Python and AI Power-Up Program Online Classroom-20250814_110429-Meeting Recording

August 14, 2025, 5:34AM

1h 28m 58s

◉ **Ajay Patel** started transcription

TJ **Tarun Jain** 0:10

So probably in Twitter, right? You might have seen many people add, hey Grov, can you explain this? So this XAI is probably a team which is under Elon Musk, right? And they have this Grov models, Grov 3, Grov 4.

Currently whenever we look at open source models, right or any reasoning based models specifically, you have something called as Arc EGI 2 benchmark.

So here if you notice graph as pretty high score compared to GPT 5, where is the scoring?

OK, that.

Bing all the time.

OK, they're still not have updated, but if we look at GPT 5 live stream right during live stream of GPT 5, they just showed their score of RKGI 2. It was a little bit lesser compared to group 4. So basically XAI also follows the same format of what open AI APIs.

Yes, right. So now what we will try to do is I'll just copy this chat completion.

I will copy the chat completion and now I will just define def chat completion. Here what I'll do is I'll just use arguments.

OK, and then what I will do is chat completion needs messages, right? So what I'll do is I'll just type print statement. I'll make it F string total message received.

And then I'll put a curly brackets. Here I'll just write length of arguments.

So what I'm trying to do is whenever we define chat completion, user needs to enter the prompts, right? That prompts. I'm not sure user how much prompts you'll write, whether it is 1-2 or three prompts that user will pass. So basically every single LLM model that we have.

Supports batching.

Right, if they don't support batching specifically, we can just add the custom component of chat completion just by copy pasting this client. Because once you add data class and you define your open edge chat rate, you can reuse any component of what that particular model does. So which we will cover in data class actually.

Margi Varmora 2:35
OK.

Tarun Jain 2:40
Right, so here what I'll do is I'll just show you one functionality of chat completion. Now I added chat completion. I have arguments. I'm giving user the flexibility to ask how many other questions you need. And here instead of using LLM, I'll just print all the messages.

Margi Varmora 2:58
Thank you.

Tarun Jain 2:58
For message in argument.

Margi Varmora 3:00
Yeah.

Tarun Jain 3:03
Message.
And whenever we call this particular chat completion, usually this should match how many arguments we have. But since we are using star, what you can do is you can just ask hi, good morning.
Then what is black hole?
And what is the meaning of life? So now if you see I'm asking 3 prompts even though I just have one single parameter here, but I'm asking three different things because I have a star here here if I print the type of.
Yeah, yes, it should be tuple.
And if I print this, it shows total message received is 3, then I have last tuple then I good morning. What is black hole? What is meaning of facts? So here I'm printing the message, but typically what you will do is you'll use an LLM call so you can just use response.

Equals to LLM response and then add that message here and then what you will do is you will typically return the response.
Uh, you understood? Uh, what's the importance of arguments today or?

4:27
Yes.

Hardip Patel 4:27
Yes.

Tarun Jain 4:30
We'll take one more example.
Um, one more example.
I'll just check for single star.
OK, So what we'll do is we'll go with keyword arguments. So in keyword arguments if you see it, we have open AI client and then you have client params. So I'll just copy this.
And here I'll define whatever I've copied and same syntax I'll do. What I'll do is how do you look now that we know if you put double a double asterisk, it is dictionary, right?
So how do we loop dictionary?

RamKrishna Bhatt 5:24
Key value.

Tarun Jain 5:27
But what is the syntax?
For if I write it will show me that I don't want to write.

RamKrishna Bhatt 5:32
4 key comma when.
Four key comma value.

Tarun Jain 5:39
4 key.

RamKrishna Bhatt 5:40
Omar Velib.

Tarun Jain 5:42
Value.
Ben.

RamKrishna Bhatt 5:46
In.
In dictionary I think.

Tarun Jain 5:51
So which is the dictionary here?

Hardip Patel 5:54
Oh, client parents.

Tarun Jain 5:57
Lane files.

RamKrishna Bhatt 5:57
I.

Tarun Jain 5:59
Ben.

Hardip Patel 6:04
We'll call in and then.

RamKrishna Bhatt 6:04
Colon.

Tarun Jain 6:07
Will we use colon or will this be something else?

Recall. So what are the functions that we have in dictionary? We have keys, we have values, then we have.

Mitesh Rathod 6:21
Items. Items.

Hardip Patel 6:22
Priya, that is 4 seconds.

Tarun Jain 6:23
It will be dot items. OK, so I'll just copy this.
And I will paste it here and whatever key we have.
User entered. I'll just say key.
And the value is value.
Or instead of value I'll the param.

Mitesh Rathod 6:49
Oh.
Item should be a function.

Tarun Jain 6:54
Oh, yeah, sorry.
So now, yeah, that's it.
What are the parameters that Open AI client has?

Hardip Patel 7:09
Model.

Mitesh Rathod 7:09
Modern.

Tarun Jain 7:13
Model.

Mitesh Rathod 7:13
And temperature.

RamKrishna Bhatt 7:14
Temperature.

Tarun Jain 7:18
I'll remove temperature. Let's remove temperature top P and top K Let's add some new this thing, some new parameters that we didn't see yesterday.

Hardip Patel 7:23
Yeah.
Um, you Max Max tokens.

Tarun Jain 7:30
OK, Max tokens. I'll add 1024.

Hardip Patel 7:35
Variance penalty.
Is it ways penalty or drop it drop it bias presence penalty?

Tarun Jain 7:42
Presents community.
Presence penalty should be somewhere between 0.5 to 1.

Hardip Patel 7:50
OK.
OK, login bias.

Tarun Jain 7:54
0.5 to 1.

Mitesh Rathod 7:55
Bobby.

Tarun Jain 7:58
Yeah, there is one logic bias. Who said this?

Mitesh Rathod 7:58
Dobby.

Hardip Patel 8:02
Got it.
Hi, Ajay Sir.

Tarun Jain 8:07
This is also a very important parameter, but you will mostly found find this to be none. Not many people uses logic bias, but when we use temperature this can be ignored and then we also have something called frequency penalty.

Mitesh Rathod 8:19
Yeah.

Tarun Jain 8:22
OK.

Hardip Patel 8:23
Date.

Tarun Jain 8:25
So frequency penalty also by default is none, but it should be somewhere around 0 point more than 0.7. If you put close to one, even that is fine. Frequency penalty is for closed source elements.

Hardip Patel 8:36
Mm.

Tarun Jain 8:40
For open source LLM, it is called reputation penalty.

Hardip Patel 8:44
OK.

Tarun Jain 8:44
Which we will cover when we will talk about how to use different parameters in open source cellulum. All right, so here if you see I define as much params I need. So if I come back to code here, I'm defining open AI client and then.
User is not sure how much params to pass right? So this params typically a user will just pass model match tokens or API key because this is the only things user will be aware of.

Hardip Patel 9:06
So.
Um.

Tarun Jain 9:14
API key SK dash dash dash. So now if I run this.
It will take all the patterns.
So this has to be in dictionary. So the key difference is whenever you're defining keyword arguments, you have to give a variable then a value for it so that it can convert into dictionary. But whenever you're passing just single argument, it is only a simple string.

Hardip Patel 9:28
Yeah.

Tarun Jain 9:42
Which will convert it into tuple.
We have list of models right? There is a function.
Here I can just print.
For.
Model in arguments print model.

Hardip Patel 10:05

Yes.

Tarun Jain 10:06

Supported models.

Then here I can *** as much as string as possible.

Hardip Patel 10:11

M.

Tarun Jain 10:16

Can you show the importance of arguments and keyword arguments now?

Hardip Patel 10:18

It.

Yeah.

Tarun Jain 10:22

OK, so this is something that you will see in most of the repo. I opened Agno because the code is very user friendly compared to Lama index or Lama index. Sorry Lama index or Lang Sin.

Hardip Patel 10:24

Yeah.

Tarun Jain 10:37

OK, so let's proceed with understanding the scope exception. I mean try and accept and file manipulation. Can everyone open the template code that we have? I'll probably share the QR code again.

OK, I'll share the link.

So here I've added some of the code understanding the scope and then exception and exception mainly try and except and then also file manipulation. I only have added just two variables on how to add append and read that we will write live.

Hardip Patel 11:44
Oh.

Tarun Jain 11:46
OK. Uh, anyone has any questions before we can proceed?

Hardip Patel 11:53
No, no.

Tarun Jain 11:54
This topic is really very important because when you're trying to building your library, most of the time we'll not be sure of what parameters to use. So I hope this will be very clear, but if this is still not pending, probably you'll have multiple examples when we do text pre processing and NLP. This is something that you'll be seeing.
That you will see most often right once we start working with the agents and drag.
OK, so we'll proceed with understanding the scope. This is again very similar to what you have in Java, then other programming languages like CC plus plus. And here if you see you have same variable one is equals to 10.

Hardip Patel 12:33
You.
OK.

Tarun Jain 12:38
If it is defined outside the function, it is referred as global variables, right? And if it is defined inside a function, it is called local variables. For example, here this a is 5 whereas this a is 10.

Hardip Patel 12:39
It is fine.

Tarun Jain 12:55
So can anyone tell me what is the output of these two now?
What is the output of this print A?

Mitesh Rathod 13:03
10 day the 10.

Margi Varmora 13:04
That's fine.

Tarun Jain 13:06
5 OK.

Mitesh Rathod 13:08
Right. Yeah.

Tarun Jain 13:16
So what about this?

Hardip Patel 13:22
10 yeah.

Mitesh Rathod 13:22
1010.

Tarun Jain 13:23
So if you don't define if you.

Hardip Patel 13:25
No.

Tarun Jain 13:30
Don't define.

Hardip Patel 13:33
To in strength.

Tarun Jain 13:33

Any value inside local it will consider.

Global variable.

Now here I'll do one more thing.

Here I will define a equals to globals. So let me just print what globals will do.

Print you.

Global start.

I just didn't eat.

Hardip Patel 14:20

Get.

OK.

Tarun Jain 14:25

We have named dog.

Hardip Patel 14:29

I guess items.

Tarun Jain 14:33

Where is items?

Hardip Patel 14:35

I think there is items uh globals dot items.

Tarun Jain 14:39

OK, OK, I got it. So it should be globals dot get whatever variable you are using here.

So now if you print here it should be 101010. You know why this is 101010 now?

Hardip Patel 14:45

Hello.

Mitesh Rathod 14:53

Yes, got it.

Tarun Jain 14:54

Because I defined AI as a global, then within my what you call within the scope one second.

Mitesh Rathod 15:01
OK.

Hardip Patel 15:03
M.
M.
OK.

Tarun Jain 15:22
OK so if you notice here I have a equals to 10. Then once I define a function which is a equals to five inside the function I'm overriding the a value again. So here now a what I'm doing is I'm trying to call global's value, then I'm doing get and then I'm using a.

Hardip Patel 15:29
It.

Tarun Jain 15:39
And once I print a whatever you return now this is a which is 10.
And since this is coming from Globus.
This is also a which is 10.
Is this clear?

Hardip Patel 15:58
Yes, yeah, yeah.

Tarun Jain 15:58
This this option. So let me tell you where this is useful. By any chance I've heard of a word called verbose by any chance, like in any workshops or anything.

RamKrishna Bhatt 16:00
Yes.

Hardip Patel 16:10

Yes.

Yeah.

Tarun Jain 16:15

OK, let me show some live example if I have data now this is.

So I'll show you. I'll tell you what I'm trying to show. So basically what will happen is in most of the agentic framework, right? Let's suppose I have a CSV file. OK, this CSV file has sales data.

Hardip Patel 16:45

6.

Tarun Jain 16:46

Like what sales you did in month of April, May, June and July. Now you have a user query.

Hardip Patel 16:47

Yeah.

Tarun Jain 16:56

So this user query is.

What is the average sales done in last?

Three months.

View analysis and a line chart. OK, so this is the user query. Now once you ask to the agent agent, what it will do is it will only generate the response.

But if you use verbose to be true.

Whatever action steps agent is taking right like this tools I'm using, then whatever reasoning it is doing in the back end, this will be visible to the user.

This will be visible to the user.

So what is visible? The locks is visible to the user like what is agent doing, what tools it is picking, then what was the response from the tool. So once this locks is visible to you if you want to extract any value. So if you see here what is the average is done in last three months.

Hardip Patel 17:56

2.

Tarun Jain 18:03

You can easily extract from agent response, but when you have to print the line chart, the image file is saved in verbose which is not rendered on the UI. So what you have to do is you have to get inside logs and then fetch the globals value to get the image.

OK, one second, I'll show the code only instead of.

I'll stop sharing my screen for one second. I'll directly show that code.

OK, uh, I'll share my screen.

Can you see the GitHub now?

Hardip Patel 19:20

Uh, yeah.

RamKrishna Bhatt 19:21

Yes.

Tarun Jain 19:21

OK, so if you see it, uh.

I I'm using an agent.

Uh, where is the agent defined?

So this is my agent. What this agent is trying to do is it will take the LLM. This LLM you can consider this to be GPT, claw, gemina, anything and then I have something called as data frame. So this data frame is nothing but my CSV file. OK and then I have verbose to be true.

Hardip Patel 19:41

You.

OK.

Tarun Jain 19:51

So what this verbose tool will do is whatever this agent is printing in the back end, it

will just showcase all the steps line by line. In simple terms it will show you the logs in the user UI. OK, so once you have this verbose to be true.
If you want to display the image, image is not directly used or image is not directly given by the agent. It will give you empty graphs. If you want to get the actual graphs, it will be inside globals right? So now if you see you're checking PLT.
PLT is nothing but plots inside globals and then you are extracting the code and saving it in a image.

Hardip Patel 20:24
No.

Tarun Jain 20:32
OK, so this globals variable is very important. So from where are we getting these globals? Either you can extract it from logs, if not you can directly extract it from any Python frameworks. So most of the time if you have logs you can extract the global variables.
But the only thing is you need to know what is the exact keyword you are searching for. Here I'm searching for A, whereas here I'm searching for PLT.
Does it make sense why I'm showing this?

Hardip Patel 21:02
Yes.

Ishan Chavda 21:04
Yes.

Tarun Jain 21:04
OK.
I'll write a line here verbose that is to extract the keys.

Hardip Patel 21:15
Yeah.

Tarun Jain 21:19
Value from the keys that's generated on runtime.

In simple words.
Extracting from logs.
OK.
Uh, is this clear right? What global variables and their local variable is?

Hardip Patel 21:46
Yeah.

Tarun Jain 21:47
OK, so now we'll go with exception, which is try and except. In some of the frameworks you'll probably look at. I mean some of the programming languages. Exception is mainly written in try and catch, right? But in Python it is try and except.
So for example, let's suppose.
I'm doing here print. I mean I'm asking for user input to enter a product price. Let's suppose for entering the product price I will enter a string. So now do you think you can add string plus number?
So let's suppose I have 300. I entered the price as a string, then I'm doing plus equals to 300.

Hardip Patel 22:25
It.

Tarun Jain 22:32
So will this print or will is this an error?

Hardip Patel 22:38
Error.

Tarun Jain 22:40
Error right. So one more thing. What you have to understand with input variable is let's suppose I'm defining a age.

Margi Varmora 22:41
Thank you.

Tarun Jain 22:49

What is your age? OK, I'll run this and for what is your age? I'm just writing 24.
If I print type of age.
What is the in type of this?

Hardip Patel 23:09

String. Oh, sorry, inter integer.

RamKrishna Bhatt 23:11

Integer.

Tarun Jain 23:14

You said string that is correct. The type of 24 is int, but if you take the type of input it is string.

Hardip Patel 23:17

Uh.

Tarun Jain 23:22

So this is very important to understand. Whenever you're asking input in Python, it will always return string.

Hardip Patel 23:31

Um.

Tarun Jain 23:34

OK, even if you ask what is your age, enter the price. Even if you enter the price saying 30,000, this type will be STR itself, right? So if you want to do type conversion in Python, let's suppose you have.

Hardip Patel 23:36

OK.

Tarun Jain 23:51

Tuple, right? Tuple equals to 10/20/30.
You can use list of. I'll make it tuple one.
It is converting it into tuple same. What I can do here is input is there right before input I can just add int.
And now if I run this, if I say what is your age, let's consider this was age.
I'm adding twenty-four type of age is integer.

Hardip Patel 24:24
But.

Tarun Jain 24:28
But let's suppose when I run this, if I ask anything else apart from integer, let's suppose I ask it's 24. Now if you print this, you'll get an error.

Hardip Patel 24:42
Yeah.

Tarun Jain 24:42
Why? Because this is fine, this is printing string, but after you print string you're converting that into integer.
So since 24 is not an integer, it is printing this particular error which is value error.
And now in order to tackle this, what I'm trying to do is I'm adding this inside try and accept block. So price is in input enter your product price and whatever price is entered you just have to pay extra 300 price plus 300 just total price. But if someone prints something else which is a string.
Or a float. I have an except block which will catch the value error right? Once you have the value error you will print this message. If you are not sure of what the error message is, you can just add except.

Hardip Patel 25:25
It.
It.
Yes.
Please.
OK.

Tarun Jain 25:35
Print.
Exception.
Except.
Those.
OK, you can just write except.
So now what I'll do is I'll just add.

Hardip Patel 25:55
Yeah.

Tarun Jain 25:59
Total price I'll return 5000.

Hardip Patel 25:59
Oh.

Tarun Jain 26:04
So what error will it?

Hardip Patel 26:05
No.
Error.

Mitesh Rathod 26:11
Value error, value error.

Tarun Jain 26:13
So if you see here when I ask input I entered 24 but I'm converting it into int. So int is returning value error right? So now what it will do is this particular line will return a value error. Sorry not this line, this line.

Hardip Patel 26:24
Yeah.

Tarun Jain 26:31

Because in input itself you are giving string. That string is converted into integer which is wrong which is value error. It will come and said please enter a valid price and now if I enter this it will tell please enter a valid price.
So try accept and then we also have something called as finally. Here what you are trying to do is you are entering a number and then you are dividing that number by 10. If I enter zero it will get inside 0 division error and it will tell error cannot divide by zero.
If you have value error then it will tell error please enter a valid number. So when will this occur? When I add anything which is string, let's suppose input enter a number I will add 10. This will give me an error.
Which is value error. If I type 0 it will get inside 0 division error and then it will print this and then we have finally no matter what happens finally statement whatever you write will be executed. So now if I print this.
I'll tell 10.
I got a value error which is please enter a valid number. Then whatever you write in finally it will be printed. This will run no matter what happens. Now if I run this again, if I give zero it will get inside 0 division error and then it will print this line.
Error cannot divide by zero.
And if I just give 10 it will then the try block and then it will directly get insert finally.
Is this clear as well or do we need one more example?

Hardip Patel 28:09

OK.

RamKrishna Bhatt 28:10

Just a small question. Is that possible to am I audible?

Mitesh Rathod 28:11

Yeah, good.

Hardip Patel 28:13

That's great.

Tarun Jain 28:14

Yeah.

Yeah, yeah, you're audible.

Hardip Patel 28:17

Yes.

RamKrishna Bhatt 28:17

OK, so just one question. Is that possible to define explicitly type for the input deck and similar to HTML we are preventing user to enter a string?

Tarun Jain 28:31

Uh, preventing in the sense.

RamKrishna Bhatt 28:33

Uh means in HTML we have uh input tag in which we will able to define a type number in which user will not able to enter any string. So is there is there any way for this input to prevent user from entering string?

Hardip Patel 28:35

You.

Tarun Jain 28:48

OK, in Python you don't have that type. I mean that you are supposed to explicitly mention only string, but you'll have to add this input statement under a conditional element block so that you can try based on these conditions, but directly using input statement that's not possible.

RamKrishna Bhatt 29:00

OK.

OK.

Thank you.

Tarun Jain 29:09
But this has to come inside if condition conditional statement to restrict it.

Hardip Patel 29:09
Yeah.
It.

RamKrishna Bhatt 29:13
OK, OK, got it.

Tarun Jain 29:21
Oh, any other question?

Hardip Patel 29:23
Rahul Kapoor's Rahul.

RamKrishna Bhatt 29:24
So one more question. So as we have seen that after entering 0 we will directly went to the exception exception in the zero division error. So does this means which means this this will be a keyword right? A zero division error for this try and catch.

Hardip Patel 29:34
It's.
It.

Tarun Jain 29:40
Yeah, yeah. So these are defined keywords. But if in case you don't know what error is occurring, you can have a common except block where you can just do print.

Hardip Patel 29:40
OK.

RamKrishna Bhatt 29:41
OK.

Hardip Patel 29:43
OK.

RamKrishna Bhatt 29:44
OK.

Hardip Patel 29:46
Yes.

RamKrishna Bhatt 29:50
OK.

Tarun Jain 29:54
Invalid choice.

Hardip Patel 29:56
So.

Tarun Jain 30:00
So even this is possible, this is just more flexibility like whatever feedback message you give, it's more related to the given error. But this is you have to figure it out by yourself kind of message.

RamKrishna Bhatt 30:01
OK.

Hardip Patel 30:01
OK.

RamKrishna Bhatt 30:09
Yeah.
OK, got it. Thanks.

Tarun Jain 30:21

We'll do one thing here. What I'll do is instead of num, I'll add dictionary.

I'll just tell.

Yeah, is 10. Sorry, 10.

Then we have B which is.

20 Now if I use numb of a.

Hardip Patel 30:45

Yeah.

Tarun Jain 30:47

Sorry, I'll use num of C So what is the error here?

What is the error message?

Margi Varmora 30:55

Yes.

Tarun Jain 30:56

It is key added.

Hardip Patel 30:57

I.

Tarun Jain 30:59

So do we have Kiera?

All right, So what will be the output now?

Hardip Patel 31:02

Invalid choice.

RamKrishna Bhatt 31:06

Evaluate trails.

Tarun Jain 31:06

If I run this, it is invalid choice. This will run no matter what happens. So if you're not sure of what the error message is, you can use except block.

OK, so now probably what we will do is we'll start with file manipulation. I have only given the input which is employee data. You have some name, then you have employee ID, department and salary and then you have file path which is employee record dot TXT.

Hardip Patel 31:28
Mhm.

Tarun Jain 31:35
So now what we have to do is I'll use three different things. One is.

Hardip Patel 31:42
OK.

Tarun Jain 31:43
Right.
And we have something called as read.

Hardip Patel 31:50
M.

Tarun Jain 31:52
And then we have append. So every single word has a shortcut command which is here. Then this is R and this is.
Right. So as of now, how many times have we seen colon we have seen for if?
Then we have seen it for Elif colon in the sense. Let's suppose I define a statement and that statement ends with colon, right? So if Elif else, then what are three more?

RamKrishna Bhatt 32:26
4.

Margi Varmora 32:27
Beth.

Tarun Jain 32:27

For while.

And of so we have total 6 that we have seen wherever we are ending the colon. Now we also have one more thing which is with.

OK, so with is a new command. When are we supposed to use with? Whenever you're supposed to open any file. So with open.

Here in open you have to give a file path name. So what is the file path?

Hardip Patel 32:50

Yes.

It's.

Tarun Jain 32:56

This is the file path which is employeerecord.txt.

Hardip Patel 32:57

No.

Hmm.

Tarun Jain 33:01

Now everyone, what you have to notice is can you see this directory icon on the collab?

Hardip Patel 33:04

M.

Mitesh Rathod 33:09

Yeah.

Tarun Jain 33:09

So I'll click on that and if I refresh I can only see sample data. There is no other file which is saved. So I'll just write with open file path. Then I have to give a mode. So what is the mode that I want to use? I can't read because there's no file, so I'll just.

Hardip Patel 33:17

Hmm.

Right.

Tarun Jain 33:29

Just use right which is W.

W and then you have to give as so as here stands for alias.

Yes stands for alias. So where will we encounter as most often? Let's suppose you have import, you have import Tensorflow.

Hardip Patel 33:57

But.

Tarun Jain 33:59

So every time instead of running Tensorflow you can have a shortcut command for this which is as TF. Now this entire Tensorflow is converted into a keyword called TF. So here also what we are trying to do is instead of writing this entire statement at once I'll write open.

Hardip Patel 34:14

HR.

Tarun Jain 34:17

Then Alias, I'll just write a keyword. OK, whenever you use yes, you have to add a new keyword following to that. So this can be file. This can be any name. OK and now what I'll do is I'll just type file dot.

Hardip Patel 34:26

OK.

OK.

Tarun Jain 34:34

Right.

this particular data, which is employed data.

So if you hover on this you will see a data type. I'll just hover on this. Can you see this? It is only taking one parameter which is STR and then it is printing int. So basically it is printing either zero or one. That means the file is saved.

Hardip Patel 34:41
Ex.
Oh.
Ajay.

Tarun Jain 34:57
And here what I will do is I will just add employee data and then write a print message saying data saved.
You understood. So we start with with and after with you have a command which is open at the file path and the mode and once you define this entire thing you have to convert that into a short form method which is as alias and define any name.

Hardip Patel 35:09
So.
M.
It.

Tarun Jain 35:23
This name can be anything. Whatever variable you are using here that only has to be used. The variable dot write and the employee data. So this employee data should be string.

Hardip Patel 35:28
OK.
Yeah.

Tarun Jain 35:35
Because write only takes string as the input. Now if I print this.

Hardip Patel 35:38
We.

TJ **Tarun Jain** 35:42
I didn't learn this.
Data saved and now if I click on directory icon if I refresh.
Whereas it saved.

Hardip Patel 35:56
Yeah.
Oh.

TJ **Tarun Jain** 36:04
OK, I'll just do LS and here I'll do cat. OK, now it has come.

Hardip Patel 36:05
OK.

TJ **Tarun Jain** 36:12
You have that particular data.

Hardip Patel 36:13
Delay.

TJ **Tarun Jain** 36:17
So this is how we define what you call writing the input into the file. This is very easy when it comes to files. Mostly we will deal with just TXT files. Why? Because for PDF we have different.

Hardip Patel 36:28
We.
OK.
Yeah.

TJ **Tarun Jain** 36:32
Frameworks. We have PDF plumber. Why am I suggesting PDF plumber? So most of the time, let's suppose you have PDFs.

Hardip Patel 36:33
OK.

TJ **Tarun Jain** 36:45
And in most of the PDF you have tables, right? Let's just imagine you have a table. Inside that table you have a value called 25 degrees Celsius. OK, you have degree. So I don't have a degree symbol in the laptop, but.
You have 25 degree. I'll just add 0 for time being. You have 25 degrees Celsius. Whenever you are working with RAG based application, you need to extract all the raw data from PDFS. So this 25 degrees Celsius will become 250 degree 250 Celsius. Right, so if you want to extract PDF in proper table format, there is one research paper which has done comparison analysis of different PDF frameworks. We have PDF plumber in that and then let me just check the spelling.
OK, there is no space in between.

Hardip Patel 37:52
Oh.

TJ **Tarun Jain** 37:53
You have PDF number, then you have unstructured IO. So unstructured IO it is both open source and you also have APIs for this. And then you also have just five PDF by PDF is the most simple best one.

Hardip Patel 37:58
Oh.

TJ **Tarun Jain** 38:09
But not many people will use it. If in case you're you need farshness and you don't have any table data then you can go with PI PDF right? When you have just text in your PDF then only you can go with PI PDF. But if you have table then either you can choose PDF number or unstructured IO. So this is for PDFS.

Hardip Patel 38:12
It.

OK.

OK.

Tarun Jain 38:29

Then for CSE you have import CSE which is completely different package again. And then for docs again you have a different framework which is docs to text and then for PPT also you have a different framework. So when it comes to opening the file.
In most of the cases it will be just TXT file.
So when will we use TXT files? Mainly again for saving blocks.
That after every single line you can save your data inside a file, which can be used for temporary purpose. But again, it's not practiced in most of the frameworks or not just frameworks, but also in your application. Most of the time you'll have your own observability tool where you will track all these applications.

Hardip Patel 39:19

Yeah.

Tarun Jain 39:36

By observability tool, in the sense you'll have something called as Langfuse, then we have something called as Phoenix.
And uh, have you guys known what is open telemetry?

Hardip Patel 39:48

A little, yes.

Tarun Jain 39:49

So when you use this saving the data, I guess not files anymore.
OK, so you understood the syntax, right? Mainly when it comes to file manipulation you'll have PXT files, but if you want to play around with PDFS, this approach you will have WB plus. So here B plus is nothing but binary.

Hardip Patel 40:16

OK.

Tarun Jain 40:22

Mod.

In file.

Hardip Patel 40:26

What?

Tarun Jain 40:28

So we have WRA, then we have WB plus which is for write binary. Then we also have RB plus which is read binary.

Hardip Patel 40:40

3.

Tarun Jain 40:43

So these are the five modes, but not many time you'll use this since we have specific frameworks for PDFCSP and docs separately. So now what we'll do is let's look an example of read file. So when you have to read your file.

Hardip Patel 40:45

OK.

Yes.

Tarun Jain 40:58

You again you can start with with open.

And here you have to define your file path.

And then now what is the mode?

What is the mode?

Mitesh Rathod 41:16

Read R.

Hardip Patel 41:20

Yeah.

Tarun Jain 41:21

Mode mode read is correct, but how do I define read? Just start then as you can again define file or since file is already defined, I'll use something else. I'll just define content. OK, content is a function.

Mitesh Rathod 41:25
But uh.

Hardip Patel 41:25
Alright.

Tarun Jain 41:36
Uh, I'll just tell F.
Then now instead of using directly function, since I want to read the content, I will save my response inside response.
And then I'll do S dot.
Reading.
And then what you can do is you can print the response.
And after every single thing you can do F dot close.
You understood this syntax. So once we open the file, we can close that particular file by writing F dot close and then response equals to F dot read. The reason why I'm saving inside a variable is because this read function will return.

Hardip Patel 42:12
So.
Yeah.
It.

Tarun Jain 42:27
String.
That is saved inside the file.
And once you return anything, you can just print the response.
It's just three to four lines of syntax.
But now let's suppose I want to add new data inside my file.
New.

Employee data.

And here I'll just give some different name Ms. Dhoni.

Hardip Patel 43:07

OK.

Tarun Jain 43:11

Employee ID will be 6, department will be sports and I'll just add one extra 0.

If I use W, whatever previous data you have will be gone. So whenever you use W, that means it will create the same file path name, but it will override the entire data.

So let's suppose I click on this and now if I do cat.

Hardip Patel 43:34

Yeah.

Tarun Jain 43:39

Employee data. This is new employee data.

So the previous data is gone, right? But if you need both previous data and as well as new data, what you can do is you can use EA. So I'll just copy this.

Hardip Patel 43:51

But.

Tarun Jain 43:59

And I'll paste it here. So if you want to append, which is to add the data in the existing file itself, what will you do?

Hardip Patel 44:00

So.

Tarun Jain 44:07

What is the mode that we have to use?

Hardip Patel 44:10

A.

Tarun Jain 44:11

Yeah, so as of now if I run this, I've added the new employee data. So here I'll just add employee data which was the earlier one and instead of data save I'll just tell appended.

Hardip Patel 44:12

But.

Tarun Jain 44:27

New data.

And then you can again do file dot close. So I'll just add add.

Add add and here the function will be same add dot write employ data so this will be same.

Hardip Patel 44:44

OK.

Yeah, close, close bracket. There needs to be.

Tarun Jain 44:49

So not quite.

So if I run this again now it will add same data again. So this is fine. I'll just run this.

You have the new information.

Hardip Patel 44:55

Yes.

OK.

Mm.

Tarun Jain 45:05

If I run this and run this again, I'll have the data again.

Is this clear? The syntax is similar. You have with and then you need to add open function. Open function takes 2 parameters. The first parameter is the file path.

Hardip Patel 45:21

Oh.

Cool.

Before.

Tarun Jain 45:31

Syntax.

So we start with with.

And open and in open we'll have file path.

And then we'll have mode. Then we define the shortcut which is alias and define any variable name.

And based on the mode you have to define what function to use.

dot what function to use. So for write and append it is write. For read it will be read or it will be read lines. So read lines. Let's suppose you have new lines that you are adding.

Then you can use relines, mainly if you have it in paragraphs, that is.

Your data is in paragraphs.

So what it will do is after every single line it will add back slash line.

Hardip Patel 46:31

So.

Tarun Jain 46:42

I'll add this inside triple quotes.

Uh, do you have any questions in files?

Hardip Patel 46:54

But.

Well.

No.

Tarun Jain 47:00

50 OK, I'll show you one more framework which is very solid when in recent days what it has. It's called Mark it down, which is by Microsoft.

Hardip Patel 47:01
It.

Mitesh Rathod 47:02
No.

Tarun Jain 47:12
I'll also share this URL in chat because it's very useful.

RamKrishna Bhatt 47:21
One question then So what if we will not do uh uh close file command so the file name will close?

Hardip Patel 47:22
Somewhere here.
See.

Tarun Jain 47:30
So nothing major will happen, but the only thing is it if anyone is supposed to. Let's suppose you're building the same file and you're deploying it in Streamlet, right? And most of the time the Streamlet files are publicly available.

RamKrishna Bhatt 47:30
M.

Tarun Jain 47:46
So you can just add new data in that particular file by just reusing that particular code. So that's the reason why most of the time you just close the information so that the logs are saved properly. So I will show you something.

RamKrishna Bhatt 47:46
OK.
OK.
OK.

Tarun Jain 48:00
So here this is the stream data. The reason I'm telling stream data example is because it's very relatable.

RamKrishna Bhatt 48:07
Mhm.

Tarun Jain 48:09
Otherwise is it not showing locks?
I'll just show one example so that it will be clear.
What is happening?
Alright, let me just check which app is currently running.
OK, meanwhile, since this is loading, what I'll do is I'll show Mark it down. So recently Microsoft released their framework called Mark it down. So this is mainly used for file manipulation itself. I'll show the code. So if you see it, you just have to import from Mark it down, mark it down.
You have to define what plugins you need and then you just have to use one single line. That's it. MD dot convert whatever file path you have. So this can be Excel, this can be PDF. It supports so much what you call path. So if you see here it supports PPT.
Docs, Excel, then XLS, PDF, TXT. When you have YouTube, there are so much file support. So what this is trying to do is once you upload a PDF or an Excel, it will convert the entire data into markdown format, right? And most of the time when you are working with AI models, mainly GPT or.

Hardip Patel 50:27
OK.

Tarun Jain 50:27
We do right now, we are using market now.

Hardip Patel 50:28
OK.
You.

TJ **Tarun Jain** 50:33
So this framework is very important. Probably I don't think so. We'll be using it in our session, but if I get a chance, I'll definitely show two or three examples in drag. So we use this in one of our product mark it down where you just upload.
PDF and that will convert it into markdown and then you add that markdown syntax for chunking and whenever I ask any question it will extract the chunk which is in markdown format. So this is very useful.

Hardip Patel 50:50
We.
Yeah.
It.

TJ **Tarun Jain** 51:04
So we can just save it somewhere so that if in case you are working with files and you need markdown format, you can reuse this framework.

Hardip Patel 51:04
OK.

TJ **Tarun Jain** 51:17
So can you see this logs here? So now what will happen is during this time most of the Streamlit tab the logs is public. So if the files is not closed, you can check all the file output in this particular logs right? So every single time after you use read.

Hardip Patel 51:21
Yeah.

RamKrishna Bhatt 51:32
OK.

TJ **Tarun Jain** 51:35
We can close that particular file. If not, there is a data leakage over here.

RamKrishna Bhatt 51:39
Mhm.

Tarun Jain 51:42
It's just to avoid the data leakage.

RamKrishna Bhatt 51:46
OK, got it.

Tarun Jain 51:54
OK, so this was basically the CSE agent. So you just upload a CSE file and ask the question on sales and all which we will also build when we work with agent because this is the project that I've added in the agenda as well. OK.
Oh, any questions we have in files?

Hardip Patel 52:13
Uh uh, about the PDF? Uh is uh PIMU PDF uh OK or?

Tarun Jain 52:20
Which one?

Hardip Patel 52:22
So like there is IMU PDF.

Tarun Jain 52:26
Ha ha. MMU works. Yeah, yeah, I remember. So that is also good. But PDF lumber is much better. I'll just show PDF.

Hardip Patel 52:29
That is a.
OK.

Tarun Jain 52:39
Comparison.

Data extraction.

Yeah, exactly.

Hardip Patel 52:57

Oh.

Tarun Jain 53:11

So you meant this one, right?

Hardip Patel 53:13

Yeah.

Tarun Jain 53:16

Yeah, so by this thing, in most of the cases it had lesser for. If you see, we usually use this PDF plumber.

But pipe MU PDF, I don't think we got much better result, but even that is there. And again this Kamal, it's loading the model which is super slow. Again, since this is focused on the their own folks, so they have better results, but it's super slow.

Hardip Patel 53:29

Thank you.

Yes.

OK.

OK.

Right. Uh, for PIMAPDFI think they added the the possibility to give LLM to parse the PDF. So that's why I was.

Tarun Jain 54:04

So we typically if it is related to tables, we either use unstructured IO. In unstructured IO you have two modes, one is fast.

Hardip Patel 54:11

On.

Tarun Jain 54:15

And one more is OCR something, so we use fast, so that is that in one product. But when we did the comparative analysis, we mostly found better results with PDF lumber. So only I directly showed that name. I don't think we got good results with.

Hardip Patel 54:16
Mhm.
OK.
OK.

Tarun Jain 54:32
PIMO PDF. This was worse PIPDF.

Hardip Patel 54:35
Obviously, uh.

Tarun Jain 54:37
Unstructured IO and PDF lumber had very close results.

Hardip Patel 54:42
And so the follow up is that how is this compared to the close options like tax rect or maybe?

Tarun Jain 54:55
Yeah, you have llama parts.
So Lamaparse is something new.
Something new.

Hardip Patel 55:01
Mm-hmm.

Tarun Jain 55:04
So again here probably if you're using table based then you can go with llama parts. If not you can go with unstructured IO as well. Unstructured IO as I said you have open source version as well. You have the API version as well. If you go with API you'll get much faster results.

Hardip Patel 55:10

OK.

OK.

Tarun Jain 55:21

But if you're going with open source, it's bit slow, but since all the document processing that we do is offline process.

Hardip Patel 55:30

OK.

Tarun Jain 55:35

So what do you mean by offline? It's like uh.

Hardip Patel 55:38

Yes.

Tarun Jain 55:40

Is over.

Asian tech.

So our structure is probably similar like one of our core.

Hardip Patel 55:51

Sachin Patel.

Mhm.

Tarun Jain 55:58

Why is it taking? There is one flow chart here which is very useful. So if you see here, there are two major components. One is offline document processing. In document processing, what they're trying to do is you extract all the data and these folks, what they're trying to do is they're converting it into a Google Docs.

Hardip Patel 55:59

Yeah, yeah.
OK.

Tarun Jain 56:17
So they have PDFS policy, Uber, but they're converting it into documents so that they can extract it properly and then they're doing metadata filtering. So if you look at this entire process, this is happening offline. It's like even if it takes too much of time, it's fine because.

Hardip Patel 56:18
Right.
Stop.

Tarun Jain 56:35
User is not able to see this. Once it is saved in Vector DB, you're just reusing the Vector DB. So in our case as well, when we use unstructured I/O, we are using open source one. It's a bit slow, but it's fine. Even if it takes one hour, it's happening in our.

Hardip Patel 56:48
OK.

Tarun Jain 56:51
Uh, local Uh setup.

Hardip Patel 56:52
Right.

Tarun Jain 56:55
So if you want fast, like if you're doing something on the real time, then you can go with the API version of unstructured IO or llama parse.

Hardip Patel 57:05
OK.

Tarun Jain 57:07

I don't think so. You can share the evals that we have, but I'm pretty sure whatever I've written here makes sense.

Hardip Patel 57:15
Next.

Tarun Jain 57:16
Because we we did get good results with this too.

Hardip Patel 57:19
That's fine, I guess.

Tarun Jain 57:23
Yeah.

Hardip Patel 57:25
Thank you.

Tarun Jain 57:25
This is for just POC.

Hardip Patel 57:32
Have you tried to use the Microsoft that document and the AWS?

Tarun Jain 57:36
Yeah, document intelligence for this one, right?

Hardip Patel 57:41
Yeah, and the text.

Tarun Jain 57:44
This right we don't use. I've used Document Intelligence loader, but the pricing didn't make sense so we didn't use it.

Hardip Patel 57:51

OK. Sorry, I am working, so that's why I'm asking so many questions.

Tarun Jain 57:52

Even this.

Yeah, we do have used document intelligence loader, but for us the pricing didn't match well, so.

Here in Marketon also they support document intelligence loader. Where did that go?

Yeah, this thing.

Hardip Patel 58:18

Yeah.

Tarun Jain 58:20

Yeah, this is something that we'll also cover in drag.

Hardip Patel 58:24

OK.

Tarun Jain 58:25

OK, so any other questions in files?

Hardip Patel 58:34

No, I guess.

Tarun Jain 58:36

OK, so I have to check out from hotel, but what we'll do is let's cover these modules and then we'll wind up. So can you see this code in template? So what we'll do is we'll start with modules. So basically modules is nothing but.

Hardip Patel 58:48

Yeah, OK.

Tarun Jain 58:56
From now on we will be using existing code which is already available.

Hardip Patel 58:58
M.
64.
Yes.

Tarun Jain 59:08
You can also call this as package or it's referred as modules and some people also call this as libraries.

Hardip Patel 59:14
Um.

Tarun Jain 59:18
Right, so we'll cover these two. One is import random and one more is import math. So here what we will do is I'll just add a range of numbers.

Hardip Patel 59:24
Yeah.

Tarun Jain 59:33
From 1 to 100. So if I want to convert this into into list, how should I convert it? What is the syntax to convert these numbers into list?

Hardip Patel 59:34
Oh.
Yeah.
OK.
Oh.
List, I guess like a star.

Tarun Jain 59:56
I just have to add.

Hardip Patel 59:56
OK, no, just the wrong list, no.

Tarun Jain 59:58
close it here. That's it. So now if I print numbers, it will go from zero to 99. Why? Because the last range will be excluded.

Hardip Patel 1:00:02
Oh.
Hello.

Tarun Jain 1:00:11
OK, so now if I print the length of numbers, what will be the length?

Hardip Patel 1:00:13
It.

Tarun Jain 1:00:18
99 Why? Because I'm starting from 1 instead of 0, so it will go from one to 99. If I just give 100 then it will be 100.

Hardip Patel 1:00:21
It.

Tarun Jain 1:00:27
then it will be 100.
Because it is starting from zero to 99.
Alright, so now if in case you want to pick random numbers from this.
Guess equals to.
Random dot choice.
So just like what we do in Open ER, right? Most of the frameworks you just have to

import. Once you import, you can use the function. So how do we check what functions that one variable has?

Mitesh Rathod 1:00:59
Yeah.

Tarun Jain 1:01:01
DIRI can just use random.
Now if you see you have choice, you have Gauss, then you have log, then you have random, then sample seed. This seed is very important. I'll tell you the importance of seed. Then we have shuffle, then we have uniform, right? So now what I'll do is I'll do random dot choice and now I'll just.
Pass whatever I showed earlier.

Hardip Patel 1:01:26
OK.

Tarun Jain 1:01:28
Which is numbers. Can anyone tell what can be printed? Like just take one random guess. If you tell correct, I'll give you a T-shirt when I come to Barodra.

Hardip Patel 1:01:39
50 Fifty, 49, 30 something.

Tarun Jain 1:01:43
OK, 49495037 last two mortgage.

RamKrishna Bhatt 1:01:49
Oh, bye.

Tarun Jain 1:01:51
Hi.

Margi Varmora 1:01:52
B.

Tarun Jain 1:01:54
Award.

Margi Varmora 1:01:56
Please.

Hardip Patel 1:01:56
12.

Margi Varmora 1:01:58
Yes.

Tarun Jain 1:01:58
3.

Margi Varmora 1:02:00
Yeah.

Hardip Patel 1:02:00
I need it's not gonna work.

Ayush Makwana 1:02:02
39.

Tarun Jain 1:02:03
OK, last.

Hardip Patel 1:02:05
M.

Tarun Jain 1:02:06
There are, I guess, 9 folks.

Margi Varmora 1:02:08
19 was very good.

Tarun Jain 1:02:09
It should be 8 numbers.
19 so 12345678. OK, we have 8 numbers. It's 84.

Hardip Patel 1:02:17
3.
Yes.

Margi Varmora 1:02:19
Is.

Hardip Patel 1:02:23
Oh.

Mitesh Rathod 1:02:24
No, no, that.

Tarun Jain 1:02:28
OK, who said three?

Hardip Patel 1:02:28
Margi Margi.

Tarun Jain 1:02:31
Yeah, probably just remember the name. I'll definitely do what he said. OK, so you have random dot choice and then whatever you give here right inside random dot choice, it should be a sequence.

Margi Varmora 1:02:34
I.

Hardip Patel 1:02:35
Yeah, it's early.

Tarun Jain 1:02:46
Random dot choice. Inside this it should be a sequence.
So if you see the syntax after random dot size you have sequence. So this sequence can be arranged.

Hardip Patel 1:02:58
M.

Tarun Jain 1:02:59
It can be a list, then it can be a tuple.
Or it can be set of numbers which is again range right? So this is the syntax for random dot choice.
Now we have seed. Random dot seed will do nothing. I'll just give 42. Now let's suppose you're working with LLM most of the time. How many of you know Kaggle?

Hardip Patel 1:03:28
Yes, what is that for data to find data sets and all?

Tarun Jain 1:03:33
Yeah.
So apart from data, you also have computations.

Hardip Patel 1:03:41
Right.

Tarun Jain 1:03:42
Right, so usually what happens in ML based models, right? Not just ML deep learning models or NLP. Most of the time it is associated with weights. So you have data, you're training the model and then you're getting the evaluation results.

Hardip Patel 1:03:45

So.

OK.

Tarun Jain 1:03:59

You have the same data. If in case you want to train the model again, you'll get different values, right? So if you give random dot seed and you define any value, it will just make sure that there is no repetition.

Hardip Patel 1:04:01

It.

OK.

Rahul.

I.

Tarun Jain 1:04:18

Or shuffle in the data training process.

Right now why have I given 42? Because it just believed that number 42 is a standard value. So if you check at this particular articles that is there, you call #42 as a pseudo number, right? So whenever you use random seed.

Hardip Patel 1:04:28

Thank you.

Tarun Jain 1:04:41

In most of the cases either you will see #42 or you will see zero. These are the only two things, but some people also use 123.

But the standard value is 42. Again, this is out of just research. There is no specific number that you can define, but they just tell that 42 has some significance and some people just use 0.

Now when are you supposed to use this? If in case you're building any ML algorithms or training it, or if you're using deep learning, deep learning models.

Hardip Patel 1:05:11

Yeah.

Yes.

Tarun Jain 1:05:19
Or if you're doing fine tuning. So in our agenda we have fine tuning. So when we do fine tuning right when we import the statements, as soon as we import the statements, we'll add random seed 42.

Hardip Patel 1:05:21
It.
It.

Tarun Jain 1:05:33
So random CDs there and then you also have a framework called Torch.
Import Torch. Torch is nothing but Pytorch.
Right, so here also you have torch dot.
Manual seat 42. So if you see that 42 comes by default.

Hardip Patel 1:05:56
Um.

Tarun Jain 1:05:56
So this is very important. Now why am I defining Torch? Because most of the deep learning models and fine tuning are based out of Pytorch. The back end is in Pytorch.
So just because if you are training and doing inference, if there is any change and if you want to avoid the change, you can add seed.
And if you're using CUDA, you know what is CUDA?

Hardip Patel 1:06:19
And the NVIDIA architecture.

Tarun Jain 1:06:23
Correct. So all the things that you're writing as of now, the code it's running on CPU, but if you want to run it on GPU then it is running on CUDA, which is a programming language, CUDA programming which is GPU programming language.
So if you are using CUDA, how do we change it into GPU? We click on runtime, then we click on change runtime type and then you select T4 GPU. So if you are running it

on T4 GPU then you can also define this torch.
dot CUDA.
And here also you'll have seed. If you see the second variable is seed, here you define 42.
Now if you look at the statements I did import, then I did torch, then you just have to use torch. Then you have to use the what you call functions it supports. Same also we did here we import random. Once we import random, whatever random supports, you just have to click.

Hardip Patel 1:07:12
OK.
Yeah.

Tarun Jain 1:07:28
On random, click on dot and select whatever functions have to use. Let's suppose you just need seed from random, so you can do from random import. I just need seed.
And now if you do seed, you can directly give 42.
You understood?

Hardip Patel 1:07:53
Yes.

Tarun Jain 1:07:53
So it's like if you need the entire family, you call from random, but if you need just a single person, you can just do from random import seed.

Mitesh Rathod 1:07:55
Yes.

Tarun Jain 1:08:03
And then you can only call that particular value.
Same for choice, but you have to define choice.

Hardip Patel 1:08:08
It.

Tarun Jain 1:08:14
So these are the two ways we define import. One is import everything at once, if not import only the specific things that you need. So this way you'll have lesser memory utilization compared to import random. Both the approaches are fine. It's not a very heavyweight framework, so you can directly use this, but when it comes to Torch.

Hardip Patel 1:08:14
Yeah.
Yeah.

Tarun Jain 1:08:34
We only import specific things for Torch and Tensorflow, and if there is any NVIDIA based libraries, we'll only use the specific functions that we need instead of the entire library.

Hardip Patel 1:08:40
6.

Tarun Jain 1:08:49
Is this clear? Now how do you define choice? Here you can define anything. I'll just add some names.
Here I won't give any swag because it's very easy to guess. There are only three option, but you got the syntax right from the library name, then import, then the functions that it supports.

Hardip Patel 1:09:05
But.
M.

Tarun Jain 1:09:14
Uh, any questions in this?

Hardip Patel 1:09:18

No, no.

Tarun Jain 1:09:19

I'll use one more. Uh, this thing. Let's suppose I have rings.

And here I'm deciding the tent.

If I print num.

You'll have zero to 10. If you want to just shuffle this, you can use something called a shuffle.

Random dot shuffle.

And then you can give them.

Now if you notice in shuffle, what is the data type it is returning?

Hardip Patel 1:09:55

Mutable sequence.

Tarun Jain 1:09:57

No, that is the parameter. So whatever you see inside brackets is parameter. So whatever you see this arrow mark right after arrow mark, if you see this thing, what is it returning?

Margi Varmora 1:10:01

Goodness.

Hardip Patel 1:10:01

It.

M.

Tarun Jain 1:10:10

So what does that mean if it is returning none?

Mitesh Rathod 1:10:10

None.

Hardip Patel 1:10:13
X will be uh mutated, mutated. I guess it's I mean num num will be changed.

Tarun Jain 1:10:18
Oh.

Hardip Patel 1:10:24
It will update the norm variable.
None.

Tarun Jain 1:10:37
This var is none, but since I'm defining random dot shuffle now whatever you see this sequence right will be gone. So now if I print num.
It's like random shuffle.
Is this clear?

Hardip Patel 1:10:57
Yes.

Tarun Jain 1:10:58
So random dot shuffle can't be assigned in any variable if you print where it is none. This is similar to what you saw in append and sort. In append and sort you can't save it inside a variable, but the original value will be shuffled based on the syntax that you use.

Hardip Patel 1:11:04
No.
Hmm.
No.

Tarun Jain 1:11:15
So if you do list dot sort and then if you print that particular list it is already sorted, but you can't assign that in a variable because the return type is none.

Hardip Patel 1:11:16
It.

Tarun Jain 1:11:27
Now where is this used? So in most of the cases, let's suppose you have data which is a CSV file. You have 100% of the data, right? So when we train the models, we convert this into training.
And testing sample. So training sample is there. Testing sample is something that we shuffle most of the time.

Hardip Patel 1:11:48
Yes.

Tarun Jain 1:11:52
So I'll just use data loaders.
Shuffle.
So if you see in data loaders you have a parameter called shuffle equals to true. So if shuffle is true, that means you will shuffle the entire list. If shuffle is false, you will not shuffle anything. This is also used in many frameworks this function, but the only thing is don't assign it in a variable.
Is this clear?

Hardip Patel 1:12:21
Yes.

Tarun Jain 1:12:22
I'll just do DIR.

Margi Varmora 1:12:22
Yes.

Mitesh Rathod 1:12:23
Yes.

Tarun Jain 1:12:26

Random.

So choice is done.

Huh. Log is also there.

So how many of you recall what is log of 10? Let's suppose this is log of base 10. This is log of base 10, and if I give 10, what is this?

Hardip Patel 1:12:40

Yeah.

The one.

Wow.

Tarun Jain 1:12:47

One log of 100.

Hardip Patel 1:12:51

10.

Yeah, 10, I guess, no.

Tarun Jain 1:12:58

Now log of 100 will be two IES. Let me just check it random dot log. OK, it will come in math.

Hardip Patel 1:13:04

6.

OK.

Tarun Jain 1:13:10

O I'll define imortmath.

Hardip Patel 1:13:12

Hmm.

Tarun Jain 1:13:14

And now what I'll do is I'll just math dot log 10. If I give 10 it is one. If I give 100 it should be two.

Hardip Patel 1:13:15
Yeah.
Yeah.
Oh, OK.

Tarun Jain 1:13:26
You know why it is true? Because 10 of base 10 will be one.
Now 100 of base 10, just take the square root. What is the sqrt 100?

Hardip Patel 1:13:39
Mhm.
Uh, 10.

Tarun Jain 1:13:41
Sorry, square root of 100 is 10, right? So square root is 2. Now if I give this as thousand, this will be 3.

Hardip Patel 1:13:46
M.
OK, I got it.

Tarun Jain 1:14:00
But if I give a value somewhere between 1:00 to 10, sorry 10 to 100, let's suppose I give 80, it will be somewhere between 1:00 to 2:00, which is 1.7 or 1.6 or 1.9.

Hardip Patel 1:14:12
MM mm.

Tarun Jain 1:14:15
If I give one, then what is the value?

Hardip Patel 1:14:18

Yep.

0.1.

Tarun Jain 1:14:21

It will be just zero.

Hardip Patel 1:14:23

Zero. Yeah, sorry.

Tarun Jain 1:14:25

If I give zero, it will be error.

Because it doesn't support 0.

Same if I use log of two.

Let me check if I have log of Ku.

If I give two, what it will be?

Hardip Patel 1:14:42

Yeah.

1.

Tarun Jain 1:14:46

4.

Who?

RamKrishna Bhatt 1:14:49

Oh.

Tarun Jain 1:14:51

6.

RamKrishna Bhatt 1:14:52

Two point something.

Tarun Jain 1:14:53

OK, 24 then what is after 4/8? Eight is 3 then 16 will be 4. Now you got it this syntax. So this we will use in some of the search techniques.

RamKrishna Bhatt 1:14:56
8.
OK.

Hardip Patel 1:15:02
Mm.

Tarun Jain 1:15:11
Or like TFIDF. This is where you use keyword search.

Hardip Patel 1:15:15
You.

Tarun Jain 1:15:16
OK, so if I do TFID formula.

Hardip Patel 1:15:17
OK.

Tarun Jain 1:15:27
You have log.
So this command again, it's just single line, but when we work with TFIDF, so why do we need TFIDF? Let's suppose I have my document. Most of the time we just do schematic representation, right? Vector search. And there is also an approach called in that which is hybrid search, which does vector search with keyword search.
So if you are searching for machine learning, it will look into the entire PDF and it will only match machine learning. But what vector search will do is if you are searching for machine learning, it will also give you what is neural networks, what is deep learning, what is machine learning.

Hardip Patel 1:15:56
OK.

Tarun Jain 1:16:09
So keyword matching is 1. Vector matching is made based on schematic meaning. How often is one word occurred with the another word occurrence. So this kind of search techniques usually have log based formulas. So for that we'll use math.
So the syntax is same. Once you import anything just do math dot I'll do power and then 10 comma 2. So if you see here it returns.
X double star Y. So you actually don't even need this thing. You can directly run.

Hardip Patel 1:16:41
Yes.

Tarun Jain 1:16:43
10 Double star 2.
Is this clear?

Hardip Patel 1:16:52
Yes, yes.

Tarun Jain 1:16:53
One more thing why where we will use this is there is something called as round.

Mitesh Rathod 1:16:55
Yeah.

Tarun Jain 1:17:01
OK, in recent latest updates of Python they got round outside. So let's suppose you have a Pi value. I will print math dot Pi.
So what is the pi value?
There is some value, but when you are working this on the UI you just need the starting two values. So what you'll do is you'll just do math dot py, you will add it inside round and then you just need starting to decimal. So now if you see 3.14.

Mitesh Rathod 1:17:18
The .144.

Hardip Patel 1:17:31
Um.

Tarun Jain 1:17:33
Earlier round was inside math.

Hardip Patel 1:17:36
Mm.

Tarun Jain 1:17:38
But now it's outside.
Oh, you got it. It's very simple. You don't have to import anything and then do DIR if in case you get confused. If not, we'll have auto complete. And once you do auto complete, we just have to use what we need and check the particular parameters, what it accepts and what it returns and we can get to know when to use this.

Hardip Patel 1:17:45
Yeah, that's it.

Tarun Jain 1:18:06
So we have few more frameworks that we will cover, but it will be under Python for data science which will be just for one day. There we will cover Pandas, Matplotlib. Matplotlib and numpy. So this is for data manipulation.

Hardip Patel 1:18:24
I.

Tarun Jain 1:18:29
This is for data visualization.

Hardip Patel 1:18:30
4.

Tarun Jain 1:18:35
And this is for.

Hardip Patel 1:18:36
You.

Tarun Jain 1:18:40
Data augmentation.
Probably this will be on Monday because tomorrow, oops, might be covered.
But in terms of basic Python, this is it.
Or any questions in basic Python?

Hardip Patel 1:19:03
No, no.

Mitesh Rathod 1:19:06
No problem.

Tarun Jain 1:19:08
Oh OK, so I have to send the GitHub URL here. Since we are maintaining 2 collab notebooks, there are so many comments that we have added in between. So I'll I'll try to merge both of them and create one single notebook with proper.

Hardip Patel 1:19:13
OK.
OK.

Tarun Jain 1:19:25
What do you call comments? Because somewhere I've written some comments and in one poll app there is some comments, so we just have to combine them.

Hardip Patel 1:19:27
Mm.

Tarun Jain 1:19:34
So this comments and all. So I'll combine them and then I'll share the GitHub URL.

Hardip Patel 1:19:37
OK.
OK, yeah.

Tarun Jain 1:19:43
Cool. Anything else or we are good to go this URL, just save it. This is very important. We mark it down.

Hardip Patel 1:19:50
Hmm.

Tarun Jain 1:19:51
It might help even if you are doing simple prompting, right? You can just use this framework. It's just four lines of code. Copy that and then use it in GPT or cloud. It doesn't have to be specifically in your code. It can be also daily usage between cloud and.
At least if you are using copilot, because whenever I'm using cloud code, before passing it to cloud code, I go to GPT, convert into to markdown and then I use augment. So augment is one of the copilot that I usually use similar to what you have Sir.
OK, uh, that's it, right?

Hardip Patel 1:20:32
All right. All right. Thank you.

Tarun Jain 1:20:33
Yeah, thanks everyone.

Mitesh Rathod 1:20:36
Thanks. Bye.
But.

Hardip Patel 1:20:37

Thank you. Bye. Thank you.

RamKrishna Bhatt 1:20:43

Thank you.

Hardip Patel 1:20:43

Right.

It's all good.

I I will look for a person. Hmm. Main. Yeah, he, you know, he understanding.

Can't be, then things will be much easier for us.

Neural network, neural.

Neural net is cover for both. See I gotta fill in the blanks.

There is no in between state.

Margi Varmora stopped transcription