathers and I am going to actually input what state you want to look for. And I'm going through this file with for row SB reader. And the whole thing is that actually I am calling my state my functions here that is expecting one parameter and I am passing one parameter. Okay? Like I, like I say I'm gonna, I'm gonna jump to keep because it's important and we can actually do it, it's quite fast explanation, but we can actually check an overview on that. You can do some testing and then basically we can actually jump to the last activity that is for you to create new files.

S1 Speaker 1 ▷ 2 40 49

Okay? So intro to gi, you see my screen comes up. Perfect. So intro to GI guys, we have been using GI as a drop, as a Dropbox, as a drive, Google drive, storage. But why the main purpose of gi, you may be asked this before. Well we're gonna see some deeper capabilities on GI and we're gonna, we're gonna see why it's so important to use GI and why all companies, well not all, but most of the companies are using gi. Okay? So GI essentially is a way for us to keep track of our work overtime, whenever we get another piece of a project working, we can save that change with kids. As you can see here, we have seen overtime we have version A, B, C, then maybe you change version B and now you just have A and T in version three you just have tea and it's saving all of your versions. Okay?

S1 Speaker 1 ▷ 2 41 57

So basically when you save something in it, it's called commit. Okay? And it represent that big point of our project where we save and describe our work. We can see this as a initial committee and then we have the master branches or the main code and we have different branches, okay? As you can, you can see here, we break something we will working on our code while working on, on our code, sorry, it is going to allow us to restore working code from the part that wasn't that that is not broke, okay? So if remembers checkins it is really important and keep version of power code. Oh this is a scenario your group has been working with Uber, it Ubers rider data that the main whole problem, main, main, whole code. And you have decide to analyze the average age of the riders, right? So the root cause for the product is called main, where you are working with Ubers rider.

S1 Speaker 1 ▷ 2 43 01

Data in our G is a remote repository and that's like your main version, right? Version one station area. You're gonna hear this in the companies. The staging area is where you edit the files. That will be part of the next commit of the next main version. You create your phone from the existing pository and you download it to your local company. After you do this, basically you're gonna GI is gonna allow us to write this code and save it with the name that you decide in this case each analysis that is at the branch of my main branch of my main pository, okay? And I am going to do some anything in my screen area and I can commit my changes in my version in my local machine.

S1 Speaker 1 ▷ 2 43 56

After I think everything is done and my age analysis can be part of my main code, then I do my me, okay? My MER is going to my code in the main version, okay? The area again is where you edit, did your files. So basically we're gonna, I'm gonna show you how to do a GI clone that maybe you know by now, get out, get commit, get, push and get pulled. Okay? We have GI status, we have GI checkout, we have GI branches, we have a lot of GI commands. But for that, I mean we will have to create another course. Of course it's not that difficult. You can go ahead and

check git documentation to find everything about it. Okay? A committing GitHub is like a snapshot of what your project or file looks like at a particular moment in time. If a file doesn't contain any changes, the file is not stored again, instead GI provides a link to the, to identical file that is previously stored. Okay? It's like a snapshot basically. So questions about GI before I show you the example, okay, pretty good. So me show you, I have this repository that I create, I show you how to create a repository. So I'm going open gi, gi, I'm go, can you see my GI badge or no? No. Okay, let me stop sharing and share my whole screen.

S1 Speaker 1 ▷ 2 45 44

Can you hang now my GI badge? Yeah. Okay, so this is my main, my main folder where I am going do my first gift and then I'm going here to my GI repository. I am going to copy this, this file, okay? Copy page here. And I am cloning my repository. It's closed so I can go to my folders here. I know where is it? So I go to Windows C users. What is user, okay? Oh yes, you go. And then I'm going here. Yes. Test. Okay, point test. I have these, these documents right here. Just like because I am working here with my data, I'm going to open another one and I am going to actually go to lesson plan, my Python file class one. And I am going to get my first, my, my variable, my Python variable.

S1 Speaker 1 ▷ 2 46 57

Okay? I am going to copy this here and I am going to get another file that I found here, the hello world and I am copy that. Hello world sir. Okay, so I have these two files. If I go to my visual studio code, I am going to close all these folders. Don't say just close everything, okay? And I am opening a new folder. Open a new folder. In this folder I am going to that specific folder where I have my test, okay? Where my test here select this and now I can see for example, yes, I trust, yes I trust my variables. Okay? So I'm working with my variables, everything looks fine, I run it, everything great. Okay? So now that everything is, is working, now I can do something. I can navigate to that repository saying the the users. Now it's not this one. So let me check where I am.

S1 Speaker 1 ▷ 2 48 27

Yes, I'm in now users then Ramo and then test, right? I'm in test and you can see that I'm in main, okay? In main, I click Ls and I can see whatever I have here. If I go to my repository here, okay, in my test part, I just see the Excel playground read me, but I don't see my Python, right? So what I will have to do here is first just to see that I have two files that they're ready to commit, but I haven't had those ones. So I'm gonna put give, okay and appoint here in order to add everything I can specify to add in a specific pile to add a specific folder or whatever. But I want to add everyth, okay? So it's adding everything. So here I go here and I go and I go check here I am gonna see GI commit and GI commit. Basically what you are going to see is how to commit your gi, okay? Commit your kit. You will have to, I dunno can using my screen because it is showing me an error now. No, I share my screen Now can you see my screen now using my my GI badge, right? Yeah. Yes. So basically I'm going to put after my GI commit M. That means it's a message, okay? GI commit and putting a message. Message. Can you

S3 Speaker 3 ▷ 2 50 24

Make your GI patch smaller? The window can't see what you're typing, you go

S1 Speaker 1 ▷ 2 50 30

A smaller window or bigger.

S3 Speaker 3 ▷ 2 50 33

Okay, so the zoom controls are covering what you're type typing. For me at least. Yeah, now it's better.

S1 Speaker 1 ▷ 2 50 41

Yeah, okay, adding new pythons, new python, and that's it, right? Keep commit, adding new Python. So if you don't have any problems, perfect, good, good. I will add to file 52 inserts because those are my lines. And then I will put get push for reaching for reaching master main. It's gonna push it and just everything is, is apparently, let me show you now I was the other screen. Let me share my, so if I go here and then I, you can see my screen, right? My repository if I refresh it and voila, I have my pythons here. Okay, that's a good push. So if I do anything to these files,

someone else create another branch and they start working in that branch and then they put something else. For example, I can actually in variables here, let me do something. I'm gonna edit my code here, right? And I am adding a new variable variable, these these things, I'm gonna do it without the, the help of GI patch. So you can see

S1 Speaker 11 ▷ 2 52 38

Variable,

S1 Speaker 1 ▷ 2 52 42

Right? So I'm gonna put commit changes, not doing it in my machine. Okay? And it's saying commit a message. Yes, update variable, I like it. Commit changes, okay, it's here. But if I go to my visual studio code, you can see my visual studio code, right? Yes. Okay. I don't see that new variable, right? I don't see those changes. So what I have to do, I need to go to this main repository and I need to put, and you think it already that that's the magic of kit, right? So basically that's it guys is we're gonna further in the course, we're gonna actually work with branches and, and how to merge a branch. GI Mer, you just git merge and then you just put the name of the branch and how to work with your main branch. One of the things as well, it's gonna work a lot when you don't want to step each other's work. So when you work in teams, each one of you can work in one branch and, and the main main branch is gonna be the one that you can all merge. Okay? So these five minutes left guys, what we are gonna do, actually questions? Sorry?

S1 Speaker 12 ▷ 2 54 24

Can you show an example of staging?

S1 Speaker 1 ▷ 2 54 28

Mm, staging exactly. It was where I was editing. When I was editing the code, I was in the staging area. So basically it's, you can be in a staging area here in Taiki or here in your local code. Okay? So it's whenever you're editing your code, you're in the staging area, okay? In the staging area, okay?

S1 Speaker 12 ▷ 2 54 55

Dating the code, okay.

S1 Speaker 1 ▷ 2 54 56

Yeah, thank you. Perfect, no problem. Oh please, if you can use this four minutes for you to do what I did, create a test folder or the folder that you have and then put some files on it. Do a GI clone, a GI A get commit and then get push and then maybe just a gi, okay, this is instructions using the repo that we just create or your own repo, of course make add the following changes as a new line of code pipe and create a new folder. Add file to the newly created folder. Add, commit and push the changes and delete the new folder. Then you can commit an app. If you, if you delete a file, you just put app and then you can push it and it's gonna delete it. Okay? Even though if it's saying app, okay, so for example, I'm going to my folder and then here I am going to close all my folders. So I don't have any problems. And here I don't like this anymore, so I'm gonna delete this. Okay? So I'm gonna go here, I am going to do keep the status to see, and you see I delete this so that the change that it can happen. So I'm gonna actually do with that again with my point of everything. I can do this right? Delete.

S3 Speaker 3 ▷ 2 56 54

So this GI commit dash uhm, is that just a comment or what are you trying, what are you doing here?

S1 Speaker 1 ▷ 2 57 02

GI commit remember is a safe that say basically the M is a message only. Okay, but GI it's important because it's saving your repository. Okay?

S3 Speaker 3 ▷ 2 57 15

Okay,

S1 Speaker 1 ▷ 2 57 16

So you save it and then I just do like it push, okay? And then I can go to my repository and here in my repository I can go here in my test and that's, you see I don't have my other pie. Pie, I delete it.

S3 Speaker 3 ▷ 2 57 39

So why do a GI ad?

S1 Speaker 1 ▷ 2 57 42

It's just, is it de the file? It's just the structure of the git comments that you're gonna be using.

S3 Speaker 3 ▷ 2 57 49

So even if I delete a file, I have to do a get ad and then git comment.

S1 Speaker 1 ▷ 2 57 53

Yes. So basically the big product here is get, get status just to check if you have any changes, get up, get commit, get push. Okay. Okay, thank you. And whenever you have changes, get pulled to always renew whatever information is in your repository because remember in your company or in the, in the industry, you, you are not working alone. So people is doing changes in, in the repository. So you just have to get the most update repository. Ok,

S1 Speaker 12 ▷ 2 58 25

Got it. Thank you.

S1 Speaker 1 ▷ 2 58 26

Any other questions guys? Basically it was done and gi I mean you can, you can play around with this, create a test folder like I did and play around. But yes. Any other question before we go?

S1 Speaker 12 ▷ 2 58 44

One quick question. Yes? Can we use the terminal, which is inbuilt in the VS code to do get clone and GI pull push?

S1 Speaker 1 ▷ 2 58 52

Yes, yes. You, you, you, you can, you just need to add the extension of here, but you, you, I prefer to do it separately to be honest because sometimes I get some problems in BS code. I think the Git extension is this one Git extension pack. That's why I don't have it installed. But you can do and, and here actually you can see the branches, you can see like, to be honest, I don't use it a lot because I do everything on GI batch and whenever I want to see my changes and everything, I use this graphical user.

S1 Speaker 12 ▷ 2 59 31

Okay, thank you.

S1 Speaker 1 ▷ 2 59 32

You can see here the story, the codes that you change. Whatever you delete you can see like a you request for for example. So yeah. Any other questions guys?

S2 Speaker 20 ▷ 2 59 51

Yes, one silly question. So when you go to get bash, you type in get, and then you get the suggestions, how do you do that?

S1 Speaker 1 ▷ 3 00 03

You type the gi and then what?

S2 Speaker 20 ▷ 3 00 05

And then you are getting all the suggestion instead of typing, get commit or push or pull. Yeah,

S1 Speaker 1 ▷ 3 00 11

Because I type it already, you use the up and then arrow in order to go the commands that you already use.

S2 Speaker 20 ▷ 3 00 18

Oh, so if if it's used, then only it gives you the suggestions.

S1 Speaker 1 ▷ 3 00 24

Take the command so you can type it faster.

S2 Speaker 20 ▷ 3 00 26

Okay, got it. Thank you.

S1 Speaker 1 ▷ 3 00 28

No problem. Any other questions before we go?

S2 Speaker 21 ▷ 3 00 35

I have a question.

S1 Speaker 1 ▷ 3 00 36

Yes.

S2 Speaker 21 ▷ 3 00 38

So if two people are colonial a repository and then they make changes at the same time and they want to push also at the same time, what happens? That case,

S1 Speaker 1 ▷ 3 01 00

To be honest, and I'm gonna be, I'm gonna do a while getting here, it's gonna go the one that pushes faster, even if it's like one, I mean both are going to be merging to the main branch, but it's gonna go the first one that if it's, I dunno, one second faster, it's gonna go that one first and then the other.

S2 Speaker 21 ▷ 3 01 33

But their changes might be different, right?

S1 Speaker 1 ▷ 3 01 37

Yes. But you can have the story of both. So basically, but that's a good question. I mean I know where are you going there? Basically in no companies, when you do a push or a merge, you always have a, a master branch administrator. So that master branch administrator is the one that is at admin all the merge that you are doing. So I haven't seen a company where you can just merge crazy things into the master branch because that is a restricted branch. You can work locally in your branch and then use your changes and then whenever it is ready, someone is reviewing what you're doing and then they are actually the ones that can approve the merge or not. So I don't know if your question was going through that part. I mean,

S2 Speaker 21 ▷ 3 02 49

I still don't know what happens to the changes. Like if I, for example, I make some changes, someone else is making some changes. Our changes are different. So if I want to push, let's say my friend has pushed earlier and I'm pushing to a, to the master, which is not the same as the bra as master branch that I was, that I pulled.

S1 Speaker 1 ▷ 3 03 18

Yes. Is because two concepts are different here. I mean okay. Okay. One is to push and the other one is to merge which one you want to, to check. Because to push, basically you want to push or add any changes you want to push.

S2 Speaker 21 ▷ 3 03 34

But if I push the, the, the, the master will be the same as what I pushed

S1 Speaker 1 ▷ 3 03 41

With any change. Okay. With the new changes. So for example, if you are in your master have a Python document that only contains one part of it and your classmate create another that same pa python, but he has two variables. And in your, in your file you add three variables different from your, from your classmate. Those five variables are going to be reflected on the main file, on the changes of the main file. Okay. But first you're gonna see the two of your classmates, then you're gonna see yours because he pushes first. Okay. So always when you're talking about push is whatever you push, push at the end. Okay. Whatever changes you did in the staging area at the end is the ones that are going to be reflected in the main file.

S2 Speaker 21 ▷ 3 04 43

Okay?

S1 Speaker 1 ▷ 3 04 44

Okay. Okay.

S2 Speaker 21 ▷ 3 04 45

Okay.

S1 Speaker 1 ▷ 3 04 46

And merging is a different story. Okay.

S2 Speaker 21 ▷ 3 04 52

Oh, so what is the difference between merging and push?

S1 Speaker 1 ▷ 3 04 55

Yes, merging, basically you have your complete file, you have another file completely. Another, oh, basically this, you have, you have completely a different file that it's a name that contains everything and you are merging this.

S2 Speaker 21 ▷ 3 05 11

Oh, okay. So you push to, you push to basically to the branch and then merge the master.

S1 Speaker 1 ▷ 3 05 17

Exactly, yes.

S2 Speaker 21 ▷ 3 05 19

Yeah, yeah,

S1 Speaker 1 ▷ 3 05 21

Yeah.

S2 Speaker 21 ▷ 3 05 22

Thank you. Perfect.

S1 Speaker 1 ▷ 3 05 23

Perfect height. Sorry to steal this questions. That's it for today's class. Thank you very much for your attention and see you for tomorrow for talking about libraries with Python. The next topic of today's of of next part, we jump into to week number three. And if you still have questions guys, please stay here. I am going to open the breakup room. I'm stop the recording now.