Python and AI Power-Up Program Online Classroom-20250815_120303-Meeting Recording

August 15, 2025, 6:33AM

1h 35m 7s

Mitesh Rathod started transcription

TJ **Tarun Jain** 0:04

So now let me rerun this.

Now I will just write C1 dot.

You have brand.

Which is your attribute?

Now I'll create a new cell which is C1 dot model.

And now I'll speak about state persistence.

So what happens in Python classes is let's suppose you have any instance variables, right? All these are instance variable. They are state persistent. These are state persistent. That means whenever you define any variable within the class.

If you are incrementing it, it will just keep on adding it. For example here it is 0 right? But you are adding that particular speed value. So even if you are defining outside this, that particular speed value will keep on increasing. So let me show you one example.

C1 dot accelerate. I will mark it as 10 now.

R as no object.

No.

OK, sorry, this should be speed.

Did you see this here? I actually wrote self dot speed but it should be. Sorry I wrote self dot num it should be self dot speed.

1:45

No.

TJ **Tarun Jain** 1:47

Because Nam is not a.

Num is not instance variable.

Hardip Patel 1:53
Instance.

Tarun Jain 1:59
Rather it is just a argument of method which is of a accelerate.
And now if I do accelerate, currently the speed is 10 kilometer. Now if I call this again, so basically what happens in function if I'm defining def of accelerate and if I'm giving some value which is numb.
And now if I am telling the same statement.
But here I have a value called speed.
Speed is 0 and then I'm doing speed plus equals to NUM.
OK so I'm just telling what's the difference between function and a method. So I'll run this. Now if I create accelerate and if I give a value called 10.
OK.
OK.
You have 10. Now what I'll do is I'll create the same function again. Now I'll give 20.
So what will be the output?

Mitesh Rathod 3:18
20.

3:19
Karti.

Tarun Jain 3:20
No for function it is. But now for class if I call this method again it will be current speed is 20 kilometer per hour. Why? Because.

Mitesh Rathod 3:22
20.
It will be 20.

Ajay Patel 3:30

Yeah.

OK.

Tarun Jain 3:35

The instance.

Variables or attributes.

Our state persisted.

So if I again call this a method and if I make it 30, now what will be the output?

Mitesh Rathod 3:56

3060.

Tarun Jain 3:59

It will be 60, sorry 101020 + 30 is 50.

Mitesh Rathod 4:01

50 Sorry, yeah 10 like 1550. So the state persistent is based on the the object we have created right? Like C1. If we have created AC2 then it will be another step.

Tarun Jain 4:11

See what?

It will be.

Mitesh Rathod 4:17

Yes, OK.

Tarun Jain 4:18

So now what we will do is let's create a new object.

Now what is self of the current?

So now whatever self is there, right? Now what is self? Self is nothing but current object. So what is our current object?

Mitesh Rathod 4:32

Yeah, yeah.

32.

Tarun Jain 4:39
That is C2. So all the persisted variables are there, right? It will follow with the C2 object. So does anyone have any better example here?

Mitesh Rathod 4:50
Uh, Mahindra Scorpio.

Tarun Jain 4:57
What's the spelling? I hope I'm correct.

Mitesh Rathod 5:01
No, that is fine.

Tarun Jain 5:04
OK, so now if I do C2 dot brand.
You'll have this thing, but you can also change this.

Mitesh Rathod 5:15
OK.

Tarun Jain 5:17
OK, so now if I do CP dot brand again.
It will be updated, so that means instance variables are.

Mitesh Rathod 5:29
Mutable.

Tarun Jain 5:30
Mutable.
OK, so now if I do C2 dot accelerate and if I just give 5, will it be 55 or will it be just 5?

Mitesh Rathod 5:42
Justify.

Tarun Jain 5:44
It will be just five, and if I make this 50, it will become fifty-five.

Mitesh Rathod 5:49
55.

Tarun Jain 5:54
Oh, is this clear?
So the major things to remember here are once you define a class, it will follow along with the init function which is also considered as constructor. So constructor is nothing but it will have the same name as the class name. So now whenever you define the object for this constructor if you define.
Any input variables, let's suppose brand and model. Whenever you are creating the object of it, you need to give the same amount of arguments. So this is this is the first thing and 2nd always every single method that you have in class it will start with self.
That means in it we'll start with self. Every single function or methods that you have inside class will also start with self. So this is the second thing. Now what are we supposed to define inside in it? We need to define attributes or you can also call it as instance variables.
And once this is done, whatever variables that you are defining, they are state persisted for the current object which is self. OK and now if I do self dot num, sorry self and then num, this num is a.
Common variable. It has nothing to do with class attributes or class methods. So now what we are doing is I'm doing self dot speed then plus equals to numb. Every single time I call accelerate it will keep on adding on self dot speed.
So you can also check the differentiator between a simple function and the method.
So these are the keywords to remember object and every single object will have attributes which is instance variable and then you have methods which is also called as behaviors.
Uh, let me just add it as.
Instance variables.
Oh, is this clear?

Hardip Patel 8:01
Yes.

Tarun Jain 8:02
OK.
So now what we will do is let's look at an example of inheritance.
So basically for inheritance what we need to have is we need to have a class parent.
So parent will have its own method, right? It will have. Let me add it as a syntax.
So you have class, then you have parent and the parent will have its own method.
Own method.
Self.
Plus.
And then what you have is you have a.
Child class which will follow what parent does right? So this class can have its own method.
Or it can have the inherited.
Method which is similar to parent which is.
Already defined.
Inside parent.
Self and then pass. So let me take one example. So we have an example.
I will define class animal.
And how do I define? Once I define class, what is the next step?

Ajay Patel 10:02
Init. Init.

Tarun Jain 10:04
You have in it and what should be the first keyword for in it?

Ajay Patel 10:07
First self self.

Hardip Patel 10:09
Now.

Mitesh Rathod 10:10
So.

Tarun Jain 10:10
OK, Sir.
And I'll tell you the importance of what this double this thing is the double under score when I talk about property. So once you have in it, what is the next thing you need to have self and then.
Uh, what is this name now? What is this name?
What is this called?

Ronak Makwana 10:36
In sense.

Mitesh Rathod 10:36
Yes.

Tarun Jain 10:38
Sense variables.
Now what is self?

Hardip Patel 10:41
Yeah.

Mitesh Rathod 10:44
Current object, current instance.

Tarun Jain 10:46
Current instance.

Hardip Patel 10:47
OK.

Tarun Jain 10:50

OK, so now what should we define? I'll just define some function, speak self.

And I'll just tell you.

Return.

If.

Self dot name.

Make some sound.

OK, So what is this called?

Mitesh Rathod 11:24

Ocean.

Tarun Jain 11:27

It's function, but what do we refer it as that class? It is called class method.

Mitesh Rathod 11:36

Yes, that works.

Tarun Jain 11:36

A function, but when it is inside class we just term it as method.

ohh Is this clear?

Ronak Makwana 11:48

Yes.

Hardip Patel 11:48

Yes.

Tarun Jain 11:50

So now let's suppose I create a child class. Uh.

What is the first line that I need to define? Let's suppose I'm taking dog. OK, what will be the first line of the dog?

Hardip Patel 12:09

Hello.

Ronak Makwana 12:11
Glass dog animal.

Hardip Patel 12:12
Mhm.

Mitesh Rathod 12:13
Dog and in bracket any.

TJ **Tarun Jain** 12:16
Dog and then.
Animal.

Mitesh Rathod 12:19
in bracket animal.

Ronak Makwana 12:19
And.

TJ **Tarun Jain** 12:21
So as as of now if you see when I define my first class.
I just defined it as class car. OK, there was no bracket. Here also if you see when I define animal, there is no bracket. Whenever you define a bracket, that means you are inheriting this particular class.
From a parent class which is already defined. Is this clear?

Mitesh Rathod 12:48
Yes.

Ronak Makwana 12:48
Yeah.

TJ **Tarun Jain** 12:48

You will also see some examples when a same class is inherited from two different class. So I'll just tell animal and I'll also say ban.

Hardip Patel 12:50
Yeah.

Mitesh Rathod 12:58
So one thing.

Tarun Jain 13:01
OK.
So the same class Dog now is inherited from animal and as well as band which is already defined.
So you'll also have some syntax where a same class is inherited from two different class.

Mitesh Rathod 13:21
OK.

Tarun Jain 13:24
So what is the first thing that I need to define?

Mitesh Rathod 13:27
And what is that?

Hardip Patel 13:29
Constructor.

Tarun Jain 13:31
I can prefer in it.
I'll just write self.
And uh, do I need to?

Hardip Patel 13:42
M.

Tarun Jain 13:43
Define anything? Probably I don't have.

Hardip Patel 13:45
Yeah, so.

Tarun Jain 13:49
When I'm defining animal, animal already has one in it. So basically dog also is any written name. So if you see this thing is the right.

Mitesh Rathod 13:52
OK.

Hardip Patel 13:54
Thanks.

Tarun Jain 14:01
I'll just remove it. This thing is already existing in dog.
By default.

Hardip Patel 14:11
OK.

Tarun Jain 14:12
So what do I mean by this? Let's suppose I want to create an object of animal. How will you create?

Hardip Patel 14:13
Yes.

Tarun Jain 14:21
How do you create the object of animals?

Hardip Patel 14:26

Uh, and while it was.

Mitesh Rathod 14:28

Animal.

Tarun Jain 14:30

Any variable. This can be any variable A equals to animal.

Hardip Patel 14:36

Mhm.

Tarun Jain 14:36

And then you have to give certain name right? So this name you can give anything that you get and you can run it. So now what you can do is you can just tell cat sorry A1 dot C.

Mitesh Rathod 14:49

Even taught me speak.

Tarun Jain 14:52

So this will tell cat make some sound, but do you think this is a very good message to give?
So it should be cat makes meow right? Or dog barks. So this is where whenever you're defining dog, it is already inheriting the constructor.

Mitesh Rathod 15:05

OK.

Tarun Jain 15:13

When you are using.
Inheritance.
It is already inheriting the constructor.
So now for instance, let's suppose I have something inside class. How will I define

dog?
So just look at the example for animal. If you want to create an object you would told A1 then animal and then you give a name. Now what I said is when you are inheriting the parent class to a child class.
The inherit, which is your constructor, is already inherited. So the same thing whatever you have here is existing here, but you don't have to define it. The above thing is already there.

Hardip Patel 15:55
Mhm.
Yes.
Yeah.

Tarun Jain 16:09
So there is no need to.
There is no need to repeat.
So now now that you know this is existing, how will you define an object?

Hardip Patel 16:20
OK.

Mitesh Rathod 16:23
So uh D1 equal to.

Tarun Jain 16:26
B1.

Mitesh Rathod 16:28
Equal to dog.

Tarun Jain 16:30
Dog and then you have name.

Mitesh Rathod 16:31
And the in bracket, yeah.

Hardip Patel 16:32

Um.

I don't know.

Tarun Jain 16:35

You understood, so if you notice.

Oh, what happened?

Hardip Patel 16:39

Does this work?

Tarun Jain 16:41

Yeah, this will work. So as I said inherit. What does the constructor have? It is self comma name. So when you're defining dog, dog needs to have a name, right? Because here you have a name.

Hardip Patel 16:54

2.

Mm.

Tarun Jain 16:57

If I don't define a name, you'll get an error.

So if you see animal in it missing one requirement positional argument which is name. Now where is this name coming from?

It's coming from here.

Mitesh Rathod 17:11

Anyone.

Hardip Patel 17:12

And now, OK, got it.

Tarun Jain 17:14

So what do I need to do? I just have to do one name. I'll just tell it coming. So now if I do D1 dot, what function does it as?

Hardip Patel 17:24
Speak.

Mitesh Rathod 17:25
Speak.

Tarun Jain 17:27
Beep.
Do I need to pass anything? No, it's empty. So self. Now what is self? What is self here?

Hardip Patel 17:35
No.

Ajay Patel 17:36
No.

Hardip Patel 17:38
OK.

Mitesh Rathod 17:39
Do you want?

Hardip Patel 17:40
OK.

Ajay Patel 17:40
Our dog instance of dog.

Hardip Patel 17:41
Instance.

Tarun Jain 17:43
Instance of dog that is D1.

Hardip Patel 17:44
Yes.

Ajay Patel 17:45
Dog.

Mitesh Rathod 17:46
Yeah.
OK.

Tarun Jain 17:49
Instance of dog is nothing but current object.
Is this clear? And do we have any parameter for speed? No. So I'll just write D1 dot speed from your box.

Hardip Patel 17:59
Yes.
OK.

Mitesh Rathod 18:02
M.

Tarun Jain 18:07
Let's look at.

Ajay Patel 18:07
So Tarun I guess in Python in Python we don't need to write an extends right? Otherwise what happened in PHP or Java also we have to write specifically right? Like if parent class is there then child class extends parent. So this is same as.

Tarun Jain 18:21
Yeah, correct. Correct. We don't need to do that.

Ajay Patel 18:25
OK.

Tarun Jain 18:27
So this thing will take care of it this particular line. The only thing that follows is all the init function which is constructed and the methods will be same as the inherited class.

Mitesh Rathod 18:41
So one question, if we are inheriting like multiple classes, let's say dog have animal and we have another class as well. So which constructor it will be called?

Tarun Jain 18:55
It will have the constructor of the first one, probably. Let's try it.

Mitesh Rathod 18:56
All right.
Like both have a Yeah, OK.

Tarun Jain 19:03
So here for band.
I'm defining a net.
Self.
I'll just dial Nishan.
Self dot nation equals to nation. So what I'm trying to do is I'm checking if dog named Tommy is banned in this particular nation or not.
Duff use ban.

Mitesh Rathod 19:35
They.

Tarun Jain 19:40
OK, it's so clever. It said India just now.

Mitesh Rathod 19:43
Yeah.

Tarun Jain 19:45
If self dot.

Mitesh Rathod 19:46
But there's things also.

Hardip Patel 19:46
It should say it should say Delhi, I guess.

Tarun Jain 19:48
It was from India.
I'll just tell you.
In yes band.
In India.

Mitesh Rathod 20:07
Mhm.
The.

Tarun Jain 20:08
So just define this. I'll tell K equals to ban India.
Then K dot is ban.

Mitesh Rathod 20:21
OK.

Tarun Jain 20:23
It is, yes, band in India. Now usually every single DOP class is there, right? You also

have something called a super this thing. Let's let's suppose you are defining def in it.

Mitesh Rathod 20:30
Mhm.

Tarun Jain 20:39
Self. So here if you see you have the super keyword. So this will tell you from which class you want to inherit it, whether you want to inherit it from animal or you want to inherit it from bank.
Uh, is it clear? I'll just run this.

Mitesh Rathod 20:53
Do we need to like how it will it will consider?

Hardip Patel 20:56
Yes.

Mitesh Rathod 20:59
Like how you consider like I need to call animal or or it is just taking a first as a by default?

Tarun Jain 21:08
So let me check if it is is ban.

Mitesh Rathod 21:08
This.
Look.

Tarun Jain 21:35
It takes two positional arguments, but three were given. Let me give it as name equals to.
Nation equals to India.

Mitesh Rathod 21:47
No nation.

Tarun Jain 21:55
I guess it has to do with super variable, super keyword, uh def in it.

Mitesh Rathod 22:06
Nation.

Tarun Jain 22:09
Super.

Mitesh Rathod 22:11
OK.

Tarun Jain 22:12
dot in it.
Nation.
Now if I do D1 dot D9.

Hardip Patel 22:25
I think instead of using super, what about if we use self on the self dot name equals to name and self dot nation equals to nation.

Tarun Jain 22:25
Dog objector.
That's it, right? Then this becomes as if you're defining for your own.

Hardip Patel 22:41
We then you are not using parents constructor.

Tarun Jain 22:43
It still work. It's like now this is not parenting. It can inherit the functions, but it's not inheriting your.

Hardip Patel 22:48

Mm.

Mm.

Mitesh Rathod 22:53

Data attributes, yeah.

Tarun Jain 23:01

It will tell less ban in India, but this is not inherited.

Hardip Patel 23:01

So.

Tarun Jain 23:07

This constructor.

Is not inherited.

Mitesh Rathod 23:13

Is a.

Mhm.

Tarun Jain 23:16

So I will look into this example. As of now if you see it is inheriting the class that it needs which is from two different classes. But how to inherit the constructor? Probably I'll just come back on this on Monday.

Mitesh Rathod 23:31

Yeah, that's fine. OK.

Tarun Jain 23:34

Because I know somewhere it has to do with super.

Hardip Patel 23:39

It.

Tarun Jain 23:40
Let's see if auto completes, give it or not.
Oh, wait, did you get something?

Mitesh Rathod 23:46
Uh, yeah, yeah, the OK.

Tarun Jain 23:52
Oh yeah, it should be band dot generate then self link.

Mitesh Rathod 23:55
OK.

Hardip Patel 23:55
No, yeah.

Mitesh Rathod 23:57
The class name dot in it OK.

Tarun Jain 23:57
So now this is in.

Hardip Patel 24:01
Yeah, the same for anyone.

Mitesh Rathod 24:02
So instead of super, yeah, instead of super, can you directly write?

Tarun Jain 24:05
But wait, I should actually not give this.

Hardip Patel 24:09
Yeah.

Tarun Jain 24:13
Oh wait, I guess I have to get it, but instead of.

Mitesh Rathod 24:15
No, of course.

Hardip Patel 24:15
Yeah, yeah.

Tarun Jain 24:18
Yeah, here you can give animal.

Mitesh Rathod 24:21
Yes, I think.

Hardip Patel 24:21
Yeah, about self you don't need. You need to be self and anyone in in a team.
Mhm.

Mitesh Rathod 24:34
OK.
Oh.

Tarun Jain 24:36
Yeah, no, probably this was first time the auto complete work.

Mitesh Rathod 24:37
OK.

Hardip Patel 24:37
Yeah.

Tarun Jain 24:43

This is for inheriting from the parent.

Plus.

Mitesh Rathod 24:52

Multiple currents I can say.

Tarun Jain 24:55

Oh.

OK.

The second alternative is.

Using super.

Function. So the syntax for super function is super dot inherit. So you don't have to give a define what you call self, you can directly define name. So now here super is referring to.

Animal which is the first one. So if I remove this.

Mitesh Rathod 25:29

Anyone. That's all.

Uh, Super should be a function, right?

Tarun Jain 25:38

Yes.

I can use uh what was another function speak.

Mitesh Rathod 25:47

Pick.

Tarun Jain 25:49

OK, so I'll come back to the syntax. So if you see parent has their own method, so child is using the narrated one. Now what we'll do is we'll define our own method of child.

Mitesh Rathod 25:49

30 bucks.

Tarun Jain 26:03
It has speak. So how many function does dog has now? Dog has speak. It has ease ban.
And what I'll do is I'll add new function called defeat.
Self.
Return what is usually dog eats.
I'll just add biscuit self name loves.
Apology.
And if I just add it, it says Tommy loves Parlegi. But if I do A1, So what is A1?
A1 is the object of animal right? But if I do a one dot eat it will be an error because eat is a function of. Sorry, eat is a method of.

Hardip Patel 26:53
Animal object.

Tarun Jain 27:05
Child, not parent.
Child can inherit parent method.
But parent.
Can't.
Inherit child method.
Uh, is this clear?

Hardip Patel 27:28
Yes.

Mitesh Rathod 27:29
Yes.

Tarun Jain 27:31
I'll just start C1 dot eight. Sorry, it's D1.
OK, so I'll just summarize what we did before I proceed to property. So the first thing is when we start with inheritance, the first class, whatever you define, which is the parent class, will not have any bracket, right? So you can have your own instance

variables, you can have your own methods, but when you are inheriting the child.
Child. What it will do is it will inherit the own methods. Also you don't.
Have to define the constructor.
But if in case this child is inheriting from two different parent, that is when you can use super function.
Or you can directly define that particular parent class dot in it. And what is the first thing you need to do? You need to give self and whatever instance variable that class is providing, you have to give that instance variable.
And child can have its own methods. So this own method is only accessible by child, parent can't inherit it and child has the flexibility to also inherit the methods which is provided in the method parent method which is bar and is ban example which was defined in animal and.
1.
Is this clear? What is instance variable? What is method? What is super function and what is inheritance? And this too is very important.
Uh, any questions before we proceed to property?

Mitesh Rathod 29:21
Yeah.
No.

Tarun Jain 29:28
OK, so property is very simple. In simple words, what property does is property is a decorator.
Which we'll cover next after property. So let's suppose you have any method and if you want that method to act as a attribute.
So what does this mean? So what is the method in dog?
What are the methods dog have?

Hardip Patel 30:09
Uh, speak and eat.
And eat.

Tarun Jain 30:12

Speak and eat. Let's suppose I define speak. I'm defining like this, right? So if I do speak colon, this is method.

Hardip Patel 30:13
And he's bad.
Yeah.

Tarun Jain 30:25
Now, how do you define attribute?

Hardip Patel 30:28
Uh, in May.

Tarun Jain 30:30
D1 dot what are what does it have name?

Hardip Patel 30:33
No.

Tarun Jain 30:35
It is Sami, right? This is attribute.
Or class instance variables.
OK, now what I want to do is I want to define D1 dot speak.
And when I do D1 dot speak, I need to get output as Tommybox.

Hardip Patel 30:55
Mm.

Tarun Jain 30:56
So if you want to do that, this is where we will define property. So let's define one example called.
One second.
Oh.
I'll just define a bank account. So what basically we'll try to do is we'll define bank account and when we define bank account I will define certain instance variables

which is attributes and in that attribute I'll have a balance button right? So every time I want to know the balance for that I will have a function but.
It should act as a attribute, so.
Class bank account. What is the first thing I need to do after class bank account?

Mitesh Rathod 31:46
Create a minute method.

Hardip Patel 31:46
Constructive.

Tarun Jain 31:48
Constructor.

Ronak Makwana 31:48
Initialize it.

Tarun Jain 31:51
Remember, since this is a parent, I'm defining constructor inherit. What is the first keyword argument?

Mitesh Rathod 31:54
Yes, yeah.
OK.

Hardip Patel 31:58
OK.

Ronak Makwana 32:01
7.

Mitesh Rathod 32:02
Sorry.

Tarun Jain 32:03

Self and I will just define balance.

And if I do self balance.

Equals to balance.

Now is this state persisted or is it not state persisted?

Ishan Chavda 32:21

Skip.

Hardip Patel 32:22

Assisted.

Tarun Jain 32:22

It's set for system.

OK, so now what I will do is I will define property.

And then what I will do is I will define a function called get balance.

Sorry method and I will add it as self here. What I need to do is I just have to return self balance. I will tell you the importance of this. First I'll show how property works, then I'll tell you why we need it by telling an example of public.

Protected and private I guess. Do we have this concept in PHP and?

Hardip Patel 33:04

Yes.

Mitesh Rathod 33:05

PHP, yeah, XS 45.

Tarun Jain 33:06

Yeah, so I'll if I yeah access modifiers mainly for I guess for Java also it is same.

Hardip Patel 33:15

Mm.

Tarun Jain 33:15

But in Python we just have two of them, which is public and private. Protected is there, but there is not much meaning in the protected. You call it as a convention. So

I'll show that example as well. For time being you have class bank.
Account. Then we are defining the constructor. In constructor we have an attribute called balance which is state persisted and now we have a property class so that whenever I want to call this method I can call it as a attribute.
Right, so let's also have one more method which is not under property. So I'll just tell I'll use the same auto complete. You have def deposit self amount and I will add self balance plus equals to amount. So now if I keep adding any amount, the self balance will keep improving.
ohh Is this clear?

Hardip Patel 34:06
Mm.

Tarun Jain 34:07
So how do I define the object now?

Mitesh Rathod 34:09
Yes.
Let's say BA equal to.

Tarun Jain 34:12
How do I define an outlet?

Ronak Makwana 34:12
Any variable. Any variable.

Tarun Jain 34:17
BA equals to.

Mitesh Rathod 34:19
Bank account.

Ronak Makwana 34:19
Bank account.

Tarun Jain 34:22
Bank account, some initial balance. I'll add it as 1000.

Mitesh Rathod 34:26
Person and.

Tarun Jain 34:28
OK, so now if I do.
BA dot balance. What is the output?

Mitesh Rathod 34:38
OK.

Tarun Jain 34:41
It's thousand. Now I'll just do VA dot deposit.
I'll add thousand.
Now instead of doing BA dot balance, what I'll do is I'll do BA dot get balance. I'm not using any bracket, I'm just calling get balance. It is 2000.

Mitesh Rathod 35:08
Oh.

Tarun Jain 35:09
And now I'll do BA dot balance. It's the same result.
Is this clear? What property is doing is whatever you have as a method that will be converted as an attribute. That means you don't have to define the brackets.

Hardip Patel 35:19
Yes.

Tarun Jain 35:32
You understood. You don't have to define the brackets, you just have to define get balance. That's it. Now why is this even useful? Usually most of the time what happens is if you have any functions or methods which is which has no.

Hardip Patel 35:35
M.

Tarun Jain 35:48
Parameters or arguments. Let's suppose you have no arguments here, right? You just have self.
So it doesn't make sense to have this as a method, so you can directly define as a attribute. So this is mainly useful when you have.
Instance variable which are protected or private. As of now what I'll do is I'll use the same example bank account.
And here I will define in it self comma balance.
And I will define self dot balance equals to balance.
So this over here is public.
OK, so public in the sense whatever you're defining as of now, which is self dot balance equals to balance. What I'll do is I'll remove this. I'll make it 500 or 5000 directly. Now this is public.
Then you have self dot single under score and then if I define balance.
4000.
This is protected.
But in Python you will always call it as.
Protected.
But by convention.
Now why is it called by convention? Because public.
And protected.
Both performs the same.
During runtime I will show an example for this.
OK, so when I say it performs same during runtime, what we usually do is.
Here if you notice if I do ba dot balance equals to 400, will it be updated?
If I do this syntax, do you think this will work?

Hardip Patel 38:01
Yeah.

Mitesh Rathod 38:02
Yes, it will work.

Tarun Jain 38:03
It will work. Now if I do VA dot balance, it will be 400. It won't increment right? Because you're overriding your instance variable, so it's 400, correct? Now if I do single bracket, let's suppose here I had single bracket.

Hardip Patel 38:03
So.

Ronak Makwana 38:03
Yes.

Hardip Patel 38:13
Very well.

Tarun Jain 38:21
Even then it will work. VA dot single balance equals to 400 even that will work. Now this single thing is there right? It is referred as protected, but why is it added single so by convention in the sense.
Whenever you notice.
This variable which starts with balance.
Only for internal purpose.
It tells user not to change it.

Hardip Patel 38:56
So.

Tarun Jain 38:56
Not to change this variable.
Or attribute.
Is this clear? Both are same. So whatever you do with public, you can do that with protected as well. But the only thing is it is defined as convention. That means for

internal team purpose, whenever you have, whenever you see this under score right, you just have to be informed that you should not modify.
This particular instance variable outside the class when you're defining the object.
This is only for self-awareness.
Is this clear? What do you mean by by convention? Both will perform the same operation. Whatever you do with public, you can do with protected. But the only thing is when you see this, don't change it outside the class. Now if you want to provide restriction, that is where you have private where you have to give double.

Hardip Patel 39:35
Mhm.
Yes.

Mitesh Rathod 39:40
OK.

Tarun Jain 39:54
Brackets.
Now this balance if I give 10,000, this is private.
That means if you do BA dot double balance equals to 300 outside when you create object it will show an error.
If you want to define this balance, this is where you do property.
property.
You have get balance and then double under score balance and then you can have the same other function def deposit self amount self dot balance plus equals sum on.
So is this clear? This code is same. The only thing is I changed.

Hardip Patel 40:39
And.

Tarun Jain 40:42
Balance to double under score.

Hardip Patel 40:44
And deposit double under score balance no the assignment.

TJ **Tarun Jain** 40:46
Deposit is also that we can do like if we want to add self dot balance.

Hardip Patel 40:53
OK.

TJ **Tarun Jain** 40:55
We can do it for all plus equals to amount.
Usually you will not have three variables with the same name. I'm just showing so that I can show the difference.

Hardip Patel 41:03
Yeah, yeah.

Ajay Patel 41:08
So Tarun, there is no keyword like for private, protected and public right? Only by under score we can identify them as a either it is a public, private or protected.

TJ **Tarun Jain** 41:10
Hello.
No, no.
Correct. No keywords. So I'll just run this code. Now I'll define BA1 equals to bank account. Do I have anything here?

Ajay Patel 41:20
OK, OK.

TJ **Tarun Jain** 41:32
Do I have any? I mean variable in self, noted. So I can find no argument.

Mitesh Rathod 41:33
No.

Ajay Patel 41:34
No, no.

Mitesh Rathod 41:36
No documents, no.

Hardip Patel 41:36
Mhm.

Ajay Patel 41:37
No blank.

Tarun Jain 41:41
So I will just define bank account one.
OK, so now if I do BA .1 balance.
Equals to 300.

Mitesh Rathod 41:59
Sorry.

Tarun Jain 42:01
This will be updated.
Now if I do BA one, if you notice here in the auto complete there is no under score balance.
Even though I said you can change it, it's not visible.

Mitesh Rathod 42:15
Yes.

Tarun Jain 42:18
Because that is trying to say that hey you can use that only within the class. Now if I do this 400 this will work.
But as per the convention, you should not do this.
So you should be self-aware, like whenever you see under score balance, you want

that particular variable to be protected. So in terms of programming, we usually don't override this particular variable, but if you actually want to provide restriction, that is where you can define it within double balance.
Now this will throw an error.
Because you can't even call this outside. You can't even assign it.
So if you just call this.
OK, this is again new variable, but it will not follow what variable you have inside.
For example if I do BA1 dot deposit.
500.
BA one dot balance. Now what will be the output of this?
What will be the output?

Hardip Patel 43:41
There it was.

Ajay Patel 43:42
900.

Ishan Chavda 43:43
Yeah.

TJ **Tarun Jain** 43:44
It was 300.

Ajay Patel 43:44
900, not 300, OK.

Ishan Chavda 43:47
OK.

TJ **Tarun Jain** 43:47
300 + 500 it should be 800.

Ajay Patel 43:48
508 hundred 800.

Tarun Jain 43:51
Now if I do double balance.
Sorry, single balance.

Ishan Chavda 44:00
OK.

Ajay Patel 44:01
400.

Ishan Chavda 44:02
2.

Tarun Jain 44:03
You have to define a variable. It was 400. Now again I'm adding 400, it will be 800.

Ajay Patel 44:10
Deposit.

Hardip Patel 44:18
You are.

Ajay Patel 44:19
There is no function under score balance.

Tarun Jain 44:24
Oh, sorry, sorry, sorry. I I I gave this thing which we should not give because I'm checking the balance rate. It's 900. Why 900? Did I add 500 or?

Hardip Patel 44:37
I think you added just DBA1 deposit 500 now.

Tarun Jain 44:41

OK, 500 if I give double equals.

And now VA dot deposit. OK, so if you see the balance, it's not updated.

Hardip Patel 44:54

Mm.

Tarun Jain 44:56

So what we did was we did BA one dot deposit.

500 initially what was the value of balance? Balance was 300.

Hardip Patel 45:12

Mhm.

Tarun Jain 45:16

Balance was.

400 if you see under score balance was 400.

Then double score balance. You can't access it.

Can't access outside, but what I did was I overrided it and I made it 500.

Now when I do BA dot deposit 500, this will be plus 500, this will be plus 500, this will be plus 500, but this is not possible.

Now if I do this again, what will be the output if I run this again?

Hardip Patel 46:00

Yeah, let's try one drink and.

Tarun Jain 46:05

So I'll just run this. So BA1 double under score balance. It is still 500 because I overrided it.

And if I do double single under score balance, what is the output?

Hardip Patel 46:22

Uh, 1400, 1400.

Tarun Jain 46:25

It's 400 and 400. If I just add balance, it is 1300.

Mitesh Rathod 46:26
What what you're looking?

Hardip Patel 46:33
Yes.

Tarun Jain 46:34
Is it clear what I meant by public then protected but by convention and public is nothing but we added by double score under score. So if I rerun this and if I rerun this line directly, it's an error.

Hardip Patel 46:52
Hmm.

Tarun Jain 46:53
Oh, and.

Hardip Patel 46:53
So like if we assign private assign value to a private like it should not work. Is it something like that? I didn't understand about the deposit because for deposit the balance should be updated.

Tarun Jain 47:12
But you can't access that outside, right? Outside class you can't access the double balance. So let's suppose if I do get balance, let me run this and let me run this.

Hardip Patel 47:15
Um.
Oh, OK, so get balance will be fine. Get balance will be updated, right?

Tarun Jain 47:27
And if I do get balance.
Of V1 dot get balance it is 10,000.
Now what I'll do is I'll just add deposit P1 dot deposit.

Hardip Patel 47:39
Yeah.

Tarun Jain 47:46
So get balance. What is it doing? It's only giving me the private one which was 10,000.

Hardip Patel 47:51
Yeah.

Tarun Jain 47:53
If I do BA1 dot deposit 500.
This should be how much 5000 + 500 which is 5500.
Oh, I wait, let me run this again.

Hardip Patel 48:12
I think I get balance should be 10,500, right?

Tarun Jain 48:14
5500
No, get balance is 10,000, right? So I just defined it. I'm just defining the bank account after deposit. Let me run this again.

Hardip Patel 48:23
But after deposit, I mean like after deposit, it should be 10,500, right? OK, OK, I got it.

Tarun Jain 48:30
It is 10,500.
So you can't run this outside.
So let me remove all these things. We will not override any variable BA1 dot. If I do double balance this is an error. OK now if I deposit more 1000.
What is VA1 dot balance now?
BA one dot Balance.

Hardip Patel 49:00
6500.

Tarun Jain 49:05
So it was 5500, it will become 6500. OK, now let's use a new one, BF1 dot single this thing, then balance.
Protected. What will this be now?

Hardip Patel 49:25
So straight 400 I guess.

Tarun Jain 49:27
So it was actually 4000.

Hardip Patel 49:31
OK, yeah, 405500.

Tarun Jain 49:31
Then I guess we added 500 and then we added 1000.
So it will be 5500. OK, now we saw public, we saw protected, but since we can't run double under score balance, since we can't run this, we have a property called get balance.
So now what this should be? It should be 11,700.

Hardip Patel 50:00
Yeah, I got it.

Tarun Jain 50:02
So if you want to access.
If you want to access private.
Instance variables.
As an attribute.
Outside the class.
Define that instance variable.

Inside.

And property.

Decorator.

Oh, is this clear?

Hardip Patel 50:51

Yes.

Tarun Jain 51:05

OK.

OK.

So let's take one example just for a quiz for bank account to definite.

So can anyone tell me the output of this?

Mitesh Rathod 52:09

Thanks. So first is 51,001 country.

Then with true what to fix? Uh.

Tarun Jain 52:14

So what will be this be?

So let's take it down so that the calculation becomes easy.

This will be how much?

Mitesh Rathod 52:25

It will be like the 505,100, 51,000.

Ajay Patel 52:29

6000.

Tarun Jain 52:30

It will be 6000 so it is 5000 then I'll add plus thousand which becomes 6000. Now I withdraw 500. So this how much will it?

Mitesh Rathod 52:32

Oh, you are the OK.
Mhm.

Ajay Patel 52:42
5500500

Mitesh Rathod 52:42
55.

Tarun Jain 52:45
Now if I run this, what will happen?

Mitesh Rathod 52:48
If you say.

Tarun Jain 52:49
This is an error.

Ajay Patel 52:50
5500.

Tarun Jain 52:52
Why is it an error?

Ajay Patel 52:58
Because we are accessing private property.

Mitesh Rathod 52:59
Property. OK, it's a property. Yeah, yeah, no, it's a property, but we are calling it as a function.

Tarun Jain 53:03
When we do property, what should we do? Correct. It's a property. We just have to do get balance. Now this will become 5500.

Mitesh Rathod 53:07
Just yeah, yeah.

Ajay Patel 53:08
Oh.

Tarun Jain 53:16
You understood. You have to keep this in mind whenever we are defining property and if you are calling it, it should be as an attribute and not as a function.
And here for the entire example I'm using private.
So if I do BA.

Mitesh Rathod 53:33
Yes.

Tarun Jain 53:35
Two and if I just add balance.
What is the output of this?

Mitesh Rathod 53:40
It would be 5555.

Tarun Jain 53:43
This is an error. I'm not using function, I'm using the.

Mitesh Rathod 53:46
Not calling, not calling. OK, yeah, yeah, it's a private use.

Tarun Jain 53:51
OK.
This attribute is an error.
And this getbalance is a property method.
I hope you understood the difference between get balance and the common attribute.

Hardip Patel 54:09
Yeah.

Mitesh Rathod 54:11
Yes, got it.

Tarun Jain 54:13
OK. So we'll take decorator as well since we already mentioned property.
So if you notice this particular keyword right at and then property, this is usually referred as decorator.
So when do we use decorators?
So most of the time when I what happens is let's suppose you are running some DB operations. So what are the different DB operations we have add into database, then we have delete.
Into database and then we have modify.
And then uh.
I'll just give this three and let's suppose you also have some logging statements, right? So whenever you're running this kind of operations, you usually have one function.
Which is initializing few things.
And most of the times, most of the times.
This function logic changes for each function, right? I'll just tell each sub function.
So this is where we have decorator.
Which is nothing but a function.
Under another.
That acts as a wrapper.
You can refer this as.
Nested function, but it's not specific as nested function, but in terms of understanding you can call it as nested function, but specifically.
It's just function another another function.
Is this clear? I'll show you the example. So basically whenever you see at anything right, let's suppose.
Act symbol.
Followed by any name which is property.

You have this particular function defined somewhere that is def.
Space property.
Self or it can be simple function. This is defined anywhere.
Then only you can utilize property.
So let's take an example so that it makes sense why I've added the DB operation, add into DB, delete into DB and as well as modify. So let me quickly open one example.
So let's suppose I have some DB operations.
And whenever you are defining functions or what you call whenever you want to define decorator.
Your first parameter will be FUNC, which is a function.
Again, I'll repeat what decorator does is decorator takes a function under another function. So the input of DB operations will be a function. OK and inside this what you can do is you can define another function. I'll just tell log.
The operations.
And I will just add some statement inside this print.
Log.
Running database.
Operation and since we are defining function, I'll show one example. Let's suppose I have a function called.
Greet name.
So this is great.
You have something called as name.
It's greet right? So this greet what I need to do is when I'm defining this particular function, since I'm passing any function variable name right, what you can do is you can just add function dot double under score name.
So let me tell you what is happening here. You have multiple operations like you have add into DB.
Then you have delete from DV.
Whenever you're defining this function, I want to add a decorator which can log all the operations, right? So this is the log operation that I'm adding and every single time I'm running this, I have a log message saying running database operation and that function name. So what is the function name that I have here? One is add into DB and one more is delete from DB.
So once I do this, what I just have to do is I have to return the function.
Return function.

And then this is the return function of. This is the return statement of.
Log the operations.
And this return here you can just add log the operations for the DB operation.
So I'll explain this code again. Once we define these two sub functions, that way it will make more sense.
So now what I will do is I will use this DB operation as a decorator. So what? What is the first keyword of decorator? What is the first symbol?

Hardip Patel 1:00:44
International.

Margi Varmora 1:00:45
Enter it.

Tarun Jain 1:00:47
Of what?

Hardip Patel 1:00:49
I I didn't.

Margi Varmora 1:00:49
Excellent.

Tarun Jain 1:00:51
At the rate. Now what? Which is the decorator here? It is DB operation.
So as per the syntax, if you see you already have def property which is defined somewhere. If you want to reuse this as a decorator, you just have to give at the rate property and inside that you can define some functions. So the first function here I have is.
Def insert data.
And here what you can do is you can define any logic that you need. But for time being since you are just logging it, I'll tell inserting data into DB. Then here you will have logic.
And then you will return something, right? You will return.
Either a message which is data inserted.

Successfully.
I will just run this. Now I'll define one more function. I will define it under DB operation itself.
Def delete.
Data.
Here I'll define.
Deleting data.
From BB.
Then the logic for that and then return.
Data deleted.
So let's go backwards now, right? Instead of starting from here, what I'll do is I'll go backwards. We'll start from def insert, right? So how do you call this function? You just have to define insert data. Now since this is a decorator, what it will do is first it will look into this decorator operation.
And what is the function here? This function is nothing but.
Insert data now.
So I'll again repeat, I'm defining insert data which is my function.
This is my function, but since I'm adding a decorator now the FUNC whatever parameter I'm adding inside DB operation is insert data. So now I will log inside the operation. So what will be the first line of answer that I have?
Once I printed this, what is the first line?

Mitesh Rathod 1:03:18
Log running is operation, database operation.

TJ **Tarun Jain** 1:03:20
Logging uh log running database operation. What is the output of this?

Mitesh Rathod 1:03:26
Uh, insert data.

Hardip Patel 1:03:27
Insert it.

TJ **Tarun Jain** 1:03:27

So since I'm passing insert data as a function, so if I do function under score under score name, the output will be the function name itself which is insert data.

Mitesh Rathod 1:03:33
Mhm.
U.
Yes.

Tarun Jain 1:03:40
OK, now I will return the function. So when I do return the function where it will return back to, it will return back to.

Mitesh Rathod 1:03:49
Insert data.

Tarun Jain 1:03:50
This thing. So here what will it print now? Return statement of.

Mitesh Rathod 1:03:55
Inserting data into DB.

Tarun Jain 1:03:58
Log the operations.
To the function that is insert data. So it will print log running database operation insert data then function which will get inside inside data. Now this will be executed. As of now it's not entering here. It will look DB operation. It will check the statements as soon as you do return.
Then the flow will come to the insert data. So if I open the drawing board, so the first step is decorator.
Input is function. Now it will execute.
What is inside the decorator?
Then it will return the same function. Now whatever logic you have inside this function.
Whatever logic you have inside this function, then this will be executed which is inserting data into DB. Since I'm not printing this function, this line needs to not be

printed data inserted successfully. This should not be printed, but again we are using collab.
If I directly run this, it will also execute data inserted successfully.
You understood the flow.
So this line should not be printed if you run on VS code, but since we are running on collab, this line is printed.

Hardip Patel 1:05:25
Yes.

Tarun Jain 1:05:32
So same you can do it.

Mitesh Rathod 1:05:33
Because of we are, uh, for one question, we are returning from the like a decorator, so that's why it should not be printed right?

Tarun Jain 1:05:43
No. So if you remember this example, if I do def greet name.
I'll tell print welcome.

Mitesh Rathod 1:05:53
Yeah, print and then we are returning.

Tarun Jain 1:05:56
And then I'm telling return name. OK, so basically if I do like this greet and I if I tell Tarun, So what should be the output of this?

Mitesh Rathod 1:06:08
So welcome and then turn just welcome, OK.

Tarun Jain 1:06:08
It should be just welcome. No, it should be just welcome, right? Because this is returning it, right? I print it. It should be just welcome if I do name equals to create.

Mitesh Rathod 1:06:17
OK, OK, OK, got it. It's a function output.
Yeah.

Tarun Jain 1:06:25
Now if you see just welcome, then you are doing name.

Mitesh Rathod 1:06:30
Yeah, yeah, makes sense. Got it.

Tarun Jain 1:06:31
So basically in collab right, if you directly call greet it will print both, but if you run the same function in VS code you will only get welcome.
Same thing is happening here. I return data inserted successfully, but I'm not printing it, I'm just written it here. So basically this should not be printed at all. If we need to print this, we have to print this function.

Mitesh Rathod 1:06:43
Yes, got it.
OK, got it.

Tarun Jain 1:06:58
Same goes for delete operation. If I do delete data, what will be the output?

Hardip Patel 1:07:07
Same log running database operation, delete data, delete data.

Tarun Jain 1:07:12
What will be this? It will be deleted.
And then it will be deleting data from DB and then data deleted.

Margi Varmora 1:07:16
now
Meeting my show.

Tarun Jain 1:07:23
Is this clear?
Or we'll let's take one more example so that it will make more sense.

Mitesh Rathod 1:07:30
Yes.

Tarun Jain 1:07:38
We'll take this last example, then probably we'll wind up.
And here what I'll do is let's just type. I want to add enter pin, so enter pin. I want my enter pin to be decorator. So how will I define my first line?
So I'll tell you what we are trying to do. So I want to do the same bank statement, a bank account statement I previously did. So for the bank account, previously we deposited the money, deposit, then withdraw.

Hardip Patel 1:07:57
OK.

Margi Varmora 1:07:58
Um.

Hardip Patel 1:08:00
Hmm.

Tarun Jain 1:08:13
I want to perform the same task which is deposit withdraw and I also want to print the statement which is the current balance.
Or you can just say view statement.
So I want to perform this three functions. For this three functions I need to have a. Validator pin, Validator pin, which is a decorator.
Only if you enter a valid pin, then only you can execute this three function. So this logic of validating pin will be inside decorator. This logic of checking if the pin is right or wrong. I want to define a decorator for it and then.
I want to use the same decorator log or what you call the log statement for all

deposit withdraw and view statement that is pin entered successfully.
So this will be my log statement for all these three functions. So why do we need decorator?
It just to avoid.
DRY Does anyone know what is the full form of DRY?

Hardip Patel 1:09:33
Don't repeat yourself.

Mitesh Rathod 1:09:34
Do not repeat yourself.

Ajay Patel 1:09:34
Do not repeat yourself.

Tarun Jain 1:09:36
Correct. So this is just to don't repeat any code files again. So now you understood what we want to do. So can anyone tell me what will be the first line?

Hardip Patel 1:09:48
The we need to take the variable team.

Tarun Jain 1:09:54
So enter pin and then variable should be.
What variable should we take?

Hardip Patel 1:09:58
Uh, any, uh, like entered entered pin. Uh, pin code. Uh, not pin code. Yeah, pin values.

Mitesh Rathod 1:09:59
OK, Bing.

Tarun Jain 1:10:07

Can we check this code again?

So I want I want this center pin to be a decorator.

Ishan Chavda 1:10:11

We need to.

Hardip Patel 1:10:20

The the function.

Mitesh Rathod 1:10:21

OK, so function.

Tarun Jain 1:10:24

Correct. So what I will do is when I define deposit withdraw and view statement, for every single function I will add the decorator which will have some kind of log statement that I want to give. So your enter pin will have.

Mitesh Rathod 1:10:24

Function as argument.

Tarun Jain 1:10:39

Function.

Hardip Patel 1:10:43

Mhm.

Tarun Jain 1:10:43

OK, now what I need to do is I want to validate this, so I'll just write def validate. So do I need to define anything inside that? How did I define log?

Hardip Patel 1:10:50

Hmm.

Uh.

Tarun Jain 1:10:55
It's a simple function, right? It's a simple function.

Mitesh Rathod 1:10:58
Just a function here.

Hardip Patel 1:10:58
Yeah.

Tarun Jain 1:11:00
So I will just define a simple function. Let's suppose my pin is. I want to take user input. How will I do it?

Mitesh Rathod 1:11:09
Input.

Hardip Patel 1:11:10
Um.

Tarun Jain 1:11:12
PIN equals to input. Enter your PIN.

Mitesh Rathod 1:11:15
Enter your green.

Hardip Patel 1:11:18
Yeah.

Mitesh Rathod 1:11:19
And convert it into the integer, so just awesome.

Tarun Jain 1:11:21
Correct. They should be integer.
Now this is data type int.

Mitesh Rathod 1:11:29
Mm.

Tarun Jain 1:11:30
So let's suppose if my pin was 1234. If I just use input, this will be 1234. But now that we have added in, it will be a integer. Now what will be the next?

Mitesh Rathod 1:11:47
And if they will need to check.

Hardip Patel 1:11:47
Check.

Tarun Jain 1:11:48
OK, this is correct. So we just have to check, but let's write it.
I'll just do if pin double equals to 1234. What do I need to do?

Mitesh Rathod 1:11:55
Hmm.
Pin equal to 1234.
Print entered successfully.

Hardip Patel 1:12:05
Yes.
Mm.

Tarun Jain 1:12:08
Blend.
Entered.
Successfully. Then if we come back to this code, we wrote this program, then should we return?

Mitesh Rathod 1:12:18
Then read a dog.

Tarun Jain 1:12:23
The function. So now what we are trying to do is you entered the pin successfully. Now do whatever you want, right? Which is do you want to deposit? Do you want to withdraw? Do you want to view the statement? So here what I'll do, I'll just write return.

Mitesh Rathod 1:12:23
The on the function to the function.

Hardip Patel 1:12:23
Yes.

Mitesh Rathod 1:12:30
Mhm.

Hardip Patel 1:12:31
M.

Mitesh Rathod 1:12:34
Hmm.

Tarun Jain 1:12:38
That particular function. So whatever logic is there, let's suppose it was deposit which I added. Now you have your pin successful. You can do whatever operation you need, which is to add the money. Else what should I do?

Mitesh Rathod 1:12:53
Invalid ping.

Hardip Patel 1:12:54
Done.

Tarun Jain 1:12:55
It is invalid pin.

Hardip Patel 1:12:59
Yeah.

Tarun Jain 1:12:59
Invalid pin.
So this return is for which function in this particular logic.

Hardip Patel 1:13:06
Uh, intervene.

Mitesh Rathod 1:13:06
The which we are calling.

Tarun Jain 1:13:10
Which is for validate. So this line return is for validate. So what should I do for enter pin?

Hardip Patel 1:13:14
Mhm.

Mitesh Rathod 1:13:15
Mhm.
Again, you need to return.

Tarun Jain 1:13:22
What should I return?

Mitesh Rathod 1:13:26
Uh, the validate function, validate function.

Tarun Jain 1:13:26
What should I read here?

Hardip Patel 1:13:27
the function name validate.

Tarun Jain 1:13:30
It should be written.

Mitesh Rathod 1:13:32
Validate.

Tarun Jain 1:13:33
Return this function which is inside it validated. Everyone understood now the flow of decorator. So we start with the decorator or what?

Mitesh Rathod 1:13:40
Oh, one more Indian district.

Hardip Patel 1:13:41
Yes.

Mitesh Rathod 1:13:45
Uh, in the last return validate there is a indentation like uh two backspace. Alignment.

Tarun Jain 1:13:56
Wait, let's give 4 everywhere.

Mitesh Rathod 1:13:58
Yeah.

Hardip Patel 1:14:01
OK.

Tarun Jain 1:14:09
Oh.

This return is for this one, then else correct. OK, so everyone understood the flow of decorator. Now this is my decorator which is enter pin. Next time when I define anything it will be at enter pin right? Now inside enter pin I need to define a logic which is.

Mitesh Rathod 1:14:24
F.

Hardip Patel 1:14:25
And.
M.

Tarun Jain 1:14:39
Should I validate or it can be any logic right? Based on what you want to log. So here I'm checking the validation which is to check the pin. If the pin matches, I'm returning a log message and then that particular function. So that function is nothing but wherever you're defining the center pin, whether it is the deposit.

Hardip Patel 1:14:46
And.

Tarun Jain 1:15:04
Or view statement.
Or withdraw.
So now if I use this enter pin for deposit if I want to run.
I just want to run a statement. So what is the current function?
So let's suppose for deposit what log message do you want to give or a common message that can be used for all the three, but you have to use a function name.

Hardip Patel 1:15:37
Yeah, I know.

Tarun Jain 1:15:40
So how will you define a function name in this line? What is the logic?

Ishan Chavda 1:15:48
And this.

Mitesh Rathod 1:15:48
Fun FULC.

Tarun Jain 1:15:49
So what we're trying to do is we have inter.

Mitesh Rathod 1:15:53
Mhm.

Tarun Jain 1:15:54
FUNC them.

Mitesh Rathod 1:15:56
Uh dot under score, under score, team under score, under score.

Tarun Jain 1:16:02
So.

Mitesh Rathod 1:16:03
But in duplicate curly places here.

Tarun Jain 1:16:05
I was just executing.
This particular line and dot dot dot back slash N and back slash N.
OK so now here I'm trying to use deposit. How will I use deposit? Now this function do we need to take any input variable?
We need to take amount. We need to take amount. But here what I'll do is just write some your statement.

Mitesh Rathod 1:16:27
Yes.

Hardip Patel 1:16:28
No.
OK.

Tarun Jain 1:16:36
Uh, depositing.
The cash then return.
Deposited.
Now what will be the output of this?

Mitesh Rathod 1:16:55
First, this will ask for the input pin number.

Tarun Jain 1:16:55
First, tell me the flow. First, what will be the flow? First, as soon as I run this, what will it ask?

Mitesh Rathod 1:17:03
It will ask for the input uh pin number.

Tarun Jain 1:17:03
It will ask me to enter the I'll add 1-2.

Hardip Patel 1:17:05
And do you have been?

Tarun Jain 1:17:09
Invalid pin. If I pin 1234, what will it do?

Mitesh Rathod 1:17:10
I did.
Enter print successfully and executing the function that is deposit.

Hardip Patel 1:17:17
Thanks.

Tarun Jain 1:17:17
Deposit then these two lines total 4 lines should be.

Mitesh Rathod 1:17:24
Correct.

Tarun Jain 1:17:28
123 and four. Same goes for view statement.
Your bank statement.
Report.

Mitesh Rathod 1:17:41
Please.

Tarun Jain 1:17:44
And then I'll return. No need to return anything in new statement usually will not return.

Mitesh Rathod 1:17:47
Yeah.

Tarun Jain 1:17:50
Now if I run this again, it will ask me a PIN.
1234.
And then that logic same goes for withdraw.

Mitesh Rathod 1:18:16
True.

Tarun Jain 1:18:21

Rise detect.

Is this clear how decorator works?

Hardip Patel 1:18:32

I guess.

Tarun Jain 1:18:33

So the only thing what you'll have to notice is this part.

This is the only part you have to think. The other thing is same. You will start with a decorator name, you will give a function, you will return whatever you're defining inside which is validate. Now how you define a validate will define how your other functions will use it.

Hardip Patel 1:18:46

M.

Mhm.

Tarun Jain 1:18:54

Right, so the only thing is this logic has to be such solid which is using multiple functions, right? Since you want to avoid repeating this code, that that is the reason why you're defining A decorator. If not, there is no need of decorators.

Mitesh Rathod 1:18:54

Mm.

Tarun Jain 1:19:11

OK, decorator only comes when you have three to four functions in your code, and all the three to four functions is dependent on one function, right? And you're reusing that code. In order to reusing of that code, you're defining this decorator. Which is property one of the example.

Mitesh Rathod 1:19:27

Yes.

And.

OK.

Tarun Jain 1:19:38
So we have few more decorators. If you see here I've added object methods, abstract methods, class methods. Here you have few inbuilt decorators that we will use.

Mitesh Rathod 1:19:45
OK.

Tarun Jain 1:19:52
And later on, when we talk about evaluations right in drag evals, we'll be using OPIC so that you can track your LLM cost latency.
And eval result. Then other parameters are there. If you want to track this, you have a simple decorator called track. So when I say track, all the logic to track your LLM calls is returning some function.
Right. And we are just reusing that track.
And then you'll have your own LLM call. You have LLM response. Here you are using open AI, let's suppose open AI LLM. So every single time you make any LLM call, you have this track function which will track your LLM cost, latency and other important feedbacks.
Feedbacks. So what do you mean by this tag? The logic is written and you are just reusing it.

Mitesh Rathod 1:20:49
OK again.

Tarun Jain 1:20:49
You understood the use of decorators here when we use inbuilt decorator and device that that is when you'll able to relate. OK, this topic was already covered. If not very rare you'll see decorators in most of the open source frameworks.

Hardip Patel 1:20:53
Yes.

Mitesh Rathod 1:20:53
Yeah.

Tarun Jain 1:21:06
If you are creating as a contributor then you will see it. But in terms of actual implementation of your own code then probably you will not notice it much. Data class also if you notice data class starts with data class which is a decorator.

Mitesh Rathod 1:21:12
Mm.

Tarun Jain 1:21:23
You saw this in Agno code as well.

Hardip Patel 1:21:25
M.

Tarun Jain 1:21:28
I have no.
Models Azure.
So this data class is also undecorated.

Hardip Patel 1:21:38
So.

Mitesh Rathod 1:21:42
OK.

Tarun Jain 1:21:44
So in most of the cases you will not find it inside a function, but it's good to know.

Mitesh Rathod 1:21:45
Hello.
Yeah, one question here.

Tarun Jain 1:21:51

Yeah, any.
Yeah.

Mitesh Rathod 1:21:54
Sorry for a decorator, what if like we want to like pass any argument to the function like into the deposit? Let's say I I want to like pass any number. So how it will consider into the decorator validate under the validate?

Tarun Jain 1:22:03
Yeah.
OK.
So suppose I have global account.
Well, so decorator will not have this variable, right? Decorator's only job is what are the common statements or common logic you need across all those, right? So all these functions that you have right deposit will have their own independent variable. This variable has nothing to do with decorator, so I'm defining global.

Mitesh Rathod 1:22:24
Mhm.
Yeah.
OK.

Tarun Jain 1:22:35
So balance plus equals to 50.
So suppose I'm defining balance equals to 100.

Hardip Patel 1:22:46
So.

Tarun Jain 1:22:48
And then I want to return.
Deposited.

Hardip Patel 1:22:55
Yes.

Tarun Jain 1:22:56
New amount.
S balance.

Mitesh Rathod 1:23:01
Balance.

Tarun Jain 1:23:07
1234.
Deposit is missing on what is cash? OK, the cash should be valid.

Hardip Patel 1:23:12
So.

Mitesh Rathod 1:23:15
Yeah, because.

Tarun Jain 1:23:18
The cash is balanced.
OK, let's do one thing. I'll remove this balance. I'll add cash plus equals to 50.

Mitesh Rathod 1:23:27
Yeah.

Hardip Patel 1:23:28
Thank you.

Mitesh Rathod 1:23:37
No, it will call from the like and from the validator decorator because on the decorator we are dating the fun, right? That fun is the required parameter.

Hardip Patel 1:23:40
Rahul.

Yeah.

So.

Tarun Jain 1:23:51
Oh, yeah, one second. Yeah.

Hardip Patel 1:23:52
We can't. We need to give ours and fewer documents in this in the function.

Tarun Jain 1:24:00
Cesar.

No, it's not working.

Hardip Patel 1:24:09
No, I I think it is a different.

In the bottom.

Mitesh Rathod 1:24:19
Right in the in the in the validate as well.

Tarun Jain 1:24:23
Yeah, validate I've added.

Hardip Patel 1:24:24
M.

Mitesh Rathod 1:24:26
No, uh, on the return, on the return.

Tarun Jain 1:24:30
No return will be approved.

Hardip Patel 1:24:31
No, I think this is enough. Yeah, this is enough. They should work papers.

Mitesh Rathod 1:24:35
OK, OK, let's try.

Tarun Jain 1:24:36
Enter pin deposit cash. Cash is a new variable. Then I'm just doing cash equals to 500.

Mitesh Rathod 1:24:41
Mhm.

Hardip Patel 1:24:42
Mhm.

Tarun Jain 1:24:45
This was 5000.

Hardip Patel 1:24:50
I'm shifting.

Tarun Jain 1:24:52
1234.

Mitesh Rathod 1:24:54
Yeah, there you go.

Tarun Jain 1:24:56
One second. Why is it 500?

Mitesh Rathod 1:25:00
We are just using a balance as a global variable.

Hardip Patel 1:25:00
The cash.

Tarun Jain 1:25:03
OK. OK. This should be cash.

Mitesh Rathod 1:25:06
So.

Tarun Jain 1:25:08
123-45500 so it should be ours.

Hardip Patel 1:25:14
Yeah.

Mitesh Rathod 1:25:14
Uh, OK, like in the Uh for the PHI get we are using a split dot dot.

Tarun Jain 1:25:21
For which one?

Mitesh Rathod 1:25:21
And the JavaScript as well in JavaScript or PHP and PHP we are using a spread operator dot, dot, dot, yes, yes.

Tarun Jain 1:25:26
Yeah, I've seen this thing in React. We have this three dots.

Mitesh Rathod 1:25:31
Yes, yes.

Tarun Jain 1:25:36
Yeah, we can add arcs.

Mitesh Rathod 1:25:38
Yes, you got it.

Tarun Jain 1:25:43
Oh, any other questions?

Hardip Patel 1:25:47
Do you know if we can add the similar argument to enter the team? I mean like what if instead of asking the team if you want to add let's say PIT or something directly? Is that possible?

Tarun Jain 1:26:07
So for which one?

Hardip Patel 1:26:09
And so like I didn't and and under under the bracket just like we have in deposit and we have a value we can enter our own names, yeah.

Tarun Jain 1:26:18
This select.
No enter pin. You can't give any brackets. It should be just this thing.

Hardip Patel 1:26:25
OK, OK.
I'm asking because there is annotation in PHP which does the seminar and we can pass the.

Tarun Jain 1:26:30
So and this is the decorator.
No, probably in abstract classes you can add arguments, but usually for decorators since the major logic is.

Hardip Patel 1:26:42
OK.

Mitesh Rathod 1:26:42
App.

Hardip Patel 1:26:46
And they might not be nice.

Tarun Jain 1:26:48
The outside function is there it there you'll not have any of parameter there you'll have function.

Hardip Patel 1:26:52
Right.
Yeah, you did good. Thank you.

Mitesh Rathod 1:26:56
Like what if like if we provide a any like function after a func, we provide any number or something in the enter pane?
Can you can we give it something? Any anything else? Yeah, no. And then here.

Tarun Jain 1:27:13
I don't think this will work.

Mitesh Rathod 1:27:13
That's it.

Hardip Patel 1:27:16
Well, I mean that.

Tarun Jain 1:27:17
Oh, this should be numbered. No, I'm giving 500.

Mitesh Rathod 1:27:22
Huh.

Tarun Jain 1:27:23
So let's do one thing here function. I'll give num.

That means here I'm giving 500 that 500M I'm giving it to deposit. OK, so basically I don't need cash here then my deposit will be empty.

Mitesh Rathod 1:27:29
You did also, yeah.
Mhm.
No, you you you need to require them.

Tarun Jain 1:27:43
No, why will I need? If you see here 500 we give OK cash is for this one, the original cash and 500 is the deposit one. But where is this been used in the code?

Hardip Patel 1:27:45
Yeah.
Mm.

Mitesh Rathod 1:27:48
Mhm.
Yes, correct.
In in in use in the like a validate like in the return. After the enter pin successfully there is a return func name num.

Tarun Jain 1:28:00
M.
OK, this number is 500 now.

Mitesh Rathod 1:28:10
Correct. So the funk is the the deposit.

Tarun Jain 1:28:13
It's 500.

Hardip Patel 1:28:15
OK, I think ideally the num the num will be just instead of num it should be adds but

the thing should be replaced by num. So like if num equals equals to 1234.
So enter pin 1234 will will not require console input. I guess if it is that it is for us.

Mitesh Rathod 1:28:38
Uh huh.

Tarun Jain 1:28:38
All right, 1234. Now we are I have instead of pin, I'll make it pin.

Mitesh Rathod 1:28:43
Mhm.

Hardip Patel 1:28:43
Oh.
Yeah, we don't need.

Mitesh Rathod 1:28:48
Calendar validate not required adds.

Tarun Jain 1:28:53
This correct PIN will come here.

Mitesh Rathod 1:28:53
It's fine to be.

Tarun Jain 1:28:56
OK, now it makes sense. Here I'll just add Rs.

Hardip Patel 1:28:59
Mhm.

Mitesh Rathod 1:29:01
Hmm.

Tarun Jain 1:29:03
Yeah, it's an error. Enter pin missing one.

Hardip Patel 1:29:06
Mhm.
Oh, uh, it might be correct. Uh, keyword argument, yeah.

Mitesh Rathod 1:29:09
Correct.
The validate.
No, the function is also required under the. Can we access current pin under the?

Tarun Jain 1:29:20
Mm.
No. How can it do you see dates from this?

Hardip Patel 1:29:25
No, we have a.

Tarun Jain 1:29:29
No, this won't work actually. I don't think we can view.

Mitesh Rathod 1:29:31
OK, OK.

Hardip Patel 1:29:33
So.

Tarun Jain 1:29:35
So wait, what is it?

Mitesh Rathod 1:29:35
Just like for experiment, for experiment only.

Tarun Jain 1:29:38
It says deposit but deposit needs we can try it for view statement.

Mitesh Rathod 1:29:45
The argument.

Tarun Jain 1:29:48
Here I'll give 1234 and then I'll give new statement.

Mitesh Rathod 1:29:48
Mhm.

Tarun Jain 1:29:55
Interview for.

Hardip Patel 1:29:56
OK, so like I said it, you don't need to. We cannot give arguments directly, but we can use function tool from tool to.

Tarun Jain 1:30:09
Oh yeah, that is operate tool only you have import from tools.

Hardip Patel 1:30:13
Yeah, yeah, that that gives the possibility to wrap the function with argument, but directly.

Tarun Jain 1:30:19
Yeah, this.
Function tools. Yeah, here we have something called us.

Hardip Patel 1:30:25
Maps.

Tarun Jain 1:30:28

So usually when we will use function to list, we'll only use it for LRU cache. We'll only use it for this. Apart from that, we'll not use function crawling to anyone. This is LRU cache is used for multiple cases as well. Even these folks, OK, I guess Langston is using it.

Mitesh Rathod 1:30:37
OK.

Hardip Patel 1:30:37
Oh.

Mitesh Rathod 1:30:39
A little kiss, OK.

Hardip Patel 1:30:40
OK.
Um.
Sure.

Tarun Jain 1:31:07
Yeah, the phone tools cased.
I remember these folks are heavily dependent on function rules.

Hardip Patel 1:31:16
Mhm.

Tarun Jain 1:31:19
Yeah, mainly it is for Kesh. That is what I know. But apart from Kesh, I've not seen on tools anywhere else.

Mitesh Rathod 1:31:30
OK.

Tarun Jain 1:31:34
Yeah, but did it do 1234?

Mitesh Rathod 1:31:34
Yeah, it's fine. Yeah.
Hmm.

Tarun Jain 1:31:37
So if I remove this thing arguments.
And if I run this for view statement it will work.

Mitesh Rathod 1:31:45
It will work.

Tarun Jain 1:31:48
Why is it telling now?

Mitesh Rathod 1:31:48
The.
Uh, you did a remove from the funk answer.

Tarun Jain 1:31:55
Did I run this?

Mitesh Rathod 1:31:58
You remote.

Tarun Jain 1:32:04
Earlier it printed something. Now it's not printing.
OK, so.
Oh, it's getting inside an event loop. It asked me to enter piper two times.

Mitesh Rathod 1:32:30
Mhm.

Tarun Jain 1:32:33

So I entered 1234.

Just give what?

Hardip Patel 1:32:38

Thanks.

OK.

Tarun Jain 1:32:44

OK, this was weird. Earlier it showed me the output. I didn't see anything.

Mitesh Rathod 1:32:45

What is that?

Tarun Jain 1:32:51

But I don't think so. We can do this.

Mitesh Rathod 1:32:54

OK, that's that's fine, yeah.

So.

Tarun Jain 1:33:00

Yeah, it is.

Mitesh Rathod 1:33:02

At the end you need to just remove the correct link from the.

Hardip Patel 1:33:19

123.

Mitesh Rathod 1:33:27

What?

Hardip Patel 1:33:30

OK.

Tarun Jain 1:33:39
Yeah.

Hardip Patel 1:33:40
OK.

Tarun Jain 1:33:40
So now the pending thing is typing extension we have already seen, but this we can cover very quickly because you know this right after append you might have seen this keyword.

Mitesh Rathod 1:33:40
Yeah.

Tarun Jain 1:33:55
In append in sort that this returns none. This thing whatever you have here is what you call typing intention then pydantic. OK, I guess till class methods we can complete on Monday.

Mitesh Rathod 1:33:55
The written statement.
Hello.

Tarun Jain 1:34:12
Then you can start with the the NLP.
So next week we'll have in person calls itself.

Hardip Patel 1:34:22
I think.

Tarun Jain 1:34:23
So I don't think we also have calendar invite for next week.

Ajay Patel 1:34:24

Thanks.

Yeah. So when are you coming to Vadodara?

Tarun Jain 1:34:29

OK, I'm coming tomorrow only.

Ajay Patel 1:34:33

Tomorrow. OK. OK.

Tarun Jain 1:34:36

Tomorrow at 8:25 night.

Ajay Patel 1:34:39

3/25.

See you. See you then.

Tarun Jain 1:34:46

Yeah, yeah.

Ajay Patel 1:34:48

Yeah.

Mitesh Rathod 1:34:49

OK. Thanks, Taran. Bye. Bye. Thank you. Yeah. Bye, bye. Thank you.

Ajay Patel 1:34:49

Bye, bye guys.

Hardip Patel 1:34:50

See you.

Tarun Jain 1:34:50

Thank you. Yeah.

Hardip Patel 1:34:52
Thank you, ma'am.

Ronak Makwana 1:34:55
Happy Independence, Akhil.

Ishan Chavda stopped transcription