

In [6]: `print(type(d))`

<class '__main__.Dog'>

Quiz - 1

In []: `class Foo:`
 `def __init__(self, name):`
 `self.name = name`

→ a = Foo("a")
→ b = Foo("b")
→ a.name = b.name
→ b.name = "c"
→ a.x = 2
→ b.x = 1
→ x = (a.x + b.x) * (a.name + b.name)

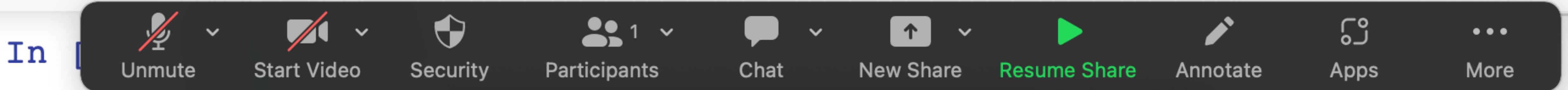
(2 + 1) * ('b' + 'c')
3 * 'bc'

In []:

b c b c b c

Dunder Methods

In []:



In [16]: `c1 = Car("Nexon", 12)
c2 = Car("Altroz", 15)`

In [17]: ~~print(c1) # can I override the print behavior?~~

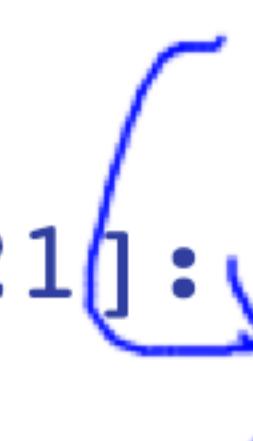

`<__main__.Car object at 0x7fe721d83c40>`

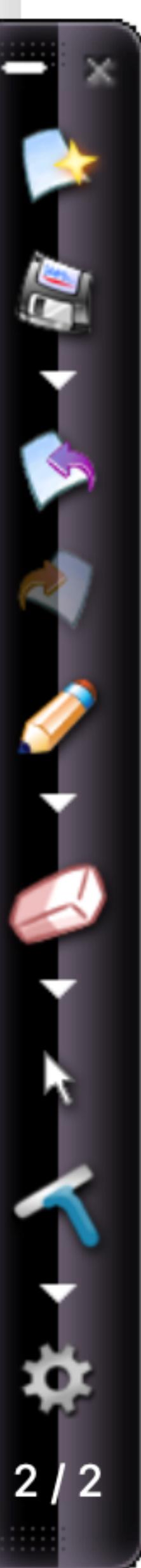
In [19]: `class Car:
 def __init__(self, name, mileage):
 self.name = name
 self.mileage = mileage

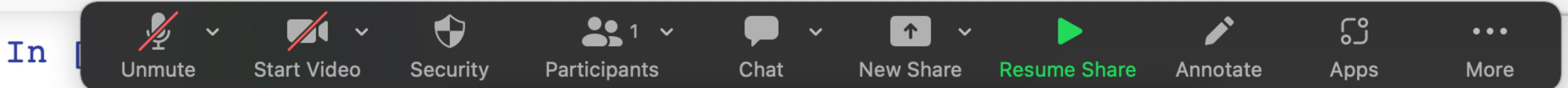
 # __str__ magic method to override print behavior
 def __str__(self):
 return f"{self.name} -> {self.mileage}"`

In [20]: `c1 = Car("Nexon", 12)
c2 = Car("Altroz", 15)`

In [21]: ~~print(c1)~~


`Nexon -> 12`





```
In [16]: c1 = Car("Nexon", 12)  
c2 = Car("Altroz", 15)
```

```
In [17]: print(c1) # can I override the print behavior?
```

```
<__main__.Car object at 0x7fe721d83c40>
```

```
In [19]: class Car:  
    def __init__(self, name, mileage):  
        self.name = name  
        self.mileage = mileage  
  
        # __str__ magic method to override print behavior  
    def __str__(self):  
        return f"{self.name} -> {self.mileage}"
```

```
In [20]: c1 = Car("Nexon", 12)  
c2 = Car("Altroz", 15)
```

```
In [21]: print(c1)
```

Nexon -> 12





Unmute



Start Video



Security



Participants



Chat



New Share



Resume Share



Annotate



Apps



More

In [23]: `print(c2)`

Your screen sharing is paused

40:55

Stop Share

Altroz has the mileage: 15

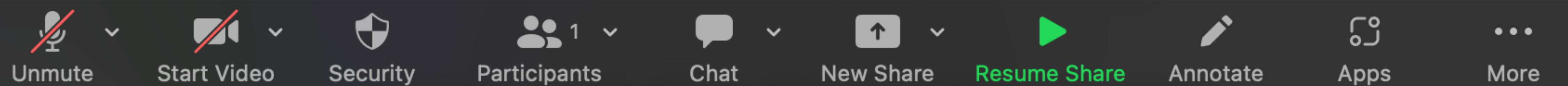
In []:

```
In [ ]: class ComplexNumber:  
        def __init__(self, real, imaginary):  
            self.real = real  
            self.imaginary = imaginary
```

real imaginary
] ↓
at + i 0
 ↓

In []:





sel Your screen sharing is paused 47:02 Stop Share
self.imaginary = imaginary

```
# overrides the print behavior of printing object of class ComplexNumber
def __str__(self):
    return f"{self.real} + i{self.imaginary}"

def __add__(obj1, obj2):
    return f"{obj1.real + obj2.real}"
```

In [50]: c1 = ComplexNumber(5, 3)

In [51]: print(c1)

5 + i3

In [52]: c2 = ComplexNumber(3, 6)
print(c2)

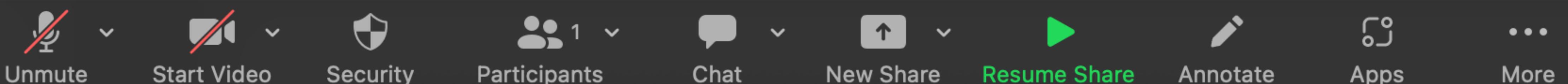
3 + i6

In [54]: c1 + c2 # same syntax of addition but different behavior

Out[54]: '8'

In []:





Your screen sharing is paused 01:05:57 Stop Share

In [83]: `c.speak()`

Meoooewww Meoooow!!!!

In [84]: `c.eat()`

I am eating!!!

In [86]: `d = Dog("Scooby Doo")`

In [87]: `print(d.name)`

Scooby Doo

In [88]: `d.speak()`

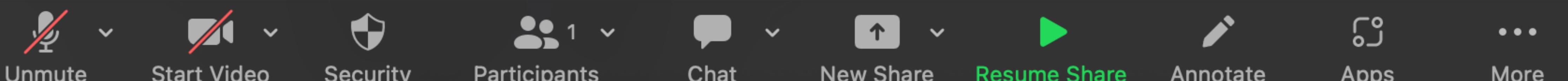
Booooow Bhhoowwww!!!!

In [90]: `d.eat()`

I am eating!!!

Common
functions





Your screen sharing is paused
01:12:11 Stop Share

```
def speak(self):
    print("Booooow Bhhoowww!!!!")
```

In [100]: c = Cat("Tom")
print(c.name)
print(c.species)

Tom

Cat

Best way??

In [101]: c.speak()

Meoooewww Meoooow!!!!

In [102]: c.eat()

I am eating!!!

In [104]: d = Dog("Scooby Doo")

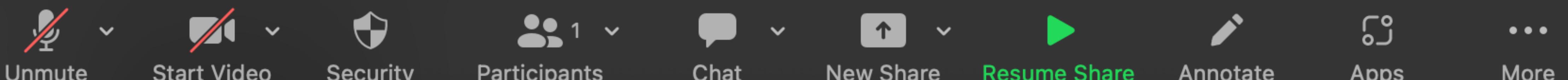
In [106]: print(d.name)
print(d.species)

Scooby Doo

Dog

In [88]: d.speak()





Your screen sharing is paused 01:28:42 Stop Share

In [96]:

```
class Animal:
    def __init__(self, species):
        self.species = species # how to initialise this
        # parent class property in the child class object

    def eat(self):
        print("I am eating!!!")
```

In [97]:

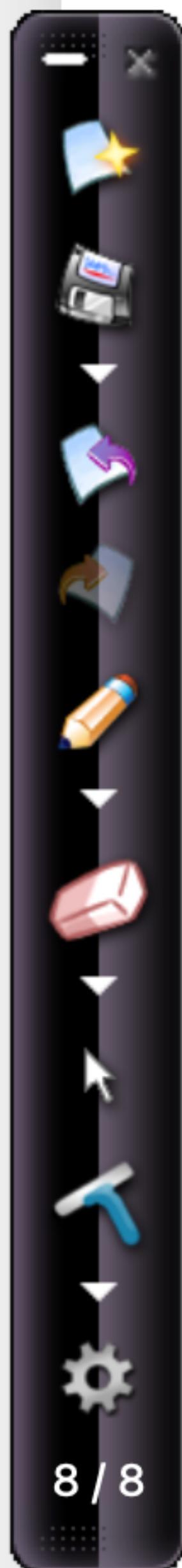
```
class Cat(Animal): # syntax to inherit
    def __init__(self, name):
        # to set species data, we need to call Animal initialiser
        Animal.__init__(self, "Cat")
        self.name = name

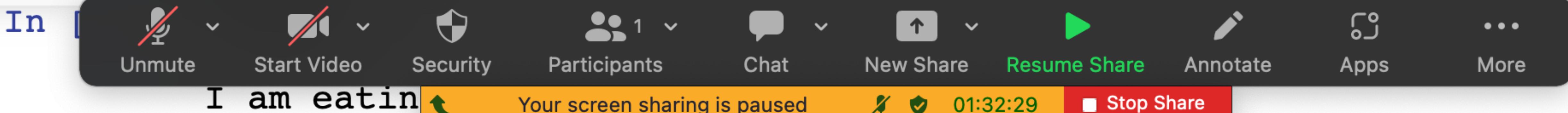
    def speak(self):
        print("Meoooewww Meoooow!!!!")
```

In [98]:

```
class Dog(Animal):
    def __init__(self, name):
        Animal.__init__(self, "Dog")
        self.name = name

    def speak(self):
        print("Boooow Bhhoowwww!!!!")
```





super

In [112]: ~~class Cat(Animal): # syntax to inherit~~
~~def __init__(self, name):~~
~~# to set specifies data, we need to call Animal initialiser~~
~~#~~
 ~~Animal.__init__(self, "Cat")~~
 ~~super().__init__("Cat")~~
 ~~self.name = name~~

~~def speak(self):~~
 ~~print("Meoooewww Meoooow!!!!")~~

In [113]: c = Cat("Tom")

--
AttributeError
t)

Input In [113], in <cell line: 1>()
----> 1 c = Cat("Tom")

Parent
init

Traceback (most recent call last)

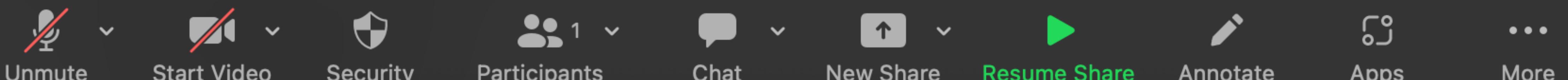
Super()
--init--

Input In [112], in Cat.__init__(self, name)

2 def __init__(self, name):

3 # to set specifies data, we need to call Animal initialis
er





Your screen sharing is paused 01:39:01 Stop Share

super

```
In [124]: class Animal:
    def __init__(self, species):
        self.species = species
        print('self inside Animal class:')
        print(self)

    def eat(self):
        print("I am eating!!!")
```

```
In [125]: class Cat(Animal): # syntax to inherit
    def __init__(self, name):
        print('self inside Cat class:')
        super().__init__("Cat") # the object is passed automatically
        self.name = name

    def speak(self):
        print("Meoooewww Meoooow!!!!")
```

```
In [126]: c = Cat("Tom")
print(c.name)

self inside Cat class:
<__main__.Cat object at 0x7fe721ed5ee0>
self inside Animal class:
```



In [168]:

ca.get_int()

Your screen sharing is paused

02:01:02

Stop Share

Interest is: 2250.0

Out[168]: 2250.0

Quiz

```
In [ ]: class Person:  
    def __init__(self, name):  
        self.name = name  
  
    def say_hello(self):  
        print(f'Hi my name is {self.name}')  
  
class Employee(Person):  
    def __init__(self, name, salary):  
        super().__init__(name)  
        self.salary = 0  
  
    def say_hello(self):  
        super().say_hello()  
        print(f'My salary is {self.salary}')  
  
e = Employee("Sahil", 10000)  
e.say_hello()
```

Unmute

Start Video

Security

Participants

Chat

New Share

Resume Share

Annotate

Apps

More

 Your screen sharing is paused 02:01:21

```
e = Employee("Sahil", 10000)
e.say_hello()
```

Hi my name is Sahil
My salary is 0

In []:

```
class Person:
    def __init__(self, name):
        self.name = name

    def say_hello(self):
        print(f'Hi my name is {self.name}')


class Employee(Person):
    def __init__(self, name, salary):
        super().__init__(name)
        self.salary = salary

    def say_hello(self):
        super().say_hello()
        print(f'My salary is {self.salary}')


e = Employee("Sahil", 10000)
e.say_hello()
```

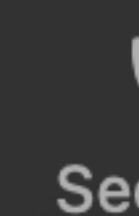
In []:

In []:

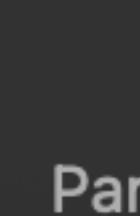


Unmute

Start Video



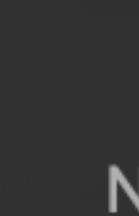
Security



Participants



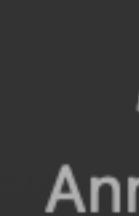
Chat



New Share



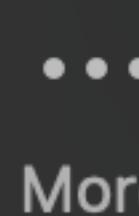
Resume Share



Annotate



Apps

In [172]: `print(ba)`

Your screen sharing is paused

02:08:05

Stop Share

Balance (Savings): 1321423473941274728413379367264378296423794623

In [173]: `class BankAccount:` `def __init__(self, balance):` `self.__balance = balance` `def deposit(self, amount):` `self.__balance += amount` `def withdraw(self, amount):` `self.__balance -= amount` `def __str__(self, account_type = 'Savings'):` `return f'Balance ({account_type}): {self.__balance}'`In [179]: `ba = BankAccount(10000)`In [180]: `print(ba)`

Balance (Savings): 10000

In [181]: `ba.__balance = 1032141847157358974289074238423740240`In [182]: `print(ba)`

Balance (Savings): 10000



Unmute

Start Video

Security

Participants

Chat

New Share

Resume Share

Annotate

Apps

More

In [184]: `print(ba2)`

Your screen sharing is paused

02:09:02

Stop Share

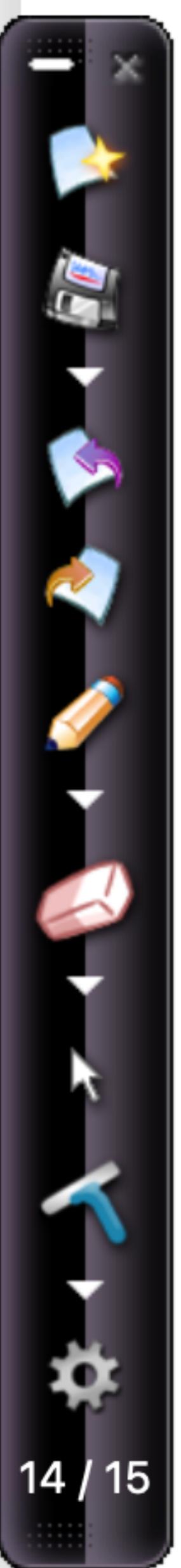
Balance (Savings): 1000

In [185]: `dir(ba2)`

Out[185]:

```
['__BankAccount__balance',
 '__class__',
 '__delattr__',
 '__dict__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__le__',
 '__lt__',
 '__module__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__setattr__',
 '__sizeof__',
 '__str__']
```

→ ClassName__varName



In [172]: `print(ba)`

Balance (Savings): 1321423473941274728413379367264378296423794623

In [173]: `class BankAccount:`

```
    def __init__(self, balance):
        self.__balance = balance
```

```
    def deposit(self, amount):
        self.__balance += amount
```

```
    def withdraw(self, amount):
        self.__balance -= amount
```

```
    def __str__(self, account_type = 'Savings'):
        return f'Balance ({account_type}): {self.__balance}'
```

- Bank Account __balance
inside the class

In [179]: `ba = BankAccount(10000)`

outside

In [180]: `print(ba)`

Balance (Savings): 10000

In [181]: `ba.__balance = 1032141847157358974289074238423740240`

???

In [182]: `print(ba)`

Balance (Savings): 10000



Unmute

Start Video

Security

Participants

Chat

New Share

Resume Share

Annotate

Apps

More

Your screen sharing is paused 02:11:45

```
def __str__(self, account_type = 'Savings'):
    return f'Balance ({account_type}): {self.__balance}'
```

In [179]: ba = BankAccount(10000)

In [180]: print(ba)

Balance (Savings): 10000

In [181]: ba.__balance = 1032141847157358974289074238423740240

In [182]: print(ba)

Balance (Savings): 10000

In [189]: print(dir(ba))

['__BankAccount__balance', '__balance__', '__class__', '__delattr__', '__dict__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattr__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__le__', '__lt__', '__module__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', '__weakref__', 'deposit', 'withdraw']

In [183]: ba2 = BankAccount(1000)



In [1/1]: ba = BankAccount()
ba.balance = 1321423473941274728413379367264378296423794623

In [172]: print(ba)

Balance (Savings): 1321423473941274728413379367264378296423794623

In [173]: class BankAccount:
 def __init__(self, balance):
 self.__balance = balance

 def deposit(self, amount):
 self.__balance += amount

 def withdraw(self, amount):
 self.__balance -= amount

 def __str__(self, account_type = 'Savings'):
 return f'Balance ({account_type}): {self.__balance}'

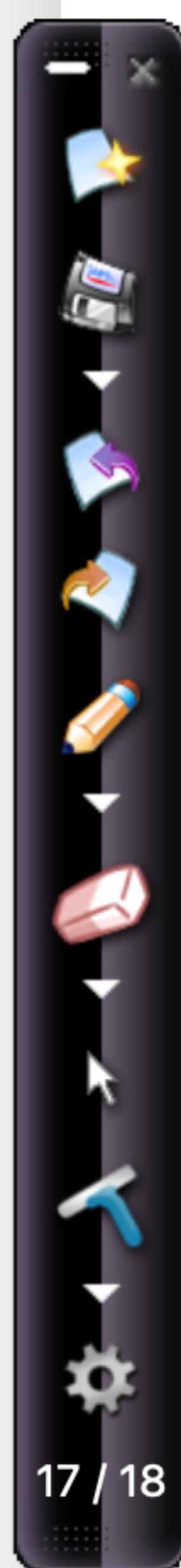
In [179]: ba = BankAccount(10000)

In [180]: print(ba)

Balance (Savings): 10000

New attr

In [181]: ba.__balance = 1032141847157358974289074238423740240



In [206]:

Unmute

Start Video

Security

Participants

Chat

New Share

Resume Share

Annotate

Apps

More

on 3 (ipykernel)

Your screen sharing is paused

02:23:52

Stop Share

In [207]: `print(ba2)`

Balance (Savings): 10328194724719241827412471

In [208]: `print(dir(ba2))``['__BankAccount__balance', '__balance', '__class__', '__delattr__', '__dict__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattr__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__le__', '__lt__', '__module__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', '__weakref__', 'deposit', 'withdraw']`In [209]: `ba2.deposit(243)`

— balance

In [210]: `print(ba2)`

Balance (Savings): 10328194724719241827412714

Doubts

In [190]: `# Next Problem Solving Session on OOPS`In [201]: `class Foo:`
 `def __init__(self):`

'__weakref__', Your screen sharing is paused 02:24:08 Stop Share
'deposit',
'withdraw']

In [187]: ba2.__BankAccount__balance = 10328194724719241827412471

In [188]: print(ba2)

Balance (Savings): 10328194724719241827412471

In [206]: ba2.__balance = 2342

In [207]: print(ba2)

Balance (Savings): 10328194724719241827412471

In [208]: print(dir(ba2))

['__BankAccount__balance', '__balance__', '__class__', '__delattr__', '__dict__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattr__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__le__', '__lt__', '__module__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', '__weakref__', 'deposit', 'withdraw']

In [209]: ba2.deposit(243)

In [210]: print(ba2)

