* Hello. Hello, everybody. Oh, there.

1:41

* Oh, no. So it's 630 now and we are right on time.

1:49

* All right. So, I mean, come on campus today and.

1:57

* Let's start so let me start with the screen in a second, but.

2:03

* Right. Okay.

2:15

* So I'm sharing the screen. And.

2:28

* All right. So we are in module five.

2:36

* Uh, you did the.

2:43

* The previous assignment. So for module module five, that's basically what we have in the agenda.

2:47

* So we will discuss the previous assignment.

2:56

* We will talk about data in a broad sense, then we will talk about Python libraries and in particular pandas.

3:01

* Then we will do the usual in-class exercise and then I will introduce the next assignment, the some comments and that will be the end of the class.

3:09

* All right. So the previous assignment that was this one was about working on.

3:22

* Who files. And doing pretty much the same thing, comparing them.

3:33

* So let's go to a pie chart and let's see an example of the.

3:39

* Luciana quoted. So it was three parts.

3:50

* The first part was on the CSB file.

3:54

* We wanted to keep it clear this.

3:59

* We want to skip the first line with the header.

4:03

* There are different ways of doing it. So one way could be to read the and I mean the the first line outside of the loop.

4:08

* The second is just initializing the counter to minus one meaning.

4:22

* Then when you will count, you will automatically skip the first one.

4:29

* So I'm initializing the counter at minus one again, not to count the first one to the header.

4:36

* Then I initialize a list with the subscriber list.

4:46

* I mean the counter of this. Could I borrow the counter for the customer and an empty list for the call that started the loop again.

4:52

* Obviously it would have been probably easier using pandas, but you didn't know.

5:06

* Officially pandas are up to today.

5:14

* Meaning we are using a C as B files and regular this.

5:18

* So I'm appending the ROE, meaning the line to the file to the lists.

5:26

* Then I'm adding one to the overall counter, checking if the in that row there is a subscriber.

5:34

* If yes, I will increase by one the counter subscribers.

5:45

* Same thing for the customers. And then I will continue till the end of the file and then I will print it out of the loop.

5:50

* The following are the last five lines of the file.

6:00

* So I'm going from 1 to 6 and then going from the end line of content to minus.

6:06

* And so that's the pointer. And then getting and printing the from the list the pointer,

6:17

* I mean the elements with you pointed by the pointer that at the very beginning will be the last one.

6:27

* And then because the.

6:36

* I mean, the point it's 1264 thing will be one meaning in the end, minus one year, minus two and so on.

6:42

* And then maybe add to the three to remove special characters and blank spaces at the beginning.

6:53

* At the end I calculated the percentage.

7:01

* I use it round with two decimal values.

7:07

* You don't need to do that, but it would look nicer. And then same thing for the subscriber.

7:13

* Print the results. Part two.

7:20

* Pretty much same thing for the other five.

7:24

* In this case, because I mean that in the first case, I was printing the last five lines here in print,

7:30

* in the first five lines, meaning I want to skip the first one because again,

7:38

* it's the one with the header and I don't want to print it and then I'm checking if it's less than five and then if so,

7:44

* I will printed the same thing, adding one to the counter.

7:55

* So for subscribers and customers and then increasing the overall count.

8:00

* Same thing calculated a percentage printing and less to.

8:06

* Part three. I'm comparing the two values.

8:12

* I mean, the two find the values for the two files. So the overall number of elements.

8:16

* And the percentage. If I run it.

8:26

* I have the last five lines of the file.

8:33

* There are 21 320 21,350 rows in the first file.

8:39

* Of those. Those are a user type customer and per subscriber percentage.

8:48

* Stay on file. First five lines. Total number distribution percentage.

9:01

* Spring riders are more than winter riders.

9:10

* During the spring. There are more customers to non-subscribers than winter.

9:14

* That's kind of understandable. So during the winter, you have more of the regular clients instead of those that I mean, just go and take a bike.

9:22

* It's winter is cold. All right.

9:40

* So that's basically it. Questions?

9:43

* Comments. Did you pretty much did the same.

9:48

* I didn't check your submission yet. So just want to be sure that is pretty much in line with what you did.

9:56

* I'm in in this assignment. The only thing.

10:07

* So let me share again. That we are a little bit more complex.

10:11

* We are defined.

10:18

* I mean, the fact that you need to keep one line so you need to initialize to either minus one or just read the first line out of the loop,

10:20

* then I mean that all the rest is pretty much straightforward.

10:33

* You could have probably done a function for dealing with the two sides,

10:39

* but we didn't introduce the functions yet that we will do today, meaning it wouldn't make sense.

10:45

* And then even with the function in the first case it was about printing.

10:51

* The last five lines in the segment was the first five lines, meaning the two were different anyway,

10:57

* but you could have a function to calculate the subscriber and the customer itself, but it wouldn't save much in terms of the processing.

11:04

* All right. So let me go now to the actual content and let me start just talking 37 slides, but we will go relatively fast on that.

11:16

* So Data, what we do in this course is pretty much extracting meaning insights from data.

11:34

* But what is data? I mean, it's kind of a generic term.

11:49

* So let's talk a little bit about which kind of data we can deal with.

11:54

* So it could be structured data like databases, data warehouses, things like that, or can be data that is totally unstructured,

12:02

* like the web, like other pieces of information, music videos, images that may not be in the structured.

12:15

* Databases or data structures.

12:33

* You may want to use it, but that's it. Most of the time when you have a.

12:37

* Structure, the systems for data, you deal with databases.

12:45

* So databases are a large integrated collection of data and they are handled by what is called a DB,

12:52

* a massive database management system that is a software that makes our lives easier to interact with the data.

13:03

* Both the ways. Writing data. Retrieving data.

13:14

* Updating data. When we work with data, there are several layers that we need to consider.

13:18

* So there is the physical layer, meaning a one in zero day an hour on your device, whatever.

13:29

* The device user could be a hard disk to be a remote storage.

13:37

* But at the very end that what you are going to have will be anyway one zero on a device.

13:43

* Then you have a how does one zero hour recorded on the physical device.

13:50

* So if you consider that you interact with that physical device more than once, you may not want to have all the data in one given place,

13:57

* but you need to have somehow a piece of software that will unlock the doors that one zero

14:11

* belonging to one particular file impositions that are optimizing the space you have on the disk.

14:18

* You have sort of cross-reference the role that for each file will point you or to the system to where physically the data are located.

14:28

* Then you have a conceptual schema of the data, what they are presenting because they are representing students.

14:43

* They are representing courses. They are representing clients.

14:52

* Representing whatever. So those are the logical view of the data that we can have.

14:55

* And then you have the views of the users.

15:04

* So if you are in a. Or the example.

15:08

* If you are in a university, you have the view of students, the view of administrative personnel, the view of the faculty.

15:15

* Each one will see only a portion of the entire data as a data base or dataset.

15:26

* So that again, you have a different linear from the very physical to the physical schema that is a layer provided by the operating system.

15:36

* Then you have the conceptual schema that is provided by the DB mass and then different views that again are provided by the DB mass,

15:51

* that order, that software. The advantages of having a DB a mass is that you basically do not need to worry too much about the data itself.

16:02

* So you just I mean, point to the logical.

16:19

* Entity and you will get the information.

16:27

* The master is also doing a bunch of other things, either letting you extract the data in the aggregation you want,

16:32

* or that this aggregation you want is also managing the integrity and the security of the dataset.

16:43

* But the integrity means you have multiple accesses.

16:53

* Again, concurrent access is an example, so you have multiple people accessing the same data, but you need to define a way to handle those situation.

17:00

* Typically there is a lock on the first one, and when the second is trying to access the same data cannot access it.

17:14

* But this lock is either handled by the DB crash recovery.

17:22

* What if in the middle of a transaction you have a power failure and you have no backup you in the power pick up unit.

17:29

* So at that point, typically DBM assets are doing all or nothing.

17:40

* Meaning either you have your transaction completed or you have nothing.

17:45

* So is not leaving in the middle. There are several of those.

17:51

* The. Those are some of the examples.

17:58

* Oracle is the one of the leaders that historically is one of the leaders, IBM, Microsoft, that are solutions that are open source.

18:02

* MySchool is what the general use, even if it is open source but is now owned by Oracle and the community is not very happy with keep using it.

18:15

* There are other options.

18:27

* I mean, the doubt is, yes, now is open.

18:33

* So it's at the certain point the Oracle can pull the plug and say, okay, it's not available anymore.

18:37

* Again, there are other options. Anyway, Dbmr is three main components.

18:45

* So a storage query process or transaction manager.

18:53

* So the storage manager is what is really dealing with the physical writing, updating.

18:56

* Deleting the data on your physical device, the query processor.

19:06

* What is it in charge for letting you talk to the data?

19:15

* Meaning creating queries, optimizing selections of data that you may have.

19:21

* Transaction manager is what is handling the transactions, meaning in error,

19:29

* either making sure that concurrent requests will be handled in the proper way.

19:36

* So those are the SC. The properties again stressing are pretty much the four characteristics that I was mentioning.

19:47

* Let's go now to the logic.

19:58

* I mean, it is a sort of a stack from the physical data to the very logical lot to the different views.

20:03

* So we are going up in this stock and we are now on the conceptual level and we are talking about a data model.

20:12

* So the model is primarily three main characteristics that are entity attributes and keys.

20:23

* The model has been created to meet the 1970s and by Peter Chen.

20:34

* And again, it's called entity relationship because entity relationships are the key ingredient of this representation.

20:42

* So in this chart you have the entities that are in the rectangular shape box.

20:52

* So there are two entities here, a shopper and item, and then you have relationships that are in the room, Boyds.

21:02

* And in this case, you have shopper buys item and then you have attributes so that are this kind of

21:12

* rounded shape boxes where the attributes are the characteristics of the entity.

21:21

* So shopper buys item item as three attributes, type price and sorts.

21:32

* So there's the basic example.

21:40

* In real life, things can be obviously way much more complicated.

21:44

* Thinks about large companies may have quite a lot of those entity and relationships making the old think a kind of complex.

21:48

* Do you really need to have that? In some cases, yes.

22:03

* So in some cases, they are required by law for the traceability of the data that you are handling.

22:06

* The Sarbanes Oxley that was created roughly 15, 20 years ago is saying that all the companies that are publicly traded,

22:20

* we really need to document processes and data that are related to money handling in a broad sense.

22:30

* So you have that you need to have a good representation of the processes and a presentation of the data has to be compliant to the requirements.

22:43

* One of the things that you may want to do with your data is normalization.

22:55

* So normalization, uh, instead of giving all the theory, let's use an example.

23:01

* So this table is from we don't needs to go,

23:09

* but the concept is still the same in this case is using Microsoft access but can be any read VMs and

23:16

* you have all that you have what is in the order you have with the supplier and with the customer.

23:27

* So you have three different things that are in the same table.

23:37

* If the certain point you have a customer who changed the address,

23:43

* you need to go through all the records, all the lines with that customer and change the address.

23:49

* It would be easier not to have three different tables, one for each one of those different objects, different entities.

23:57

* So you have one table for the order, one table for the supplier and one table for the customer.

24:07

* And then you have a way with the keys to jump from one to the other and eventually recreate what you have, what you had in the previous table.

24:15

* So that's a form of normalization that when you do when you work with the other basis, you want to have your data normalized.

24:26

* There are different stages of normalization, but what we need that in this case, in this course,

24:38

* that's for using this generic way of doing a normalization at the school or systems and enterprises.

24:46

* We don't have records on diplomats is and on as you have at the graduate level and I created an undergraduate course for.

24:55

* They the engineering that is rooted in as well.

25:09

* But it is the language that is used by relational databases.

25:17

* But it is again, undergraduate, I think is ISC to 25.

25:21

* But I mean, and since we do have courses on the computer science side of the of the Institute.

25:27

* When you have multiple of those databases each one Fogg used on one of the characteristics of the.

25:38

* Of your business, then you need to have something that will let you have an integrated view.

25:51

* So those a collection of databases are called data warehouses with different logical

25:58

* pieces of the data of your company in that you need then to have something,

26:12

* another piece of software that will help you put together the different pieces to have one single view.

26:22

* So this is the EDL that stands for extract transformer and loader that will take pieces from the different databases and provide an integrated view.

26:29

* EPL, I mean, is it a big business?

26:43

* There are companies that are thriving on it.

26:47

* Um. We mentioned that one of the features of the beams is their ability to interact with the data,

26:52

* to interact with the day that you need to have a language to deal with it.

27:01

* Edgar called the created the relational model and Escuela structured the query

27:09

* language and it is what we use to interact with the relational database.

27:17

* It's. Again that we don't have a course at the graduate level on actual ISC to 25 is the only court that is spending quite some time not on that but.

27:23

* As well is a language meaning like all the languages, like Python.

27:45

* It has all the structures that you expect in any language.

27:51

* Most of the time, you don't write entire programs in as well.

27:57

* You do only the portion that is directly related to the.

28:04

* Interaction with the database.

28:11

* But you could. What I normally do in my programs, when I have a database in board, I write the logic in Python and then I call that the query.

28:14

* Just asking one specific value that is the result of the processing that they that they did with Python.

28:27

* But again, it's basically up to you where you want to place the logic.

28:38

* If most on the python side, the moth on this other side or all in, you have all the possible degrees of where to place the logic.

28:45

* Um. I would keep those. Uh, again, there are several databases.

29:01

* My Askew and Postgres are two of the most commonly used as open source.

29:09

* Again, I using my ask you a lot, but they are pretty much the same.

29:15

* Keep in mind that the scale is sort of a standard, but it is not 100% standard.

29:20

* Meaning each one of the different VMs is may have a special instruction, special features that are working on Dave system and on Dave system.

29:32

* So this is intentional to lock the clients on that particular would be a mess, but you have a good number of common actual statements.

29:46

* So if you write your ask well in a basic format, it will run on all the DB masses.

30:01

* Now. Not everything is on relational databases because a relational database is a required, required structure.

30:14

* When you have data that is not structured, then relational database may not be what you really need.

30:31

* So the database that are not relational are called know as well, meaning a database as well.

30:40

* Assuming that all the relational are using as well, that is pretty much true.

30:50

* But the reality is that there are more and more unstructured data that would fit poorly on relational databases.

30:56

* So the main reason for people looking for non relational solutions is because the relational

31:08

* is they have limited flexibility because they have a rigid schema that you predefined.

31:19

* Think about the Excel you need to have the data that are in the format that is defined by your header to I mean upload new data.

31:26

* If you don't have it, then it is not going to fit.

31:40

* Um, some companies are using a sort of a combined solution, so you have something for the frontend and something for the backend.

31:44

* Facebook has a relational database for the frontend that's called Cassandra.

31:56

* They created it and then they made it open source.

32:03

* And then for the backend they have a relational database.

32:08

* I don't know exactly what they have, but is a relational one meaning information about, I don't know your characteristics.

32:13

* So the network of friends you have the about yourself and all the rest is on a relational database.

32:25

* But the content that you provide, the pictures, I don't know.

32:37

* Details, thanks.

32:45

* Then we go into Cassandra as a non structured and then there is software taking from the unstructured and populate or update the structure.

32:47

* There is no real definition of those known as well.

33:05

* Generally speaking, they are just non. As you well know, the election of I generally use MongoDB as a not relational database.

33:09

* I use MongoDB because the internal structure is based on JSON.

33:20

* That is a format for file.

33:28

* So that is pretty common in exchange of information between servers.

33:31

* Meaning if I have a if I upload, let's say tweets from Twitter,

33:38

* I will receive JSON files and I can just dump those JSON files to a MongoDB and they will fit right away.

33:48

* When you work with those relational databases, you don't have the same features that you have with the relational meaning.

34:00

* For example, you do not have a control of the integrity of the transaction, meaning those controls would be up to you.

34:11

* So you need to take care of the inconsistencies on the integrity of the other transactions and all the other good things that we are in.

34:20

* The list of the capabilities of a relational database is pretty much what you have on the web.

34:30

* So you get information based on best effort so they can be there or cannot be there.

34:40

* So it's up to your code doing it in the proper way.

34:47

* MongoDB Again,

34:52

* it's pretty much 1 to 1 with the relational databases making the transformation from the extraction of data from MongoDB database to a relational one.

34:54

* Pretty straightforward. And yet it is clear that another element that could be of interest is the MapReduce.

35:11

* So MapReduce is a technique for handling relational data databases, where you take the data, you map it with a B, it's a key value.

35:24

* And then when you want to retrieve, you reduce the.

35:43

* Pretty much you merge and sort the key values players and get the results as.

35:51

* I mean really using the the the differences that you get in this emergence of Hadoop

36:00

* is using the MapReduce approach in uni for a loop is not only a DMS like but is more

36:13

* a distributed resource manager that can be files but can be I mean processing resources

36:25

* so so we use it to put together a bunch of a mac mini that we had from a sponsor.

36:35

* So instead of running them individually, we can, we connected them with a layer.

36:46

* All of Hadoop, managing those resources for a sort of a parallel processing hub well was supposed to revolve, shoot up, revolutionize the industry.

36:53

* Um, but if you look at the numbers, the number of Hadoop, uh, systems didn't grow dramatically.

37:09

* The main reason is because when you need more storage capabilities, you go to the cloud.

37:25

* So you go to Amazon, you go to Google, you go to to Microsoft.

37:35

* They do have Hadoop for managing their.

37:42

* Storage system, but you don't see it. So you may use Hadoop without knowing it, but the number of systems is not so big as that.

37:49

* Initially, we don't. When you work with data, one essential step is the preparation.

38:02

* So you want to be sure that you prepare the data in the proper way.

38:11

* Data preparation in the entire process of extracting knowledge in a broad sense from the data can take up to 70 80% of the total time.

38:15

* That data preparation is essential because at the very end,

38:29

* the quality of the results of your analysis will be meaning the quality of the decisions that you will take based on the analysis.

38:33

* It's pretty much highly correlated with the quality of data that you provided.

38:48

* But if you have either a not enough data or a low quality, the analysis cannot be good.

38:54

* Generally speaking, there are three main steps in the pre-processing the data cleaning,

39:03

* meaning you need to deal with the missing values, you need to deal with the outliers.

39:14

* So those things or inconsistencies are all part of the cleaning.

39:22

* Then you need to do data integration. So the more sources you have, the more you know of the domain that you are analyzing.

39:28

* Markets today are so good in pinpointing our needs because they can put together a.

39:40

* Pieces of information about us from different sources so they can put together the beats that we visited through different tracking systems.

39:50

* Facebook is doing a lot of it.

40:06

* Sometimes they have pieces of software in your computer, and when you visit the page so that are in their network, they basically read those pieces.

40:10

* So they actually read the cache memory on your computer, downloading the entire history of your browsing.

40:27

* So that's the page that you visited.

40:37

* But then you have information from the credit cards on your spending patterns and then you have information on a,

40:41

* I don't know, easy pass on the, the, the traveling that you did.

40:51

* Then you have information from your phone provider and so on.

40:58

* I mean, different sources, different formats.

41:05

* But then when you do the integration, you could really go into the repeat in my own name with all the characteristics.

41:09

* And marketers can do that because there are privacy restrictions.

41:20

* They don't go up to that, but they could. So this is just to say, how important is data integration?

41:27

* Without data integration,

41:37

* you cannot really know enough of someone or some problem because one source will give you only a portion of the information doing the integration.

41:40

* It can be complex because the data may not have the same format.

41:54

* Format is not just the format, one is CSP and one less or something else, or one is from a database, a certain type and one another type.

41:59

* But sometimes that can be in the way the entities are named.

42:13

* So just a stupid, simple example.

42:20

* A street can be as Dot Street, the old world as the dot media in different ways.

42:24

* Stephen So can be Stephen's conceit of technology can be Steven's to a New Jersey and so on.

42:31

* I mean that you need to sort it out. I mean, the second part is more on the cleaning, but the first one, the three, is a good example.

42:41

* So that's an easy way. But sometimes you have even more complex than that.

42:50

* You may not have the names of the entity, then you need to extract the what the entities are from the context where you are.

42:56

* So those are issues with integration and transformation.

43:06

* We mentioned the normalization and we also want to aggregate different entities of different attributes that that you have.

43:11

* So if you have, I don't know, an event that is happening over time and you have the number of records and the duration,

43:20

* but then if you do the number of runs in that unit of time, you have more of an indication of what's going on.

43:31

* So that's an example of aggregation of data.

43:40

* So cleaning the integration transformation, those are the three elements that are really important when you do the pre-processing.

43:44

* So that's basically it. Let me stop for a second on this fourth part that is related to data.

43:56

* In a broad sense, questions so far.

44:09

* All right. So let's move on and let's go now to functions.

44:19

* Let me share the screen again and let's talk now about functions.

44:27

* So we mentioned briefly today about the possibility to kind of put together parts of the code, creating an entity that can be used multiple times.

44:35

* So those are functions.

44:51

* So functions are different types.

44:56

* So that are functions that are built in in Python functions that we can import in our code, functions that we can create.

45:00

* So examples of builtin functions are range some land.

45:10

* So those are building functions in Python.

45:17

* You can create your own if you create your own.

45:23

* You use death to define a name that is your own name, and then in parentheses the parameters that you are passing to the function.

45:27

* In this case, I'm passing nothing. And you have defined.

45:42

* Hello? Hello. I mean, like in the conditions.

45:48

* You had the colon and the indentation and then you have it in this case, print, high print.

45:54

* You might not do much more than that, though, when you call in your program.

46:00

* Hello. You will get the high gym. Then you bring something else.

46:05

* Then you call again. And we do again at the same thing.

46:10

* So basically it's a way to write once and use multiple times.

46:15

* So that's the basic reason why you want the functions.

46:22

* That's another example. So you have you define a function to calculate the area of a triangle and you pass the height and the with the

46:30

* two dimensions and you calculate the area and it is you apply the formula and you retard the value of the area.

46:41

* So if you pass those two values to the function like this one, then you will get the area of three triangle Iguala.

46:52

* I mean, you apply the formula, then you change the values, you use the same function and you will get a different value.

47:05

* So obviously if you use it just once, it would it may make much sense to use a function.

47:16

* If you use it multiple times, then yes, there is reason.

47:24

* That's another example. So square root.

47:30

* Same thing. This is a little bit more complex.

47:34

* And then when you want to use it, you just call it and pass as an argument parameter to the value that you want the function to use.

47:38

* Again. That's another example.

47:53

* This is a function with some ifs, so you will pass to the function, the language.

47:57

* If the language is yes, it will print.

48:05

* Hola. If else. If ops is f are not French.

48:09

* Bonjour. Otherwise is. Hello. So if you pass.

48:16

* If you call greet yen, you would get hello.

48:20

* If you do greet if our is.

48:24

* Bonjour. So he's not returning anything.

48:29

* In this case, this function is returning. Hello?

48:33

* I mean, that's a basic stupid function, but, I mean, explaining how it works.

48:38

* So if you. Hola, Greta.

48:47

* It will retain. Hello. And you will have the first one.

48:50

* Hello, Glen. Hello. Sending this ceremony one. You can do what we did here without return.

48:56

* With the return meaning you can pass up the language and then return either a whole bunch of hello and then that's what you have.

49:07

* So it's pretty much the same, but structured in a different way.

49:19

* Just as a recap. So you have in this case, the function is called Max and you pass a law.

49:26

* So the argument that whatever you want to call it is a award that will be passed to the function.

49:35

* The function will do whatever it's supposed to do and then will return W as a result.

49:45

* You can pass multiple parameters to a function.

49:55

* The parameter will be separated by comma.

49:59

* You can have a default value for the arguments.

50:04

* So in this case the function is make sandwich and you have meat cheese.

50:09

* That's the default value of cheddar bread.

50:16

* The default value. Right. If you. And then with those parameters that you print, I will have a sandwich.

50:19

* Me, cheese bread. So if you pass, we go instead of meat and you will have this outcome.

50:26

* Same thing. If you change the value, you will get the first one.

50:36

* I mean, then you will get whatever is the value.

50:40

* Like in the first example. If you pass two parameters,

50:44

* the second part of the first parameter will go into the argument that with no

50:49

* default and this tag on would go instead of the default if it was a third one,

50:57

* that will go to the third one. You can have a variable number of arguments.

51:05

* I generally don't use it because it can be a little bit confusing.

51:13

* But you could. Why and when you use functions, so you use functions for sure for a sort of an economy of scale.

51:17

* So you have the square root, you call the square root multiple times.

51:31

* You write it one once and then you call up multiple times.

51:36

* You can also use the function search to better organize your code, having sort of logical elements of what people use to code.

51:41

* With object oriented approach that's more familiar.

51:52

* There are other ways to doing it. We will say with the functions.

51:57

* Another important element is the difference between local and global variables.

52:06

* So local variables are variables that are, let's say, in a function and we in the function.

52:11

* When you go out of the function, they will have no value.

52:18

* So that's an example. So you have this function up with those values inside the function.

52:22

* I define that a as global. So outside of the function, not assign a, b, x, y, those values.

52:30

* Then I print before calling the function A, B, X and y, and then that's what I get.

52:40

* So I have one, two, three, four, that's what I have.

52:52

* Then I pass the 25 and 50 to the function and then I print after.

52:56

* So if I go I mean, when a calling I'm sorry, when I'm calling the function, I go into the function, I will assign those values.

53:09

* And then that's what I have. A is five, B is.

53:19

* Then I flip the X and Y, meaning instead of 25 and 50, I have 1525.

53:24

* Then I stepped out of the function print after a because a global the value that I set

53:32

* for a in the function that changed the from 1 to 5 meaning outside of the function.

53:43

* Now I have a stay at five, but b is not global.

53:53

* So when I'm out of the function would go back to the original value and same for the other things.

53:59

* So local, global sometimes can be used.

54:08

* Let's talk now about the imported functions.

54:16

* So you import function from libraries.

54:21

* So in Python, there are several thousand libraries that can really make your life easier.

54:24

* Pretty much you have libraries for anything that you may think about.

54:30

* No, quite so. But pretty much. Obviously is not solving your specific problem but is addressing the needs that they got.

54:37

* A million years ago when I was starting working in artificial intelligence, I was in what is called expert systems, so rule based systems.

54:45

* But we also did something that is pretty much in the real mode, machine learning.

55:01

* We did the algorithm some. There is one algorithm that is decision trees.

55:06

* We manually created the algorithm in our code.

55:14

* So that was either a Fortran was lisp was ops five and we created our own libraries.

55:18

* Now I import a library in my python code.

55:28

* I call the function for the decision tree passing the parameters and then they get the result.

55:34

* So writing a decision tree can be several lines of code now with two lines that import the library and call the function.

55:43

* And then so that's something that is making lives easier for coders, not solving the big problems of the world, but it's solving our small problems.

55:54

* So for example, you have a library that is called math, so you import the library and then you bring to the square root.

56:11

* So security is a function part of the library.

56:19

* So you call it the as the name of the function.

56:25

* And then you pass the argument and it's and you get the result.

56:29

* There are different ways for doing it. So the first one is what we.

56:34

* So you can. Import to only the function that they need.

56:39

* And then at that point, you don't need to write the math dope as Quixote,

56:46

* but just as are keeping in mind that when you import the function, you import the your the old function into your code.

56:51

* Some of the functions are big. When something is big, your code will become big.

57:02

* Meaning if you also deal with large data, you might run out of memory.

57:08

* So you may want to say importing only what you really need of big libraries.

57:15

* You can also use the nicknames. I mean, in this case doesn't make much sense because you are basically saving one character only.

57:21

* But if you have libraries with longer name than using a three character as nickname may be useful.

57:30

* But again, that's an application. And those are links.

57:40

* So let's talk about the I mean, we mentioned the building functions.

57:47

* Those are some of the libraries that are very common in Python.

57:54

* Amoeba is one that is kind of the most commonly used library for plotting for a basic.

58:00

* Graphs so you can have a. I mean, you imported with Matlock Lockley.

58:13

* But again, this case being longer P.A. is kind of useful.

58:21

* So in this case, it's sort of arch of a parable.

58:25

* You can do plotline. You can have parameters.

58:33

* Call auto market and style of line scatterplot, but shop by shot the histogram integrated view so you can have one single metadata.

58:37

* With subplots, you define the different subplots and you have one single view.

58:52

* Noon pie is another very popular.

58:59

* A common library is the root is the foundation of most of the scientific numerical libraries.

59:02

* One of the main characteristics is the use of another variable type that is array.

59:17

* So in canvas you have a linker kind of comparing arrays with lists.

59:29

* They are they look the same, but they are different.

59:37

* The primary arrays are used to deal with that exceeds so think an array as a mavericks and then can be one dimension multiple dimensions,

59:41

* but pretty much is what you really need if you want to use linear algebra pretty much.

59:56

* Those are examples. And thus it's probably the most common library in dealing with data,

1:00:07

* not because it is doing a lot of hardcore data science, but because it's creating data structures.

1:00:19

* Then you use the data structures for doing something else.

1:00:29

* Keeping in mind that is based on a known pie, meaning embedded in pandas that are all the hearts that you can get from that.

1:00:35

* From the nearby. So let's talk briefly on pandas and then you will do exercises on that and we will probably talk a little bit more next class.

1:00:50

* So the name is not from the animal, but it's from Python Data Analysis Library.

1:01:05

* Um, there are a lot of similarities between the general concept of table that can be as well Table Excel spreadsheet and the PANDAS data structures.

1:01:14

* There are two types of data structure.

1:01:32

* One series. One is a one dimensional data frames, two dimensional a most of the time we call data frame, not no matter what.

1:01:36

* So the name series is less used even if one dimensional, but it will be more besides pandas.

1:01:50

* We the library has been created for one specific company.

1:02:01

* It was in finance. And then the order asked that after a few years of use,

1:02:11

* I mean the exclusive use by this company to make it the open source and we are all happy that they agreed.

1:02:19

* In doing so you can import into pandas pretty much any form of data.

1:02:28

* So can be. This can be.

1:02:38

* She has defiance. It can be fired from a database so all of those can be imported into pandas.

1:02:46

* Dictionaries are a kind of 1 to 1.

1:03:02

* But again, dictionaries are a form of data structure.

1:03:05

* So I will not do much more than that on the pandas.

1:03:11

* You will have the slides. You will do exercises.

1:03:18

* I don't want to skip the the in-class exercise because it's already 730 and change.

1:03:22

* And I want to make sure that you will have the opportunity to do it.

1:03:35

* All right. Let me go for a second here and share this screen again.

1:03:43

* And. So the in-class exercise has two parts.

1:03:55

* So it starts with asking the user to input three numbers.

1:04:05

* You want to perform an automatic check on the numbers, and you want to use a function that you will create to do the checking.

1:04:14

* So no, each one is number or try except by the function doing for all of them.

1:04:25

* Then with those three numbers, the first part is calculate the factorial as a function for the first number that is being it.

1:04:39

* Then the second to you will use to calculate the floating value of the solution in the equation.

1:04:50

* The linear equation a multiplied by x plus b equals zero, where A and B are the second and third numbers.

1:05:00

* The second and third order of the three numbers, meaning X is equal to minus B divided by eight.

1:05:13

* So you want to write your own function for both the factorial and the linear equation.

1:05:23

* And you do not want to use it for this particular exercise except in a library.

1:05:30

* So let me stop sharing and let me make sure that you have it.

1:05:39

* Okay. I published the. The solution on the previous assignment.

1:05:54

* And. Okay.

1:06:05

* So. And publish the in-class exercise.

1:06:11

* Let me create the breakout rooms.

1:06:18

* Okay. There are four breakout rooms with two or three participants per room.

1:06:28

* Creating and opening the rooms and give you 15 minutes to work on it between 15 and 20 minutes.

1:06:34

* And then we will reconvene and we will discuss the solution and then I will introduce the next assignment.

1:06:46

* It is a little bit tricky and so I need to explain what to do.

1:06:54

* So opening the rooms, I will pause the recording and I will resume it after a 1520 minutes.

1:07:00

* Meaning in the recording. So we'll come back.

1:07:12

* Let's give people another 15 seconds.

1:07:17

* And so comments. Get in there. All right.

1:07:28

* So any volunteer.

1:07:37

* Anyone want to share what you did?

1:07:42

* Nope. All right, so let me share my solution.

1:07:50

* Make sure to screen. All right.

1:08:00

* So again, my solution is not the solution, but the solution.

1:08:07

* So just to be clear, as usual, obviously, I had more time to refine it, to present in the proper way.

1:08:14

* You had up 10 minutes or so. So that's not possible to be done in 10 minutes for most of us.

1:08:24

* But that's the way it is. So comments over here again, comments can be on multiple quotation symbols or single astorga.

1:08:32

* This is the function to check if the value is numeric.

1:08:47

* Is basically doing is getting the value then is trying floating of the could be into same think

1:08:52

* floating for the value if there is no error meaning is going here and or with no problem with it.

1:09:01

* Why, if there is an error will go into the accept and with have down and.

1:09:09

* This is the function for the factorial.

1:09:17

* So taking the number, checking if it's zero, the factorial zero is one and is returning one if the number is zero.

1:09:21

* Otherwise it will return recursively.

1:09:30

* The value of the factorization until the end of the loop quoted.

1:09:35

* Over here, I'm defining the function not to calculate the linear equation.

1:09:44

* And that's the formula, passing a list of numbers.

1:09:50

* So instead of individual numbers, they pass the entire.

1:09:55

* So what is this top number? So I initialize as an endless list of numbers so that the user will provide this.

1:09:59

* So we know that there will be three numbers. So I'm counting them up initializing to zero.

1:10:08

* And then while counter is less or equal to two,

1:10:16

* I will look for asking for a new numbers with just telling the user how many numbers he already entered.

1:10:20

* For each one I'm calling the is numeric function getting the result and either going back to asking again or asking the new one,

1:10:31

* then predicting the factorial. Just calling the function up, passing the first value, and then solving the linear equation.

1:10:47

* Passing the entire list. Keeping in mind that the function would take the second and the third values only.

1:10:59

* And then printing the solution and the process the mirror at the.

1:11:11

* But number, let's say it's five, then 22, 66, whatever it is.

1:11:20

* So I have a. I mean, just not is the fact that he's kind of a useful tool for the user to know where you are.

1:11:28

* So it's one out of three, two out of three, three out of three factorial solution.

1:11:38

* They are recreation and the liberals and that's it.

1:11:45

* Questions. All right.

1:11:51

* So let me go now to.

1:11:57

* The next assignment. Admin closed this section.

1:12:03

* Right. Share the screen again.

1:12:12

* Okay. The assignment is again on the CD biker bike sharing program in New York City.

1:12:19

* And in this case, we are going to compare two different finds, one from November 2021 and 1st November 2016.

1:12:29

* That. When you compare data from different periods.

1:12:39

* So even if they want to represent the same think or they may be different.

1:12:44

* So and this is the case. So let me go to the two files.

1:12:49

* So those are the two files that we have.

1:12:54

* So this one is the. Newest one.

1:12:58

* This is the oldest one. So in the older one, you have.

1:13:05

* Repudiation that is not in the new one.

1:13:13

* You have the user type that is either a subscriber and a customer where a subscriber is a subscriber,

1:13:18

* customer is someone vocationally like this guy here occasionally getting the bike in the new one.

1:13:28

* You don't have subscriber and customer, but you have a casual or member.

1:13:37

* So member a subscriber. Casual is customer.

1:13:44

* So again, semantically is the same thing but has been represented that in different ways.

1:13:49

* So again, you have two different files collected from Jersey City, one in 2016, one in 2021.

1:13:57

* They are pretty much doing the same job, but the format is slightly different.

1:14:09

* So that's probably the single most complex part of the assignment.

1:14:17

* So then, I mean, the things to do are kind of similar to what you did in the previous one.

1:14:24

* Uh, the indication that and giving you here are not using pandas, but you can use pandas at this point.

1:14:33

* If you don't use pandas, then you use lists. So each line meaning.

1:14:44

* Each one of those. Let's say this one up would be.

1:14:50

* One element of your list and with if you call the entire data set data then would be data want to do because you have

1:14:56

* to file to read the data to would be the third line that you will read and it will look something like this one.

1:15:09

* I mean, that is not from those files, but it will look similar to this one.

1:15:19

* So if you want the data, you need to go into number two, in this case, zero.

1:15:25

* If it is the fifth entry, it will be data for zero, meaning you are pointing to the first element or the fifth row.

1:15:36

* So that list of lists, that is kind of a usual thing to do.

1:15:46

* But again, if you use pandas, you can do it in different ways.

1:15:56

* So you want to create a function called the print details that will loop into the list,

1:15:59

* that will collect the the daily trip duration that, again, you have in one file.

1:16:07

* The actual duration. And for the other one you don't meaning either you take for both the difference between and in the started,

1:16:18

* or you take that for the second one and the trip duration for the first one is up to you.

1:16:33

* So. You, Luper, you take that reiteration and then you will.

1:16:41

* Create a blank line, you will print the average daily trip duration for the five most popular starting and ending stations.

1:16:53

* The number of members and casual users. So again, that duration is different to five members and casual.

1:17:04

* That's the way a user types are called in.

1:17:15

* The new one for the old one is a different way, so you need to create the same language.

1:17:20

* Compare that the results from the two files, the comparison that should create the elements to evaluate the ridership.

1:17:27

* I mean, how the ridership changed over the two years.

1:17:38

* And then you will print the end of the processing.

1:17:42

* You will write one two page report based on the data hub over to explain how the ridership changed.

1:17:46

* So that's pretty much it.

1:17:59

* Again, the most there are two complexities.

1:18:04

* One is the fact that the two fights are structured in different ways with the same informational content, but presented in two different ways.

1:18:10

* And the second, if you usurp and assert that is the complexity of working with pandas for the first time.

1:18:24

* Again, you can read into Pan thus. And if you don't use Pan thus working with the least obvious, that can be with some level of complexity.

1:18:32

* Okay. So let me. Publish everything.

1:18:48

* So you have the assignment.

1:18:56

* Okay. I would publish it. Now I can publish right away.

1:19:01

* I published the in class.

1:19:07

* Exercise. And I'm publishing all the material and the recording will be available as soon as a zoom will allow us.

1:19:13

* All right. So questions.

1:19:32

* Any professor who's having trouble accessing the pre pre-work module for module five, the pre class assignment.

1:19:37

* I think the ungraded email on there will be.

1:19:48

* Yeah. I um. I made it available.

1:19:54

* Can you check is available now? Yeah, I just do it.

1:19:58

* But I have to do that. Yeah. Yeah.

1:20:04

* I mean, that that's absolutely fine. I mean, also considering going into the zero four is not a problem at all.

1:20:07

* I'm sorry if sometimes I don't get back to you by the million things, but I definitely care for your comments and that definitely.

1:20:15

* That's fine. I assume I assume that you accidentally deleted it or anything.