Python and AI Power-Up Program Offline Class-20250819_191740-Meeting Recording

August 19, 2025, 1:47PM

2h 7m 27s

Ajay Patel started transcription

Tarun Jain 0:04

One is pydantic which we didn't cover last, I mean in the this thing and then we have meta classes and once meta classes is done we have multi threading, multiprocessing and multiprogramming. So basically what I've done is I've added all the code bases today in the GitHub. So whatever we did so far it's in the notebooks.

Folks, only this topic, multithreading, multiprocessing and asynchronous will run it on VS code, right? So if we want to proceed from collab to any IDE, right? For example, let's suppose I have VS code and if you guys are using Jetbrains, you'll have Pycharm. So how do you proceed in using VS code right? First we need to create environment variables. Then if you need any external libraries, how do you set up external libraries? And there is also a package, right? How many of you know about PIP PIP? Huh. So how do you install that? So these are the three major things one needs to understand once we do these three things, probably tomorrow once we start using NLTK, right? So if we notice tomorrow we'll start with NLP. So for NLP.

Mainly you have this library called NLTK. So NLTK will run on Collab, but our goal is to run NLTK on VS code. So VS code by default will not have import NLTK. So if you write this line you'll get an error, right? So today we'll try to set up.

Python environment, try to install some dummy library and try to execute one or two code. Once it works then we'll have multithreading and multiprocessing and asynchronous programming. So this will be the main agenda for today. So again let's just revise.

What we did yesterday. So what were the topics like? Mainly it was.

Typing and.

Which example is this?

Before data classes I took something right string representation. So here if I run this. What is the first line of output you'll get?

So what's your mood right now? Are you guys sleepy or are you guys tired? Hungry. Tired. Tired. Angry. Mixed mode. I'll just write mixed tired.

And.
So.
So what will be the output? If I print M, what will be the output? Move right now in upper case, so M capital. This entire thing will be in capital, right? So I'll just write mixed.
No, only I you here you have to give user input.
Is this clear? What if I remove this?
It will print with those I object ID and address location. It will give a hash symbol with a memory address.
Is this clear? So if I want to know what is the actual memory address, what is the alternative command?
ID of 1010.
Which is an object still here. Is it clear? This is for string representation. Now let's proceed with the second option. OK, I don't know why this was not cleared. So what example is this for?
Typing extension typing right typing or typing extension. You can call it both. So basically what typing extension does is you already have a library called typing. There are few just few methods which you have in typing extension which are not available in typing, but 99 or 90% of the functions are already available in.
Typing, right. So these are two frameworks or two libraries that you can play around, right? If it is not there in typing, you check in typing extension. So what are the basic data types that we covered?
Tuple list, Tuple list. Apart from those, apart from basic data types, we saw three new keywords. Tuple list, type, type bar, type bar from typing import.
Type bar. When do we use type bar? Let's suppose I'm defining a function generates. I don't know what data type user will enter, whether it is float, whether it is string. So I'll just keep it generic, which is type bar right? And what were the two other things? Typed it OK that you can add it here.
If not, you can use it from typing extension as well. What else?
And we had union and optional and any. So these are some of the data types which are basic data types which were covered in the previous session, right? We had int. Float type and other things. Apart from those data types we also have these things. So now what is the purpose of optional?
We can have. So basically most of the time you define certain function and you don't expect user to enter any input variables for that. For example, not everyone gives

temperature value. You have top P, top K, frequency penalty. Tons of parameters are there in every single library, right? We are taking.
Example of Openair because in Openair we have tons of input variables. So here now I'm defining A variable called name and now if I don't give any name then it's fine. It will take some optional alternative which is name or guest.
Uh, now what happens here? It's like.
None or guest?
So basically it will take guest. So if I give Tarun or guest, it will pick Tarun. Is this clear? The order order matters, right? So if you're comparing none with a string, none has no value.
So obviously it will go to the alternative, but if here you already have a string, it will pick the first one. This is order right? And if I print this, what is the output now?
Hello guest. Hello guest. OK, so now coming to the data class example, when it comes to data class example, what are the three input or three parameters which are true?
You don't have to define what are those.
True or false? No, no, by default. True by default, which is 3?
So just think of an example of class. In class what do you define which is ignored in data classes?
Constructor. Constructor. And what is that called?
One is init which is ignored in data classes. So we have data class example here which is cart. Have I defined any init function? No, I'm not defined. Then you have REPR, so REPR.
Obviously we don't define this in most of our use cases or whatever program we have written, but in many programs, REPR is written by some of the developers, right? But when it comes to data classes, REPR is already existing. So how is REPR written?
Syntax of REPR. Let's suppose I print REPR of self, not self in the sense. Let's suppose I define cart and I will print this cart. So what will be the output of this cart?
Oh, OK, so wait, I'll rephrase my question.
Correct. So here data class is there. Data class already has REPR, correct? That is how we write REPR right? So now that we know REPR is already there.
Here what I'm trying to ask is what is the output format? For example if I print cart? So let's suppose I print card. What is the output of this card now? But what is the first keyword of string representation?

So let's suppose I have a basic class class animal. OK, and this is data class now.
And here what I'm doing is I'm defining name equals to.
STR. This is my class.
Oh, the the first. Uh, it will be the first. Uh, the name parameter name. What is that keyword?
So what I'm trying to say is if I create the object of animal now members will be there right animal and in bracket name equals to correct animal in bracket name equals to yeah whatever I give right name correct.
So here's a equals to animal and if I give name equals to dog.
Now if I print here, it will be animal name equals to dog. So I was just asking the format of how RPR is written. OK so.
Init is true. REPR is true. Now frozen by default is false. Now what are we doing now here? I'm giving frozen to be true in simple word. What is frozen? It's a read only. So what will be the output of this now?
So I'll just run this and this is what I've defined.
Will this line work? No.
Will work, no. Think again. We did experiment this. Yes, yes, exactly. It will work. This will work. Now what is this? You guys remember the syntax, right? Default factory?
If I have the value, let's suppose I'm defining name equals to Tarun. I have name of that cart and if I know what needs to be given, it will be field, then it will be default and it will be value. If I don't know the value, I will keep it as default factory equals to data type.
So the difference between default factory and default is default has to have a value. Default factory. I don't know the value. I'm giving the data type. Like you mean to say that instead of list in default factory we will add. Let's say if it is a it will be some apple and all.
Something like that. So it will be like this default.
OK, so will this work? This one will not work. This is an error. Now what will be the output of this?
Book, list of book, list of book and this one will be 1000, 1000.
So I will run this. I'll get an error. Where is this getting error in this line, right? So I'll just comment this. Now what will be the output? Book and thousand. So is this clear? Yes. OK, so this is last example.
We already discussed this.
What am I printing? I'm printing an object. So when I'm printing an object, what will

be triggered string representation?
X equals to 5, Y equals to 10. Is this clear? So we'll start with pydantic. Before we go to pandanic, let's just understand the difference between typing data classes and pydantic, right? So if you look at typing, what is 1 drawback in typing?
I define a function, I define a type and then I define certain logic and in 90% of the time what is Python? Python is dynamically typed. You will have user just adding any single things he will needs and obviously he'll get an error.
And if you want to avoid all these things, I said we can add assert or if not we can add some kind of checks. So what Pyrantic will do is it solves the drawbacks of typing which will add the validation. So in Pyrantic basically what we do is we define certain schema.
We define a schema which is nothing but our base model.
And here you have to give these validation checks.
That will allow the class.
To be parsed or not?
So this is just to have drawbacks of typing. Typing is OK, I'm defining certain data type and user has to use float only. So if it doesn't use float even though it will work. So float in the sense I need to have some values. I will give string even if you give string that.
Function will work, so that is 1 drawback of typing. Now pydantic will have validation sex. Where is it commonly used?
If you're defining any fast API endpoints.
For endpoints, if you want to add certain text you can add and if you need Jason parsing, let's suppose you have large language model and you need LLM to give 4 Jason keys every single time you run that code. So those four Jason keys you can define. What is the key of it?
And what is the description? Like what kind of output are you supposed to generate? You will only have key. The value is supposed to be generated by. Who will generate the value?
LLM, LLM. So what we're trying to do is OK, I have an LLM which is open AI. I want open AI to generate Jason as a response every single time. So when you're generating Jason, you only have key value is something that open AI needs to generate.
So if Open AI has to generate any value, it needs to know what is the key and what is the purpose of that key. So key and description. All those things is defined inside

pydentic, so it's used for Jason parsing. So let's look at one of the example.
Just like how we used data classes, what was the first line of data classes? You import it from data classes, import data class comma field. We also have from pydantics. Import instead of data class. Here you have base model comma. You also have field. It should be field or.
Let me just check.
It should be capital field. Can you see the first line?
First line if you see it's field capital F whereas in data classes it is small F.
So first what we will do is let's define a dummy typing function. For same typing function, let's define a pyrantic based model so that we can understand what's the difference between that.
So I'll just do from typing.
Import list and then I'll just define some dummy class which is prices. So how do I how do I define prices? So let's suppose I have a list, all the list. What should be the type of it if I say prices?
OK.
List and float, so I'll just add items.
Then you have list float.
And let's suppose here I want to print total price.
So what will be the data type of total price?
No, no, no float. Either you can give it float or you can give it as any float.
Return of sum of items.
Now here what user can do is user will add total prices.
And you can give.
List under.0.
200 now 200 is a integer then 300.0.
So it will consider this and here what I'll do is I will just add one line.
Which is items of 0, sorry one and I will print it just so I can show the difference between pydantic and typing. So my first index is a integer.
So integer 600. OK, so now what I'll do is I'll just define pydantic. So for pydantic, usually you have to define a class and I will define invoice. You can also define price.
Here again, what kind of syntax are we supposed to follow?
So the syntax in the sense formatting way you have to write this base model in the no. So there is some kind of writing style writing style. I meant it should be camel case, right?

So base model and here you have to give base model.
So now what is this base model?
I mean what kind of syntax is this? Inheritance. It is inheritance.
So here you can just define prices of.
List.
Plot.
And then?
You can have this function. If not, what we'll do is.
I will just leave this here.
Or.
Now inside this price what you can do is you can define all the values. Can you see it?
So the syntax is same as what we define in data classes, but the only thing is instead of decorator data class we are giving base model.
So instead of this.
Inheriting it from.
So this entire thing is there, right? We are writing it in a this particular format, but both of these two details are different. The purpose of data class is to have alternative for how you define your classes, right? But when it comes to Pyrantic, it's mainly used for validation and.
Parsing whether you have to parse it or not. So both have different what you call.
Both have different purpose to solve. I'll just remove this.
Now inside price I'll just define prices 100.0200300.0. So now if I print price.
What is the second index?
What was it supposed to be? It should be integer. But what this pyrantic will do is hey, you are supposed to give it float. You are given integer. I'll just convert that into float. But what if?
String will also be passed. So The thing is let's suppose I'm supposed to give string.
Just keep it here as a pro.
This you can. There is no issue with string conversion to tuple is simple. This is just a number. The issue will arise when you make this as STR. I'll tell items.
Altel purchased.
OK, so now if you look at list here, if I give under.0 I'll tell.
Apple.
Then here I'll just add orange. Now this under.0 will be converted into list.
OK.

OK, wait since float.
Here what I'll do is I'll make it Apple. I'll revert the changes.
Just a second.
Because string, even if you give 100, it will convert that into normal float. But if you look at other way around, for example what I meant is float to string. So let's suppose a equals to 100.
.0 If I convert this into string, this is not an error, so it will just add 100.0 inside a brackets. The same thing will happen, but if I give float, So what I usually did was.
I defined float and here I have unread.
So this will also work. It will make 100.0 but here if I give any value this is when the issue occurs because you can't change apple into float.
Value added. So now again what I'm trying to do here is I'm reverting back to prices which is in float. Instead of having a value I'll give Apple.
100.0 then again orange.
Then I'll get a validation error. Why? The purpose of Pydantic is mainly these two things. One is I have to pass certain validation checks and 2nd I want to tell whether I can parse this or not, right? So these are the two purpose that Pydantic solves.
And above this what people have done is they've used Pyrantic for Jason Parsing.
Pyrantic team is very what you call active. Also you can say they release their two product. One is Logfire if you want to build observability tool.
Then you have Logfire which is completely open source. Again, it will track everything, all your loggings. If you're using LLM, it will also track your LLM cost per token per call, right? Logfire. So and they also have their own agentic framework as well, which is Pydentic AI. Again, it's best.
Use when it comes to production right? As soon as it got released, the first keyword of Pydantic AI was we are production ready right? Because Pydantic is running Python for almost some 10 or five years now. Every single frameworks that we see right, it's based out of Pydantic.
Because they know how to build this framework. So they also released their agentic framework probably just last year and the first thing what they released was, hey, we are production ready and then they teamed up with some of the or other partners to build certain proof of concept.
Open source. You can say that, right? Huh. Open source. They're mainly run on NGOs. So there there are three major products. One is the validation one. This you don't call as a product. So logfire, they have some kind of pricing as well.

Pydentic AI is a agentic framework, so all these three belongs to the same community. So they mainly run on NGO support. If not, Python happens, right? Most of the funds goes to these folks. Initially when Fast API was like that, but then Fast API opened their Fast API labs.
Or cloud something. So this is similar to what fast API used to do.
I was asking if it's Replit or Replit. Oh, sorry, there are actually two companies, right? Replit. So this one I meant. OK, Replit.
OK, let's take two more examples which will show what are the different ways you can add validation. There are some of the very what you call some of the arguments are there where you can also define hey if you give an age.
Usually age should be between 18 to 64. If you encounter someone from less than 18, you can tell hey you are not valid to use this particular account. So those kind of validation check also can give and if you want to give any parameter.
Let's suppose price. If someone gives more than some amount, you should tell hey you gave a very high amount, it should be less than and if someone gives some amount which is very low and if you want it to be greater, you can also give that less than greater than than the range.
So let's look at those examples as well.
For like we have data class and we have identic. Let's say we are using animal. Can we have data class annotated on identic as a base model? I mean that should also be possible.
So we can can we use it as a parser validator? But this won't actually make sense. Usually if we define this right, what we usually do is we first define one product separately so you don't directly involve that with the pyrontic.
So once you define that particular class inside data class, you can just inherit that in prices. So instead of directly using like this.
You'll have to create two different classes. OK, so that's why I was asking. So let's say if we are using animal, then I'll have data class separate, then I will have bidentic model as well.
Data class separate and I mean like I will have data class of an animal. OK a a class which annotates as a data class. Animal is under data class correct? Will be a class over. Huh. It's the right class which is which is from pedanting.
Which identity? Then let's say if I'm using database, so different class, correct? Is it? No, that is fine. So usually what we usually do is let's take the best example of invoice, right? Invoice. Usually what you're supposed to extract, you need to extract

line items in line items.
You might have multiple keys. OK, so one is you have ID, ID, SKU, you have SKU, you'll have quantity and you'll have price. This is just one parenty class. You'll have one more parenty class which will take name.
E-mail address, then vendor's name, vendor e-mail address, customer name, customer e-mail address. This is second pedantic class. Now you'll have third pedantic class where you will merge the customer information and invoice item. So even though everything can be in one classes, usually we don't define that in one class.
Just to have a structured, structured format.
This you will see the format. OK, so it it is separate class. We have three separate huh class of the same yeah entity. So I'll just show one example how you do have that example.
But it's two pyrantic classes. The similar format can be followed for data class data class.
So here if you notice you have line item where you have description, quantity, unit prices and total prices. I'll come to what field is, but you understood what field is right from data classes. Now description is the key.
I'm telling LLM what is the use of this key and you're just defining the description. So here you have line item which is your first pedantic. Then you have one more invoice data. Here if you notice you have list of line item.
This class is already defined, so this is OK to have two different classes. So now here what will happen is you'll have all these things in a key. Now then this will be a single key with a list of a class exactly. So here you can have database.
Then you can have one top layer of data class just to define your code base. So data class is to define your functions as well. Yeah, pydentic is not used for defining functions. OK, so an action.
Will be in data class, correct? But the attributes validation will happen in pydentic. Yeah, and the state will be in data class.
State state SQL as your model state in the sense state as in like invoice #1, invoice #2, invoice #3 something like that. Yeah if you are if you wanted to add in the keys that has to be generated then.
It will be in pyrantic. OK, but if you know it's already defined properly in your database, then it will be in data class because you have to use those values and you have to use a functions. Here you can't define methods. OK, so you have data class

separate, but you can link this pyrantic to those classes.
But it's OK to have multiple classes. That's not enough. Yeah. Thank you. OK, so we'll take one example quickly.
I'll define class.
User. Now I will probably not write. Can you just tell me the first line? I want to define a class user and then I need to define three variables. One is user ID, name and e-mail.
We're using Python T plus.
User base model.
Uh, what are the user ID?
Let's keep it integer so that we can see the validation happening.
Yeah.
And.
Is this fine? So now here what you can do is you can define user one user.
User ID equals to one, username equals to.
Arun and male equals to.
This will work OK now.
I have user 2. What this user to do is this.
If I give 123 it will work. Why? Because you can convert 123 into string to integer. But if I give 123 a.
You can't convert this into integer because my data type is integer. If I give 123 and I'll just tell Rahul Rahul had dummy so this will work now if I print user 2.
This 123 is a integer. How do we extract this? You can just add same logic. OK, this logic is same. I define type.
Which is integer as per what we have defined in validation. Here if I give VA then it's an error and what is that error name? Validation error. So what you have to observe is this is common error you will every time you will face when you are working with pydantic right? Validation is common.
How can you resolve it? So if you see input should be valid integer unable to parse. So parse keyword and validation are two common keywords in parenter and what does it say? Input value is 123 a type is string but it should be integer.
So this error will be common even when we work with LLMS and when we tell to parse it in Jason, either you'll get Jason decode error. Yeah, I'm not convert. I can't convert into to Jason. If not, you'll get validation because most of the time LLM will forget what.

Data type to use, right? So these are two common. One is validation and one more is Jason decode error. That is, I can't convert it into Jason.
OK, so we'll take one last example. So this example is like what if I want to give certain conditions that age needs to be between 18 to 64 and?
Correct. So we can have certain pattern even that is possible. So I'll define class.
And here I'll just define customer.
Base model. Let's also take field right? Because I want to give field. If you notice here, field should be capital, whereas in data class field is in small.
Till you're you're able to understand it. It's very simple logic. Just define a class, inherit it with a base model, define attributes. This attributes is mainly used for validation and this is solving the drawbacks of typing.
Because in typing it was just dynamically typing anything and you were able to parse it. But here we can't parse it. There are checks, so let's add one more layer to those checks. I'll just add class customer.
First line is base model.
And this customer has a name.
I I also have a AD.
Field so in JavaScript I remember you have dot dot dot triple dot right? So triple dot in the sense to ignore if I'm not wrong. So to unpack means like whatever the.
And then in the function argument. OK, so here also what you have is in field. First parameter is there, right? If you notice it is default. OK, that means you can give any name for ID. If you don't want to give any name, you just give triple.
dot dot dot which is usually called this ellipsis.
Just ignore like I want to give description but I don't have any description to give. So if you want to give description we can just define that.
So you can say this is the user customer ID, right? You can define that. If you don't want to define this, you can just give triple dot.
Hi you can define it as an empty string but empty string might sometimes give validation error. So just to bypass it we give triple dots. This type we also see before React also has triple dot if I'm not wrong. What is the purpose there?
Just take the data previous data. OK, if we are we have an error then we can use. Let me just check what it is called in pydantic. Usually in pydantic it's like if you don't define a description, you just define triple dots.
Uh, that titled answer.
I just it's to ignore or add the description.

Can I give?
Description.
Instead of.
We will test this one. So I'll for one of the value I'll give a description and if not I'll give any data type as per it suggested. So the first keyword is data type or description here. Then what you have to do is you have to define a check.
So the check is what should the ID be? If it is an integer, you will obviously need it to be more than 0, so you have greater than.
So what is GTGT stands for greater than 0 so I need this particular ID value to be more than 0 and then I have name STR same field dot dot dot.
And then if you see for name also it has given 2 new variables which is minimum length of the customer needs to be at least three and maximum length should be. 30.
And then I'll also consider age, which is obviously int field, then triple dot.
Then if you see uh.
You have greater than 18.
And what is the second thing greater than and one more is less than.
Lt. less than less than I'll give 64.
And here what I will do is I will also add one more thing, I'll add dummy age.
Here instead of this triple dot, I'll give default.
Default equals to 20. So what did it say? No description is metadata. While triple dot mark is required, you can remove. You can place it with default value which is default.
And I'm giving it defaults.
And here value. Maybe it should be instead of description it is give a default value.
And then you can have e-mail. So for e-mail it has given a rejects pattern. So basically this rejects is replaced as pattern.
So can you notice there is one auto complete? I'll just place pattern. If we get confused we can just come to.
Field over on this and you will see some of the keywords.
So you have GT which is greater than and then you have LD, then you have minimum length and then what else did we had pattern. So pattern is basically you have to define.
Regular expression. Is this clear what we're trying to do? First we defined hey this is the validations and these are the data types. Now we are just adding one more additional layer where we are defining field and the first parameter inside field is

define a default value. If you don't have a default value.
I just have three triple dot and then for ID I have a condition that hey no matter what uh value you pass it should be greater than zero. That means I can't give -1 and same name name.
Needs to be between 3 or 30. You can't just give something called as yay right? Many people have some short form yay. Same goes with age. Now I'm giving dummy age as well just to check what is supposed to happen and then you have e-mail.
E-mail if you have certain pattern, let's suppose you are defining phone number and you want only Indian folks to join, you can have plus 91 then DDD. So D is nothing but digit in digits. So here also if you see you're checking.
Of alphanumeric which is capital small letter and digit and then you have at then again you have alphanumeric digit then dot then alphanumeric digit and it should end with dollars.
So this is again regular expression.
Now if I define customer one.
ID is -1, name is unknown.
Age is five.
Or else let me give some valid one which is 19 and e-mail is.
Uh.
Unk.test.com I'm not defining dummy edge. Why? Because it has a default value, so I'll just run this.
What is the error? Input should be greater than 0. So if I make this one, if I run this, it should not throw any error. Why? Because my dummy age.
Should be 20.
dot.
Damage.
It's 20. Now here only if instead of unknown, if I give VA it should again give validation error. What does it say? String should have at least three characters. Type is string two short input value a input type, so at least three characters. So all these. Meaningful messages there. Most of the time you'll only get this error in Pydantic or Jason decode error.
Is this clear?
We are importing this field as from from the base model, right? Yes, yes. So I'll just import that here.

From Piedantic.
You just have to use import base model. What is base model? It is this one, then comma capital F wait for some time, you'll get filled. So this field needs to be used. Similarly for data classes you have from data.
Classes then you have import data class and small F.
E-mail e-mail also has e-mail STR. Which one e-mail field has e-mail STR as a import. Yeah. Oh, this one. Yeah. OK, yeah, for my. You should have. Yeah. So this is coming from Pydentic.
It's more like a that is link in Java with validation schema. There is a library in Java like so this purpose is also these three things only schema, validation and parse these three things.
And mainly it's for fast API endpoints and fast API. Mainly it's this thing JSON parsing. If we are using any LLM with some kind of schema, hey this key needs to be there and this value I will give you description value you have to generate so that. That thing is added in base models.
OK, so we have last thing to cover which is metaclasses. This can be validation like function like this. No no I mean like.
Let's say JSL. Let's say my I have something like KID, just like ID, but it is but we also want to validate if KID.
It has a dash ID. I mean like if ID is 10, then K ID should be K-10 conditional K-10. Yeah, so that is very custom and concatenate. Yeah, OK, those things I don't think. Examples and set. Can we have? Wait, you can add that as a pattern, right? Because you're saying K Yeah, under score. That means that's a pattern. Yeah, but then that pattern also needs ID, so it will be like pattern.
You can define star, very star star. Yeah. Then it should be specific KK let's say yeah K then you have under score. Yeah yeah. So this side will remove since K dash is there. Yeah here you'll have.
0 to 9.
Yeah, but uh, it needs to be ID.
It's what if it needs to be same as ID?
The.
So my main intention is can we have something? Yeah, you can have right like if you see it, you have oh function. Yeah, so that a function is a validator, custom validation. A closer.
Yeah, that's something that connection. Mainly this is for attributes. I'm not sure. OK,

OK.
But let's just ask.
Very tricky.
Very tricky.
Validation which can only be done via function can we have?
There are so many keywords, right? We'll just ask which argument is useful for that.
Yeah.
Oh yeah, you what you can do is I remembered.
Here when you import field you also had field validator. So this field validator is a function so then decorator not function. I meant you have add field validator.
Here you define your logic.
And it can be any condition.
Define condition.
And if this condition is passed, you can give true.
We should be used. Oh wait, actually it might be under the function.
Field name. Wait, let me check this example and come back because this was deprecated in one of the framework. So what I'm asking is let's say if in e-mail I want to check if this this domain of that e-mail.
Has uh is is valid or?
We can do the DNS check on it, right? Is it a valid domain? I guess I don't think we'll only have like we'll just check a basic thing, not an external no, but like.
The.
Oh, what is the question?
So example I'd like to understand is from other fields like age. That's what I'm saying.
Like age and country are two different I parameter. What do we say members?
Instance variables. OK, same as what you asked, right? Yes, yes. So there are two things. Probably one, we have to add this validator or field validator. I'm not sure whether this is version one or version 2. So if you want to check that, right?
I'll just show that how you can do. You can just print import pyrantic.
Let's suppose in Pydantic version one you had some feature, but in version two it was remote and you're not sure what version you're using, so you can just import any statements, then add pydantic dot double under score version double under score.
So I'm using second version, so I'm using validator so I'll just check is Indian.
Is my condition. I want to check whether this particular person is Indian or not, and again I'll have some logic.

And return India. Sorry through.
And this what?
OK, there is some deprecation error. Probably I'll just see which library it is using. It needs to be under this, just under this class only.
Validator e-mail.
Where is that past now?
Like class scan the field validator and we oh OK, OK.
So here usually what you are trying to do is.
It's is it field validator or is it validator? Field validator?
Field validator, but I'll come to this example. Simply we'll avoid time, but let me just note this down.
We can get and no. Yeah, that's possible, but just the logic, yeah.
Because it shows deprecated error.
Define custom. This we can take in quiz. Define custom condition.
4.
The validation of.
Attribute and I will just attach this.
OK.
Is this clear? Pyrantic? Yes. Now let's look at into meta classes. Meta classes is 1 optional topic, but we have already covered meta classes, right? That function we are using probably from the day one. Probably that was our first function.
And that first function itself was a meta class whatever we saw. So here meta classes basically is like you can define your own this kind of operators right here what we did, we defined a double under score, we add REPR.
And we had in it. What else did we had in this? If you remember, we also had something called as data classes params.
And uh, when we saw annotated, we also had metadata.
Right now, can anyone recall if I use that function and if I give any value output it was printing a class?
So basically meta classes is classes of class. OK, so now when I used that function, the output was a class. Can anyone recall what function that was?
So I'll repeat again. I use a function. I will give any value inside that bracket. When I print that, the output is a class and then it has some value under score metadata.
No function, no function. That function was our first function that we used. DIR name no DIR we used in the second day.

Only when we came for import statements.
Is it last for name? No ID? No ID is for day one. ID we looked into day one. That's correct. Print statement we looked into day one. That's correct. What are the two other functions?
ID print is done, DIR is done, two are pending.
Yeah.
I'll just define its type. So what does type display? Meta classes is the classes of the class.
So what is type? Let's let's suppose I define type and I give 10. What is the output class int?
So if I do print.
It is class and then it isn't. So huh. So basically type is a meta class.
Type is one of the example of meta class, right? So what we will do is let's define a meta class. So basically when we usually define meta classes, it's like what is that class about? It's just to give certain.
Representation. Not just representation, but what is the purpose of that class? Most of the time what happens is you have tons of classes which is available in a library. So meta classes is just to have to check hey what is this class even doing right? Is this inherited? Does it have any build based class?
So you don't actually have to look at the entire code base, you can just run a separate script. So what does DIR do? DIR you give any statement, you give a variable, it shows your entire thing. Now variable correct methods you have. You also have this.
This kind of methods as well. So DIR will print the entire thing, whereas what class meta class will do is it will check four things. One is whether it is it has any arguments. Second, it will check is there any inherited base?
And then the other arguments and the name of the class. So these are the four inputs that we usually define in meta class. So let's define that class mymeta which is simple type. So now as I said.
When we are defining type classes, I mean meta classes, which function is a meta class?
No, which method? I mean function, it's type. So in order to define meta classes, just define a class, define any name, then open brackets, just write type.
Is this fine? It's similar to what we have done for base base model, but for base model you were supposed to import here. As we know type itself is a meta class, so

we just have to use type here.
And now what you have to do is you can have your own name which is new double under score. Is this one or two?
1-2 and then again 1-2. Here you only need to have 4 parameters as I said.
Which is class. I'll just give CLS. Then you have name and then you have base. Base is mainly used to check if it is inherited or not. If it is not inherited, base will be none.
OK, so base and then you'll have arguments.
Arguments in the sense attributes.
So usually these four things will be there. You will never use meta classes, but the only thing is you will find meta classes in some of the test scripts to see what kind of classes are there and what is the type of these classes and all, right?
So you have type again I as I mentioned you have a class inside that class you have a type inside that you can use new or call. So directly you will never use a meta class.
I'll show one example.
My different screen is visible, right? OK, so there is a library called Pytorch.
And I will also open langchain. Probably I'll just add custom components for LLM.
Sorry.
And get done.
They have this example. No. OK, so I'll just write Pytorch and then I'll add call function.
So basically if you see her.
When you define Pytorch, right, I'll just tell you why Pytorch is used. Pytorch, it's like you want to train your own neural networks, right? You have your data and you're building neural networks. So whenever you want to train neural network, you have a class which is class NN.
NN stands for neural network which is a class of Pytorch. Inside this NN class you have a function called under score under score call under score under score and inside that under under score call you have to give a class and then you have to give a parameter.
Which tells you till how long you need to train this particular model. So that call method is part of the class. What is that class name NN. So same here what you're trying to do is we are defining new. So inside this new you have CLS, you have name, you have basis and you have attributes only four parameters.
So now what we can do is I will just print.
CLS.

CLS then rint name.
Name. First we'll see what are these parameters, then we will look at one of the example.
Print basis.
Vs. So as I said, this is not inherited. I mean whatever class I use if it is not inherited, if the class I use my meta is not inherited.
Then basis is none.
And then print.
Attributes, attributes and then you can just return the super function. So in super if you notice you're defining new and you just have to use the same parameters.
OK, so now what you need to do is, as I mentioned, metadata metaclasses is mainly used for classes of classes. Let's define one class.
I will define class example.
There is a attribute called meta class. Then just copy my meta. This my meta should have.
Type inside this.
Is this clear how we are defining it? I'll just repeat this. So when we are defining meta classes, you have to use a parameter called type and it has only four parameters, class, name, basis and attributes. Basis is for inheritance attributes if it has any values.
So now if I want to use this and I want to check any class that I have, I'll just use class example meta classes equals to my meta. My meta is something that is already defined. Now here what you can do is you can have any attributes.
So what are the attributes? X integer equals to 10 is an attribute. What is attribute? Instance variables.
Name string equals to unknown.
And you can also have any methods, right? So def LO.
I'll just return greet.
So now when I print this, what it has to do is it has to print this for statements.
So what is class class main my meta and then you have name equals to example. So this is name. Then you have basis. What is basis? It's empty and then you have attributes. In attributes you'll have all the annotations.
So this is basically not used in what code you write, but in most of the testing scripts or any debugging scripts you'll find meta classes.
Is this clear? For development you will never use this. For development you will just have either this one which will be under data class.

Or you'll have pirantic. Then you can use that and then you can check your meta classes. This is just to get this metadata, whatever you have here. So it's in the name itself. Meta classes get the metadata of the class class of the class. Is this clear in code? In development we will never use this.
But in order to get all these parameters, just as a argument you can say or context, we will use meta class.
So the only things that you have to learn from intermediate Python is data class Pydantic and what was the other thing? Typing. In typing mainly it is type dict then annotated.
And you have type path. So these are the few important things right? In data classes, frozen is something that you'll have to experiment with.
Till here, is this clear? We'll probably proceed with IDE.
Let me show you one example of basis. Here basis is empty if you see right because example is not inherited. We'll take one example of inherited, then we'll proceed with VS code and I have the ready made code for multi threading and asynchronous programming.
You cannot put meta class on inherited classes.
Huh.
We cannot put meta class on inherited classes. Yeah, we can put. I'll show that now example.
So I do have one example.
We'll take the same example of animal. Did you guys get the URL?
So you can just check github.com AI with Tarun then Python AI workshop.
Don't copy paste the code. Obviously I'll not push any code until the sessions are done.
OK, so here I'll take one example called class animal and what did we had in class animal? We were just telling what sound it makes right? So deaf greet.
Self.
Return.
Makes some sound.
Now here what I will do is I will define my meta class.
Type. So you you you only need to have to tell what are we supposed to do inside my meta.
Dimeter.
Basically, you just have to.

Tirth 1:11:05
under score under score new, yeah.

Tarun Jain 1:11:05
Define new.
How many parameter does it take 4 CLS name, basis and attributes?
If you want to print this, you can print it. But here what I will do is I just have to use the final statement which is return. OK, so I will just comment this out.
And now what we can do is I will define class dog. How do I inherit it?
In bracket animal then how do I make this as? How do I call this dog into metadata?
Not metadata, I mean metaclass.

Tirth 1:11:53
Meta class equal to my meta.

Tarun Jain 1:11:54
So first thing is we want to inherit which everyone of you told you have class dog animal. How do I also pass my meta class here?
Comma meta class equals to my meta.
So you have to give bracket animal, then comma. You have meta class equals to married meta.
And then here doc can have a new function which is greet self.
Return bugs.
So if I print this now, what will be the output?
What is the first line of the output class name? And what will be the name? No name is dog. This is class name.
Then basis probably you don't know, but it will have from where it was in edited and in attributes will have form methods whatever you have.
So if you see you have class class main my meta, the name is dog. Then now in basis if you see you have class main animal. So this class whatever have dog it's been extracted from animal.
And then you have all these information attributes. Now where is this actually used?
Not in the development, but in most of the tree structure that you can say like I have this class, can you give me the metadata of it? Then I'll feed it to the LLM.

So one of them are doing it. I actually wanted to show that startup as well. I don't know if they're still working or not because this product was actually closed because of.
What do you call?
OK, so you basically you have copilots right? In copilot you have your own ID. Best example you guys are using cursor. So in between this was very popular with. Cursor. This was open sourced. So this was actually the first copilot which was open sourced. The name is changed before it was called HID AIDE. Now they have changed the name because this product is closed. So what this product used to do is it was similar to Cursor and it was at top in SWE bench.
But since many people started using the tokens, I don't know what happened. It got closed. So they had one blog article. I don't know if they still have it.
Oh.
Archive.
With this one.
Are they using? Do you know Monte Carlo tree search?
So Monte Carlo tree search is one of the tree search technique. So I'll just tell what currently I want to show you. Let's suppose you want to build a copilot right? In copilot you are generating code for one file which is app dot PY.
In this app dot PY you have some import statements which is redirecting to you in a different file. So that different file if you want to get the schema metadata schema. So this information whatever you have here is passed inside that particular Monte Carlo research technique.
This is the schema which is named dog. I want to debug or generate code for this class and this class is now inherited from animal. So you should also look into where this particular file is and if there is any changes that is supposed to be done, you have to do that as well. So again this.
Things are used in LLMS and one of the search technology was Monte Carlo research. I don't know where the blog is.
Of building Monte Carlo.
It is nothing.
But it's a very old product. It's was.
Huh. Can you see the research paper SW bench enhancing software? So Monte Carlo tree search. So usually there is a framework for this which is called tree sitter.
So this is the framework. You can open any code editor which is open source. This

framework will be there, right? You can check the requirements dot TXT file. You'll find this file which is three setter or you'll find something called as EST.
So EST is nothing but.
Correct. So there is called Hider, Hider GitHub. So if you see this all started back in 2023. You had Eider, you had HED. So this came in, then you had Cursor and all. So all these things were open source, right? You got to know what frameworks are used. To map it, the major challenge in Copilot is if you are generating files in one file, is this file dependent in a different file, right? You need to get that logic to write test cases or you want to generate product documentation. Now I want to generate product documentation for one file.
How will I generate it? I want to know from which file it is dependent on, so that is where meta classes are used and meta classes are used in this frameworks like KST, tree setter. I'll just open requirements dot TXT.
So this is hider and inside hider probably it should be there.
EST you have EST and now if I add 3 seater you have three seater. So for C programming you have. Then if you see three seater is very popular framework, you have three seater and then you have Python version of it.
So these are mainly used for LLM to get more context similar to what we saw in annotated.
So as I said, meta classes was again optional topic, but since most of you use Cursor, I just wanted to show how Cursor is actually built in the back end and everything started with hider only hider. Hid was also open sourced, but they just closed it. Right. And this libraries are pretty old, like in 20/23/20/24 it was started. I don't know if they are still active. OK, last week they made some change. So basically you can just add either in your terminal.
But now you have better tools, so obviously not many people are using it. But when it came out at first, every people were exploring it and then Cursor got boomed. This is before the Cursor era probably.
OK.
So if you see it, probably it started very old. It was in 2023 itself two years ago.
Is this clear? This one more example when you have basis which is animal?
So this will means execute. So what will be the order of if if we are getting console log? So how we are getting this? So basically when it comes to tree sitter right in tree sitter you might have seen grip command.
Grip. So grip is a Linux command. Let's suppose you want to search anything in.

Let's so I'll take one actual example. You have to generate a button. If you click on that button it will trigger LLM call. So you have LLM as a keyword LLM dot invoke. So now this invoke function where is it been called from?
Now this invoke is just a function which is part of some class. Now it will go to the class. It will extract the metadata of it or metaclass so that logic is usually written in tree setter where tree setter is getting that relationship node schema.
So in that schema, you'll have all these details. I'm not sure if our product has locks, but we do have one.
So basically what we built was we built product documentation. So in this product documentation you upload any GitHub file you need and you just choose for which folder you need product documentation. For example, we have all these things right? Beyondlm.applenet.com we just build the framework. Whatever you see on the left hand side, source, embeddings, auto retrievers, not even a single line of text was written by us. So this entire details whatever you have.
Is LLM generated and you'll see common headings right? First you have what that is about and then you'll have some sub headings and then you'll have code snippet. So we had a template that we defined. So once you define the template it will just you have to pass folder.
You add that folder, you generate the code. So if I show beyond LLM code base.
So let's suppose I click on SRC beyond LLM. I click on memory or I click on retrievers, right? Once I click on retrievers, it will generate auto retriever script. Same goes for if I click on.
Embeddings. How much embeddings has? It has total 7 to 8 embeddings and that thing if I open embeddings it will have all the embeddings. Look at the format you have pip install, then params, then code example, pip install params code example. Tip install params for example. So the initial draft was completely generated by. Yeah, we also open sourced it.
OK, the tap is deleted. So basically you just have to select and it send it. We did that for open AGI as well. So if you see the format is same, you just this component is the right. This component was AI generated same year as well.
This core components getting started. We had human touch advanced rag. Also we had human touch only core components. We didn't write anything.
So there we used EAST. So what EAST does is it will get all the context as much as possible, get the relationship node where it is dependent. Once it get that context, we feed to the LLM and that LLM will generate the final response.

We just import and we use grep. So now what grep will do is it will search here whatever you're doing, it's just getting you some string.
So from string, if you want to extract certain code base, you have to use grep. So grep is a Linux command right to find something that was used here. Where is that hider?
I guess grip is there. I'll just confirm if it is.
GREP so can you see this grep AST. So grep is to find AST is to get the tree tree node. So this library is very useful. We even use this AST grep and.
This three sitter.
If you're building anything related to code agents, these are the libraries.
But this is not anything special. This is just to get context what we did with annotated.
OK, so everyone has VS code probably will wind up early itself, but I wanted to show how to set up the code.
If you have Cursor, we can use Cursor also. Anything works. So which ID are you using? I'll probably avoid Cursor at least for workshops. OK, even VOS code has auto complete, but let's see.
I will use this.
I'll also pause this hider if anyone wants to use but now cloud code is better, but still if you want to experiment how hider works.
For developers, this is actually very useful.
Since it is open source right, you can choose any LLM that you need. But when it comes to cloud code, again as I said, cloud code is better. Cloud code can only be used if you have premium account. Then you have Gemini CLI where you can only use Gemini related LLM codecs. Hider choose any LLM that you need.
Right. So you have that flexibility just like cursor in cursor also you can choose, right? But it's good to experiment. I don't think it's that great now, but before it used to be great.
OK, so let's suppose I'll just open terminal.
We'll just set up one project. We will look into one threading example and then we'll wind up.
It should hardly take 5 minutes.
OK.
So I'll just create a new folder called Desktop.
Testing.

Are you on the same page?
So we are in just a new folder, just list the directory. It's completely empty.
I'll try to zoom this more.
Is this visible? Yes.
OK, so now what we need to do is every single time you work with any Python framework, whether you want to clone anything or what, first thing what you have to do is you have to create one environment variable. So the syntax. OK, we have different OS I guess.
So how many of you are using Linux, Linux and all of you are using Linux then fine. So if you're using Linux and Mac, it's the same syntax. If you're using Windows, it's a different syntax, right? So you give Python 3.
Is this working first of all?
Python 3 are able to get. So now what you can do is Python 3 hyphen M.
Then give VNVVNV is nothing but virtual environment and then define any name. So usually people give only two names. One is PPNV if not PNV. So I'll go with PPNV.
So I'll repeat again Python 3 slash M then VNV should be same whatever you have written which is virtual environment then last keyboard whatever you have it can be anything. In my case it is PPNV. You can have ENV or anything.
And once you run it, are you able to run it?
To those who are using Linux, there is one more command which is virtual ENV then PPNV. You can also try this, but if first line work, don't run the second line. Did anyone get any error?
After right running first line, OK PIP needs to be installed. Can you try second one? Virtually ENV, PPNV.
No, the second one.
There is second line. The second line I've written as virtual ENV space PPNV. This is for Linux users.
The first line will work for the Mac users. What is the error? You can also try this.
Pip version version.
Thanks. OK, so probably we'll have to install it. It does start before. It's a Python, this, this, but we'll have to install it. Yeah, thanks.
Share the command. I think you might need to change it to Python.
I guess they have Sudo app install number right? No, no, no. Python dash M it install.
No, no, not that. So if you're a DMV, right, you don't have to run that.
But can you just let me?

VIRTUI oh, can you hear me please?
OK, one more time. OK, I guess it's not there. We'll have to install it in there. Oh, you shared the comment. Yeah, I keep. I don't know if it works.
Python 3 dash dash version.
Please.
They work. I think 3.8 at least for our use case will work, at least for today for rag agents.
Oh, here it work.
I guess the is not installed. I'll just show the command for Linux.
For Mac, how many people are using that?
How do you work? OK.
Ubuntu only, right? Linux everyone are using, yeah.
Yeah.
No, we notice it should be still straightforward.
What?
No.
OK.
Oh, you can try this command on legus.
Can we try this one?
So first this sudo apt get installed Python 3 pip and then use this line.
Pip uh Python 3-M pip install upgrade pip.
There are two comments, not just one.
So this is the first one. So get installed Python and then try this thing.
So actually it showed in suggestion. Actually when you got the development has the KBT get me I mean Python, they might need to. No, this is.
I well, actually.
Thank you.
That is going to.
No, so first you have to write this way.
OK.
Oh, everyone have different.
I will take my.
Uh, here it goes. 3.10 it goes.
Are you going to run it for 3.1 this way? I think there's up of the three double.
OK, can you do LS? OK.

I'll let you. This is not bad.
I want to print the Python 3, the three with one to print each other. So that's not a exciton. So now our game is now in piping.
It's so you keep Windows, not more directed package download for.
107 which is.
Sorry.
OK.
We go to a second if you get the.
Yeah, I can hear them now.
If we find that we keep on.
Right.
And we've installed 3D, new three of the three.
No, because I mean, I don't.
OK.
The the second comments, so I guess quick is already downloaded.
But.
It's something.
Good.
2.
So we're running.
So you'll just have to run one file. I'll have three files in the scripts, whatever I've shared.
OK, so I'll look clear. Can everyone of you see PPNV at least to those who have done? To those who are getting errors, what you can do is you can at least note down the starting three steps. Then we can get into the code. So what was the first thing you write Python 3 hyphen M then V&V.
And some name. OK, after that is done, you just have to write source.
After source, whatever first name is there, right? That has to be in your system. Whenever you created virtual environment, whatever name you give, that name needs to be used.
PPNV and after PPNV you have something called as bin and after bin you have activate.
So PPNVB tab then it will be autocomplete then activate autocomplete and as soon as you run source PPNV bin activate you will have PPNV here.
So now what is trying to happen is in your system there might be chances you might

have run pip install open AI even without running in a environment variable. That open AI version is working in your entire system but.
When you're working on any project, you might need some different version, right? Let's suppose you created a new project, you will have open AI version. So if you looked at either.
So let's suppose I'm using tree sitter 0.23.2 version. Now 0.23.3 broke something and you're using that particular version in your system that will break your system as well if in case you're not maintaining a virtual environment.
So now what happens is as soon as you create a virtual environment and if you install any libraries, it will stick with that particular version itself until you upgrade it. So if I open this now, I'm inside virtual environment.
If I do pip freeze.
It's empty. If I deactivate, I'm I'm deactivating, I'm coming out now if I do pip freeze. I have some frameworks right? Python frameworks. If you see I have QR code which is 8.2 so there might be high chances if I use this particular ENV file after one month. At least three or four frameworks will have different version and that version might not work with my project. So it's always good whenever you're building any new project, first create a virtual environment, install the packages, whatever is used for that framework, have the dependency version. So this is dependency version so that you don't.
Get any dependency version when you are deploying this project. So when you deploy you need the package name and version name. So what is the first step? Create the environment then source PPNV bin activate.
And now if I do pip freeze, it is empty. So now we are creating a fresh project. Whatever library we need, we will install it here. So now what I'll do is I'll install pip. Requests.
Sorry, pip install.
Requests.
Now if I do pip freeze.
I have some frameworks.
All libraries.
So what does this pip install request do? So usually we use this postman right? Most of your developers like get command, put command, post command. So we can use request in Python. Let's suppose you have a local host. What is the get command of it? What is the post command if you want to do?

Do simple testing. You can write a Python script, get the output information right so we can use request. So why am I using this request? Now I'm using multithreading. In multithreading and multiprocessing there might be chances.
Let's suppose you want to download 10 images. So what is ideal way? Download image one, wait, download image 2, wait, download image 3, wait or download all 10 images simultaneously. What is the best ideal way?
Simultaneously. So what we will look now is we will look at an example of multithreading and then once we look at the syntax we will look at one example where you're downloading images simultaneously. OK, so instead of.
Start, stop, start, stop. It will be start, start, stop, stop. Is this clear?
OK, so I have an example which is very simple.
And you'll have to notice some changes that I say here in multithreading. First thing what we have to do is we need to import statement. Now what does import do? You're using this particular package. Inside this package you might have multiple. Functions right? You have active count, you have current thread, you have enumerate. There are tons of framework. So where can we relate this to? We can relate this to import math. I do import math. What are at least two functions that we looked into math?
Some no Matt dot sum is not there.
So we can directly use some.
Maybe.
Absolute py. Absolute is there. Py is there. We actually used py. We used log round. Actually in Python it's outside now before it was inside. So these are a few functions, right? So how do we use it? Import then math.
dot blah blah blah, whatever that function is. So same thing we are doing now. From now on, every single code that we write, we will have import statements every single code. Earlier we were just working with Python syntax. From now we will be using libraries.
And these are some common statements you'll have. Either you will see import some library or you will see from math dot import SQRT.
So from this framework import this particular function instead of getting all the functions right. So these are some things that you will see from all the sessions that you have now which is import.
Now what I have is I have a very dummy worker. Inside this dummy worker I have a name and he will start his work. I'm giving certain delay and he will end his work.

There is nothing special about this function.
So this worker has some name, right? I'm just giving some name and that name will work on something. It will wait for some time and then it will end his work. Is this clear? Now this time dot sleep 0.1 it's like a delay.
Now time is a library, sleep is a function and then it is taking a float. So here if you give one that means it will wait for one second. If you give two it will wait for two seconds, but I just want to give it 0.1 so that I can.
Even 0.1 second difference can show the difference like 2 workers are working at same time instead of work complete, work complete and then once you complete this your function is done.
So this is a simple logic. There is nothing used here apart from time dot sleep. This is a simple print statement with a format string. Simple print statement with a format string and then you have worker with some name. Now is where you have your actual code. You start your thread.
Thread trending dot thread is the class and inside that you are defining your attributes. Now target is nothing but a function. So what is the function that we have here? Worker. So I'll repeat.
When it comes to threading, the first parameter is worker and then you have argument. Now what is this argument?
Stream. This is name. Atyantik is a name. If you miss this comma, it's an error. OK, so you have to have comma over there.
So basically if you give arguments, it takes multiple arguments. So comma we can. So without comma it is string, with comma it is tuple and yeah arguments tuple tuple.
So this is not tuple, it's not a string. Same goes for here as well. If you you can remove this and you can try it right. I hope you have the GitHub for it. We can just copy the inference dot PY code. Now once you define it for both of it, I want to start. Both the thread at the same time, right? I define my threads. Both of them have their functions. They have their own attributes. I want them to start at once and once they start I want them to stop at once, which is join. Once they join, you have to complete the code. Is this clear?
I'm defining the thread, start it, stop it. That's it. Now usually if you run this code in simple ways, let's suppose I use a function I call worker, I give Atyantik.
And then I do worker.
Google right? So ignore this thing, ignore this. What will be the output of just this function? What will be the output of this?

Thread Atyantik started, then thread Atyantik finished. Then you'll have thread Google started starting thread Google finished. This is what these two will print. Now what we are doing is we want both of them to run at same time.
O I'll just come to the cell.
Here only I'll come.
Deactivate CD desktop.
OK, these comments everyone knows I did nothing special. I did Python 3 virtual environment. Then I just entered into it right now. Here my code is inside inference dot PY right? I will just run this Python inference dot PY.
So now if you see thread Atyantik starting, Google is also starting at same time and then Google finished, Atyantik finished. Remember one thing, there is no order for thread. Whoever is starting it, he can finish first or he can finish again. There is no order.
In thread there is no order. OK, so now let's suppose you have 100 images. You're downloading all 100 images at same time. If the first image that you give, if it is downloading at last, that doesn't matter. All the images that you have at least you'll have five or six images that will run at same time. They will complete the task.
There might be one image which is of huge MB. It will take time, it will end sometimes. So in thread the first logic is there is an order. So if three workers are starting at same time, anyone can start and complete based on its own logic.
Is this clear? This was a simple code. I'll take one example, then we'll wind up.
Actually I wanted to finish early today, but.
Let's see. Is this clear? You have the code, I'm just removing it so that I can show an actual example.
OK.
So this is an actual example. Here what we have is this is the same line of code. Now time is also we have already seen.
Are these two lines same?
Now here I have import request. So what did I say for request? You can use get post all those comments right? Now here what I'm trying to do is I'm defining a function which is download image. OK so for download image what I need to pass is just the URL.
OK, so here the URL is.
Obviously it will be one piece.
OK so this is the image. What I need to do is I just have to get this image. Either you

can download it, you can download or you can just print source code. Source code should be 200. Is this clear?
So I have print starting download from this URL then have image request dot get URL content OK and if not you can remove this.
Image dot content here you can also print.
F.
Source code of this URL.
Is.
Image.
dot source code, I mean status code. So is this clear what we're trying to do? So these three lines are same, we are just importing. Then I have a function. I want to run multiple URLs at same time in a thread.
And the logic is you use request dot get at that particular path and check the source code and probably you can write the logic to download the image locally. But here I'm just checking whether that URL is right or wrong and then you have two images. Both the images are original images which have copy pasted and once you do that I'm starting the time just to check. If not this will complicate things, I'll just remove it.
Till here is it clear I defined a function. I have two URLs which needs to be checked and downloaded and now you are starting the thread. So what is the logic to write thread?
You have an import statement which is trending threading after threading thread and what is the first parameter? Target is equals to function name, right? Target equals to function name that is.
Download image arguments should be URL comma. You understood the comma. Why comma is needed? If you don't give comma, you'll get an error.
Then T dot start. So here I'm running a loop. If you see I'm starting it and then I'm just appending it and once that is done I want to join them together. This is the same logic but instead of running it separately.
Where is that?
Yeah, but.
I'm up.
So here I have script inference dot PY. So here instead of running it outside, I'm just running it in a loop. OK.
So let me just run this code.
Now what should be the output?

It should be please.
So it should be starting download from URL. The second line also should be starting download. You got it because I want both to happen at same time. If it's not happening, what is the line I need to add? I just have to add.
Time dot some sleep which can be .1 or it can be 0.05 as well, but I don't want to add any delay. But sometimes if you put load rate that will also cause an issue. But let's just comment this for time being. We'll observe the output if it works.
Then we don't have to give any delay, so I'll write Python.
Inference dot PY.
Now why is it request? You understood request is not there. I need to install it. So last time when I did pip install that was in different folder just for showing you here I'm in actual project folder.
So pip install requests.
This is for my new ENV that I had created.
Hi if I do pip freeze I'll have that pip freeze usually is inside your requirements dot txt. Now I don't have any requirements. Once your project is done, do pip freeze, copy this, paste it in your file. If not you can do this also.
That's requirements dot TXT then.
Cat requirements, cat pip freeze or the best is just do pip freeze.
Requirements dot txt. Is this clear? And then you can look at requirements dot txt.
So now I'll just do Python inference dot PY.
So if you see starting download, what is the second line?
Starting download, starting download, then you have source code, then you have source code. That means both of them are downloading at same time. I mean both the images. You can add five more images and you can test this, but the logic is same. What is the logic?
Start with threading, threading, threading dot thread target equals to function name. Then if that function if it is taking any arguments, give that in.
Yeah, GS arguments. And if you have just one one parameter, make sure you add comma. I'll remove comma now, OK.
I'll clear if I give Python inference dot PY it's an error message. If you see type error download image takes one positional argument but 135 were given. I didn't even give one of the parameter but it's telling that you have given 135.
So here just give comma and if you run this again it will work. So if you have one parameter you that particular URL comma. If you have two URL then comma

whatever.
Which one?
Here, like let's say. So let's suppose it should be keyword arguments then. Huh. If it is named, it will be KWRGS, but here it is the RGS.
It's tuppled.
So this was one example of threading and the syntax. You also have one more example in the code I've shared that is something that for you to try. So for multithreading it is mainly used for IU related which is input and output. Same logic if you use for multiprocessing it is for GPU or CPU related.
And then we have asynchronous which we will cover probably tomorrow and tomorrow we'll start with that quiz of field validator or validator.
I will start bit early 510 minutes.
And the URL is AI with Tarun Python AI workshop.
Any questions we have?
Multi threading is something that won't get in the first attempt. Yeah, actually The thing is that maybe I can understand reading that stuff, but this three form there is no.
These people are what is? So these three concepts that we have right the three concepts that I show now we usually, I mean we AI researchers, we usually never work with those libraries like multithreading.
Then asynchronous programming. Since you'll build fast API endpoints, just because fast API needs to be easy for you, we added asynchronous programming. So for asynchronous programming, the logic of threading will make that concept easy.
So these three concepts are just for those.

◉ **Margi Varmora** stopped transcription