* Hello. Hello, everybody. It's 629.

1:13

* Hey. Let's wait.

1:21

* Two more seconds. So I hope the assignment was not too bad.

1:26

* And I hope that you have time to review the material for this class.

1:35

* But we will have plenty of time for going through all of it anyway.

1:42

* And. I think we can start.

1:48

* It's 630 now. Oh, right.

1:53

* Okay. So it's February the 14th and I the Happy Valentine.

2:00

* If you celebrate, I am not particularly in to death, but that's fine.

2:08

* Too commercial. But that's me. Happy Valentine, anyway.

2:15

* 631 Now. So we are starting. We will cover the usual topics.

2:23

* So we will talk about a little bit more by Donna.

2:32

* We will add some elements that are more related on the management side,

2:38

* but we will talk about the themes in software development and I will share some experience and then we will do an in-class exercise.

2:44

* So we will review that.

2:57

* We will discuss the new assignment that is going to be a little bit more complex, but that's basically the spirit of the course.

3:00

* So the idea is to start the very low profile, easy assignments,

3:12

* and then build up a class by class up to the midterm when after that it will be a little bit more challenging

3:19

* also for people with some background in coding because we'll be more on applying coding to cases,

3:34

* situations, problems, topics, and that would be more the real goal of the entire course.

3:46

* All right. So let me start sharing the screen.

3:56

* And let me go first, as usual.

4:02

* Cause this wouldn't go here. Okay, so we will.

4:09

* We are. In Module four Software Development Lifecycle.

4:17

* A dictionary is another structure. In Python, you had some material to read.

4:28

* We will we will talk about those up in with the support of some slides.

4:39

* I hope you are not having additional issues with the material.

4:48

* Again, it was a little bit of messy.

4:53

* The canvas shell that we use and in part was my responsibility.

4:58

* I mean, it was my responsibilities, my course. But I physically messed up a little bit thinking that you're replacing, I mean,

5:07

* the new material that was kind of replacing what was there and what not.

5:20

* Campbell said that he did it. He did not. But anyway, so I really hope that now is working well.

5:28

* If it's not working, send an email to me to show you and that we will address that.

5:34

* All right. So let's start with.

5:42

* With the previous assignment.

5:49

* So the assignment was pretty straightforward. That was actually there was two components of the assignment.

5:54

* So one was the coder and the other one was the.

6:02

* So that was those things was on testing.

6:13

* So on the testing, that's pretty much always a little bit of an issue with some of the students because they don't really get that what I was asking.

6:20

* So I try to be more clear up in the way I introduce the exercise, but sometimes students still have a little bit of issues.

6:34

* So just to be very clear, the testing is for testing.

6:54

* The script that you would write. So he's not a genetic theoretical testing, but he's testing the program that you are going to write for you, Europe.

7:02

* Excuse me. Oh, right. Okay.

7:18

* So. So using the template that I gave you.

7:22

* You have a program goal that is I mean, that description that you already have,

7:34

* the program will check that user input if it is a number that is multiple of five.

7:39

* The strategy, meaning what are the logical steps that you would write again?

7:48

* It's defining the logical step that will be translated into the language that you will use.

8:00

* Meaning it could be a python, it could be R, a could be any language.

8:08

* So the program will ask the user for input.

8:14

* The user input will be checked first for the willingness to proceed.

8:17

* If the user will type done, will exit the program, will then test if the input is an integer.

8:24

* If not, the will go back and then it will check if the input is a multiple of five.

8:34

* If not, the will go back.

8:40

* Otherwise we print a positive message.

8:46

* This is a flowchart. Is the representation in the blocks, the logical blocks of the logic of the program.

8:51

* So generally speaking, you have the square blocks that are assignments, things to do.

9:02

* So ask user input. Those robots are the question that you will ask.

9:12

* So the conditional statements. So as a user input is done.

9:20

* Yes. And goodbye. And is the end of the program done?

9:28

* No. I am asking if is integer. If not the you and the message will be you enter a non integer value and you go back to EPS.

9:32

* You ask is this a multiple of five?

9:41

* If no you the message will be you enter a number that is now the number of five, a multiple of five and you go back if yes, you print.

9:44

* Okay. Uh. In reality, there should be a going back here that is not in this chart.

9:55

* So for each one of the alternatives, you want to have a value that will test that particular branch of the shot.

10:05

* So if the input is done in testing this conditional here, if the input is done, is done, then the output will be good by.

10:18

* If the input is x, y, z, then the output and testing this, the output will be under non integer value if it is a 61.

10:30

* And testing this condition here you enter a number that is another multiple of five.

10:44

* 95 will be okay. So with that, I tested every single conditional branch of my chart that it's basically in plain English this testing strategy.

10:50

* So that was the testing portion. Questions on this.

11:09

* All right. So let me go back to sharing it and let me go to the code.

11:20

* So does the code. The usual, while true.

11:28

* Lupo. So I'm asking the user to enter a number or a done to quit the.

11:32

* I'm checking if the input is done.

11:43

* It doesn't break. If it breaks, it will go here and it will print goodbye.

11:47

* If not, the will try to transform the input that is stringer into number an integer.

11:54

* If we get an error, we go into acceptor.

12:04

* Will print that your input was not a valid number of cents.

12:09

* Please try again and then continue.

12:12

* Meaning we go back to the beginning of the loop.

12:16

* If there is no except no error, I'm asking if.

12:21

* The value is less or equal to zero.

12:28

* I mean, that's something that is not in the flowchart that is an additional check was not required.

12:32

* But if it is negative, please enter a value greater than zero.

12:39

* Then I'm asking if is a multiple of five. If a is not a multiple of five, then you will get the message and go back.

12:46

* I'm checking also if it's more than 1000 again, you're not being asked to do that.

13:00

* If it is a print, please enter a value that's more than 1000.

13:11

* Continue. If I mean that all the ifs are negative, then you will print under that amount of sense.

13:17

* If you run it. So if you have a 22.

13:29

* You have not enter multiple of five.

13:36

* If you do. A certain number of hives.

13:40

* And if you do, minus two.

13:47

* And then if you do 55, you will get the message.

13:52

* And if you do, quit the. Oops.

13:58

* It's done. It will be goodbye. Sorry about that. Okay.

14:09

* So that's basically what was in the in the code, the questions on that.

14:13

* All right, so let's, uh. Anyone has any question on the assignment?

14:27

* I have a question, Professor. Go ahead. For the homework.

14:33

* Are we. Are we required to test of a floating number like a decimal?

14:36

* Well, I try to be as much clear as possible in the requirement.

14:43

* If it is not required by the requirements, you don't need to test.

14:53

* So in this case, the two tests less than zero, greater than 1000,

15:00

* that we are not required and you are not requested to do it for the floating or the integer?

15:05

* Well, it's pretty much the same in this case.

15:16

* So that being said again. So in this case I use the integer, but I could do the same with floating.

15:22

* So you don't.

15:36

* I mean that either one is fine.

15:40

* I mean, there is no difference. In this case, I know that is.

15:44

* I mean, I want to have an integer because if it has to be a multiple of five, it should be an integer.

15:49

* And that's why I used integer. But I mean that the problem would work as well if instead of Integer, I would have a float.

15:57

* Oh, I guess my question was do if I used the intention.

16:10

* Well, but I didn't, I wasn't explicit in the I guess that the testing portion in the bottom, I didn't put like 10.5 or anything like that.

16:13

* Is that is that fine or would we have points off taken for that?

16:23

* No, no, that's okay. Is absolutely.

16:28

* Because I got points taken off for, for not testing like 10.5.

16:32

* I will check that with you. I mean, again, if you're not required to do a test, you don't you don't have to do it.

16:38

* Then if you do it and it's working, that's good.

16:52

* If you do and it's not working and it is not good, then you will get some points off.

16:55

* But I mean, in this case, no one was asking for shaking her.

17:01

* If her assertion not meaning.

17:08

* I will take your. Great. And actually, that's not the best.

17:13

* All right. Still, I'm Professor. Yes.

17:24

* I just want to make sure I had the same thing. I had a note that said Float not tested and I had to take it off as well.

17:28

* Okay. I'm not sure if anybody else had that same issue myself also.

17:36

* Yes. Okay. I will review all of you.

17:42

* Okay. Thank you for telling me. Yep. All right.

17:51

* Okay. So that will be fixed. That again.

17:55

* If is not in the requirements, you don't need to do it.

17:59

* So I really tried to be as much clear as possible in the requirements just to avoid the misunderstanding of any kind.

18:04

* But things can happen. All right, so let's move on and let's go now.

18:17

* To the lights. It's 647.

18:27

* We are definitely on time. So we will cover two different topics, two very different topics.

18:32

* So one will be on working in teams.

18:40

* So all of you, most of you are professionals.

18:44

* You most likely work in teams. You know what the dynamics are.

18:50

* But I just want to revise to review with you some of the key points of working in

18:55

* teams in particular when you are developing software is engineering management.

19:02

* You may be in a position of managing people and in particular managing people who are developing software.

19:11

* So knowing how those environment may work and what are some of the key elements could be useful to you or to some of you, or for most of you.

19:20

* When you develop any organization, you have pretty much the same problems.

19:36

* So you need to hire people. You need to keep people on track.

19:43

* You need to motivate people. You need to eventually replace people.

19:48

* You need to solve the issues. I mean, the most recent software team that I managed, it was few years.

19:55

* I mean, apart from research in a company, so was a few years ago I was managing as a co-CEO and CTO.

20:04

* Not huge, but a sizable team of developers in a technology company in Central Florida.

20:19

* But the mission of the company was managing the tickets.

20:30

* So we manage the tickets for a bunch of clients, including Disney.

20:40

* So Florida. Orlando. Disney. That's why we were there.

20:46

* And some of the issues are.

20:53

* So I use that example because I had situations that are very common and they are kind of a good example on how those things can happen.

20:56

* So I will use that example a few times in these presentations.

21:09

* So when you have a. An organization on a project you may want to have a project manager or a team are doing the the management of the project.

21:14

* That can be called a million different ways, but pretty much in the one setting,

21:29

* the time being the drummer of the rock band, I wrote for Up with McKinsey for about two years.

21:35

* The company was very, very focused on organizations.

21:51

* Typically, what they recommend the most is a mavericks organization, meaning you have people,

21:58

* teams committing certain functions within the company and serving across the organization as a subject matter expert on that particular field.

22:07

* That's the Mavericks organization opposed to the hierarchical organization where you have each team with all the components.

22:21

* So let's say you have a team developing user interfaces, each team meaning each group working on one particular project within the company,

22:34

* they have their own sub team working on the user interface.

22:46

* The two organizations. They have pros and cons like everything in life.

22:53

* So the pros of a hierarchical organization is that you have more control of the resources you can use because you have your entire structure.

22:59

* The cons on the company side is that you have duplications because the same user interface expertise is in that the multiple teams.

23:13

* So. Hierarchy is okay in some cases, but you lose in terms of the optimization of the resources.

23:30

* The Matrix organization on the other end.

23:41

* It's great for optimizing resources, but the very end that each unit will become, it will serve multiple teams,

23:45

* meaning each team does not have a full control of all the components of the project,

23:59

* but they need to rely on an external to the team, internal or the company part for a function.

24:08

* One of the advantages of the Mavericks organization is that at the certain point you can outsource some of the components,

24:21

* meaning the you have a more, let's say, cluster of competencies than then in a second round out of a reorganization, I'd say you were a hierarchical.

24:31

* You move to mavericks. But people look. We'd say in competencies that we stay within the companies.

24:49

* And the certain point, you may decide that some of the functions are not core functions for your organization, and at that point you take them out.

24:57

* So I was working in a telecommunication company and in a certain point they outsource a

25:09

* good portion of the information technology because they decided that was not strategic.

25:18

* So they pretty much kept the the network and the marketing not so sure that they would be a good choice because

25:24

* I.T. is developing software that is developing the real differentiation elements for your company and outsourcing,

25:42

* that is, cannot be easy. And now that issue is also.

25:51

* I mean, when you outsource something, you want to reduce the costs,

25:59

* meaning the vendor providing the service will have a very strict policy of what they can do for you.

26:04

* If you are for anything cut. It is outside of the scope of work or they will charge you more.

26:11

* So at the very end you may lose control on the budget.

26:18

* So I mean, you don't need to go all the way to outsource everything that is outsourced by the Mavericks organization.

26:27

* That could be a preliminary step toward not the outsourcing.

26:38

* That's another example. When you develop software, you may want to have teams with high specialization in one particular area.

26:44

* So you always have a project manager setting the time, but then you have the different competencies.

26:56

* A little bit different is a. The support team or customer care.

27:09

* So the structure of a customer care is pretty much the same in many organizations, so not all and is in that honor on three levels.

27:18

* So the first level of the customer level one is basically the customer is calling and you have someone

27:32

* or something or you can have a both doing that will provide answers based on a script or a flowchart.

27:43

* So that's basically what you have at level one.

27:55

* Level one, no, the three levels is the one with the lowest cost because you can have an automatic system,

27:59

* you can have a people with no particular training.

28:06

* But the answer is that this level one can provide that may not be enough.

28:11

* So if they are not enough, then at that point that you go to the next level, the level two,

28:19

* where you have a people in the team that are with a better training and they are trained to functionally address the problems.

28:24

* So they don't know the code, but they know of functionally how the code is working.

28:36

* So when you go to level two, you have more course meaning.

28:43

* If you measure her the course by the time spent on each case, multiply that by the cost of the resort itself.

28:48

* Then that level to add will have more time and resources that will cost more.

29:00

* So. But they can see the meaning. They may not be able to address the problems that the user may have at the point that you go to level three,

29:09

* where you have a people trained on the software knowing the software and they are able to check if there is a bug,

29:20

* if there is something that was supposed to work in a way but is not working the way supposed to be.

29:32

* So obviously at that point, it will require more time and resources that are more expensive.

29:39

* So obviously you want to have up in level one as much powerful as possible, but that's basically the way it is generally organized.

29:47

* When you set up. Um, software development team.

30:02

* Uh, you go through the recruiting phase that can be intense sometimes.

30:08

* So you want to check, obviously the technical skills,

30:18

* but you want to check also the compatibility of the candidate with the rest of the team and with the culture of the company.

30:27

* I always use an example of the same people that it was mentioning the experience in central Florida.

30:37

* I will. I'm hired. I mean, I set up the hiring process looking for hiring.

30:48

* The team leader of the developers and I went through several candidates.

30:58

* At a certain point, the one candidate was my favorite.

31:09

* We had other people doing interviews with the candidate and then we hired him.

31:14

* From the technical standpoint, he checked all the boxes, but then at a certain point,

31:22

* his starting asking questions like all the people in the team are wearing shorts.

31:29

* I don't feel comfortable wearing shorts. I wear khakis.

31:36

* Is this a problem? I said, you know, a lot at Central Florida, people tend to do that.

31:41

* Where would you feel more comfortable with?

31:49

* Then the second time was a. I came to the office very early, 630, 7:00, but then I want to leave.

31:54

* 330 4:00. But the rest of the team is coming late.

32:03

* The 930, even 10:00, leaving late in the evening.

32:08

* Can we fix that? I said, you know what?

32:15

* Again, you are the team leader.

32:19

* You are supposed to be the first to come and the last to leave.

32:22

* So you want to be as much as possible.

32:27

* You want to spend as much as possible time with your people.

32:31

* So all of this is to say that you have the technical skills and what we call the soft skills.

32:37

* So in this case, this individual was not really compatible with the way we were working.

32:42

* So fortunately, we had a probation period.

32:49

* We used that period to replace this candidate with another one.

32:55

* So again, we had, I don't know, probably a total of five or six interviews, but we were not able to spot it.

33:06

* So right now it is a see it sequence.

33:16

* We are hiding faculty and we are having the same issue, I think, by the way, researchers and are having the same issues.

33:23

* So the technical aspects are important, but not the only important factor.

33:33

* So when you set up a theme.

33:42

* It's like a life cycle.

33:47

* So you have the formation of the team and the development of the team and the maintenance of the team.

33:51

* It's basically, again, like a life cycle in software development or in any product development.

33:56

* Each one of those faces has issues as a critical element.

34:05

* So when you create the team, you need to be sure that you again check not only the technical skills,

34:15

* but also the compatibility, the social skills that the individual needs to have.

34:27

* And when you work in a consulting company, I worked in consulting for quite a while.

34:36

* You may have a lot of projects coming and going and you have teams that you need to create each time.

34:43

* Sometimes, Sir is a small team. Some other times, depending on the size of the contractor, there can be a large team.

34:54

* So at that point, the time can be from few months to several years.

35:02

* I mean, if he's serious, you will have the same problem over and over because again, people can leave the company for any reason and replacements.

35:12

* They contract with the client can change. And you need to adapt somehow.

35:24

* When you create the team in particular, if you're working for a consulting company,

35:32

* you want to be sure that you maximize your revenues or your networks and you maximize the quality that you are providing.

35:40

* So you want to have a people with the right technical skills at the lowest cost possible with the maximum compatibility within the team.

35:55

* It can be a difficult problem.

36:11

* So some of the company, KPMG, is one that developed a software sort of a creating the teams.

36:14

* I mean, keep in mind that if you are a small consulting company, that's a minor problem.

36:26

* But if you are a large consulting company, international,

36:32

* sometimes you have team members that are coming from different countries because there are skills that

36:39

* are very specific and you don't have many of those in all the places across your what we call manage.

36:45

* So developing a software for that information may be a good investment that.

36:57

* We mentioned soft skills.

37:05

* That is something really important to keep in mind, that there are other elements that that on the soft skills side, but not probably a skill.

37:07

* So you need to read the candidate.

37:18

* You need to understand what are their personal ambitions and let's say longer term goals.

37:22

* And you want to be sure that those goals are aligned with what you can offer in that particular case.

37:32

* We have case it can be a general higher in the company or hiring someone parttime on one particular project.

37:42

* So let me keep all of that. So I really hope that this gave you a little bit of insight on working in teams.

37:56

* Again, the assignments that we are doing in this course are all individuals.

38:08

* The reason why I use individual assignments instead of team assignments, that seems to be more in line with what I'm talking.

38:16

* Today's is because I want to be sure that each one of you as exactly a good knowledge of what we are using in this course,

38:24

* meaning Python and the use of Python for some verticals or applications.

38:38

* But generally speaking, when you develop software, you don't develop software by yourself unless it is something as more or less specific and.

38:47

* And amendments apart from special cases doesn't really happen.

38:57

* Okay, so let's move on and let's go now.

39:04

* It's seven on seven and let's go to.

39:11

* The second part of the lecture that is a on Python that we will talk about the input output dictionaries and tuples.

39:18

* So two different topics. One, handling files and the other other types of variables that we are going to use in our coding.

39:28

* So, uh, unless you do something very specific, unless you ask everything to be type the by the user that you need to read.

39:43

* Five. So you need to take that data from the outside world, bring it to your script, process it, and then generate some output.

39:58

* So that's a normal way of doing things. Defiance that we will use the most.

40:11

* That would be text files or a comma separated value of the files.

40:17

* They are both, in essence, text files with the different skills we find so it can be handled by Excel or a similar province.

40:23

* So they are intrinsically matter that exceeds while the text files.

40:36

* I mean, that proper text file is like what you have in your screen that could be in a file and that would be at the five.

40:42

* So where each line is a record.

40:52

* So when you start working with the file with Python, you start with a statement that is open.

40:57

* So the open function in Python is the one creating and handle a pointer to the file that is in your computer.

41:08

* That's an example of a text file.

41:21

* So you have each line that is a recorder in the file.

41:24

* And each time you read one line, you will read, for example, the first line is this from Steven, whatever, that 2008.

41:32

* And the last one will be this early. So each one is a line when you read it.

41:44

* When you open, if I let the statement that open, that is required.

41:51

* So you create a variable that will contain the pointer to it to define.

41:57

* So we call it handle.

42:05

* So is actually a pointer and the statement that Auburn has a two parameter.

42:09

* One is the name of the file. The name of the file can be a variable or can be a string.

42:19

* And then the mode, meaning how you want to use the file can be for reading, for writing.

42:26

* So if it's a reading, it's our writing is w surprise.

42:32

* And if you are reading the file you don't need to specify.

42:37

* Ah, because the default for the reading for open is reading.

42:43

* But if you open for writing, then you need to specify that you.

42:52

* So that's an example you have open.

42:58

* This is the FILA. That was something similar to what you saw in the previous slide.

43:03

* You printed that what you get is really this obscure value.

43:09

* That is a pointer to the file.

43:15

* Other example, you have a text file with the three lions, so the quick brown and so on, a blank and goodbye.

43:25

* So you open the file with R, but again, this is redundant.

43:34

* You can remove it. Then I start a counter.

43:40

* I initialize the counter to zero and then insert the loop.

43:45

* So loop on the file on the content of the file.

43:49

* So the variable that I use in the loop is called the text line.

43:54

* Anything that handles in the pointer that I created in the first line.

44:00

* So the first iteration, it will have the first line and I'm increasing the counter by one and then I print the value of the counter and the line go,

44:05

* meaning the first round the line account that would be one, anybody would be the first line,

44:19

* then the second one will be a blank and third would be goodbye.

44:27

* Again, power is optional. You may or may not use it, and the handle is essential for the process.

44:36

* That's another example. We used in the previous example, the counter.

44:48

* That's a repetition. You can read the whole file.

44:57

* So you open the file again. In this case, I didn't specify the R, I mean I mean the mode, the meaning of the mode is R.

45:06

* Then I read the with the statement, the file, read the entire file into a single variable.

45:15

* And then if I bring Lang asking for the length of the content of the variable, it will be the number of lines.

45:22

* And then if I want to have the fourth part of the fourth, I mean the entire variable,

45:33

* meaning the first line I can use the partitioning method that we saw in the previous class.

45:43

* Another example. You have a bunch of lines.

45:55

* So this is a. Collection of tweets that I downloaded a few years ago.

45:58

* And that's an example of the of the user.

46:09

* So in this case, I opened the file that saw the file again, this one.

46:14

* So the goal was to counter or to have a list of all of the elements, starting with an aspect.

46:22

* So I initialize the list of elements with Ashtanga to zero.

46:38

* Then I loop into the file. I split the file to get a list.

46:47

* So again, just as a reminder, I am reading the entire string, entire line as a string.

46:56

* And then when split, I split the string into a list where the elements are the words.

47:04

* Because I didn't specify anything here.

47:17

* Meaning the criteria for splitting the variable.

47:20

* The string is this piece. If there is a space will be two different elements of the string.

47:29

* So at this point I have a list. So the list will contain all the words that you have in each line will line a good time.

47:39

* Then I set a loop within the list of words and I'm asking if the word starts with Ashtanga.

47:51

* If is starting with the hashtag, I will append the word to the list of stocks that they created as empty.

48:04

* When I finish this loop, I will analyze the entire line.

48:14

* When I finish the outer loop, I will complete analyzing the entire file.

48:19

* And then I will print the first ten. So in a loop, I could just printed the way it is, but it wouldn't look nice.

48:27

* And that's what you got. That's an example of application opening and analyzing content.

48:36

* We know. Continue. That is a way to do nothing and go back to the top of the loop.

48:45

* We can use if F in or not in.

48:57

* Again for selecting portion of the file or I'm in favor of the line within the file.

49:04

* Splitter. We mentioned that if you have a bad Ebola like this one.

49:16

* So that's what entities are missing.

49:23

* It's just to the code. So you split that by exclamation mark,

49:26

* meaning the string will be transformed into a list where each element is extracted from the original string based on the character, exclamation mark.

49:32

* So you have the first one, the first element, this would be Hello, no exclamation mark, because this is the element that is defining the separation.

49:48

* It's this one. Then the second will be this piece up to the op called this.

50:04

* And then the last one. If you don't specify anything, it will be split by space, meaning each element will be a word,

50:12

* keeping in mind that you are keeping all the punctuation that you can have in it.

50:29

* Because I mean, unless you remove it before doing it, the python will not know what the punctuation can be.

50:37

* Combining reading skills wi fi and a split the.

50:52

* So you open. You have this file with name and nationality.

50:59

* Then you open the file. You print nationalities.

51:04

* Then you start the looper into the file and you print the.

51:10

* I mean, you take the string, you split the string creating a list, and you take the second element.

51:19

* So because the elements are separated by comma in split, I specify comma.

51:30

* So karma is the character that I'm using to split the original string.

51:37

* And then I take the second element that is not number one.

51:45

* So printing nationalities at the first round is a joint American splitting, taking America.

51:49

* And then I have a space because even if we do not see it after each character, after each line,

51:57

* you have a special character that is new line trailing and she is the is a text file with the special character new line.

52:10

* I'm telling whatever is the area code that I'm using that you need to go to the next line.

52:22

* So you are just like eating right up. So but that's a character meaning when you go that next hydration that you will have a space.

52:31

* So if you want to remove it, then you need to use another functional strip.

52:44

* Strip is basically eliminating blanks and special characters left and right.

52:51

* If you do strip are it will be from the right. The strip l will be from the left.

53:00

* If you don't say anything. Just strip will take those characters out, both left and right.

53:07

* So if you do that, then you will remove the spaces that you have between the lines.

53:16

* Writing, If I look, is kind of similar.

53:27

* So you have the open with the W telling a side that in this case you are writing a file, not reading your file.

53:30

* I mean, one could be the default. You cannot not have two defaults.

53:44

* The default is R, meaning if you want to write, you need to write to add the W as a parameter.

53:48

* You read the first line. You want to add the new line because you want to keep it.

53:59

* Then if you want a separation, you would write a new line party.

54:05

* And then the third line, and that's what you have.

54:10

* And then you close it with Python, a two point X clause that was essentially after white writing.

54:13

* If you didn't have the the closer the actual right statement will never happen with python three is not essential.

54:22

* Other genera use it. Um.

54:36

* Let me skip that and then you have some information on these 3.6.

54:42

* If you replace 3.6 with whatever is the version that you are using, you will go there.

54:48

* If you water on other types of variables.

54:58

* We will talk about dictionaries and tuples. So dictionaries are a sort of data structure is a sort of archaic data structure in Python.

55:03

* We saw Lisa. So Lisa, again, defined by the square brackets.

55:23

* And inside you have elements that are separated by a comma.

55:30

* Elements can be strings can be other variables of any kind, including list.

55:36

* So you can have a list of these. I think we saw an example in the previous class.

55:46

* You start from zero to the end, meaning if you want to get to the third element, you will address it as a two.

55:54

* So two will be C. So the elements are ordered, meaning they will keep the order you use to create them unless you change it in a sort of explicit way.

56:03

* Again inside, you can have all possible types of elements.

56:23

* That's another example. These are mutable.

56:31

* Meaning you can change that.

56:37

* And in this case is a string.

56:42

* You have a banana. You cannot replace values.

56:47

* I think you final three. You could. And we mentioned that that you can change from upper to lower.

56:52

* We saw some of the functions last time in the list.

57:06

* You can change elements. In this case, I replaced the third element that was 26 to 28.

57:12

* You can get the length of the length of the list using land.

57:24

* You can create a list giving of numbers and giving the range, but you can specify the starting,

57:32

* the number of steps, how you want the numbers to be separated and what is the end.

57:41

* Or you can just do like in this case, four and you will have four elements starting from zero.

57:49

* There are quite a lot of additional functions that are built in in Python.

57:58

* So we may learn. You have Max. I mean some that they have the meaning that you can get.

58:05

* You can split again. You can split the variables pretty much any time, including strings.

58:17

* We mention that. We mention a little bit more of the split function.

58:27

* And now we have two bills, that the two bills are pretty much similar to this with two main differences.

58:34

* One, they have regular brackets instead of square brackets.

58:47

* And the second is they are immutable, meaning you cannot change the value of one of the elements.

58:53

* So once you create them, they will stay the way they are.

59:02

* So in this case, I'm addressing the second element.

59:06

* That would be if I tried to replace that B with an X, with an x, I will get an error.

59:09

* So Y and when you want to use tuples, so you want to use tuples.

59:18

* If you want to be sure that a value will stay the same across the entire program.

59:26

* So you do not want accidental change of the content.

59:32

* That's why you use a tuples. How many times you are going to use tuples?

59:38

* Not many, probably. But just keep in mind that that's an option.

59:44

* Dictionaries. As I was mentioning before, they are sort of a data structure, so they have a key and a valuable.

59:51

* So in this case you have the variable, the dictionary population, the so dictionaries are in brackets and you have a key in this case is USA,

1:00:03

* Italy, Japan, and you have the number and a column separating the key from the value and the number is the value for that key.

1:00:18

* So in this case is the population of a I mean is not exactly updated U.S.A. 380,000,380.

1:00:31

* Now Italy 59, Japan 127.

1:00:43

* So if you want to know one element, for example, Italy, you just do population Italy and you will get 59.

1:00:48

* So where we go here, like in this case do print populations in Japan and you will get you will get 127.

1:00:57

* You can get the keys into a list.

1:01:11

* So population keys should not specify anything.

1:01:15

* You have all of them. You can do the same with the values.

1:01:19

* So you can transform a dictionary into two lists.

1:01:23

* One list will be the list of keys.

1:01:31

* One is that list of values. The dictionaries are immutable and they are not ordered.

1:01:35

* Meaning if you have a. If you have the same dictionary you printed, not necessarily.

1:01:47

* You have the same order that you used in the infinite wisdom.

1:01:59

* The creators of Python, they thought that you don't need to keep the order because at the very end that you have a key and whatever the element is,

1:02:06

* you will pick it using the key.

1:02:18

* So if you do, if you create the list and you print in the most likely you will not get the same order that you used to create it.

1:02:21

* So if you want to add an element, you can do it.

1:02:31

* So you do population's name, other dictionary in the square brackets, the key and the value.

1:02:36

* And then when you printed the I mean, if we go anywhere, not necessarily at the end, if it was a list and you use append that it will go at the end.

1:02:43

* In dictionaries you don't have an upper end and there is nothing like a pen.

1:02:53

* Loops so you can loop into dictionaries.

1:03:01

* So. You can loop by element, by keys.

1:03:07

* You can retrieve elements using the key.

1:03:17

* That's another exemplar. I was that.

1:03:23

* So let's say you have a text file like this one.

1:03:35

* So we will read that one line at a time, split each line into words and we want to add one to the value.

1:03:38

* I mean that it's counting the number of times each word is appearing in the text.

1:03:51

* So you have one word. If the word is already in your dictionary, you will add one to the counter.

1:03:58

* If the word is not in the dictionary, you will create the entry for that word.

1:04:09

* So. Keep this in mind because you will use it.

1:04:23

* In the in-class assignment. So you have a file, you are opening the file.

1:04:29

* So again, is related to this file.

1:04:37

* In re again, you don't need to specify r you initialize the to empty a dictionary and you start reading the file.

1:04:42

* So one line at a time.

1:04:54

* Then what you do is basically strip the line left and right by special characters and blank spaces, and then you split it by space.

1:04:58

* So you will get. This line in a list where the first element will be in the last element would be the and with all the others.

1:05:13

* Then once you have the list, you start looping into the list.

1:05:27

* So you try if what counter there is a dictionary of that particular word, meaning you are using that particular word as a key in the dictionary.

1:05:34

* The first round up. The dictionary is empty, meaning when you try to address that particular element, you will get an error because there is no.

1:05:45

* Let's say you are in. In. There is no in.

1:05:58

* In the dictionary. So you will get an error.

1:06:03

* You will go here and you will create the first entry.

1:06:07

* So in the first round in that particular file, the first line that will be in fill, whatever is the end, and then the first word will be in.

1:06:10

* You will get an error. When you try to execute this statement, you will go into except the new will create a new entry.

1:06:22

* That will be an element where the key is that particular word in in this case and the value will be one.

1:06:31

* When you go to the second word, then the second word may be there, maybe not.

1:06:44

* The certain point that would be awarded there was already there.

1:06:49

* And then at that point that you will add one to the counter.

1:06:52

* So when you finish board the loops you will have.

1:06:56

* A dictionary with all the unique words and the number of times that they appear in the text.

1:07:03

* And then you create a list of the first ten, and then you loop it and printed.

1:07:10

* And that's what you got. But in this case, we have key and value.

1:07:23

* And that's what you have. Um, counting objects.

1:07:29

* Again, there's something that.

1:07:36

* I don't want to spend too much time because it's kind of similar to the previous one.

1:07:41

* Just briefly. So. You know, is that replacing it with that one?

1:07:47

* Okay. So that's basically it.

1:07:54

* Let me see if you have questions.

1:08:01

* Just have one quick question on the homework. Sure, yeah.

1:08:09

* For the procedure, it says to skip a line.

1:08:13

* I assume that we. Are we counting the first line in the files as the total line count, or are we ignoring it?

1:08:18

* Well, there are two ways that you can do that.

1:08:28

* One is to use a counter that is starting, let's say, mean that you normally initialize counter to zero.

1:08:33

* In this case, if you initialize the counter to minus one, then at that point the first line will not be counted.

1:08:44

* And that's one way, the other way. You can read the out of the loop, the first line, and then in the loop you will start reading after the first line.

1:08:54

* So those are two options that you may have down the road.

1:09:07

* We will use a different way to read the fine into variables.

1:09:12

* So we will talk next week about the. Data structures and in particular, pandas with those structures.

1:09:19

* You can just read the keeping the header, but in this case you don't have that meaning.

1:09:31

* You need to do either one. So either you use an account that you initialize the account that in a different way or you'll be out of the loop.

1:09:40

* The first line. Okay.

1:09:53

* So if we read or exclude the first line, it's still correct then, right?

1:09:56

* Yeah. Okay. Just making sure. I was trying to understand it fully, but to avoid any point loss.

1:10:01

* Thank you. Sure. Sure. Absolutely. Okay.

1:10:08

* So it's, uh, 730.

1:10:12

* 830. Um. It led us to 15 minutes of in-class exercise.

1:10:15

* So let me go. Let me share the screen.

1:10:29

* And let me go here. So the in-class exercise is a two portion one.

1:10:38

* Read the file. That is, name the names.

1:10:48

* The C containing a list of names. Count how many unique names that are in the file and print the results on the screen.

1:10:52

* Part two. You want to print the name that is use the the most in the input file.

1:11:04

* Keep in mind what we did.

1:11:14

* What we did here. We'd be pretty much the same.

1:11:22

* I mean, just consider that those prints are supposed to have parentheses and they have not in this example,

1:11:28

* but all the rest would be pretty much the same. Okay.

1:11:37

* So let me. Publish.

1:11:42

* This one, right? Yep. Let me publish that in class an exercise.

1:11:51

* With the file. Let me stop sharing.

1:12:00

* Create breakout rooms. So we have three breakout rooms.

1:12:06

* I'm opening it is a 740.

1:12:14

* You have about 10 minutes just to go into it.

1:12:19

* So I don't expect you to finish in 10 minutes, but you will have a sense of it.

1:12:24

* Okay. All the rooms are open.

1:12:32

* I'm posing the recording. Let's resume recording.

1:12:36

* So welcome back. Anyone want to share what you did?

1:12:43

* Again. You don't have to, but there is no judgment and there is no grading or nothing.

1:12:52

* Okay. If not, let me.

1:13:02

* Go here and let me go to.

1:13:09

* Resolutions. It's pretty much in line with what you saw in one of the slides.

1:13:15

* So I initialize the counter as I'm in dictionaries to zero.

1:13:23

* I initialize to a mean to empty dictionary to zero number or max frequency and the name for the word with the maximum frequency to a blank, please.

1:13:32

* Blank. Valuable.

1:13:48

* I think so. So and then there's the loop again.

1:13:54

* There are several different ways to go into a fight,

1:14:00

* and so that's another one that's so opening the file and the loop will continue till we reach the end of the file.

1:14:06

* So reading a line and then in the line that other loop stripping left and right.

1:14:19

* Then if the line meaning that particular name is in the dictionary,

1:14:29

* then I will add the one to the value and the dictionary meaning into the key, so that I will add the one to the value of that particular key.

1:14:41

* And then I'm checking if the value is greater or equal to maximal frequency.

1:14:55

* If yes, I will replace that value with the new value and I will add the name as the name with maximum frequency.

1:15:04

* Otherwise, I will initialize the value for that particular key to one.

1:15:17

* And then I will keep reading. But when I finish, I will get the entire fine process.

1:15:24

* And in those variables, the values that you will see in a moment.

1:15:31

* So there's the value of the dictionary.

1:15:38

* So you have the names and the occurrence.

1:15:43

* And then I used those two variables to get those two numbers.

1:15:48

* So those there's 54 is coming out of the number max three and this LE is coming from the name most frequencies.

1:15:57

* And it's basically. Okay.

1:16:15

* So for it's 757.

1:16:19

* I don't want to keep you for too long. So.

1:16:22

* Next assignment is a little bit more complex.

1:16:29

* So there are a few parts to the assignment.

1:16:33

* The whole assignment would be about those two files.

1:16:40

* So you have both the sides are related to the combined program and they are from different periods.

1:16:45

* So one in January, February 2016.

1:16:53

* And the other is April May 2016. So the structure is the same year, the duration, the day time, sort of time and a bunch of other information.

1:16:57

* So those are the files are relatively big, but you don't care much about the length because it will be either ten or 10,000.

1:17:11

* That million will be different, but the number is not going to affect much.

1:17:24

* So the first one, you will read the first file that is seeded by New York City City by January, February 2016.

1:17:30

* And you will do some processing. So before we move on, I want to drive your attention to one point.

1:17:43

* So if you look at in this case, that's the this is in classic set, say three meaning, is this guy here?

1:17:50

* And the file has to be in the same directory.

1:18:04

* So when the file is in the same directory, I don't need to specify the path for the file.

1:18:08

* Specifying the path for the file is not what you want.

1:18:17

* Because if you change a computer, meaning I don't know,

1:18:22

* you go to from a mac to a window so different structure that you move the file the

1:18:28

* program for one rectory to another one and then you need to do the same for the contents.

1:18:36

* I mean, generally speaking, I strongly encourage you to have files, data files and program files in the same directory.

1:18:43

* So I'm not going to take points off for the time being, but down the road I may do it.

1:18:56

* You don't want to use data structure management libraries like pandas.

1:19:05

* Not for now. You will use in in two classes, but not now.

1:19:11

* So you will basically go through the file and you will keep the count or number of lines to the count or number of lines with the customer.

1:19:19

* But just to be sure, you have a user type that can be a subscriber or customer.

1:19:32

* So you want to count the number of customers and the number of subscribers and each one will have a separate count.

1:19:41

* So N0 and zero or the number of lines and one or the number of customers and two for the number of subscribers.

1:19:50

* And you will print the last five lines. Then after processing, you want to calculate the percentage of customers on the totals.

1:19:59

* So you have the number of customers who have the total. You do the you divide the two and you will get by somehow.

1:20:09

* Then you will get the percentage. And then you will print the file as the end zero lines of those and one as a user

1:20:18

* type as customer end to end user type subscriber customer are the 1% of the total.

1:20:29

* But two is pretty much the same for the other five. So you will calculate, I mean, we already use from end zero two and two,

1:20:39

* two and three will be the counter for the number of lines, the second one and four for the customer and five for the subscribers.

1:20:50

* Same thing. You calculate the percentage.

1:21:01

* So this one was Z one.

1:21:05

* This is Z two. Part three.

1:21:09

* You want to compare those results? So you want to check what is the five that is bigger?

1:21:14

* What is the file with more customers or non subscribers?

1:21:20

* And then you want to write one page or interpretation.

1:21:27

* So what is an interpretation? Its narrative is plain English.

1:21:31

* You have one file that is from winter, one that is for spring.

1:21:37

* There must be differences. So are there more customers during the winter than in the spring?

1:21:45

* So look at the results that you extract, that you get, the mean you get from the analysis,

1:21:52

* meaning at those end zero two and five and use those to create the scoring.

1:22:03

* I mean, to write those few lines.

1:22:12

* These one page of interpretation again is a narrative describing, explaining in plain English the results of your scripts.

1:22:16

* Keep in mind that this sort of storytelling will be something that we will use a lot in most of the remaining assignments.

1:22:26

* We are not into writing code for writing code.

1:22:38

* We are on writing code to get insights from the data on the domain that we are analyzing.

1:22:44

* We are an engineer in management. So we want to use the tools to take a more informed decision so we can decide, okay.

1:22:55

* During the winter you have more subscribers and let's do a marketing campaign to.

1:23:05

* Incentivize non subscribers to user bicycle.

1:23:18

* I'm assuming that will have an effect in New York with the freezing temperatures anyway.

1:23:25

* But I mean it just to use the numbers to take decisions.

1:23:32

* But you want to have a narrative describing what are the findings.

1:23:38

* So you want to submit the three parts in I mean, the initial parts with the analysis of the first one,

1:23:44

* the second with the analysis of the second phyla,

1:23:58

* and then the comparison in one single piece, we file with the interpretation in a separate doc for PDF files.

1:24:03

* And that's basically it. So let me make sure that everything has been published.

1:24:17

* So I published that in class.

1:24:28

* Solution. And you have the assignment.

1:24:34

* So you should be good to go. Okay.

1:24:42

* Questions? Quick question, Vanessa.

1:24:50

* Yes, yes. That's a four part one. N0, if N0 equals N1 plus N2, then there's a good probability that we did it correctly.

1:24:56

* Is that right? Yes. I mean, keep in mind that it is possible that there are some blanks.

1:25:05

* Note some errors in the input or the file.

1:25:14

* But assuming there's no blanks, assuming there's no blanks, then means that there is no blank.

1:25:21

* Yes. Okay. Got it. Thank you. Sure. Other questions.

1:25:26

* All right. So thank you for staying with me.

1:25:36

* Still ahead on six. See you next week.

1:25:40

* If you have questions again, send me or you an email and we press that.