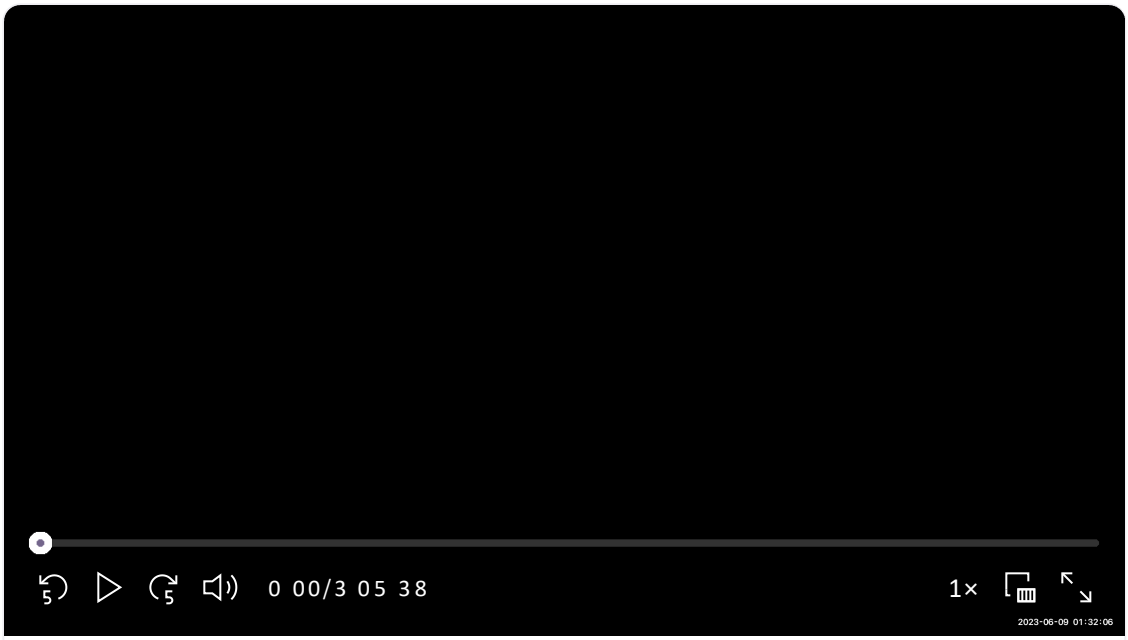
# Day 8 Reading, Writing, and Pyrithmetic




Select text to create highlights Learn more

## Transcript

S1 Speaker 1 0 00

We are gonna review, you have, there's some activities to review loops do as well, some wild loops and, and actually to do some inputs from users and to run it. And, well, we have a phone class today. It's a challenging class to be honest, because we are gonna be reviewing as well how to read files with Python, with the function in Pi in Python, it's, it's not that difficult to, to, to get along, but if it's new, it's gonna be a, it's gonna be a challenge. Okay. Because it's, it's, there are some new concepts there. Okay. And so let's start with today's class. I'm gonna

start sharing my screen and please let me know when you can see my screen.

S1 Speaker 1 ▷ 0 59

Can you see my screen? Yes. Yep. Perfect. Okay. I'm gonna put it here in a slideshow mode and well, basically, yes. Okay, so this class, like I mentioned, I am going to be showing you how to read and write in Python, but we're, we are gonna see some functions, we're gonna see modules, we're gonna see sips on how to put lists together. But we are gonna do some, some activities. Like I told you, this is Python checkup. It's again, a warmup activity. Not in all of the classes. We're gonna have warmup activities, but because it's Python and we are starting with Python, that's why we, we do this often when we start with these classes, right? So, oh, yes, these activities for you guys. We have two minutes to complete this activity. Basically, you have to create a simple Python command line application that does this thing, right?

S1 Speaker 1 ▷ 2 11

Print, hello user, then ask, what is your name? Then respond, hello, user. Then ask what's your favorite number? Oh, sorry for that. And, and well, you will have more about it. I'm gonna end up the, I'm gonna, you're gonna, we're gonna, you're gonna have more about it if you go here to Python day two and the students quick a cup, okay? And you're gonna see the instructions here in GitLab. Okay? So we have 10 minutes to work on these guys. Please start working on this and then we can review together. Okay? Remember, you can run your pipes in git in, in the terminal. You can do your pythons scene in some text editor that you choose. You don't have to have installed BS Visual Studio Code. And today's class, I'm gonna, I'm gonna do it in Visual Studio, but if you need it, I can do it as well in GI Batch, no problem at, okay, so let's work in this. We have 10 minutes to work and then we can review this. Okay? If you have any questions or anything like that, just let me know. And, and we're gonna, we're gonna all remain here in this part of the class, okay? So after that we, we, we, I can open the breakout rooms. So, perfect guys, let's start, I mean, all clear from copy fold clear in that matter. Yeah. Perfect.

S3 Speaker 3 ▷ 3 59

Hey, hug, Hugo, how do you open up a folder and make a file through vs code again?

S1 Speaker 1 ▷ 4 05

Eh, can you show your screen and I can walk you through that process number. Okay. Basically there you have open a folder, open a folder there, then you go where you have your folders, where you're going to be working. Module two. Yeah, I think it's that one. Yeah, Python, I think No

S3 Speaker 3 ▷ 4 37

Activities. And then

S1 Speaker 1 ▷ 4 38

We can just, no, you went to Vva. You went to vva.

S3 Speaker 3 ▷ 4 41

Okay. This is where I wanna be anyways.

S1 Speaker 1 ▷ 4 44

Okay, no problem. Thank you.

S4 Speaker 4 ▷ 4 54

Hi, go. Where do I find the activity?

S1 Speaker 1 ▷ 4 58

The activity will be in GitLab?

S4 Speaker 4 ▷ 5 01

No, I mean I, on Python, on vbs, I mean Visual Studio.

S1 Speaker 1 ▷ 5 08

Oh, in Visual Studio Code. Okay. So basic, you want to share your screen?

S4 Speaker 4 ▷ 5 13

Yeah, I'm, I'm looking for it. Yeah, yeah. Here?

S1 Speaker 1 ▷ 5 16

Yeah. Oh, okay. So it will be Python day two. In day two activities. And in activities. You go to the first activity of today's class, SST U checkup.

S4 Speaker 4 ▷ 5 30

Okay.

S1 Speaker 1 ▷ 5 31

Yeah.

S4 Speaker 4 ▷ 5 33

IPhone. Can I share my screen?

S1 Speaker 1 ▷ 5 43

Yes, please.

S4 Speaker 4 ▷ 5 45

Okay.

S1 Speaker 1 ▷ 5 59

Do you see it? I see it. Okay. Wait, so it's day two, you're in day one. Oh, this is, oh, okay. Here? Yeah. Okay. And first, first one? Yeah, quick check. Okay, that's, thank you. No problem.

S5 Speaker 5 ▷ 6 21

Hug, quick question. Once you take the input of the user, what are you printing out? Again, that response statement is a little unclear and the read me for me,

S1 Speaker 1 ▷ 6 34

Let me check there J because I read it. So once, can you repeat your question?

S5 Speaker 5 ▷ 6 40

So after you take the user's input for the name, what are you responding? What are you giving?

S1 Speaker 1 ▷ 6 46

You respond. Hello? Plus the username.

S5 Speaker 5 ▷ 6 49

Okay. Alright, because on my readme it looks a little different, so I do not understand what that is.

S1 Speaker 1 ▷ 6 54

Oh, okay. Yeah.

S5 Speaker 5 ▷ 6 55

Cool. Thank you.

S1 Speaker 1 ▷ 6 56

No problem.

S5 Speaker 5 ▷ 7 05

And I'm sorry, what? What about the VS code folder? I didn't understand that. Are we supposed to do anything specific there? I'm sorry, I joined a couple minutes late, so I think I missed your instruction.

S1 Speaker 1 ▷ 7 16

No, the, we have just some questions about how to open the, the folder, but no, we don't have to do like anything specific. Specific, yeah.

S5 Speaker 5 ▷ 7 27

Got it. Thank you.

S1 Speaker 1 ▷ 7 32

Hugo, can we use F instead of the plus sign? Totally. Hello. Plus name. Okay. Totally. Thanks. So, perfect guys, we actually have a time up. No, Tori, we have three more minutes to work on this activity and I just wanted to give you a heads up. No, time up, heads up.

S6 Speaker 6 ▷ 10 47

Okay. Can you help me? Can I share my screen?

S1 Speaker 1 ▷ 10 51

Yes, please.

S6 Speaker 6 ▷ 10 55

So when I try to run, it's

S1 Speaker 1 ▷ 11 03

When you try to run, it says you don't have, yes, you don't have the extension fine markdown station then what's that? Hmm? Oh no, you have to open your pythons because you are trying to run in the, you are trying to run the R. So open, no, open the program that you want to run your Python program that you want to run on bs, please.

S6 Speaker 6 ▷ 11 35

I'm not understanding, sorry.

S7 Speaker 7 ▷ 11 37

So go ahead and click the top left corner to access your files at vbs.

S1 Speaker 1 ▷ 11 41

Exactly. Well

S7 Speaker 7 ▷ 11 47

A little bit further down.

S1 Speaker 1 ▷ 11 49

Yes. To the left, down, down, down, down.

S7 Speaker 7 ▷ 11 54

No, so if you, if you click on your vbs Yeah. And then in the top left corner of the actual vbs where you see this? Yeah,

S1 Speaker 1 ▷ 12 02

Here that one. Yeah. And then open it. Oh,

S6 Speaker 6 ▷ 12 07

Oh, okay. Now, okay, I got it. Thank you.

S1 Speaker 1 ▷ 12 09

No,

S1 Speaker 1 ▷ 13 11

Perfect guide. I know that some of you are just starting but time's up of these 10 minutes we're gonna share the, the solution of this exercise. So don't worry, you can check it out there. And I'm gonna, I'm gonna start sharing my screen so you can see that. Okay, let me put it here. So can you see my screen on top? Perfect. Okay, so here they, they say print. Hello user. So this is a basic thing. Print Hello user, remember the hello work. Then you're gonna take your first input. How you do that, you define a variable and then that variable that is equal to input and then whatever input you want to take, right? What is your name? And then you put your name. Then you're gonna respond back with the user's input by printing high plus name in this case variable plus something else that you want to put as a stream. Okay? Then you're gonna take in the user's favorite number again with the favorite number declaration of this variable input. And inside your input is going to be a phrase asking for that favorite number. And then you have these conditions if, and then you have to do it inte here because this is a stream, remember? So into your favorite number is less than seven. Print this, it's equal to seven. Print this Ls, print this. Okay, let me run it.

S1 Speaker 1 ▷ 14 49

Hold on. You see it? What's your name? Hugo. Enter. Hi Hugo, what's your favorite number? Think my favorite number to be honest, I dunno, I never thought about it. 10. Your favorite number is higher than and that's it. That's what the program should be there. Questions

S8 Speaker 8 ▷ 15 16

Hugo go?

S1 Speaker 1 ▷ 15 17

Yes.

S8 Speaker 8 ▷ 15 17

Could we, could we put that in with the input and still be able to run or do we have to do it in the if statement only

S1 Speaker 1 ▷ 15 28

Again? Again. Again. Three which input? Yeah,

S8 Speaker 8 ▷ 15 32

Yeah. So you are writing int with F statements below, but can we put with the input with the favorite number int over there? Yeah.

S1 Speaker 1 ▷ 15 41

Oh yes. Line

S8 Speaker 8 ▷ 15 41

Number 11.

S1 Speaker 1 ▷ 15 43

Yes. Yes you can do that as well.

S8 Speaker 8 ▷ 15 46

It just means that user has to input in the int not string, right?

S1 Speaker 1 ▷ 15 52

Exactly. Exactly. And then you don't have to put this in your conditions.

S8 Speaker 8 ▷ 15 58

Okay? Right. Okay, thank you.

S1 Speaker 1 ▷ 16 00

No problem with that. Yep. Anyone else? Any other questions? Okay, perfect guys. So the next thing that we're, now it's more review. Okay? But in this case we're gonna review simple loops. Okay? I know we saw this last class, but this is an important concept guide because you're gonna be using quite often. So it's quite important for you to to review it, to understand it, and then to practice with it. So as you know for loop move through a given ranger of number. And as you know when you just give one number, let me do do it like this. So who can tell me what is going to print here in the line four with these four x in range? 10

S8 Speaker 8 ▷ 17 07

Zero to nine to nine.

S1 Speaker 1 ▷ 17 09

Yes. Thank you very much for that because we start from zero and we don't take this number. The number that we put here, the la the second one, who can tell me what I'm going to be printing?

S8 Speaker 8 ▷ 17 25

20 to 29.

S1 Speaker 1 ▷ 17 27

Perfect. Yes, because again, when I am doing like this range of numbers, I'm going to take the first one but not the last one. I mean not this one but the last before that one is the one that I'm gonna be printing. Perfect guys. Good. Who can tell me what I am going to print in my next loop?

S9 Speaker 9 ▷ 17 59

All the, no, all the peanut butter text. Basically peanut butter chili time is now.

S1 Speaker 1 ▷ 18 08

Thank you very much. Pad? Yes, that sounds pretty good. Perfect. Yes guys, so who can tell me what is the wild loop and what I am going to be printing here?

S1 Speaker 10 ▷ 18 30

It is just gonna keep printing. We Mary go rounds are great and it's gonna keep asking you if you want to go again and it's gonna do this forever.

S1 Speaker 1 ▷ 18 39

Yes, thank you very much. He's gonna do that while you passing. Yes. Okay. But yes. Good guys give, give yourself a round for plus you survey. Great. I'm gonna run it so you can see that all that you said to me is actually true. And let's see first zero to nine as you can see it here, let me just put this. Four zero to nine then 20 to 29. Then peanut butter, jelly pine is now right? And then the last part, the Y loop is gonna be X equal yes while X equal. Equal, yes. Remember to do an equal equal in loop you have to put to do an equal you have to put equal. Equal, okay, so print we Mary go round or grade X is gonna be equal input. Would you like to go on the merry go round again? And he is waiting for my output.

If I put like this, it's gonna keep going. If I do like this, he is gonna go out because it is waiting exactly that string with the capital letters. Perfect. Guys, any questions about this?

S1 Speaker 11 ▷ 20 06

Oh, go. Can you very quick the the while for me please?

S1 Speaker 1 ▷ 20 10

Yes, no problem. So basically in the while loop it's a look that is gonna be happening until the condition that you state met, okay? While you keep inputting yes, in the machine, I am going to be doing the loop. Okay? So you are not defining the length of the loop like in this ones for word, in word for example. You are not defining that because here words it, it has a defined length, right? You are defining a land here. Now here you are saying wild X that you define up here it's equal. Yes. Do this part. Okay. When you are doing wild, normally you expect an input from the machine because that input is the only one that is gonna break the loop. Okay? So that's why they put X is gonna be equal input. Whatever you put here is gonna convert in the new X and while it is gonna evaluate that this is equal to yes and if it's then I'm gonna print again. If not, I'm going to go out that clear.

S1 Speaker 12 ▷ 21 33

Hugo, if we put an X equal, yes, if we instead of change the variable for Y, it should work, right?

S1 Speaker 1 ▷ 21 40

Totally. Yes, you can put whatever you want to put here. It's string.

S1 Speaker 12 ▷ 21 45

So the X doesn't carry on when once they put in a the loop as a four x in range, it doesn't carry on on the conditionals. Is that right?

S1 Speaker 1 ▷ 21 55

In a y loop? No, that's right.

S1 Speaker 12 ▷ 21 58

Okay, thanks.

S1 Speaker 1 ▷ 22 00

Yeah, no problem. Any other questions? Yes, if

S1 Speaker 13 ▷ 22 04

If, would it work here instead of while? Could you just put F

S1 Speaker 1 ▷ 22 10

In instead of while?

S1 Speaker 13 ▷ 22 14

Yeah,

S1 Speaker 1 ▷ 22 16

Eh, well that will, that wouldn't be a loop, right? That just will be a conditional that it will run one time only. Okay, so that's why you use the while because the Python recognized that as a loop. Okay, that was clear or not?

S1 Speaker 13 ▷ 22 42

Oh yeah. Got it.

S1 Speaker 1 ▷ 22 44

Yep. Okay. Perfect. Any other questions? Perfect guys. Okay, so let's continue for the next activity of today's class. That was a quick review. Thank you guys for starting the recording. Thank you for that great help. And I'm gonna start sharing my screen so you can check where you can get your activity. Please, please, please, please, please, please if you're in the same tone as I am. Okay, so activities, title, day two activities, then after activities you are here, keep in Candice for recap, open this one and you're gonna have the instructions here. Okay? Yes. Can you see instructions there guys?

S1 Speaker 1 ▷ 23 53

Thumbs up please. Are you on the same page as I am? Thumbs up. Denise, you have a question? You have your rights, you have a hand? No, I'm sorry. No. Okay, perfect. Perfect, perfect, perfect. Okay, so great. We have 15 minutes to work in this one guys. So let's start working this

activity of today's class. What about I I want to, I want to show you some, okay, there is a bonus part there, but you can see you can do it or not or you can skip it. But I want to show you how this is going to work.

S1 Speaker 1 ▷ 24 48

Okay? Can you see my screen? Yes. Perfect. Basically when I run my call, it is gonna put me the list of all my candies with the index and then it's gonna ask you which can, would you like to bring home input the number of the candidate you want. Example, I will say I really like guilt and something was wrong here. Oh yeah, no worries. Go back to activities, then I have to do zero three. Yeah. And then I can write five. And then kit in candy store. Kit in candy store. Oh yeah, so list, okay, that's why I'm just trying to run my Python. Don't, don't get like, okay, you, what are you doing? Python hit. Yeah, so you see that it runs now?

S1 Speaker 1 ▷ 26 38

Yes. Thumbs up. Okay, so here you just put Skittles, like I say, and then input the number of of the candy you want. Okay? Another one. There you are. And seven are burst and I want to kick that as well. That's it. So I brought home with me and is gonna give you the names, skills, star, bus, m and ms and KitKat. Okay? That's what the program does, the program needs to do. So please start working on this and if you have any questions, just let me know. We have 15 minutes to work in this activity and then we can check together. Oh, now I am going to open the breakout rooms. Oh, the breakout rooms are open actually, thank you very much for that. But we can actually create one more room cause we are 48. Oh I, I actually bring Bill and Kiki back. Well,

S4 Speaker 4 ▷ 29 02

Hugo, I have a question

S1 Speaker 1 ▷ 29 04

Please.

S4 Speaker 4 ▷ 29 06

Can I share my screen?

S1 Speaker 1 ▷ 29 08

Yes.

S4 Speaker 4 ▷ 29 09

Okay, how do I get out of here?

S1 Speaker 1 ▷ 29 30

Go to file to the top, top, top, top, top, top. More, more, more, more, more. Up, up, up, up, up, up. Oh, here. Yes. File. And then close folder or you go with the shortcut. Yeah, that's it. Open a new folder

S4 Speaker 4 ▷ 29 51

From, from here as well.

S1 Speaker 1 ▷ 29 52

Oh, in the blue Now where you say visual studio code editing above you have new file and open just below new file. It's a folder new file. And then yes, that's a folder one. So you can open that one.

S4 Speaker 4 ▷ 30 08

Okay.

S1 Speaker 1 ▷ 30 11

Or cancel that. Cancel that. I'm going to show you something. Yeah, cancel that. And there, down there you have recent right where actually your folder where you're working is on, so you can click there.

S4 Speaker 4 ▷ 30 28

I I don't see it

S1 Speaker 1 ▷ 30 30

Recent.

S4 Speaker 4 ▷ 30 31

Oh, ok. Ok. So

S1 Speaker 1 ▷ 30 34

No top. Yeah. Yeah, click that and then you can go, well strike where you were working, but if you want to open a new folder, close that folder now.

S4 Speaker 4 ▷ 30 46

Yeah, this is the, this is the first activity.

S1 Speaker 1 ▷ 30 49

Yeah, exactly. You want to open another one, right? Yeah,

S4 Speaker 4 ▷ 30 52

I wanna go back and then open the other one.

S1 Speaker 1 ▷ 30 54

So file,

S4 Speaker 4 ▷ 30 57

Okay,

S1 Speaker 1 ▷ 30 58

File close clo close folder to the bottom.

S4 Speaker 4 ▷ 31 05

Okay.

S1 Speaker 1 ▷ 31 07

And then open in the folder just below. Just below the new file? Yes. Not new file. The one that is down.

S4 Speaker 4 ▷ 31 19

Oh, okay.

S1 Speaker 1 ▷ 31 21

No, up, up, up, up, up, up either, yes. Open folder.

S4 Speaker 4 ▷ 31 27

Okay.

S1 Speaker 1 ▷ 31 28

And then just try to find your,

S4 Speaker 4 ▷ 31 32

And do I have to do these steps every time I close and open a file?

S1 Speaker 1 ▷ 31 38

Yes.

S4 Speaker 4 ▷ 31 39

Okay.

S1 Speaker 1 ▷ 31 40

That's how you do it. Thank you. That's how you do it in Visual Studio.

S4 Speaker 4 ▷ 31 44

Okay, thank you.

S1 Speaker 1 ▷ 31 45

No problem.

S4 Speaker 4 ▷ 33 23

I have a quick question. How do I see the read me file and the unsold file next to each other? Okay, because I can see one at a time. Can I do both next to each other?

S1 Speaker 1 ▷ 33 37

Yes. You just select the file that you want to see next to the other one and just drag it next to the other file on your window.

S4 Speaker 4 ▷ 33 46

Oh, okay. Okay. Thank you.

S1 Speaker 1 ▷ 33 48

No problem.

S1 Speaker 1 ▷ 38 28

So we have five more minutes guys, and then we can check the activity. Okay, perfect guys, thumb up. So let's review the activity together. Please Let me know if you can see my screen. Thumbs up? Yeah. Okay. Perfect, perfect. Okay. And you can hear me right? Right. Okay, well one thing that we are gonna, that we're gonna discuss here in this general quote, I am going to explain it, but there are three, four loops being used in this activity. Okay? One loop is gonna print out the original can list. A

second loop is gonna collect all the candies choices that the user has. And a third loop is actually going to print the final list of choices to the screen. Okay? We're adding candies to the candy card list that we are gonna see it. The selection variable, that is the select variable. This variable, the selection variable has to be cast as an integer.

S1 Speaker 1 ▷ 44 16

Okay? Why? Because as you can see I am getting a string here and I need to append integers in a list. I can append integer, I can append strings, yes. But I have to append by index in the list. So I can relate one list and the other list. Okay, so basically here is it, we have our first list that you have it there in your own soft version with all the candies. This variable is going to let us choose how many times or how many candies the user can select. If you change this, it's gonna be 3, 2, 4, whatever you want to put here. Then I am actually putting an mt r a because here in the candy cart is where I am going to be saving all my candies. Okay? Any candy that I want to save, it's gonna be show there. Okay? Then this is my, for my first loop, I am going to get this list with me.

S1 Speaker 1 ▷ 45 23

So for candy, remember this, this can be, I gave whatever you want to put in cand list. Please have to refer to my cand list. Okay? Print. And I am using the F string here. Stream cand list index. Why the index? Because I want to print the numbers. 1, 0, 1, 2, 3, 4, the indexes of this list. That's why I am using index dot index of what? Of my eye of my candy. Okay. And bring that candy out. Okay, so we are going to put here, as you can see in a squares, in a square bracket, my index and next to my square bracket, I'm going to put the name of the, is that clear? Thumbs up,

S1 Speaker 11 ▷ 46 26

What's the F?

S1 Speaker 1 ▷ 46 28

Okay, perfect. Great. So second loop guys. Second loop is going to collect all the candy choices that the user actually put. Okay? So I am going to print which candy would you like to bring home for X in range

allowance? Allowance is this one. As we know in range we take from zero to four, including five. So from X in range allowance, select and creating a selection variable input. Input the number of candies that you want. So this is going to look as many times as X is going to run. And it's going to be a pending to my candy card. That is an empty list. It's gonna be a pending cand list. Okay, can list, is this by the index integer? Selected by the index. Okay. And the third loop is just to show whatever I just get. So print, I broke home with me for cand and I'm just gonna print my, my list. Candy card, print candy. Okay, questions here guys,

S1 Speaker 12 ▷ 48 05

Can you scroll up a little bit? Your code?

S1 Speaker 1 ▷ 48 09

Good. Nice to hear that. You sure? No questions?

S1 Speaker 12 ▷ 48 15

Yeah, can you scroll a little bit up

S1 Speaker 1 ▷ 48 17

Just to see? Are you talking and I'm not hearing what you are asking.

S1 Speaker 12 ▷ 48 22

Oh, can you hear me?

S1 Speaker 1 ▷ 48 24

Okay, let me check what's going on here because I can't hear you. Oh,

S1 Speaker 12 ▷ 48 33

You can't hear me?

S1 Speaker 1 ▷ 48 35

Can you speak again, Denise, just to see if I can hear you. Yeah,

S1 Speaker 12 ▷ 48 39

I'm, I'm, I'm talking

S1 Speaker 1 ▷ 48 42

Oh and no, I can't hear you. Okay, let me do something here. Here. Oh, what happened here with the light? Okay, lemme check here. Okay, can you speak Denise again?

S1 Speaker 12 ▷ 49 05

Yeah, can you hear me?

S1 Speaker 1 ▷ 49 06

Oh, I can hear you now hear.

S1 Speaker 12 ▷ 49 08

Awesome. Thank you. No, all good. Will you be, so can just let, can I see your, the beginning of your code?

S1 Speaker 1 ▷ 49 18

Yes. No,

S1 Speaker 12 ▷ 49 19

Yeah, after the candy card. Yeah, with the U. So you didn't use the wild loop?

S1 Speaker 1 ▷ 49 27

No, because I'm not doing the bonus one. In the bonus one. You have to use the wild loop. Yeah.

S1 Speaker 12 ▷ 49 33

Oh, so the, okay, got it. So y can you scroll a little bit down? I just wanna make sure cause I kind of get here a little bit late. All right, thanks so much.

S1 Speaker 1 ▷ 49 47

No problem.

S1 Speaker 11 ▷ 49 49

We go, quick question. Can you hear me on the top where you put like F So F what does it mean

S1 Speaker 1 ▷ 50 01

That is the F string? So instead of me changing everything to an an string, an integer or anything like that, I just will put all my variables, like in this case candy for example, that is an integer into the, into my print statement. The print statement needs to be a string in this case because I am con the strings with indexes, as you can see here, zero sneaker one Kit Kat. Okay. So the F string is gonna help me to not put plus and then transform this candy into an SST R into a string. Okay.

S1 Speaker 11 ▷ 50 44

Okay. And the index part means the, the 1, 2, 3, the 0, 1, 2, 3.

S1 Speaker 1 ▷ 50 52

Yeah, exactly.

S1 Speaker 11 ▷ 50 54

Okay, thank you.

S1 Speaker 1 ▷ 50 56

Lugo,

S1 Speaker 12 ▷ 50 56

I have a question. Why do, how come if you didn't specify the index at the beginning? Like zero sneakers, one skid Kat? It's actually showing the index on your terminal

S1 Speaker 1 ▷ 51 10

Because I am specifying it here

S1 Speaker 12 ▷ 51 13

So you don't have to specify it at the beginning. Cuz I did the list and I put the index at the very top in the beginning. You

S1 Speaker 1 ▷ 51 20

Don't? Yes you don't because the user is on, I mean if you do it, it's gonna work because it's a string and is reading another string. But I am actually using indexes because when the user choose an index, 1, 2, 3, it's gonna, the machine is gonna the index and it's gonna relate with the can name. So basically that's why I am, I am doing in,

S1 Speaker 12 ▷ 51 44

Oh, okay, I see. May I see your code at the beginning? Just to see if you keep it and then Oh, you put it on the list. I see. All right, thank you.

S1 Speaker 1 ▷ 51 54

No problem. Let me turn out the light right now. Much better. Perfect guys. So the next part that we're gonna work in is actually sharing this. And let me mm, start sharing my, this one. So can you see my screen?

S1 Speaker 12 ▷ 52 33

Okay, do, are we now gonna go through the bonus for the wall loop?

S1 Speaker 1 ▷ 52 38

Oh yes, yes, yes. Let's go through the bonus. Yes, let me share my screen. Let me open the bonus one. Perfect. So for the bonus, the only thing that we have to do everything is actually the same, the same thing actually here we are converting the string in the eye in order to get the index here. So the bonus is that I am going to define an answer that is yes. So while answer is equal, equal yes, then I am going to print this statement. Which cand would you like to bring home? Then the same input of the user. Then I am going to append to my cand list. Remember on my last code, my cand list is the empty, no sorry, my cand list is this one, right? And then the answer is gonna be input. Would you like to make another selection, yes or no? If the user select yes, I'm going to do exactly the same thing. So as you can see here, my allowance time is not being used here, right? Because I am be looping while the customers wants to keep bringing Candice home. Okay, perfect. Okay, so the next, yes,

S1 Speaker 12 ▷ 54 08

I'm so sorry, I just got a little with the i is the, I means the index. So you don't have to specify the index or it is just a random variable that you, for I,

S1 Speaker 1 ▷ 54 22

Here, actually what I am doing is I am printing right the, this part, the index, by only printing the I, because it's gonna be printing, remember I

is gonna be printing 0, 1, 2, 3, 4, 5 is gonna go through all the list. Okay? So I don't want, I don't need to specify the index because the eye is gonna give me that index.

S1 Speaker 12 ▷ 54 51

So the eye is just, is a reference of the index or it is just a random variable for that not to go through the whole list of

S1 Speaker 1 ▷ 54 59

Exactly the is the, is my variable is this eye? Yeah, I see.

S1 Speaker 12 ▷ 55 05

Okay, got it. Thank you.

S1 Speaker 1 ▷ 55 07

No problem. Okay, so let's go to the next activity. The next activity basically little bit more challenging. This one. This one, it's called the house of ice. I dunno what this but, well, I don't know why I can share my screen. You see my screen? Yes. Yeah. Okay, perfect. Oh, in this activity, house of pies, you will build an order form that display a list of pies, and then prompts user to make a selection. It will continue to prompt for selections until the user decides to end the process. Part one, create an order form that displays a list of pies to the user in the following week. Following way, welcome to the house of pies. Here are the options that one you're gonna have. Okay? Then prompt the user to enter the number of pies through like order after that. So the order with gray, well have that pie name write out for you and then ask, ask if they would like to make another order. If so, repeat the process Once the user is done, purchasing pies in the total number of pies. Order bonus part, modify the application so that at the conclusion of the transaction, the users purchase are listed out with the total count broken by API bean, blah, blah, blah. Questions about these guys?

S1 Speaker 1 ▷ 56 57

Again, you have these instructions on GitLab, so let's please start working on this activity. We have 20 minutes and then we can review. Okay?

S1 Speaker 14 ▷ 1 00 08

Well as TAs are in the breakout rooms if you have any questions.

S1 Speaker 1 ▷ 1 00 14

Yep. Thank you Bill. Thank you Bill. Guys, keep working, keep working on what you're doing. I just want to show you what the code is going to do when you print, when you run it. Okay, let me, so can you see my screen? Thumbs up? Yep. Perfect. Okay, so the code should be doing this when you run it, you're gonna print that list, which will you like? I will say blueberry. Blueberry enter. Great. We will have that blueberry right out for you. Would you like to make another purchase? Yes. Yes. Okay. And I'm gonna get a second one, okay?

S1 Speaker 1 ▷ 1 03 13

And after this is gonna ask me if I want to do another purchase, I'm gonna say no and it's gonna say your purchase, you purchase a total. Okay? That's it with the bundles. The bundles will print the list with the, with the numbers of pies that you, that you bought. Okay? Just wanted to share that part. Four more minutes guys, and we can review the guys up. And I am going actually start presenting the solution. Who finished this activity? Thumbs up. Perfect. Great. Okay, good guys, we have some, some of you that finish it. Congratulations. And let me share my screen. It was a challenging one. And on top, if you can see my screen please. Yeah. Okay, perfect. So here the thing was to create a white loop. This white loop is going to be adding or or running until one condition is met.

S1 Speaker 1 ▷ 1 16 56

In this case is Y. So it just keep, keep buying pie. So we define that y, that it's gonna be shopping equals y. Then I'm going to, to put my empty list here, five purchases. Then I'm going to create my list. These leads is going to contain empires, okay? Of different flavors. So I know that I am going to save my purchases here in my list, my empty list, that is five purchase. Okay? So I'm gonna print or display my initial message. Welcome to the house of pies. Here are our pies. Okay, so I am going to start my loop while shopping is equal, equal i shopping equal, equal I.

Okay, I am going to start printing my list of pies or my flavors. So print the lines, then all my pies, right? And the user is going to see the pie, pie choice input, which would you like? So here I am actually asking to the user to input, which by the user can choose here. So I am going to add any input in here, okay? By purchase pen to buy choice, okay, append to buy choice.

S1 Speaker 1 ▷ 1 18 33

So sorry, in by purchase I am going to append my pie choice, okay? After that I, I have to do a, I have to do the match with the index and the pies because here I am just adding the string. I am just adding the string and it's not making any sense if I put 1, 2, 3, or because I am actually choosing an index. Okay? So in order to do that, I have to use grade. We will have that I list, okay, converting my PI choice, because at the end this PI choice is going to be an integer. So remember, in order, in order for me to work with indexes, I have to put everything as an index. Okay? Integer by choice minus one. Why do you think here is minus one? Any lucky guess?

S1 Speaker 15 ▷ 1 19 37

Because the code is reading from zero and not from

S1 Speaker 1 ▷ 1 19 39

One. Exactly. Adrian, the code is reading from zero. So I'm starting here, if I start with one minus one is gonna be zero. And then my code is going to understand index zero. Okay? So write out for you then I am going to create a variable that says shopping. Because this variable is gonna ask the user if, if he, if the user wants to keep shopping, right? It just, yes, well then I am going to print you purchase a total of string because it's a string you can work with f a string as well. The length of pipe purchases that I have here in these T, okay? The bonus part, the bonus part is quite the same. The only thing is that pie purchases is gonna start with zeros. Okay? Why zeros? Because I am going to be adding to my pies. Okay?

S1 Speaker 1 ▷ 1 20 45

So that's why I start, I don't start with an MP list. I start with zeros, 10 zeros because I have 10 different pipes. I'm going to do exactly the same

thing, okay? With the choice. But here, well the choice in index, I am putting it in a separate variable. Why? Because I'm going to create a counter, okay? Choice index is gonna be my interior pie choice minus one because of my index. And then I am going to add a pie to the pie list by finding the matching index and adding a one to its value here. Basically I'm going to put pie purchase by purchase is this one that contains an array of zeros, eh, choice index. Okay? Remember that I can put like 0 1, 2, 3 and this plus one. This is the same as doing these guys, so don't get confused. Okay? I am actually copy paste this one and is the one, if I do is the same one if I do this, okay, so it's exactly the same one, but we are saving one piece of the goal.

S1 Speaker 1 ▷ 1 22 00

So I'm adding here one every time I select one of the pies, okay? The pie purchase variable is gonna be adding one to the index that is looping in my wild loop, okay? The index that is in this variable and looping in my pie choice, the pie choice that the user is putting. Ok. Then the the the question grade, we'll have that pie list choice index right out of you, right out for you. Again, I'm gonna ask you if you want to keep buying, I'm gonna keep, guess I'm gonna do exactly the same thing and I'm gonna be adding my purchases here. I'm gonna be summing up depending on the position is the pie flavor that I'm gonna be choosing at the end, I have to do a full loop through my full pie list, okay? Index in range length of pie list, range of pie list because I'm gonna get 0 1, 2, 3, 4, 5 until nine account is gonna be my, I transform everything into a string five purchase pie index. Okay? This is going to be my index, basically it's gonna be a number, okay? But that's why I am converting this into a string pie name as well is gonna be my pie list in this case because my pie list contain the names, okay? I purchase contains the integers and the pie list contains the names. So at the end I just put it together and I get this part.

S1 Speaker 1 ▷ 1 23 56

Okay, questions here guys.

S1 Speaker 12 ▷ 1 24 00

I have a quick question. I I didn't catch the CROs for the pie, for the pie purchase. That's so because you already got something in your cart, right? So they already have choose like a pie or or two or three. So why is a pie purchase in this shouldn't be adding up for what they already have put like choose from

S1 Speaker 1 ▷ 1 24 31

Choose, remember that the choose is going to be actually to select the flavor, okay? Is not adding like the number of pies that you select. Okay? So the choose basically is you selecting a pecan, you're selecting an apple crisp, you're selecting again a pecan, you're selecting a blueberry, okay? And these zeros they need with this sum up, they need to increase by index. So as soon as you select one pecan for example, and then another apple crease is gonna be pecan one apple crease one, then you select another pecan. If you select another pecan because EAs in index zero, index zero is gonna increase it, it contains a value that says one. So it's gonna increase plus one. So it's gonna be two. Okay.

S1 Speaker 12 ▷ 1 25 27

Yeah, I guess like, so you have to state that pipe purchase in zeros for to, because like you said, we are not adding, but that's the way the Python counts I guess the how

S1 Speaker 1 ▷ 1 25 41

Many times. Yes. Remember that when we, we talk about when for loops, when we say if we want to count, if we're gonna start counting something, we define a count variable that is gonna be equaled zero.

S1 Speaker 12 ▷ 1 25 55

Yes.

S1 Speaker 1 ▷ 1 25 56

Right? So imagine this count variable but in a list.

S1 Speaker 12 ▷ 1 26 01

Oh, I see.

S1 Speaker 1 ▷ 1 26 03

Okay. Yeah. Perfect. Any other questions guys? Okay, great. So let's go to our next activity of today's class. Let me open my next activity of today's class. This is a cool activity actually. And yep, perfect. So we're gonna see this plantation again and using my screen thumbs up. Perfect. Okay, so it's about reading text files in pythons. So now we're talking about the different topic. Python can read in data from external text files and perform tasks on that data. Python actually can read text files, query files, databases, files, whatever most whatever data you want to read with pipe. So basically in order to do some reading, it's, it's quite mm a structure. Okay, we're, we're gonna call it that. This is the algorithm to read files with pipe, okay? This is the step by step to read a file with pipe without any libraries, without any just pure pipe.

S1 Speaker 1 ▷ 1 27 47

Okay? So we are going to go over the syntax and the purpose of the code covering different points that when you deal with external files, Python requires precise directions on what path to follow to reach that desire pile. So basically we're gonna talk about resource here and once we have that, we have, we have to take account that Python works with different operated systems. So you have different paths, you have different ways of writing it, so you have to write it, right? Right. For example, windows machines will often use forward slashes to separate the folders. Mac devices will use back slashes, okay? So with it's part of our step one and it's a special syntax blocks that it's gonna allow us to perform operations, eh? That requires to reset our code, our block of codes completely, okay? The width is going to help us to start the operation and then close or clean that operation. Okay? So I can close that file. Open is gonna be attached with the file pad, read write. You're gonna see how we are gonna specify R for read, write for, right? I mean W for right and r w for read and write permissions. Okay? And the text do read, well gonna read the entire file, it's a string and we can read it with a form loop as well. So let me show you now how this is, how this looks.

S1 Speaker 1 ▷ 1 29 45

And I want you to make sure this one, can you see my screen? Can you see my, my visual code? My code? Yes. Perfect. Okay, so remember step by step I am going to define my path I is gonna be equal to because I am

in Windows resources. I have a folder here inside my instruction basic read that, it's called resources. And then in resources I have a, a document that it's called input txt. The input txt, it contains only this strings. This is the first line, this is the second line, this is the last line. Okay, I'm, I'm just gonna put it here next to, so, so you have Jira, okay, then I'm going to use width. Remember the width is gonna help us to open the code of block and then close it open. Then I am selecting this variable. This file R is for reading only as text.

S1 Speaker 1 ▷ 1 30 47

I can name this variable as any as Hugo as laptop, ASCA as whatever. But text it's more convenience for me. So I am naming all these as text so I don't need to put all this inside here. Okay? So then I print the store as a reference to a file stream that says, and then I'm gonna read it right lines in order. In order for me to read it, I have to store it in a bio lines is gonna be equal to X. Read that this is a function read and then print all the lines. So as soon as I am reading this, guess what? It's gonna give me an error.

S1 Speaker 1 ▷ 1 31 36

Hahaha. The arrow says I don't have that file, right? Not such a, I know why because I am actually in a, in in the 3 0 3 for Keith in candy store. So I am going back, going back and that it CD and I am going to zero five and I am going to the solve version. Okay, now we talking. So now I'm going to read it and again is not working solve I am, oh because I am in this so I need to go back. This is important guys. It depends on the machine because you get so many errors when these kind of things happen. It depends a lot and voila, it depends a lot on the path that you are selecting for your, for your file. Okay? So in this case, I have to be exactly in this label, in this label of my, of my hierarchy of folders because if not, I am not going to be able to read resources and inside resources the input of text like just happened to me. Okay? Now that I am in that label and I, I'm sure that that resources exist, then I can questions so far guys, I

S1 Speaker 12 ▷ 1 33 26

Have a question for the lines. It's just, you can put any word, it's just a variable or it's something that you specify somewhere

S1 Speaker 1 ▷ 1 33 35

Here.

S1 Speaker 12 ▷ 1 33 38

Okay? So that's, so it is just because you write that if you would've had put something else, you have to put that exactly I guess like my question, if it has to match for what you have in your text,

S1 Speaker 1 ▷ 1 33 53

Yes, exactly, because I am reading this file, I am reading this file and I am printing this file into my con into my third. Yes, it it has to match. Yeah. So

S1 Speaker 12 ▷ 1 34 05

If you say this is the first try, this is the second try, this is the third try, you will have to put call out, try equals read the line and your terminal,

S1 Speaker 1 ▷ 1 34 20

No, actually no because remember this what is doing is reading this file, this file it. Go input here. Whatever I put in this file, this code is going to read.

S1 Speaker 12 ▷ 1 34 33

Yeah, I get what the path is. It's just like for that lines equal text that read. So lines could be anything.

S1 Speaker 1 ▷ 1 34 41

Oh yes, exactly. You can put whatever you want to put here in lines. Yes,

S1 Speaker 12 ▷ 1 34 46

See? Okay, got it.

S1 Speaker 1 ▷ 1 34 48

No, any other questions? Okay, great guys. So let's go to the next topic of today's class. The next topic is actually just a really simple one to be honest, because it's called modus. Okay, modus, let me open modus,

what is modules? Well, as you know, in Python you have a lot of built-in functions, right? But when you are coding, you will realize that you need to add external modules. These external modules are going to help you to perform a specific tasks. As you can see here, for example, this, this module random is helping me to generate random numbers. This module string is helping me to a custom method. The assist letters for example, okay?

S1 Speaker 1 ▷ 1 36 02

When you are going to use a module, you have to use this import, okay? Into Python. It's very simple. You just put import in our code and then you have your module working. Okay? As of today, everything that we did, it's it come already in the package with Python, okay? So there is no need to install anything new for the moment. Everything is embedded in Python and we can use, okay, so you are going to see that most of your programs or all of your programs are going to start always with importing modules, okay? Because you will need it actually next week we're gonna start looking at pandas and programming with other modules, powerful tools that help us and simplify our lives. Okay? So if I actually run this code, you are going to see that, oh yeah, I run this code. You are going to think that the first one is print string assist letters, and this is my assist format letter, a, B, C, D, G, T, and then capitalize. Okay? I need that to specify the machine, what kind of keyboard I want to use or what kind of standard letters I want to use. Okay? Then for X in range 10 print random dot Randi ran is a module in my function random that is going to give me a random in a range that I specified. So as you can see, we're gonna print four to 1, 2, 5, randomized numbers from one to 10. So it's gonna print nine, okay? Zero, one to 2, 4, 5, 6, 7, 8. No questions guys.

S1 Speaker 12 ▷ 1 38 22

So are they like libraries

S1 Speaker 1 ▷ 1 38 25

Or not necessarily exactly. They're libraries. We call them modules, but they are libraries exactly what you said. Yeah. Okay.

S9 Speaker 9 ▷ 1 38 41

One question.

S1 Speaker 1 ▷ 1 38 42

Yes.

S9 Speaker 9 ▷ 1 38 45

Yeah. So what did the four look at here? Because if, if you would have added print, random, random 1, 2, 1, comma 10, will it not give the random numbers? What did the four look at here?

S1 Speaker 1 ▷ 1 39 04

Well, the, the for loop, actually what it's doing is giving you or or looping through the range of them, but you actually catch it correctly part because I am not using any egg here as you can see here, right? So I am not using the for loop basically I can just print this one and it's gonna randomize from one to 10. Okay? They just put the for loop because inside that for loop you can select the range and specific range you can OMI Randy, and then just put random, okay?

S9 Speaker 9 ▷ 1 39 47

Okay, got it. Thank you.

S1 Speaker 1 ▷ 1 39 49

No problem. Any other questions guys?

S5 Speaker 5 ▷ 1 39 54

Hugo?

S1 Speaker 1 ▷ 1 39 55

Yes. Yeah, yeah,

S5 Speaker 5 ▷ 1 39 56

If you already are putting one to 10 in the Randi, what is the point of putting range again?

S1 Speaker 1 ▷ 1 40 03

Oh yes, no point here. I'm putting this, yeah,

S5 Speaker 5 ▷ 1 40 06

So you can just move that statement and it'll run just

S1 Speaker 1 ▷ 1 40 09

Fine. Yes, you can remove that statement and you can just print this one. Okay. But this is just an example for us. If we don't want to use Rand it, we can use X and it will randomize the range of 10.

S5 Speaker 5 ▷ 1 40 22

Okay? So, oh, you are saying if you don't use Rand it and you just say random of X, then it's going take the range from zero to nine and just print numbers randomly?

S1 Speaker 1 ▷ 1 40 31

Yes. Random X, exactly.

S5 Speaker 5 ▷ 1 40 35

Okay, got it. Alright. But if you are using Rand int then you don't have to specify. You

S1 Speaker 1 ▷ 1 40 39

Don't have to use, you can remove.

S5 Speaker 5 ▷ 1 40 42

Thank you.

S1 Speaker 1 ▷ 1 40 44

No problem. Any other questions guys? Okay, great. So well basically it is not an activity, but here the next part you will have to go this built-in module for Python by yourself. You will have to check it out. You will have to see what these modules do. You can do it in Python, you can do it however you want to do it, but yes, it's just for you, for your knowledge and, and for your documentation. Okay? So perfect guys, let's take 15 minute break and then we can come back for the second part of the class. Okay? Thank you guys. If you have any questions, please let us know.

S5 Speaker 5 ▷ 1 41 43

You go, I do have a question.

S1 Speaker 1 ▷ 1 41 45

Yes. Yay.

S5 Speaker 5 ▷ 1 41 47

Can I share my screen?

S1 Speaker 1 ▷ 1 41 48

Yes, of course. Thank

S5 Speaker 5 ▷ 1 41 50

You. All right. So here I've been trying to run this, but is it because of where I am?

S1 Speaker 1 ▷ 1 42 09

Okay. I have to see exactly.

S1 Speaker 12 ▷ 1 42 12

Aye you're using a Mac, right?

S5 Speaker 5 ▷ 1 42 14

Yeah,

S1 Speaker 12 ▷ 1 42 15

I think is there the forward lash, he's mentioned that we have to use backlash.

S1 Speaker 1 ▷ 1 42 20

Exactly. Then is, yeah,

S5 Speaker 5 ▷ 1 42 23

One second. It gets very hard. Okay, let me just

S1 Speaker 1 ▷ 1 42 30

Stop clicking stop. Yeah,

S5 Speaker 5 ▷ 1 42 32

I am gonna do that because it's, I'm, it's very hard for me to reach the top.

S1 Speaker 1 ▷ 1 42 37

Oh, okay. Yeah, when you're sharing screen, right?

S5 Speaker 5 ▷ 1 42 40

Yeah, I I

S1 Speaker 1 ▷ 1 42 41

Like a bunch of buttons. Yeah,

S5 Speaker 5 ▷ 1 42 43

Do anything cause it's hard. Okay. Alright. I've used the okay forward slash I still see the same error. Denise can share what you've put in your input on the message. Maybe,

S1 Speaker 1 ▷ 1 43 08

I mean, I can, I can help you. Yeah. Or share your screen.

S5 Speaker 5 ▷ 1 43 12

Yeah, sure. Can you see my screen, Hugo?

S1 Speaker 1 ▷ 1 43 24

Yes, I can see your screen.

S5 Speaker 5 ▷ 1 43 26

Okay. Is that what she asked me to do? The resources and input?

S1 Speaker 1 ▷ 1 43 30

Yeah, exactly. Just go to your explorer so I can see your folders please.

S5 Speaker 5 ▷ 1 43 35

Yeah.

S1 Speaker 1 ▷ 1 43 37

Okay. So in your folders you have resolved. Okay, perfect. Let me see what you are putting on the,

S5 Speaker 5 ▷ 1 43 45

Should I change my report? Here,

S1 Speaker 1 ▷ 1 43 49

Let me check where you are. You are basically in ins basic read and you are in the solve. Go back, go back to the just zero five in No, I I mean in the, in the terminal. In the terminal. In the

S5 Speaker 5 ▷ 1 44 05

Terminal.

S1 Speaker 1 ▷ 1 44 06

I can see that you are in read file solution dot pi. Am I right?

S5 Speaker 5 ▷ 1 44 11

Right?

S1 Speaker 1 ▷ 1 44 12

Yes. Go back. So you stay only in the folder of ins. Basic read.

S5 Speaker 5 ▷ 1 44 20

So I just went back one folder. Let me just try it again.

S1 Speaker 1 ▷ 1 44 28

No, go back. One more folder.

S5 Speaker 5 ▷ 1 44 30

Go back one more folder. Okay. All right, let me just try this without sharing the screen.

S7 Speaker 7 ▷ 1 44 43

Just so you know, there, there is a way to hide floating meeting controls too when you're sharing your zoom so it doesn't have that thing up top.

S5 Speaker 5 ▷ 1 44 51

How do I do that Andrew? There

S7 Speaker 7 ▷ 1 44 52

There's like a three little dot thing when you're sharing your screen on the far right and it'll drop down a couple of meeting options and you can hide. There's an option to hide meeting controls.

S5 Speaker 5 ▷ 1 45 02

Oh, like that. Okay. Hide floating meeting controls. Okay, so wait, I still don't see it happening. So debug remind me. Still saying read solution.buy. I have tried going back.

S1 Speaker 1 ▷ 1 45 26

Try to delete the two dots that you have there. This one? Yes, those slots and as well the, the backslash. Try to delete that backslash.

S5 Speaker 5 ▷ 1 45 42

Okay.

S1 Speaker 1 ▷ 1 45 46

Okay. Try to run it. I'm not, you are actually in the folder there.

S5 Speaker 5 ▷ 1 45 53

I don't know why I am there. So if I just, okay, I have to run.

S1 Speaker 1 ▷ 1 46 03

Okay, I can see, I can see that you're in the folder there. Okay.

S5 Speaker 5 ▷ 1 46 11

Read file load solution. I have to basically be here zero

S1 Speaker 1 ▷ 1 46 15

Exactly there.

S5 Speaker 5 ▷ 1 46 16

I don't, I have done that, but it's not taking me there, so I don't know if I have to go from here. That's just, I'm

S1 Speaker 1 ▷ 1 46 27

I just, same issue too. I'm not sure why. Can you try

S7 Speaker 7 ▷ 1 46 31

Clicking on

S1 Speaker 1 ▷ 1 46 32

The Oh, you got it.

S5 Speaker 5 ▷ 1 46 44

I'm not sure if this is what I'm supposed to be doing, but, okay, so I'm here. I don't, not sure if it's gonna work though. It still doesn't work. I don't know what's going on.

S8 Speaker 8 ▷ 1 46 59

Can you try the back slash to forward slash in the file resources?

S1 Speaker 1 ▷ 1 47 05

Yes, I was on, yeah,

S8 Speaker 8 ▷ 1 47 08

We're on Mac, right?

S5 Speaker 5 ▷ 1 47 10

Yeah,

S1 Speaker 1 ▷ 1 47 12

Still not right?

S5 Speaker 5 ▷ 1 47 13

Still not working. I don't understand what's going on here. So it takes me back to the base folder. I don't know why it does that. I am not sure if I do the explorer. I am here. So I have to be here.

S1 Speaker 1 ▷ 1 47 34

Yes, you have to be in,

S5 Speaker 5 ▷ 1 47 35

I have to be here.

S1 Speaker 1 ▷ 1 47 36

Zero five. Yeah. Got And you are there, right?

S5 Speaker 5 ▷ 1 47 40

Mm.

S1 Speaker 1 ▷ 1 47 41

It's because I can't see your terminal. I can't see where exactly you are. You

S5 Speaker 5 ▷ 1 47 46

Can't see me. Tom, is this better to,

S1 Speaker 1 ▷ 1 47 50

Let me just hide my, my comment.

S5 Speaker 5 ▷ 1 47 53

Great. Okay. This is too big.

S1 Speaker 12 ▷ 1 47 56

Yeah. Did you make a copy out of this folder out of your repos and place it somewhere else?

S5 Speaker 5 ▷ 1 48 02

I'm sorry,

S1 Speaker 12 ▷ 1 48 03

This is to your, your original repos, right?

S5 Speaker 5 ▷ 1 48 06

Yeah.

S1 Speaker 12 ▷ 1 48 07

Oh, okay. All right. No, nevermind.

S5 Speaker 5 ▷ 1 48 14

Oh, it's taking me back to this repo and you

S1 Speaker 14 ▷ 1 48 18

Clicking. Go ahead.

S7 Speaker 7 ▷ 1 48 20

Oh, I was just gonna say, if you type Ls again, you're in that back in the home directory basically. The home

S5 Speaker 5 ▷ 1 48 25

Directory. Yeah. I'm going back to the home directory. This is the home directory that it's taking me back to. But when I run the program it says I am at the, so I'm, I'm here. When I run it, it shows me that it's running here and not from this folder. So I don't understand how

S1 Speaker 14 ▷ 1 48 49

Can, can you right click on that?

S5 Speaker 5 ▷ 1 48 51

Can I Right click on what?

S1 Speaker 14 ▷ 1 48 53

On the that file, the python file.

S5 Speaker 5 ▷ 1 48 56

Okay.

S1 Speaker 14 ▷ 1 48 58

Open in integrated terminal. I think that can try that Python name of the file.

S5 Speaker 5 ▷ 1 49 11

Oh, you're saying, okay.

S1 Speaker 14 ▷ 1 49 17

Read file solution.

S7 Speaker 7 ▷ 1 49 18

You'll need to add those two dots again though.

S1 Speaker 1 ▷ 1 49 21

Yep.

S7 Speaker 7 ▷ 1 49 22

So you go back a directory,

S1 Speaker 1 ▷ 1 49 23

Two dots

S5 Speaker 5 ▷ 1 49 27

And

S1 Speaker 1 ▷ 1 49 29

Yeah. Is that

S1 Speaker 12 ▷ 1 49 30

I think the back slash I don't know, it didn't work for me either. I'm, I'm in the same boat, the new hire.

S5 Speaker 5 ▷ 1 49 38

God. Okay, let's get to the bottom of this. Trying to understand what's going on here. Okay,

S1 Speaker 1 ▷ 1 49 50

Try to put Ls there.

S5 Speaker 5 ▷ 1 49 53

No such file or a directory. Can I?

S7 Speaker 7 ▷ 1 49 56

It looks like it didn't save. It looks like, will you right click the exact same. Sorry. The exact same thing you did with the terminal. I think it didn't save

S1 Speaker 1 ▷ 1 50 04

It didn't save it? Yes, it didn't save it. Save it.

S5 Speaker 5 ▷ 1 50 08

Okay. Now it's, should I change the I? Yes.

S7 Speaker 7 ▷ 1 50 14

Now

S5 Speaker 5 ▷ 1 50 14

Change. Okay, this is alright. Sorry people. I feel like I'm taking up a lot of time but ah, that's it. Finally. But why is it not running in? Okay, if I just do it this way? See it gives me an error.

S1 Speaker 1 ▷ 1 50 35

Yes. It goes to a different path.

S1 Speaker 12 ▷ 1 50 37

Yeah, me too.

S5 Speaker 5 ▷ 1 50 39

So I think what Bill suggested is probably the better one to do where he said open and integrated terminal and then just run the file. I should just stop this first of all. Okay. No, I have to, if I do it this way it works. But if I do it the regular way for some reason I dot know what's going on there. Yeah.

S1 Speaker 16 ▷ 1 51 10

Can you do it the other way just for a second? I couldn't read that Follow path. It going by too fast.

S5 Speaker 5 ▷ 1 51 17

Oh, okay. Should I just run it as it is?

S1 Speaker 16 ▷ 1 51 21

Just however you did to get that, that error? Yeah.

S5 Speaker 5 ▷ 1 51 23

Okay.

S1 Speaker 16 ▷ 1 51 24

Okay. Just gimme a second.

S5 Speaker 5 ▷ 1 51 26

Sure. Do you want me to ex like

S1 Speaker 16 ▷ 1 51 31

Yeah. Yeah. So one thing that might help is some of the settings that deal with where files are executed when you're just running using the run command. So go to settings.

S1 Speaker 12 ▷ 1 51 48

The settings on, on vs. Code

S1 Speaker 16 ▷ 1 51 51

Vs code. Yes.

S1 Speaker 12 ▷ 1 51 53

Okay.

S5 Speaker 5 ▷ 1 51 55

I just saw them. Where did it go?

S1 Speaker 16 ▷ 1 51 59

There's the, the little wheel at the bottom left.

S5 Speaker 5 ▷ 1 52 03

Oh yeah. Okay. Right.

S1 Speaker 16 ▷ 1 52 04

Go to settings. Cause I'm just curious what it's here. And Python you can search for, I believe go down to Python.

S5 Speaker 5 ▷ 1 52 22

I'm sorry.

S1 Speaker 16 ▷ 1 52 23

Python actually just, yeah. And search for execute and file dur or execute.

S5 Speaker 5 ▷ 1 52 33

Just say Python execute or just execute?

S1 Speaker 16 ▷ 1 52 35

Just Just execute. Okay. And if you want, you can set it, see on the right, you can set it to always execute the file where it, because when you run it, sometimes it'll, it'll, it'll run Python in a Python location. Run that file rather than going to the location of the file and running Python there. So click, click that button.

S5 Speaker 5 ▷ 1 52 58

This one?

S1 Speaker 16 ▷ 1 52 59

Yeah. And then go, go back out and try to run it again from just using the run button.

S1 Speaker 12 ▷ 1 53 05

Do you think I should do it the same for Jupiter? It says Jupyter interactive window text editor, execute selection, or just Python

S5 Speaker 5 ▷ 1 53 13

Exit out of testing.

S1 Speaker 16 ▷ 1 53 15

Yeah, I, I, I don't see You can just close the settings window. Yeah, if, if you go up and up to the left a little bit on the tab there's a clo there's an X for the settings window.

S5 Speaker 5 ▷ 1 53 32

I'm sorry, I'm

S1 Speaker 16 ▷ 1 53 33

So u up a little bit on the tabs, there's the tabs and you go to all the way to the right, to your settings window. Not, not, not your bar at the top, but like the tab, the settings tab that you opened up, you can exit X out of it.

S1 Speaker 12 ▷ 1 53 45

It works. Andrew, thank you so much.

S5 Speaker 5 ▷ 1 53 49

I'm not able to, cause

S1 Speaker 16 ▷ 1 53 49

It's, it's running the way VS. Code is set up. It's running those files from the location of Python, not from where the files located. Okay.

S5 Speaker 5 ▷ 1 53 59

Alright.

S1 Speaker 16 ▷ 1 53 59

And this should be able to get you to run it from

S5 Speaker 5 ▷ 1 54 04

Hmm.

S1 Speaker 16 ▷ 1 54 06

Min any place. But

S5 Speaker 5 ▷ 1 54 11

I wonder

S1 Speaker 16 ▷ 1 54 11

If it's saved.

S5 Speaker 5 ▷ 1 54 13

It's saying the same thing.

S1 Speaker 16 ▷ 1 54 15

Okay. Go back to your, your settings again and search for execute and scroll down. Scroll up. Okay. Now go back to the file.

S5 Speaker 5 ▷ 1 54 37

Okay.

S1 Speaker 16 ▷ 1 54 43

Can you exit out of this terminal, like delete the terminal or if possible I can take control of your screen real fast just to check on

S5 Speaker 5 ▷ 1 54 52

Something. Yeah, you can do that. Sure. Or,

S1 Speaker 16 ▷ 1 54 58

And go back to your file.

S5 Speaker 5 ▷ 1 55 01

Okay. One second. So,

S1 Speaker 16 ▷ 1 55 07

Because I, I know we'll have to change that setting down the road cause it's really fickle with where it's ex, where it runs the file from. Yeah.

S5 Speaker 5 ▷ 1 55 15

It

S1 Speaker 16 ▷ 1 55 15

Using the run button. Now in the terminal command line, you can go to the file and run it. But when Visual Studio has its own little settings on where it runs things,

S5 Speaker 5 ▷ 1 55 26

Why is this showing up like this?

S1 Speaker 16 ▷ 1 55 29

I don't know. Let's go back to that.

S5 Speaker 5 ▷ 1 55 31

Okay.

S1 Speaker 16 ▷ 1 55 32

And use the run button.

S5 Speaker 5 ▷ 1 55 35

So it says a good repository with

S1 Speaker 16 ▷ 1 55 37

You. You can just, you can just close that for now.

S5 Speaker 5 ▷ 1 55 40

Okay. All right. And you want me to run it right?

S1 Speaker 16 ▷ 1 55 45

Yeah.

S5 Speaker 5 ▷ 1 55 47

As a Python file.

S1 Speaker 16 ▷ 1 55 50

Okay. Okay. That is kind of weird.

S5 Speaker 5 ▷ 1 55 54

Hmm. So if I go back to the settings and we have already ensured that I have that.

S1 Speaker 16 ▷ 1 56 03

Yeah. Yeah. So it's, it's not gonna, it's, it's something else with the file path, but you can run it from the condo, which is, or from the, the, the base, which is weird.

S5 Speaker 5 ▷ 1 56 14

Yeah, I can run it from the base. I'm not even,

S1 Speaker 16 ▷ 1 56 16

If you go down to the, go down to the terminal and in the terminal do an Ls cd into solved and run it right there. Just Python read, read, file solution up high

S5 Speaker 5 ▷ 1 56 35

I think. Sure. It's working.

S1 Speaker 16 ▷ 1 56 37

Hit tab.

S5 Speaker 5 ▷ 1 56 39

Oh yeah. Okay, cool. Yeah. Okay. It works.

S1 Speaker 16 ▷ 1 56 44

Okay. So it's, it's just something else with that. We'll have to look at it at some point.

S5 Speaker 5 ▷ 1 56 49

Mm. Okay. Maybe I'll get in touch after the class so that we don't

S1 Speaker 16 ▷ 1 56 55

Yeah, yeah. We can look at it at in office hours. Cuz I, I wanna know what, what's going on there so I can help the students out. Yeah, the same issue.

S5 Speaker 5 ▷ 1 57 02

Thanks Andrew. I'll really help. Okay, perfect. And yeah, that's it. Now how do I get back to the controls? How do I go back to, okay.

S1 Speaker 1 ▷ 1 59 06

Okay. Perfect guys, I'm top. Let's go back to the last part of the class. Okay. So let's jump to the next part. And can you see my screen? Oh sorry. Cause take the, I'm gonna take the screen now. Okay. Can you see my screen thumb up? Yes. Perfect. Okay. And then, okay, perfect. So in the data industry, we are gonna talk about now reading in C S V files, it's quite the same of reading text files. These are text files but with a different different file name. Okay. So in the data industry you will encounter file knowns as CSVs a lot. What is a C S V? Basically a C S V, it's a coma separated values coma, separated values file. Basically it's a file that is separated by commas. Well, it's separating the values by commas. Okay. You can see here first name, last name, and phone I hear down, you can see Janet Bald.

S1 Speaker 1 ▷ 2 00 52

So basically each of these one suggests that it's corresponding to the first line. That is the header. Okay. So each row needs to be all located on a new line in each column. And it's gonna be separating by comm. Like I said, you can read CSVs like endless CSVs in. Yeah. And it looks like an Excel sheet because it looks everything like, like, like the Excel sheet where it's A C S B. Okay. Main difference is that C S B does not have any limit on the data. You can put as many data as you want and and Excel you have, you have links. Okay. So there is a module called CSV in Python where this model can help us to read external CSVs and we can perform different operations with data. So there is another model that it's called OS that basically help us to adapt Python to any operated system that we are gonna use. Okay.

S1 Speaker 1 ▷ 2 02 13

And here the CSB reader we are going to use instead of text do read. Okay. One, one other thing is that we are going to have ERs, the TER because we are going, we're going to read a CSV is gonna be a comma.

Okay? And it's a parameters that we actually put. So let me show you any questions so far about reading a tsv? No. Okay, perfect. So let me go to my visual studio code. Can you see my visual studio code? Okay. So what I'm gonna do is actually show my video panel and hide out my floating control controls. Okay? Perfect. So I'm gonna close this one, this one, and I'm gonna go exactly where I have my C S B.

S1 Speaker 1 ▷ 2 03 26

So you can see here, this is my code for c s csb, right? I am going to import this model that is called us, the one that I just explained. And I'm going to import my, you see my screen? Yes. Right. Perfect. Okay. And I'm gonna import my csb. So now I am going to create my path. Remember in the read text, I create my file equal to, well, here it's a different typing. Okay. This typing include os pat join. So os I'm using my model os, this model include path, okay? And the path includes join. So I can join as many things as, as I want to join in my path. So basically I can put resources and contact. Okay? Because I am going to be in, in here resources and entire resources. I have three contacts. What in context, the thing that I show you.

S1 Speaker 1 ▷ 2 04 31

First name, last name, phone. Okay, so we have two methods to read it. Okay? One is width, open is not, that one is better than the other one. It's just we have two methods, or we can have three or four depending on how you code, but is the same steps. Okay? I'm going to go through this one with open CSB path and name that as a variable TSB file. Okay? Then I'm going to create my variable CSB reader, TSB dot reader. And I am going to read that CSB file, er like, like I explained before, with a coma printing, this CSB reader is gonna print an object, okay? And this object, I can treat that object. Now I can read it because I extract it and I can read it with Python. So I'm gonna put the head first. CSB either is gonna be equal to next. The next, basically it's, it's a module that is gonna help me go to the next available line. Okay? I don't have to read each one individually, I can just put the next one. It's gonna go the next item on the iterator. On the CSB reader. The CSB reader is my object. Okay?

S5 Speaker 5 ▷ 2 06 07

Does it add like a four loop or lines?

S1 Speaker 1 ▷ 2 06 13

Yes, it, it can, but this is specifically for reading CSVs. Okay. You cannot use it outside there.

S5 Speaker 5 ▷ 2 06 22

Got it. For CSVs, if you have to change from one line to another, you can use next.

S1 Speaker 1 ▷ 2 06 27

Exactly. Thank you. So now we're gonna print the headers because I am actually using headers and the CSB header is going to be marked by, okay, now I I, I want to print all my row. So for row in CSB reader, CSB reader, print, row, and that's it. And this is not going to work because I am actually in basic. So let me go to that activities and then go to the zero eight and this will work. That's it, right? I print my header the first line, and then I print everything. As you can see, one important part here, everything is printed in the list. So I have a list of lists. Each line is a li is a list as you can see. And I can look through that individual list and I can get individual date except from questions. Guys,

S1 Speaker 12 ▷ 2 07 40

I have a very silly questions. So when you have the reader, the header, can you, is first name, last name, and phone, do you have to call out those? Are they actually not line up with the data per se? If we want to just get a last name and we just set that variable, can we call the last name only?

S1 Speaker 1 ▷ 2 08 06

Yes we can. Good question Denise. Denise. Because basically I am returning a list. Okay? And in, in this list, what you can notice index zero, index one and index two.

S1 Speaker 12 ▷ 2 08 18

So you will have to put like, if you wanna call the last name, but can you call it from the whole column or get, or like, or for specific row,

S1 Speaker 1 ▷ 2 08 30

You call specifically this list that it's coming in, in this header and then you just select the index that you want to see.

S1 Speaker 12 ▷ 2 08 41

But that's just in that line, in the header, I'm talking about column and I'm talking if, if I want to bring, for example, Henbury or Henbury, whatever, if I call out index one and then 0 1 2 index 1 0 2, index one,

S1 Speaker 1 ▷ 2 09 08

Or index

S1 Speaker 12 ▷ 2 09 08

1 0 2. So basically to get just Handbury last name, how do you take, how do you take that last name from, from there?

S1 Speaker 1 ▷ 2 09 19

So basically you want to get just the last name with all the, with all the lines that contains only the last name Ma, right?

S1 Speaker 12 ▷ 2 09 28

Yeah. So just one last name. So basically I'm taking as a reference the index, which is number one last name, right?

S1 Speaker 1 ▷ 2 09 36

Exactly. You just called it, you just called that index.

S1 Speaker 12 ▷ 2 09 39

Okay. So now I want to take him from that whole column. How can I do that?

S1 Speaker 1 ▷ 2 09 48

Okay, we are returning a list of lists. Okay? So this list of list contains, this is index zero, this is index one, this is index two, this is index three, okay, this is index four. Oh,

S1 Speaker 12 ▷ 2 10 05

Okay. I

S1 Speaker 1 ▷ 2 10 06

Will, we will have to go into index 0 1 2. And inside that list I have to go into zero one to get only Hamburg.

S1 Speaker 12 ▷ 2 10 18

So the index is first to get to get Hamburg is index 1 0 1 2 and then one,

S1 Speaker 1 ▷ 2 10 34

No, because this is a list of lists. Okay, when I'm talking about this is that it, look at this as a big list, as a just one list, right? Okay. Inside that list you have this index zero, index one. Yes. Index two. Okay. Okay, next. So in order to bring up the ary list, you will have to do index two and then inside index two you have to put zero and one.

S1 Speaker 12 ▷ 2 11 05

Okay, got it. So that'll be in square brackets two coma one,

S1 Speaker 1 ▷ 2 11 11

No, it will be square brackets two a square brackets one.

S1 Speaker 12 ▷ 2 11 16

Oh, got it. Thank you.

S1 Speaker 1 ▷ 2 11 18

No problem. Any other questions? Okay, perfect. So we can jump to our next activity of today's class. That is, you're gonna do something similar, but you're gonna read comic book C S V, okay? If you go to the rhythmia, here's the RHYTHMIA activity. In this activity, you'll create an application that search, they provide c s V file for a specific graphic noble title, and then return the title, publish your name and the year it was published. Okay, here are the instructions guys. We have actually 10 minutes to work in this activity and then we can see the, the solution, okay? So please, if you have any questions, just let us know. We're gonna be here and the room breakout rooms are open as well. Guys, two more minutes

and we are going to start checking the solution. If you have any questions, remember breakout rooms are open or you can ask directly here.

S2 Speaker 2 ▷ 2 22 18

I

S1 Speaker 1 ▷ 2 23 41

Perfect guys. So let's, lemme share my screen. The solution thankful the, the hearing screen is working fine. Is they, can you see my screen on top? Yeah. Perfect. Okay. So the activity ask you to do a search for comic books. And this search actually help us to find a specific title, right? So what it does is that when I run this code, it says, what title are you looking for? So if I put four,

S1 Speaker 1 ▷ 2 24 26

If I put here the print or something like that, it, let's see, it says the print was published by Diamond Marvel in 2010, right? If I put why say sorry about this. We don't seem to have what you're looking for. So let's see what is doing. We import this two that we talk about already. This book is, this variable book is actually the one that is getting the title of the book from the user. Okay? Then I'm gonna set the path for the file. Remember OS pad join and then where I have my comic book csb, if I open my comic book csb, I'm gonna going to see that that is really big, it's huge and I have a, something that looks like a header with title, other title I S B I I S B N and blah blah blah blah blah. In the, in the exercise they told you to look for the title.

S1 Speaker 1 ▷ 2 25 35

So it's index zero here as you can see, and the other one is index eight and nine. So it's publisher and date of publication. Okay? That's why we are going to use zero eight and nine then found files. I have to set this variable because I have to deal when without a conditional I have to deal. If I don't find any specific title, then what, what I am going to do. Okay? That's why I'm heading up this variable to bullion fold. So I'm gonna open my CSB with open csb, but this encoding don't take the, don't think about it much. UT F is just a way of reading certain files. UTF is the most

recommended ENC coding, but you maybe face some other encodings that you can Google that without any problems so you can learn more about it. So I am going to actually put everything in this name, in this variable name that's called CSB file for my CSB reader is gonna be CSB do reader and I will have inside my CSB file and my ter really important comm.

S1 Speaker 1 ▷ 2 26 52

Then I'm going to do a loop for row in CSB reader that is variable if row index zero equal equal book whatever I put here. I mean whatever I put here in book, then I am going to print that row, row index zero plus what? Published by row index A in row index nine. Okay, so what is this doing? It is gonna go first to the first line that is this one title other, I didn't find any, any title that looks like name of book. Then I go into the next line. So it's going to this line and then the next line and then the next line. So on. Okay, I set up my variable to confirm we have to find a video true, because if I found a video then this found is going to be equal true, and I can keep looping. Okay? If not, then I put the condition that is gonna say if found false, then print this state questions, guys. Perfect, no questions. Great. Okay. Okay, that's good. So let's move to the next. I have a question. Good? Yes. Kiki,

S1 Speaker 17 ▷ 2 28 22

If you go back to the comic book, the csv

S1 Speaker 1 ▷ 2 28 26

Yeah.

S1 Speaker 17 ▷ 2 28 27

Are we considering information in in columns as rose is? Is that what

S1 Speaker 1 ▷ 2 28 37

Justine? Rose, Justine enroll Justin Justin Row separated by a comm,

S1 Speaker 17 ▷ 2 28 46

Enrolled separated by a comm.

S1 Speaker 1 ▷ 2 28 48

So basically this role is a list of data. This is my index zero, my index one, my index two, and so on. Okay. Then I finish going through that row and I start in the next row. Okay?

S1 Speaker 17 ▷ 2 29 07

Okay. So, okay, don't worry, I, I can ask my question maybe after the class,

S1 Speaker 1 ▷ 2 29 19

Okay? Okay, Kiki, no problem at all. Any other questions?

S1 Speaker 18 ▷ 2 29 25

Ugo, you put the, our what did the row zero, row eight and the end of the code? Yeah, so that's where's the comic books? So that's the, that's because of the exact index title, other title on the first row?

S1 Speaker 1 ▷ 2 29 48

Yeah, this is index zero, index one. I mean all this is index one, index two, index three, right? I I, I'm reading all the rows guys, I'm not reading only the, the ones that doesn't say title here. I'm breathing 1, 2, 3, 4, 5. Okay, Kelvin,

S1 Speaker 18 ▷ 2 30 13

Okay?

S1 Speaker 1 ▷ 2 30 13

Yep. Perfect. Any other questions? Okay, great guys. So in that case, eh, let's jump to our next activity of the class. Basically what we can do as well is we can read sbs, but we can write sbs. It's not a huge deal here. The only thing is that you are going to put the name of the folder where you want to put your output. Then the name of your file here, you name your file, make sure that your file ends with CSB because you are importing a csb, but you can call it here, Hugo Ka, whatever name you want to give to that csb. It's almost exactly the same. I am going to actually open comic books one here and you're gonna see that in the comic one in the week open. We don't use this W okay? We use the R for read. Okay?

S1 Speaker 1 ▷ 2 31 24

So that's why, let me close this one. I'm gonna put, put that comma W as a CSB file, okay? In with open. Then I am going to create my writer re remember about CSB reader. Well, I'm going to write my CSB writer CSB dot writer CSB file. And then I put my millimeter, that is a coma, okay? Then I write the first row CSB writer write row that function of my C SB writer. Okay? Because remember I'm writing the c sb file that is related with the CSB writer. So I can use write row, write row. I'm going to put a list. This list is gonna contain something that is, that looks like a header, first name, last name, and S f n. And I am going to write another row that is gonna say the first name, the second name, and the social security number. So when I run this part, I am going to generate a file here inside my output. Well, it's not there, so let me write it. Okay? Area. Okay, so that's it. So I open my output. Okay, so you can see this new file that contains list one with first name and the other one with the name and type first questions about the writing. Guys, maybe you are seeing my screen, right? Yes. Yeah. Oh, perfect. Okay. Questions.

S1 Speaker 12 ▷ 2 33 17

Can you scroll up a little bit in your code? I'm just curious about how, so you just do, I thought we were gonna do a wild loop. No, that's loops. I mean four loops. Sorry.

S1 Speaker 1 ▷ 2 33 32

No, we actually, we don't need any loops now because we are just writing the directly. I mean, if you want to write like a data frame or something like that, you have a table already, you can write a fur loop. But in this case, for this example specifically, we are just writing it directly to, to rose. Okay? Please.

S1 Speaker 19 ▷ 2 33 56

And Denise, this is the next assignment. This isn't the one we were working on in the breakout room, by the way.

S1 Speaker 12 ▷ 2 34 01

Oh, but that's why I was so confused. Yeah, yeah. All right, thank you.

S1 Speaker 1 ▷ 2 34 06

Oh, okay. Yes. This is a new topic, actually, the writing.

S5 Speaker 5 ▷ 2 34 11

Hugo, one question, can you just show us the solution for the comic book please? Just once

S1 Speaker 1 ▷ 2 34 17

Please. Yes, we can, we can actually publish that one. No problem at all. Thank you. Okay, yes, let's publish this one so we can take a look on this. Okay, perfect guys, no questions about the writing. Yes. Question. My question

S1 Speaker 18 ▷ 2 34 39

Is just about the, the resolved solutions. Will you publish all tonight?

S1 Speaker 1 ▷ 2 34 45

Yes.

S1 Speaker 18 ▷ 2 34 46

Okay,

S1 Speaker 1 ▷ 2 34 47

Cool. Yes, they're gonna be tonight

S1 Speaker 18 ▷ 2 34 48

By the end of the class, cuz I, I usually just pull, push everything.

S1 Speaker 1 ▷ 2 34 53

Okay, that's good. Perfect. So in that case, let me pass to the next topic of this class. That is tip again, it doesn't have too much to understand here in tip. It is just one thing. We have this two, these all friends of us, this all model friends, and then I'm gonna define three list, three different list indexes, list employees list and department list. Okay? So then I'm going to put everything together how I am going to do that with something that is called zip. Okay? So I'm going to save this one in a variable and as I say, this variable that is called rooster equal to tip, to put these three list together, the index, the voice, and the department.

Okay? Once they are si then I can print the contents of each robots. Okay? So I'm gonna have four employees in Bruster print employee close.

S1 Speaker 1 ▷ 2 36 01

This one actually and an employee is gonna be my eye from this router. Okay? So I'm gonna print indexes and employees and department, all these I am going to print, okay? Then I am going to save the output file pad somewhere else. Output file is gonna be to os pat and the name of my file that ends with that csv. Then I am going to write something. I'm going to write with open output file W for write as asset data file. And then I am going to start doing, my writer, writer is gonna be equal to CSB writer, that's my writer of that date file. And then I am going to put my roles writer write row, I can put everything, analyst, index, employees and department altogether. Or I can put my writer write Rose Ruler. Okay? And I do something like this. Index, employee department index is 1, 2, 4. Employee is gonna be Michael, Dwight, Meredith, and Kelly and department is gonna be both sales, sales and hr. Yeah. Questions guys Here actually,

S1 Speaker 12 ▷ 2 37 34

But the, the data file, it's, it's just the format that you are displaying all your data.

S1 Speaker 1 ▷ 2 37 44

The data file is the name of the ba, the file that I'm open.

S1 Speaker 12 ▷ 2 37 51

Okay?

S1 Speaker 1 ▷ 2 37 52

This, this whole is named as data file. Okay, got it. Thank you. No problem.

S2 Speaker 20 ▷ 2 38 01

This question, so zip is basically creating a two dimensional array, a topple,

S1 Speaker 1 ▷ 2 38 06

Exactly. Jeff, like this. Perfect. Good catch there. Any other questions?

S5 Speaker 5 ▷ 2 38 19

And how you, when you write to a file, are you always writing to the top of the file? Like when you add index employee department?

S1 Speaker 1 ▷ 2 38 30

Yes, in that order, you can change the order.

S5 Speaker 5 ▷ 2 38 34

You can't change the order, right? It, it'll go as the top one. Okay. Alright, thank you.

S1 Speaker 1 ▷ 2 38 39

No problem. Other questions? Okay, perfect. Oh, in that case guys, I am just going to go through the last topic of today's class. So I can give you, if I take like five minutes, I can give you 15 minutes for you to talk the, the activity, the student censorship. So because these functions actually are very, you are gonna be familiar with it because you saw something like that in, in bba. So basically in order to write the function in, in, in Python, you will put the F, okay, functions are going to help us to not repeat code. So for example, here is definition of this function that the name of the function is print. Hello. And then you have to write it like this, brackets and blue points. What what this function is doing is printing this message with Hello. Okay, so whenever I call the function, you call the function like this print, hello.

S1 Speaker 1 ▷ 2 39 53

And then you put the brackets, you're gonna print Hello. Okay? In your function, you cannot argument, you can define the function, the name of the function, and then you put an argument here inside the argument is expecting a name, okay? So my printing, it will be hello plus name, okay? Plus another string. So when I'm calling this function, print name, print name, I am going to add, because it's expecting a variable that it's name, I am going to add my name Bob Smith. So is gonna print hello Bob Smith. Okay, then the, oh, so this is like the, the prime use case for functions.

Okay? So imagine you have to run the same code, the same code for these two lists, list one and list two. So you don't have to run, you don't have to write all these two times, right? For each list, what you are going to do is you're gonna find a function that is gonna call list information and is gonna contain an argument of simple lists.

S1 Speaker 1 ▷ 2 41 03

Okay? You can put as many lists as you want. And what is this function doing is printing the value within the list are and is doing a for loop of that value in that argument. Simple list and print that value. And as well is gonna give you a message that says print the length of the list is you're gonna transform this to a string. So you can concatenate it and then you're gonna bring the length of the simple list and at the end you just call the list within the function list information. List one, that's the name of my variable list information, list two, that's the name of my part. Okay? And you can read it here first is, hello. Hello. Both three. The value within the list are 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. And the length of the list is 10. And the value within the list is the other list. And the length, the length of the list is five. Okay? That's the main purpose. You don't have to write this code twice. You use the code inside the function and then you call the ar you, you make sure that you have the argument and then you use call the party within your function.

S1 Speaker 1 ▷ 2 42 25

Questions all functions. Guys, I

S1 Speaker 12 ▷ 2 42 30

Have a quick question. What exactly def does then? So I kind of put to, yeah, I'm confused. What the def function does.

S1 Speaker 1 ▷ 2 42 41

DEF is basically to write a function in Python. So whenever you want to write a function in Python, you have to start with def, then you put the name of the function. So basically the structure is, it is like this. When you write, when you write a function in Python, you write first F, that's one thing. Then you write the name of the function, okay? Then you write

the parentheses and you end up with the two points, okay? That's how you write the function.

S1 Speaker 12 ▷ 2 43 28

Got it? Thank you.

S1 Speaker 1 ▷ 2 43 30

No problem. Any other questions? Perfect guys. Okay, can

S1 Speaker 12 ▷ 2 43 39

You actually, can you actually go back, judge, I just wanna have that what you, the example that you wrote, just, okay, no problem. Thank you. Thanks.

S1 Speaker 1 ▷ 2 43 47

Yeah,

S1 Speaker 1 ▷ 2 43 50

Bye. Okay, perfect guys, so we have the rest of our time to work in this activity that is called S st use sensor sip. I am going to put the RHYTHMIA here. Basically what we have to do, it's again, we're gonna use the census data, okay? We are gonna create a Python application that read in the data from 2019 census, then store the content of place, population per capita income and poverty count into a Python list, okay? Then zip this list together into a single pupil people, remember people and permeable. And then you're gonna write the contents of your extracted date into A C S V. Make sure to include the title of these columns in your C S V. You may have to create these titles, okay? Bonus, find the poverty rate. Include this in your final output, converting the rate to a string, and include a percentage at the end of that string. And part of the string is associated with place, separating it into county and state. So we can serve both in separate columns and they give you some and references questions about the activity, guys. Okay, perfect. So let's start working in the activity and maybe with, okay, three minutes we can explain the solution. Okay? Again, if you have questions, please go with to the breakout rooms or stay here and ask as many questions you need. Guys,

S1 Speaker 16 ▷ 2 45 35

How many minutes was that before we go over the solution?

S1 Speaker 1 ▷ 2 45 41

We have 15 min, well 15 minutes and then we can go to, yeah, we are gonna take three minutes of your time. Yes. Perfect guys, time up. So let me share my screen there.

S1 Speaker 1 ▷ 2 59 13

Okay, can you see my screen guys? It's gonna take a few minutes, more of the class. Perfect. So let me explain a little bit. We have these sensors, a starter, as you can see, it's a whole bunch of data with names. It can be like a county state, some coordinate and, and well, we don't understand very well this, this state, and it's right in instruction. They give us like the headers on what to do. So basically they are, we are doing this to model imports. Then we are going to put my, my path here in census C, S, B. Then I'm going to create my list to store the data, place population income, poverty count, poverty rate, county and estate, right? So I am creating seven list here, empty list where I'm going to store data. Then I'm going to open my census starter data.

S1 Speaker 1 ▷ 3 00 16

And now that I open it, I say I store it in a C S B file, then I do the, the reader, CSB reader with the millimeter. Then I'm gonna look through this file, okay? For row in c SB reader plays app pen row zero in index zero, that is gonna be this one. Okay? Appended in place. Then the one appended in population, the the four appended in income, eight in poverty count. Okay? Then in in percent and poverty rate, I am going to actually do some arithmetic here. I am going to do a percent. First I'm going to round my in row. Number eight is my poverty count. I am going to divide this, my population in hundred, and my roundup is gonna contain two blessings and that I am going to append it in my poverty rate with a percentage number percentage stream symbol.

S1 Speaker 1 ▷ 3 01 29

Okay? Now I'm going to split my place into country and state because my place, I put it here together, okay? As you can see it here. So a split place, row zero, got split by the coma. Row zero, row index zero split by a coma. So I am splitting this by a, okay, be because this is the place. So in county I am going to append the index zero. And in a state I'm going to append index one. Okay? Now I am going to ship all my lists together. So tip all my list together. I am going to do an output file because I'm going to write a file with open output file W as data file. My writer is gonna be my C S B writer data file. And then writer is gonna be writer row. What role? Now this is going to be my first role play population per capita income. Poverty count is my head, right? And then write that writer, that writer wrote in clinic, C, I, V. And my final looks like this place, population per capita income, poverty count, poverty rate, and county and state. You look, the percentage that we add, we separate the county and the state and all these working questions. Guys, before we go,

S9 Speaker 9 ▷ 3 03 19

Yes. One question.

S1 Speaker 1 ▷ 3 03 21

Yes.

S9 Speaker 9 ▷ 3 03 22

Can you scroll up, up please? Yeah, yeah, that's, that's it. That's it. Yeah. Yeah. So in line number 33, why is percent a string?

S1 Speaker 1 ▷ 3 03 34

Oh, I just want to add that string into my table here. I just want to add that,

S9 Speaker 9 ▷ 3 03 45

Okay? Okay. No B because the answer to post and equal to would come out to be in decimals, right? And my understanding is string is a text, right? So how would that string percent will take it as 15.06 or any, any decimal value?

S1 Speaker 1 ▷ 3 04 09

Oh, because it's a percentage in a decimal value here. So basically this is a percentage with decimal value. Okay? It's not that it afloat or is representing the percentage without the percentage symbol. So basically this is getting a percentage, that's why we are multiplying that times a hundred. Okay? Mm. Convert this into a percentage and to represent that it's a percentage. We put this as a stream.

S9 Speaker 9 ▷ 3 04 40

Okay? Got it. Thank you.

S1 Speaker 1 ▷ 3 04 44

Yeah, perfect guys, I'm gonna close the class now and I'm gonna open the office hours because I stole six minutes of your time. So thank you very much. If you don't have questions, you can go. Thank you very much, have a great weekend and see you on Monday for the last class of Python. And if you still have questions, please stay here. We have four breakout rooms with the TAs helping in each of the rooms, and I'm gonna be here in the mail, okay? So you can choose where you want to go or if you want to stay here. And if not, thank you very much guys and have a great, thank you

S5 Speaker 5 ▷ 3 05 29

Hugo. Quick question.

S1 Speaker 1 ▷ 3 05 30

Yes? Yeah,

S5 Speaker 5 ▷ 3 05 32

For the first module, I think I'm, I'm lagging to submit the challenge.