

**Whose Method Is It Anyway?**

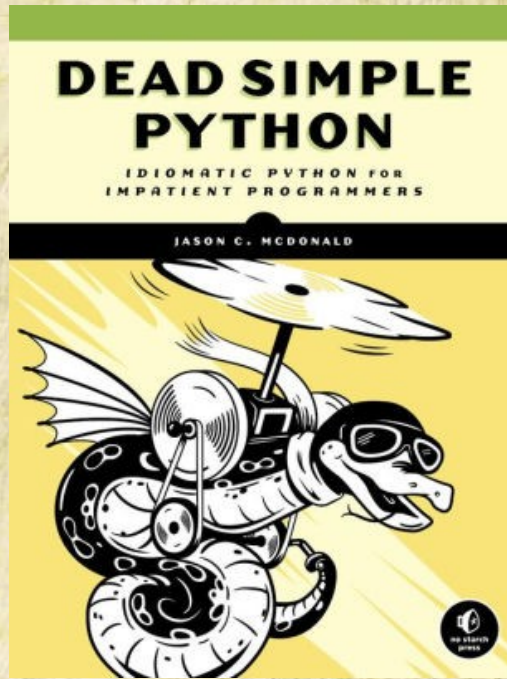
# Multiple Inheritance

Which method gets called?

# Multiple Inheritance

Which method gets called?





# *Dead Simple Python*

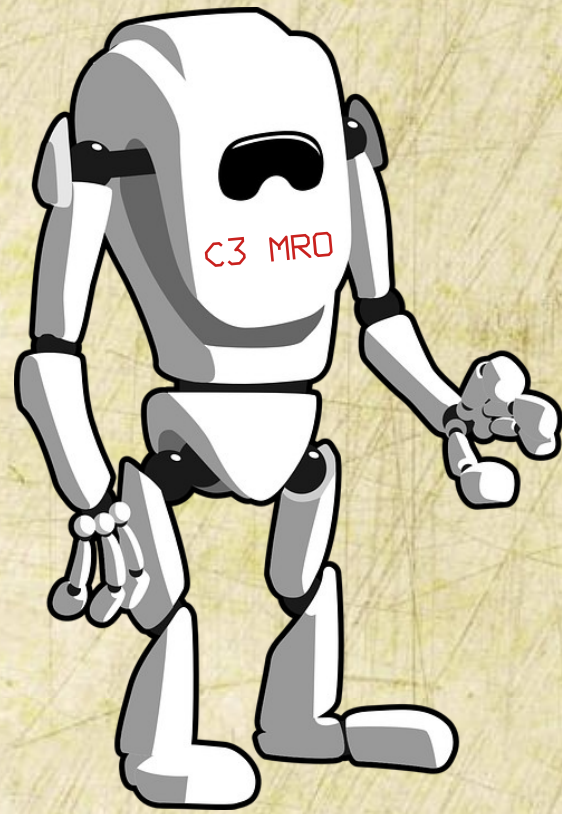
No Starch Press

2021

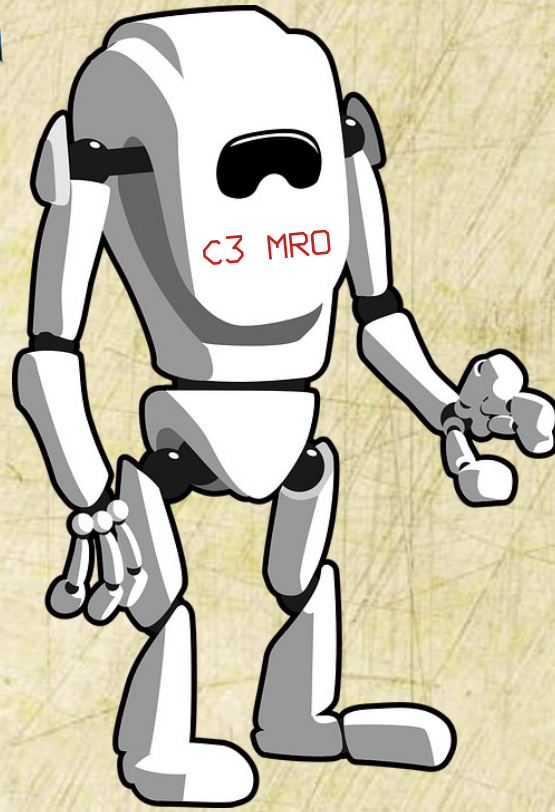
# C3 Method Resolution Order

C3 MRO



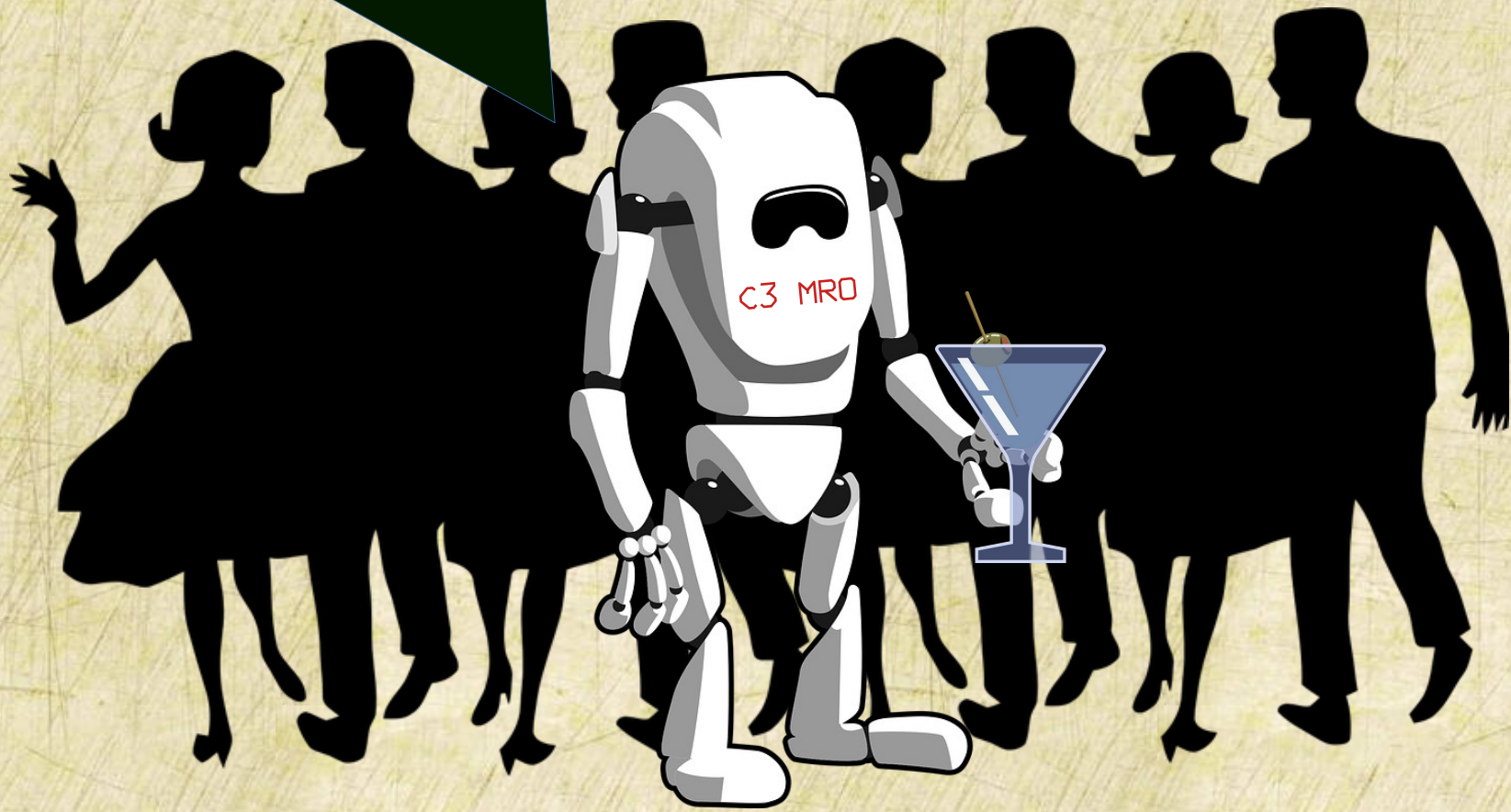


# Superclass Linearization



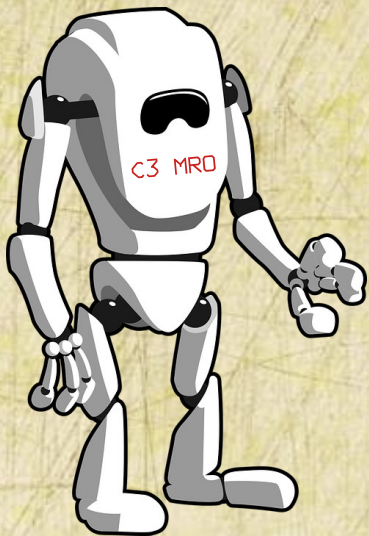


# Superclass Linearization

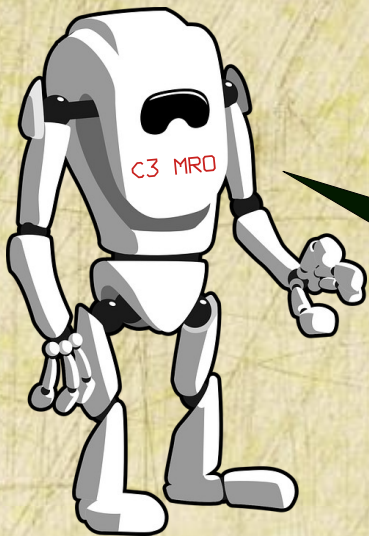




class ( ):

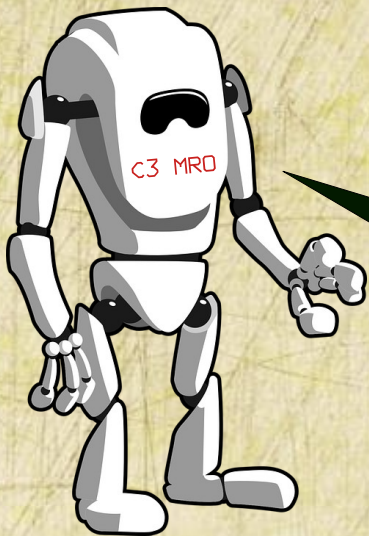


class ( ):

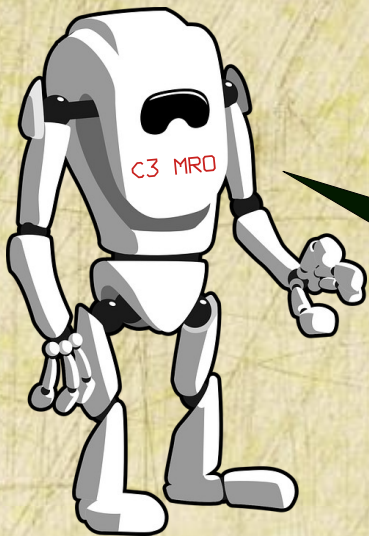




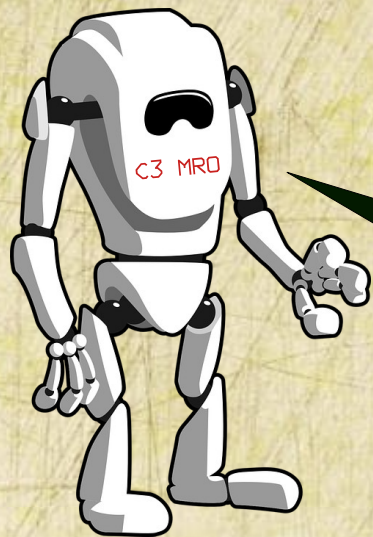
class ( ):



class ( ):

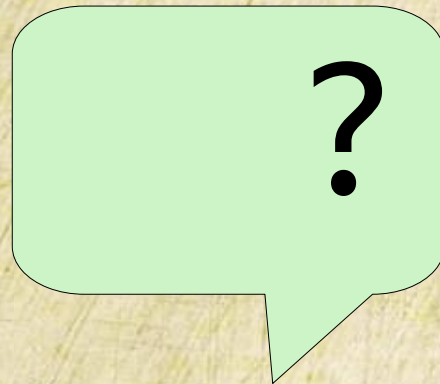




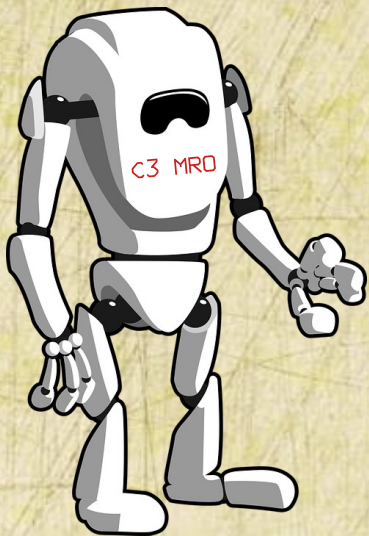


.

( )

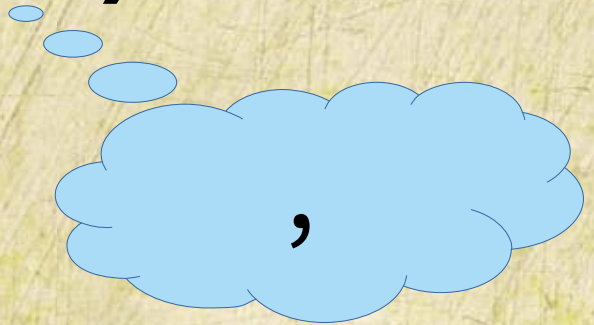
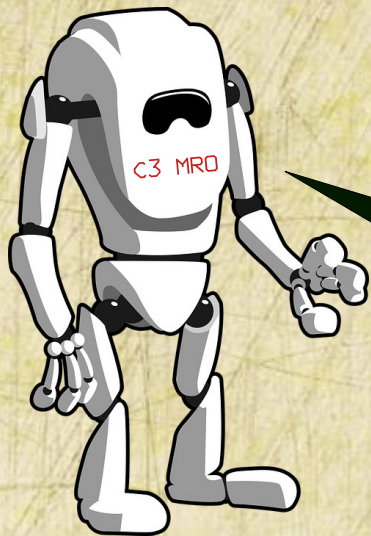


class ( ):

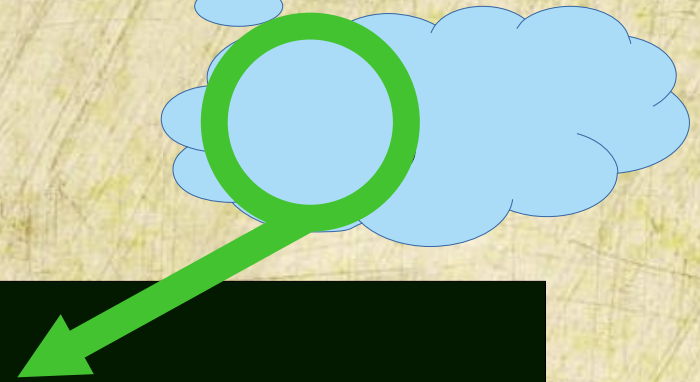
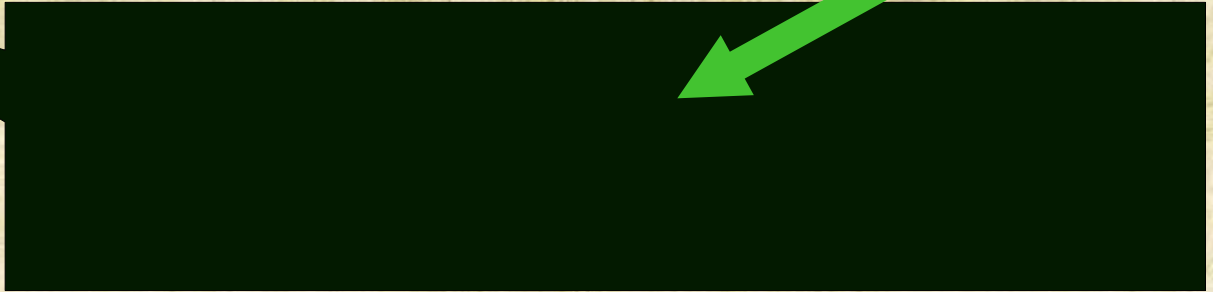
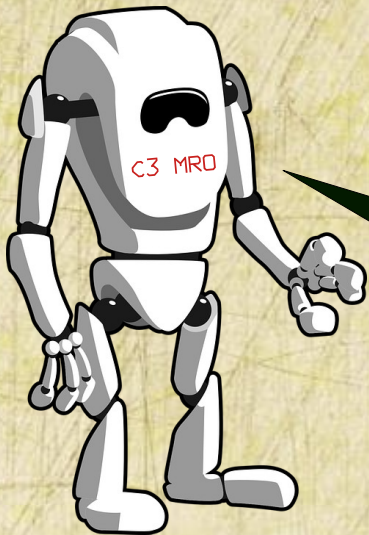




class ( ):

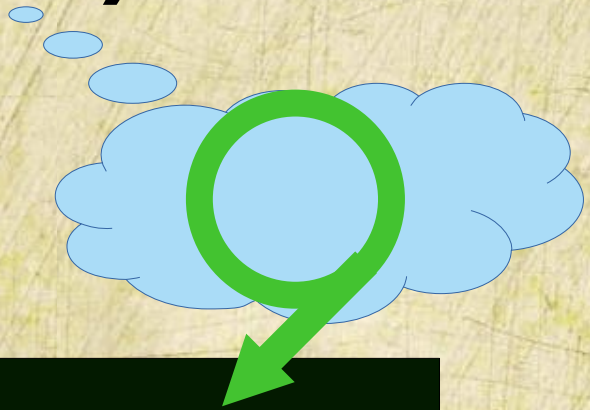
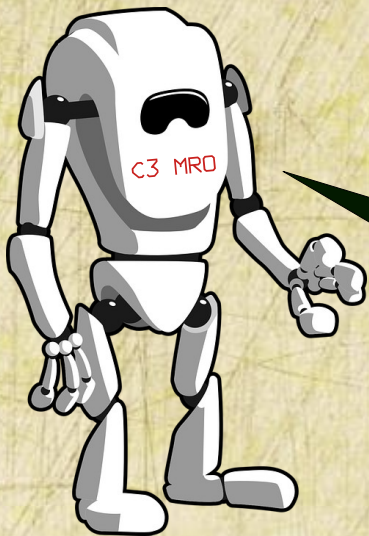


class ( ):

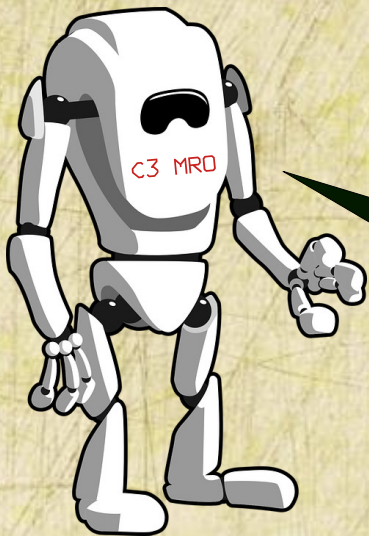




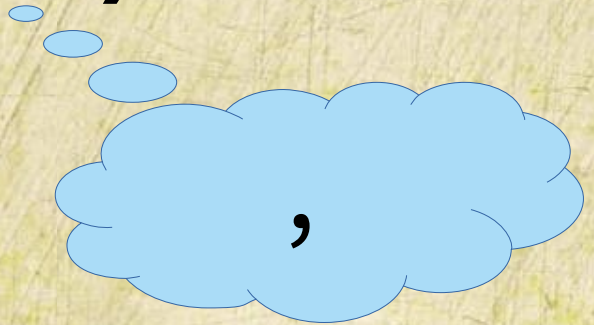
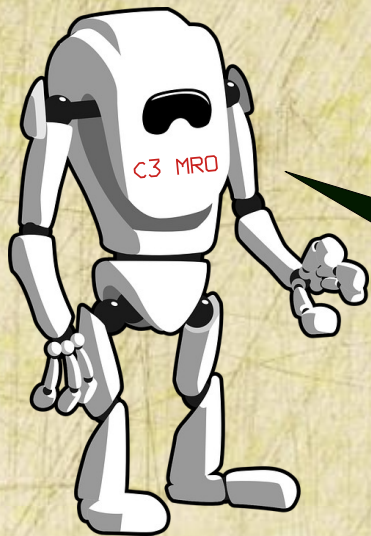
class ( ):



class ( ):

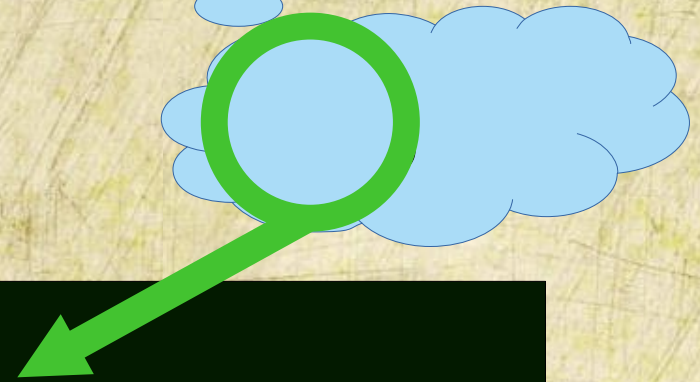
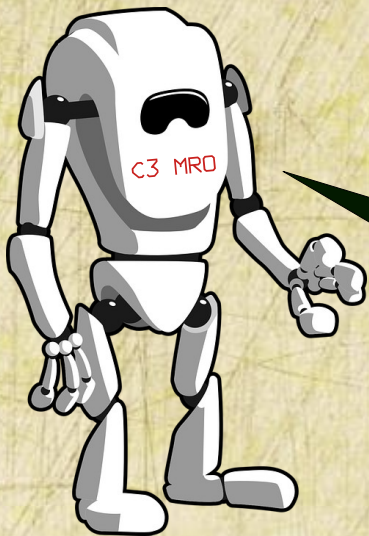


class ( ):

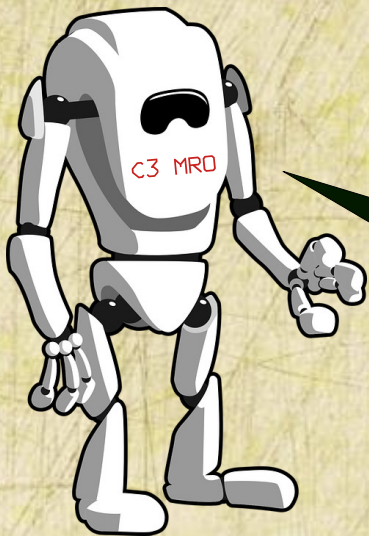




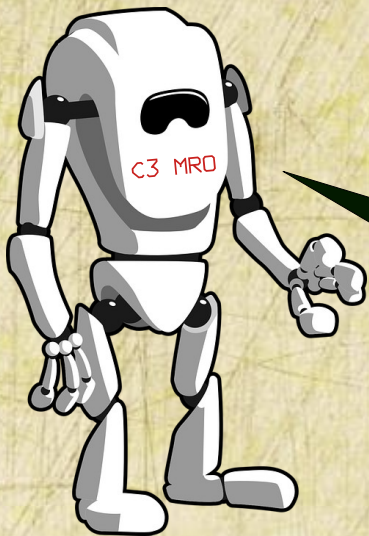
class ( ):



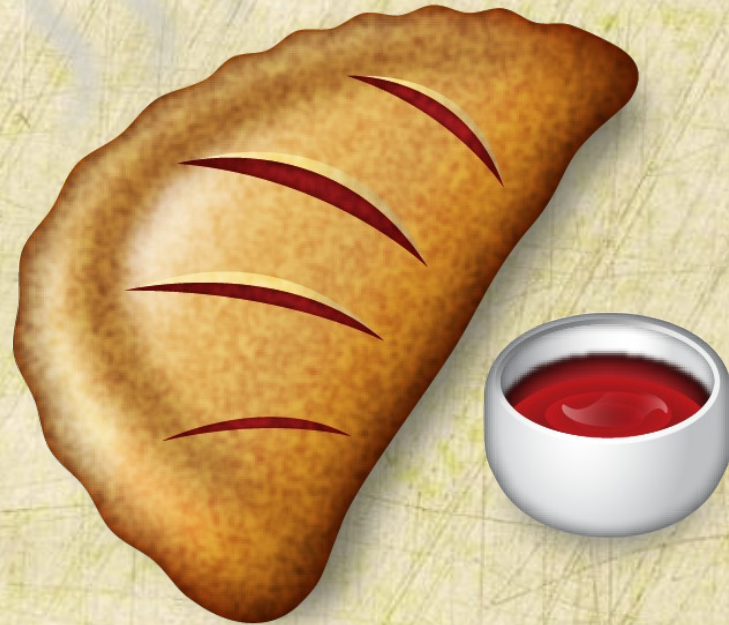
class ( ):



class ( ):

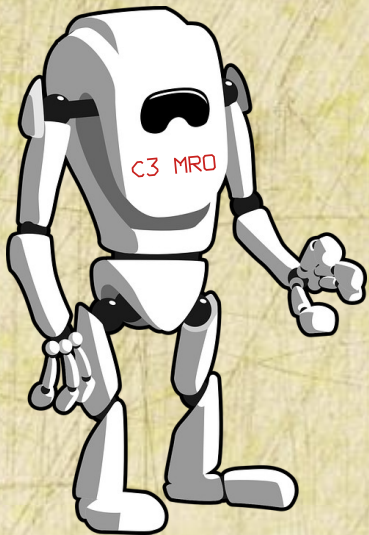






Credit: Kevin McCauley, McCauley Creative LLC  
@kmccauley  
Used with Permission

class 🥟 ( , ):





class

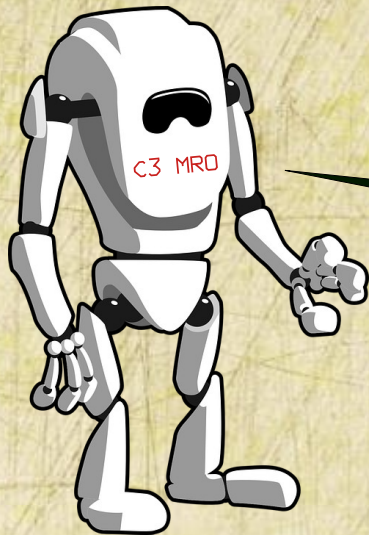
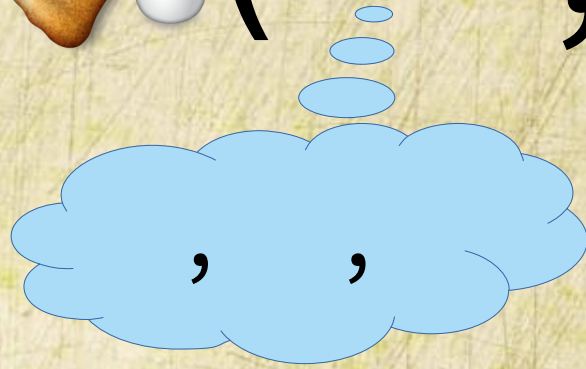


(

,

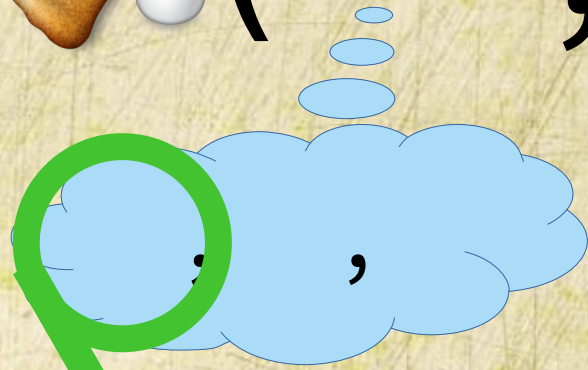
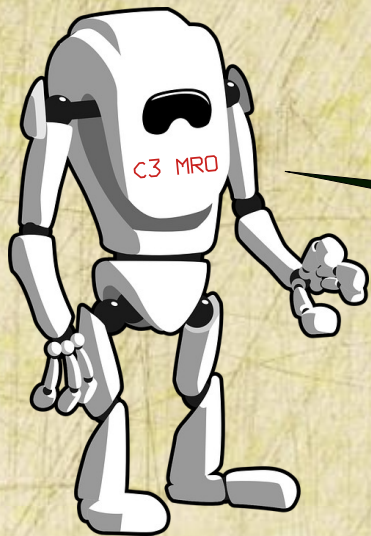
)

:

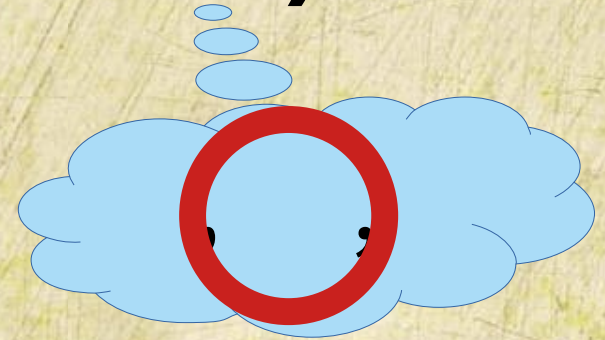
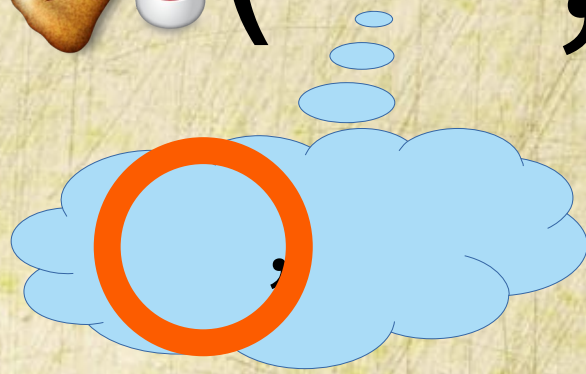
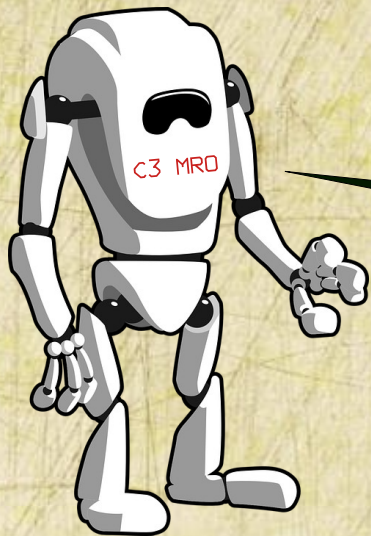




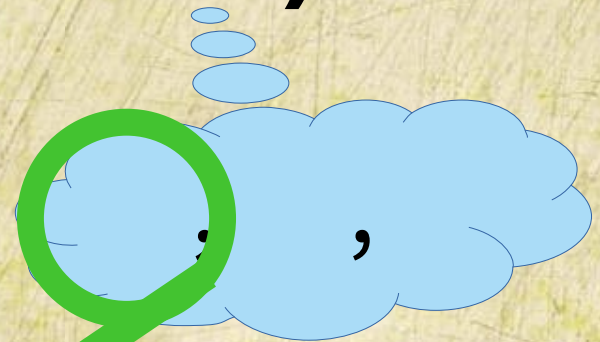
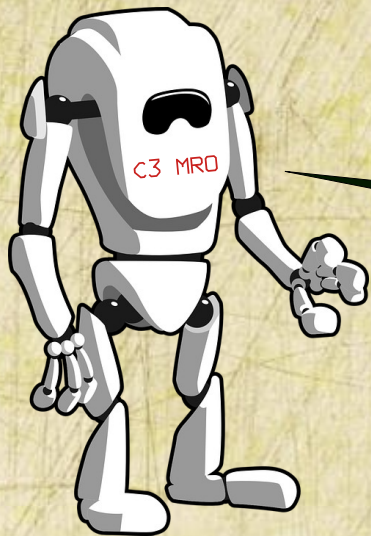
class 🥟 ( , ) :



class 🥟 ( , ) :

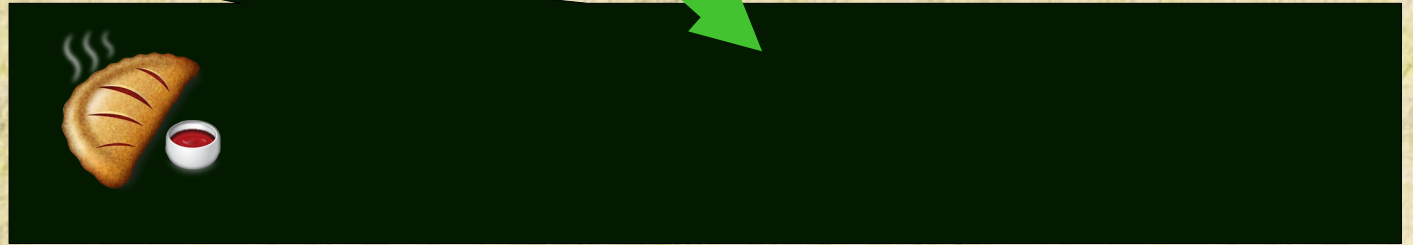
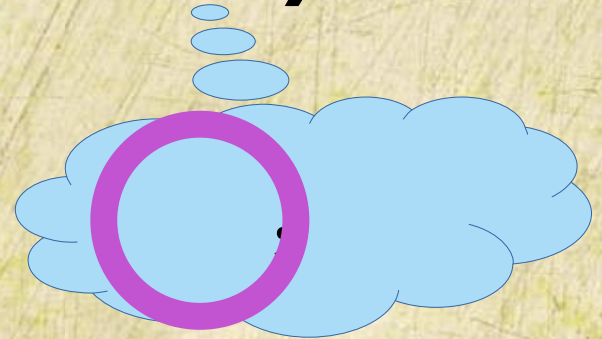
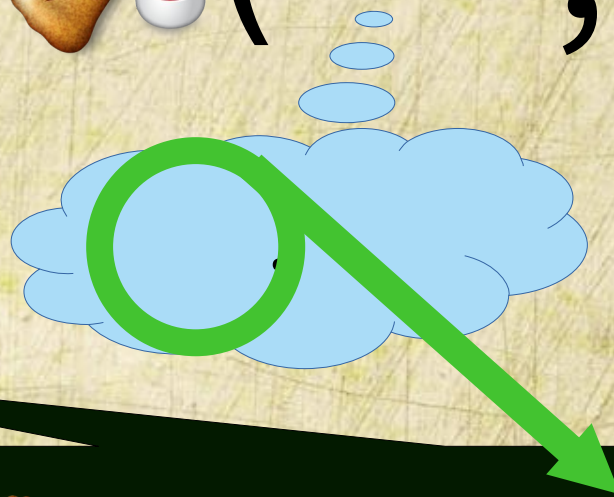
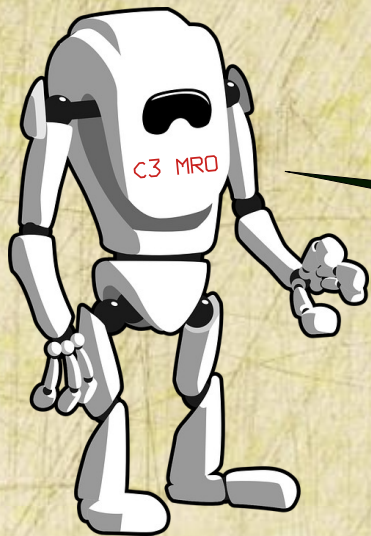


class 🥟 ( , ) :

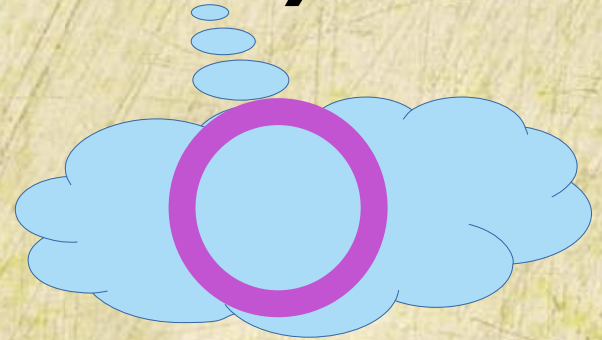
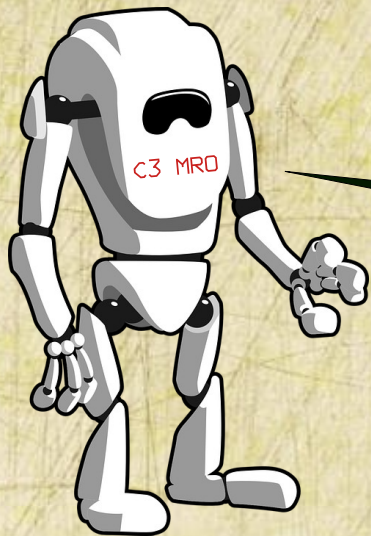




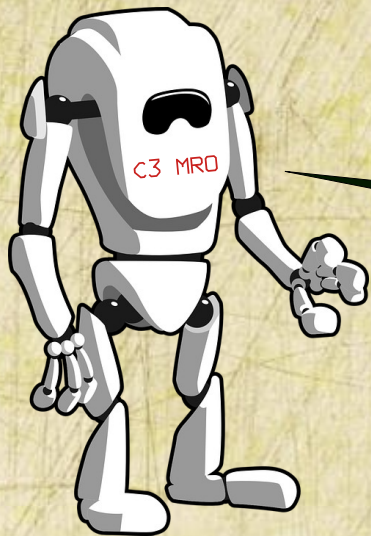
class 🥟 ( , ) :



class 🥟 ( , ) :



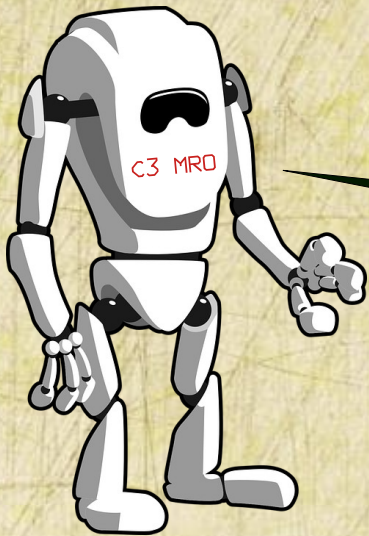
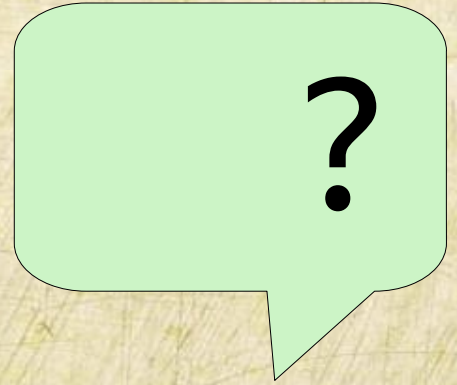
class 🥟 ( , ) :

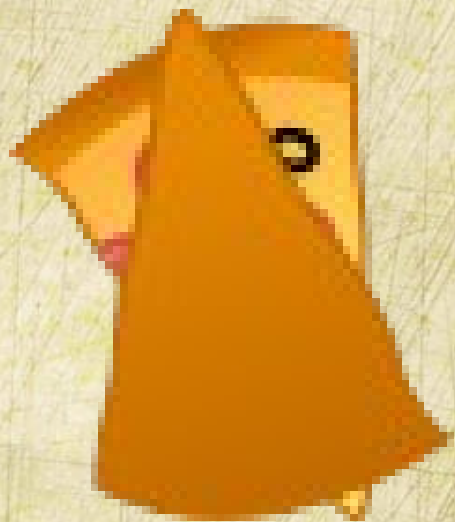






( )





class

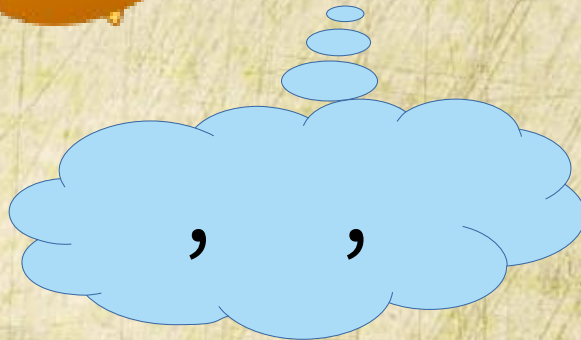
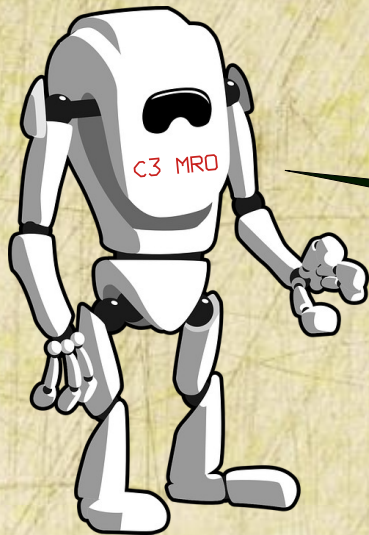


(

,

)

:





class

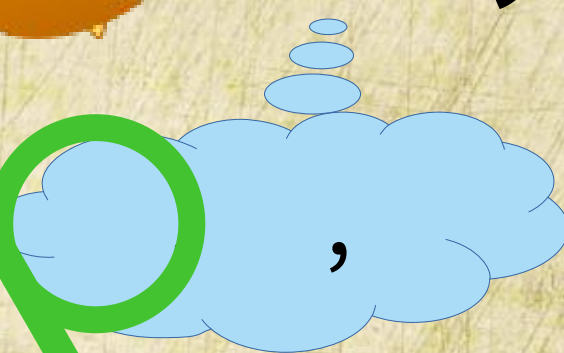
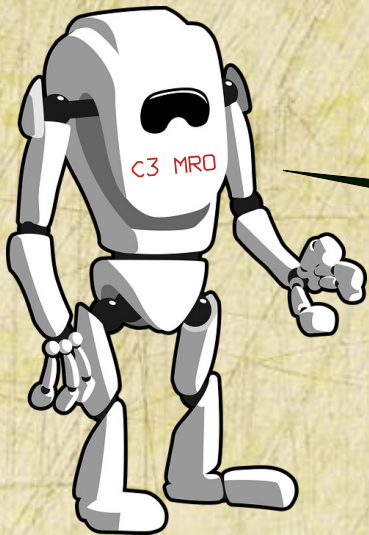


(

,

)

:



class

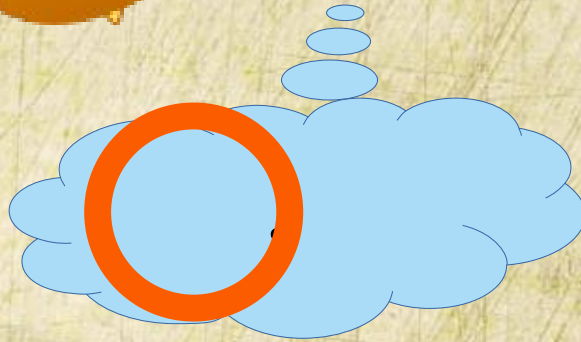
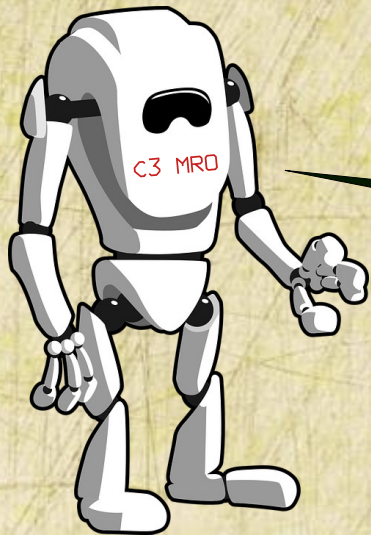


(

,

)

:



class

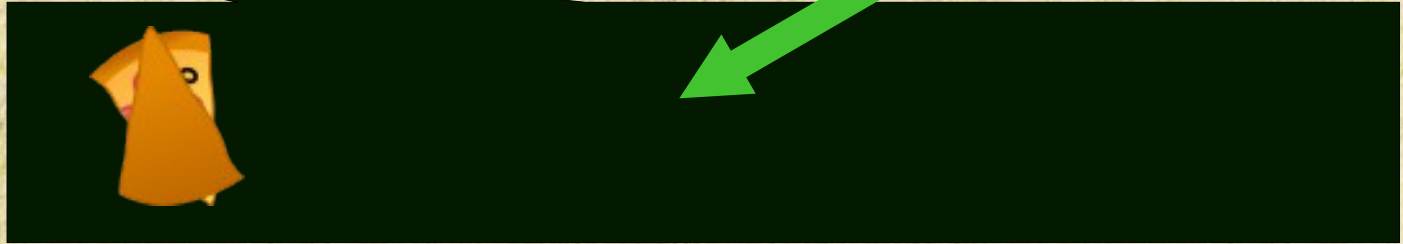
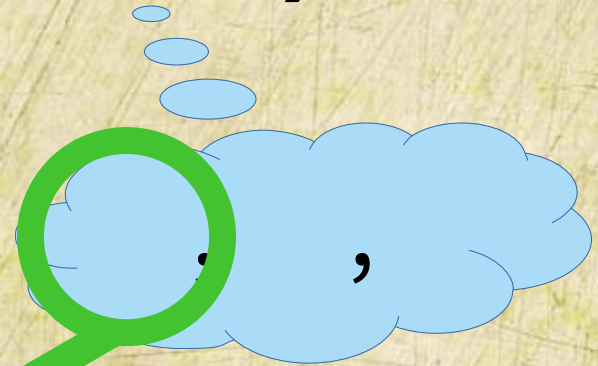
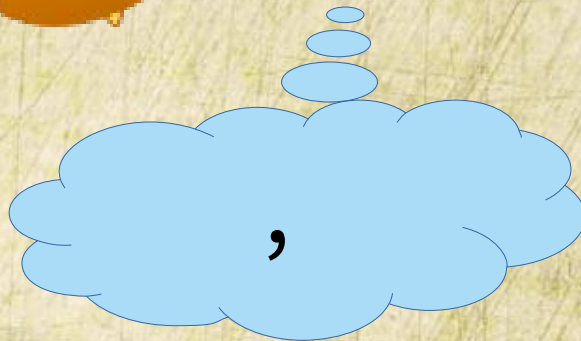
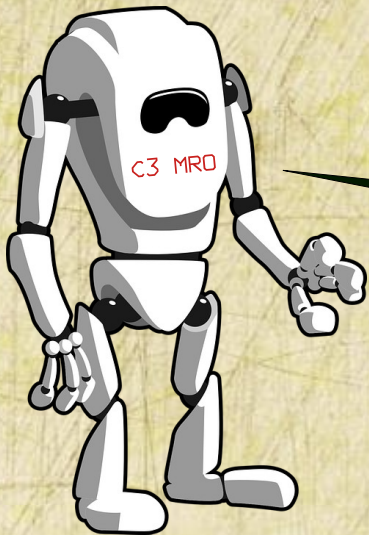


(

,

)

:





class

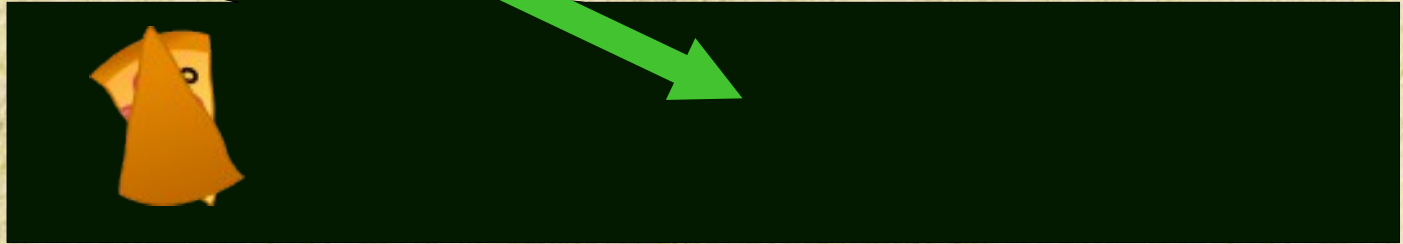
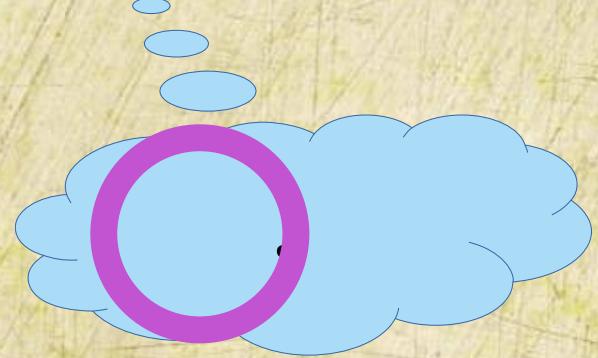
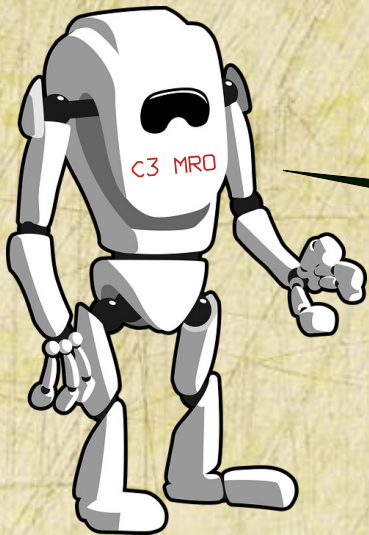


(

,

)

:



class

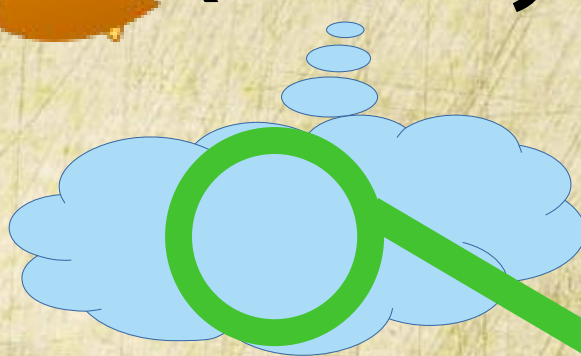
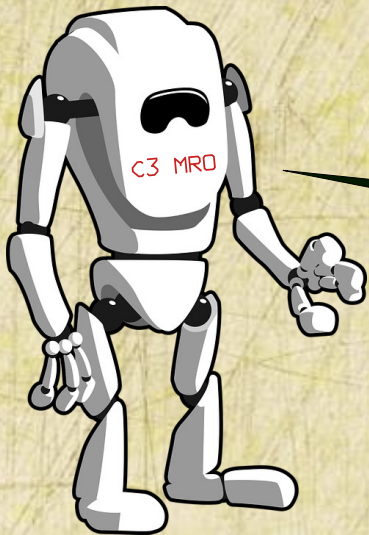


(

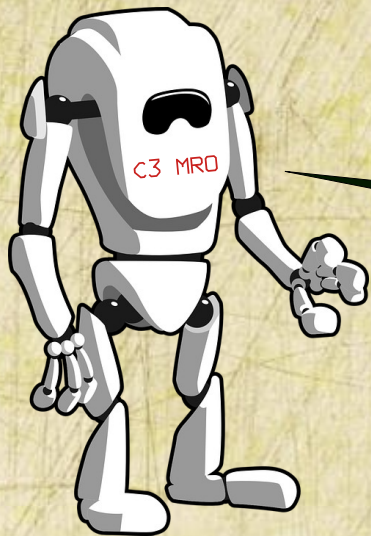
,

)

:

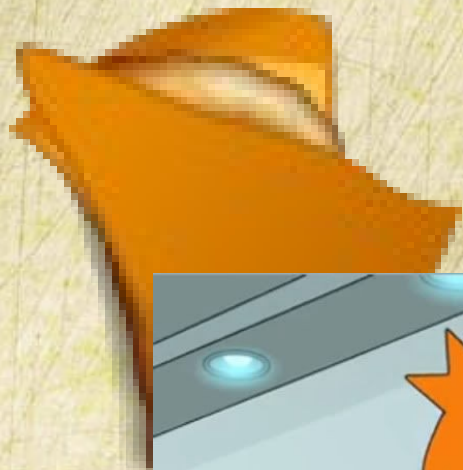


class  ( , ) :







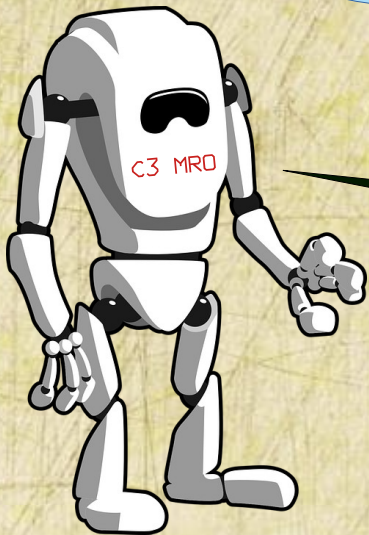
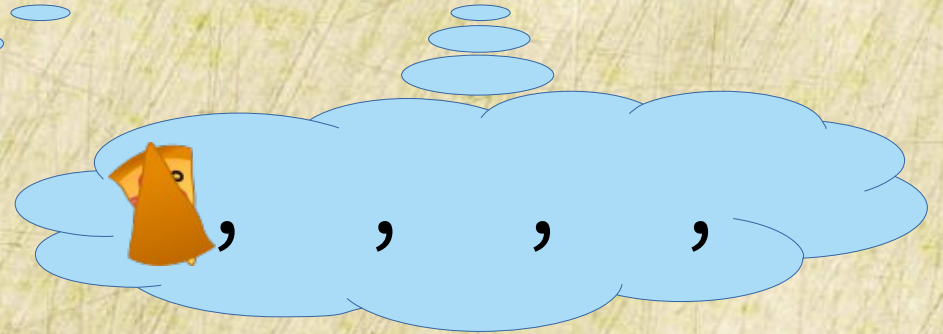
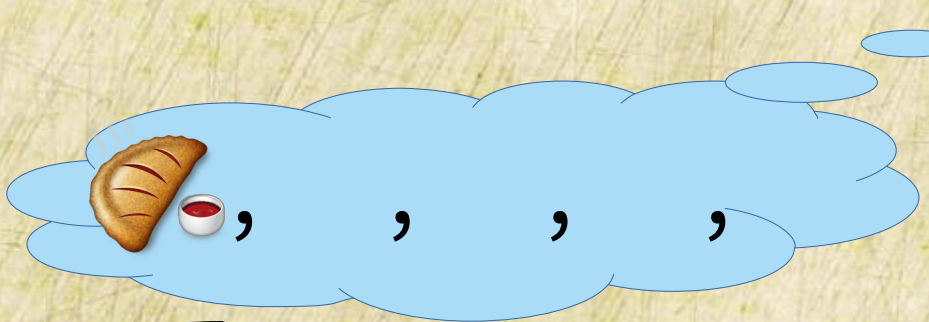


**SHUT UP AND TAKE MY MONEY**




class

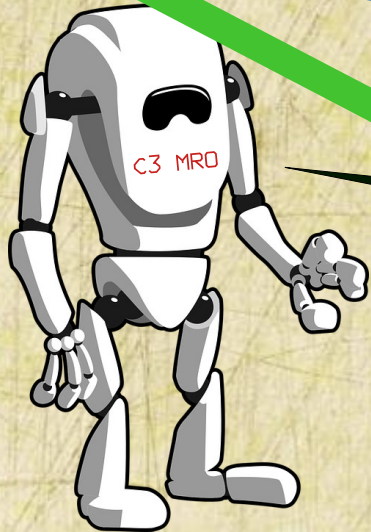
( , ) :






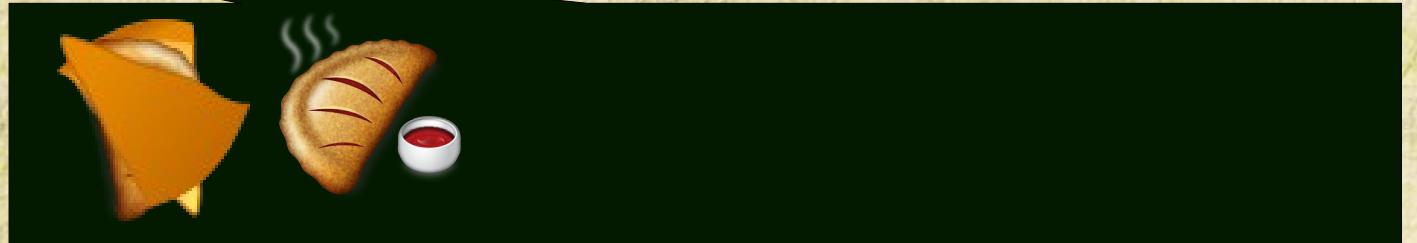
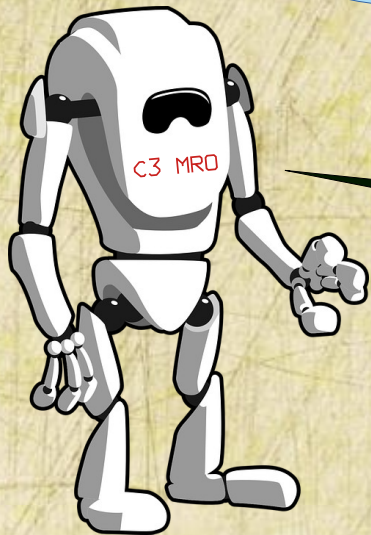
class

(  (   ,  ) :



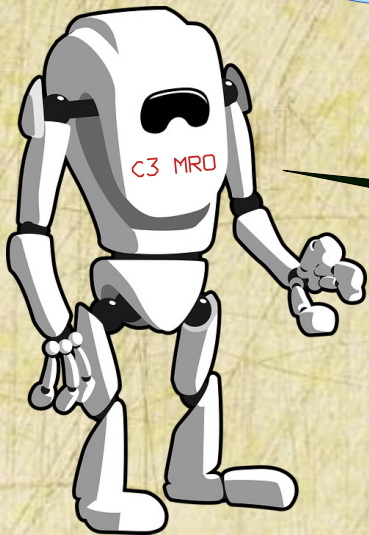
class

 (  ,  ) :



class

( 🥟 , 🥟 ) :



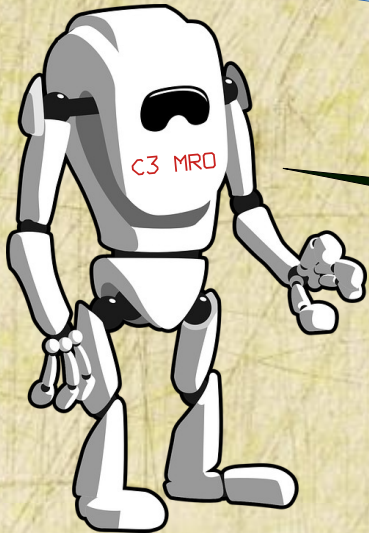


class


( 🥟 , 🥟 ) :

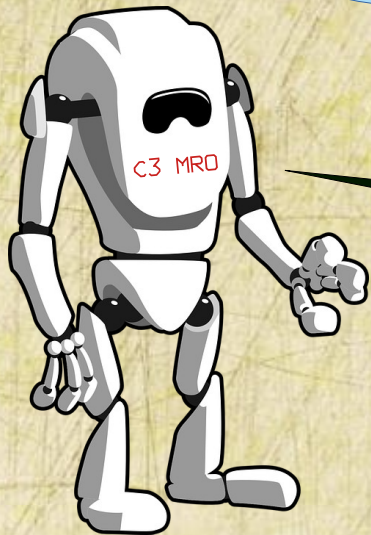
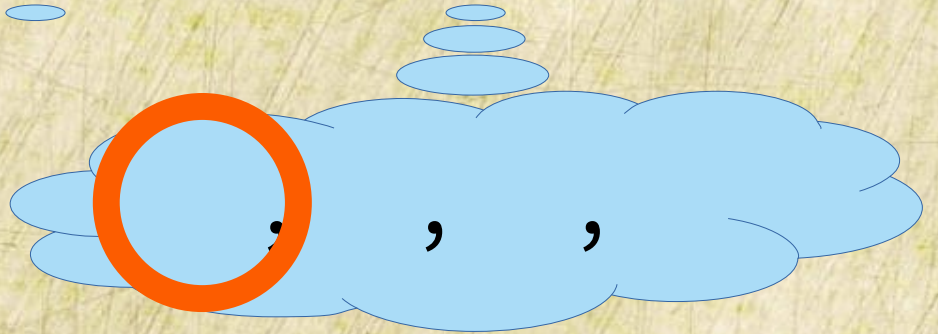
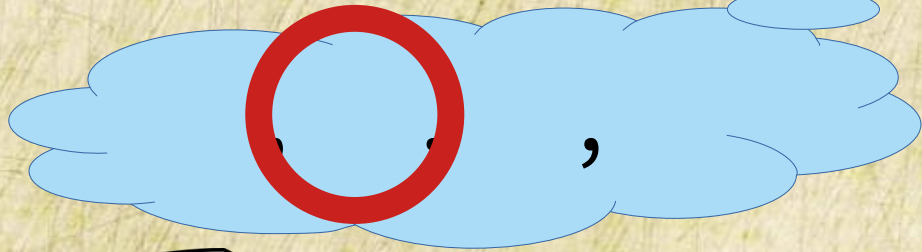
🥟 , 🥟

🥟 , 🥟



class

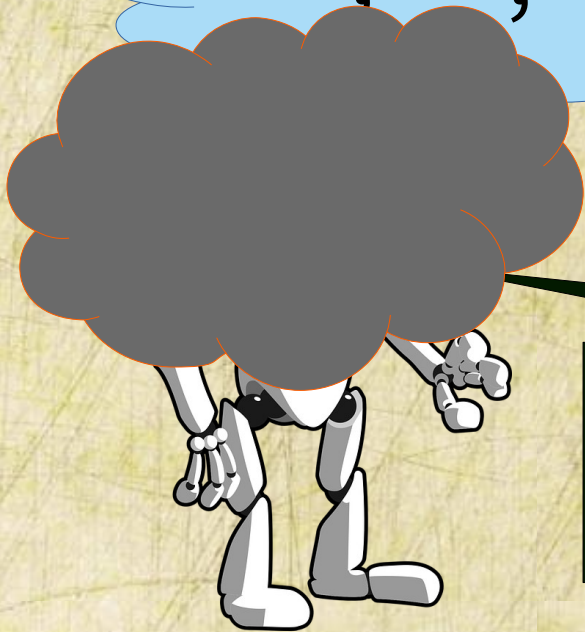
 (  ,  ) :



class

( 🥟 , 🥟 ) :

**TypeError: Cannot create a consistent method resolution**






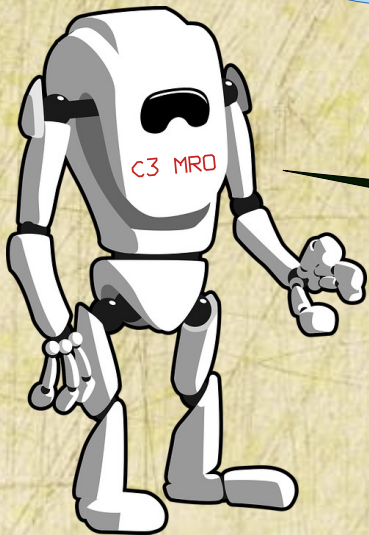
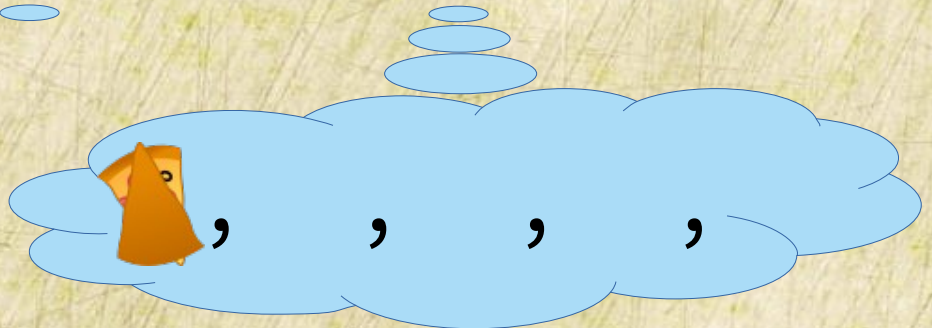
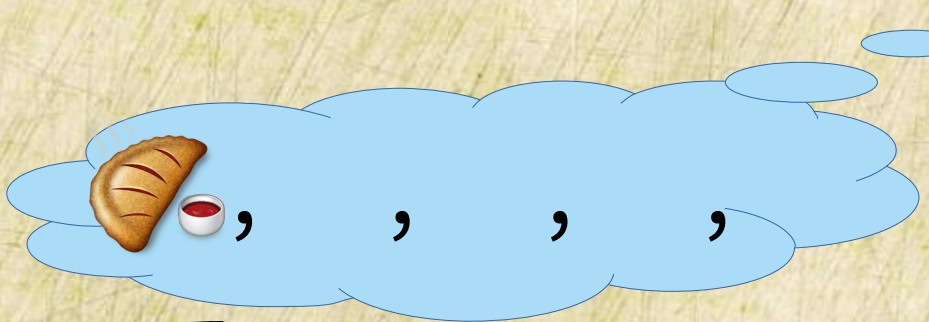
class

( 🥟 🥫 , 🥟 ) :




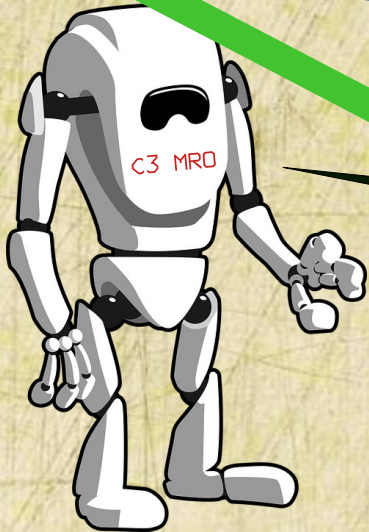
class

( , ) :



class

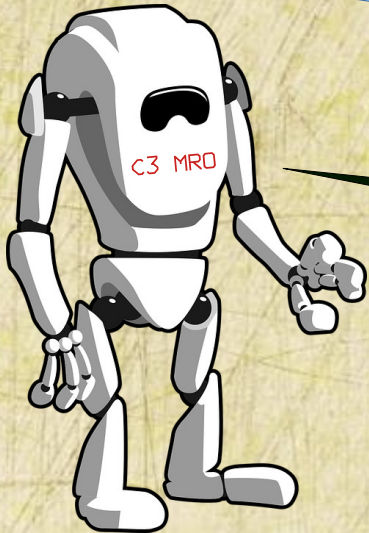
( ( ,  ) :





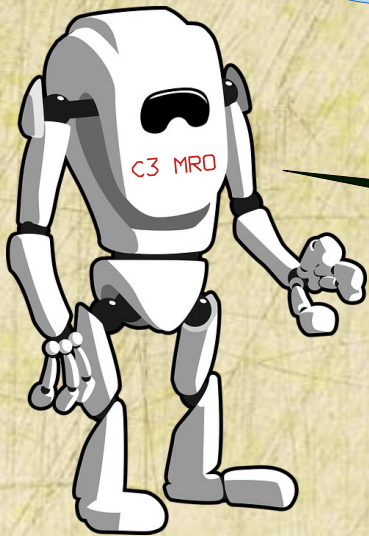
class

( 🥟 , 🥟 ) :



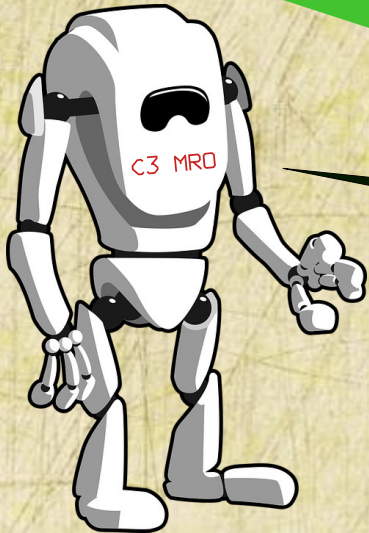
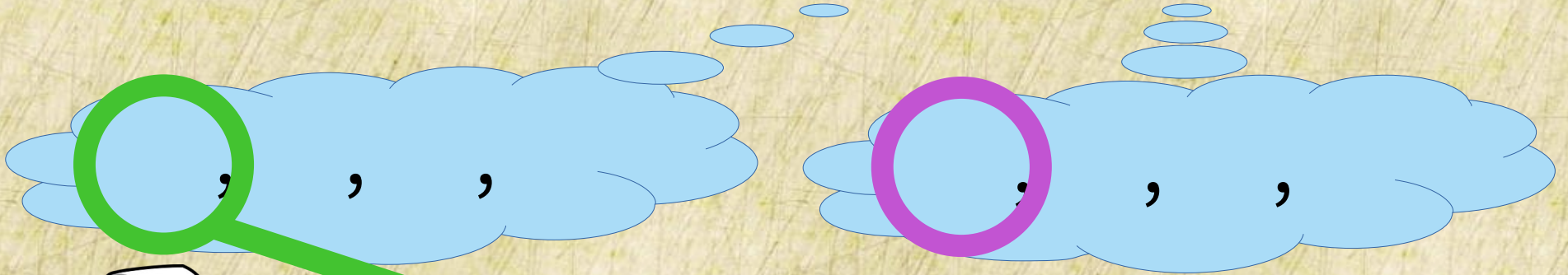
class

( 🥟 , 🥟 ) :



class

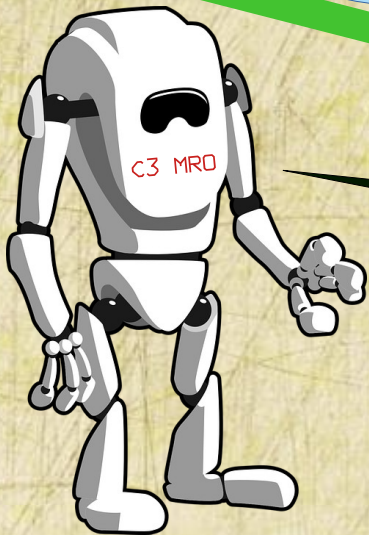
( 🥟 , 🥟 ) :





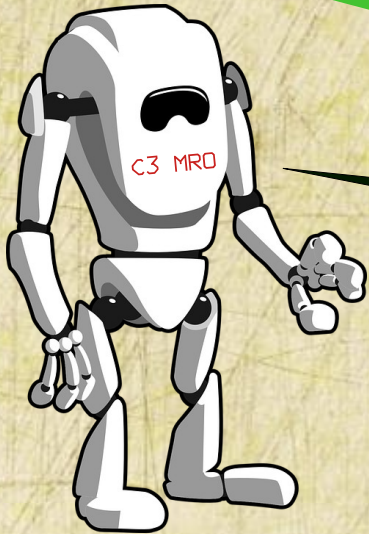
class

( 🥟 , 🥟 ) :



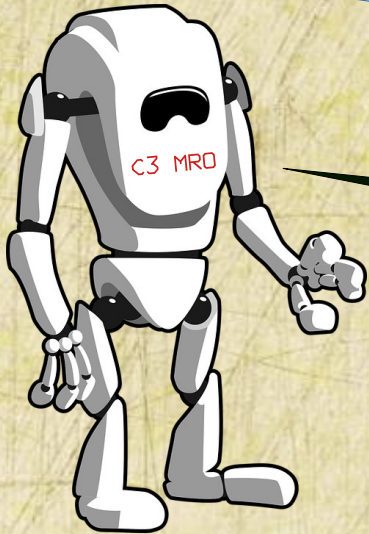
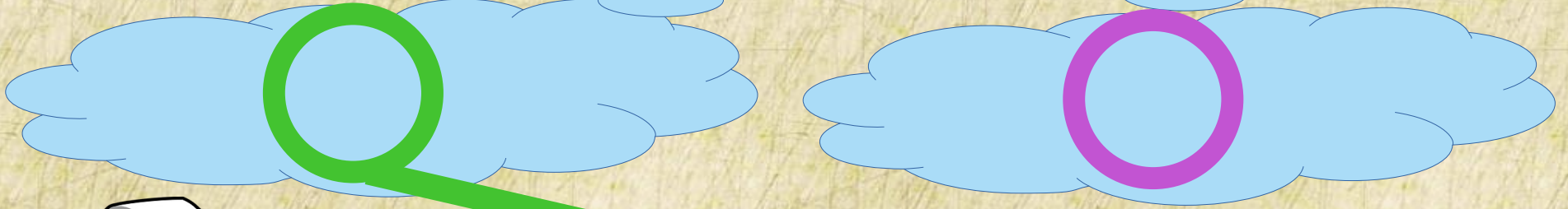
class

( 🥟 , 🥟 ) :



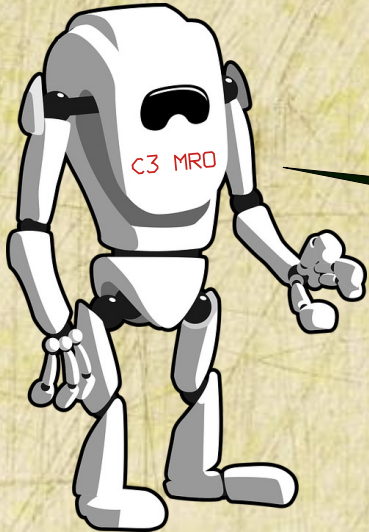
class

( 🥟 , 🥟 ) :





class 🌮 ( 🥟🍷 , 🌮 ) :



class



(



,



)

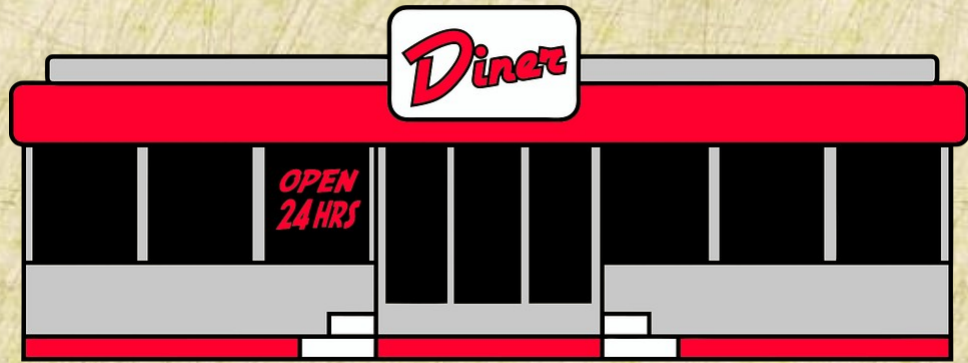
:

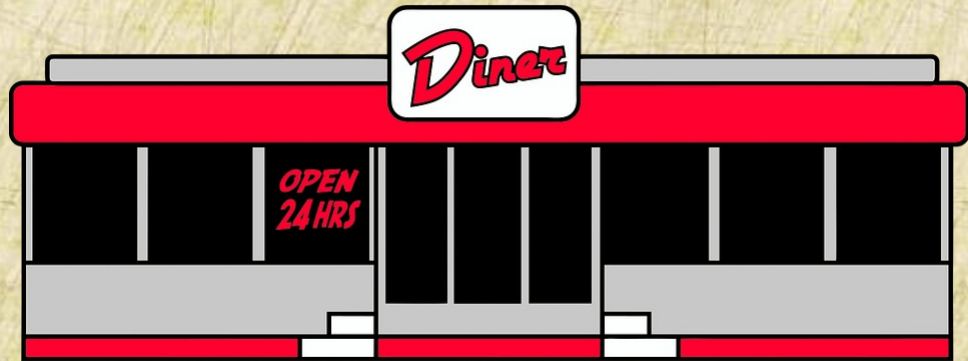
```
def (self):  
    . (self)
```

# Mixins

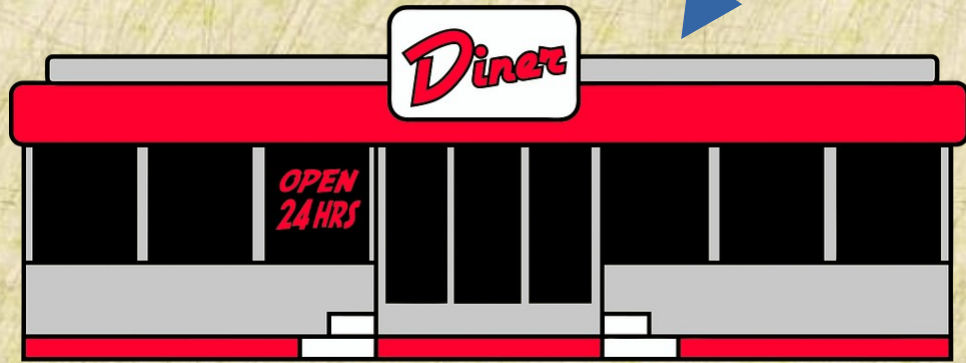
Applied multiple inheritance





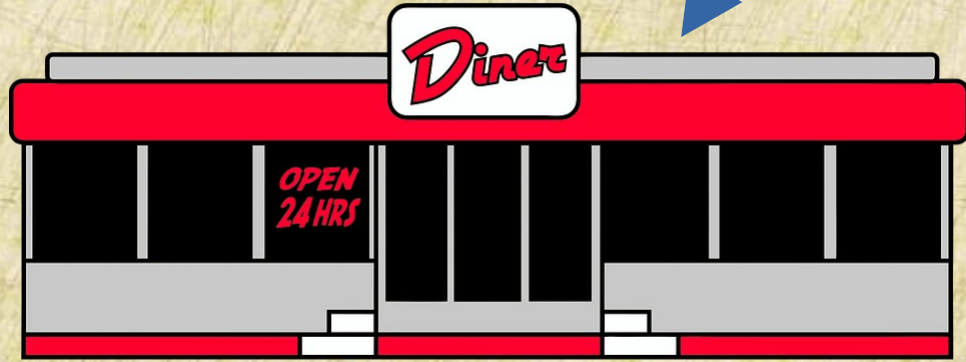


# Classes

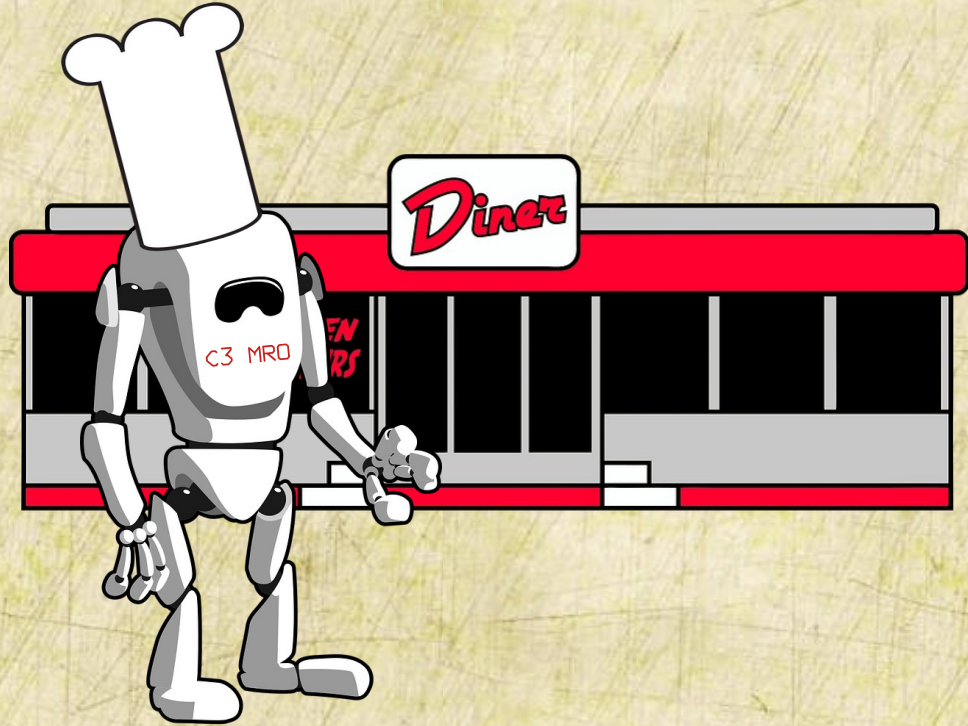


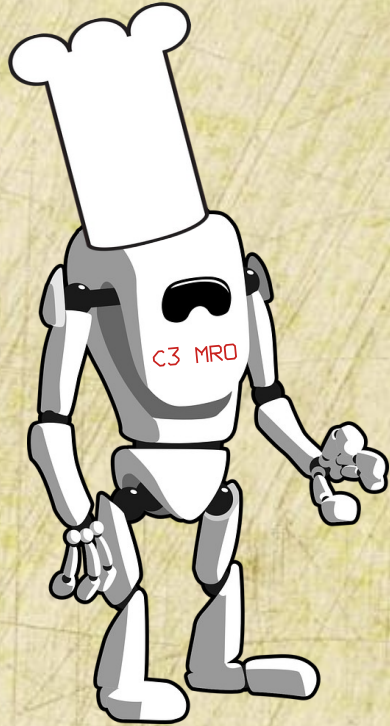


# Classes



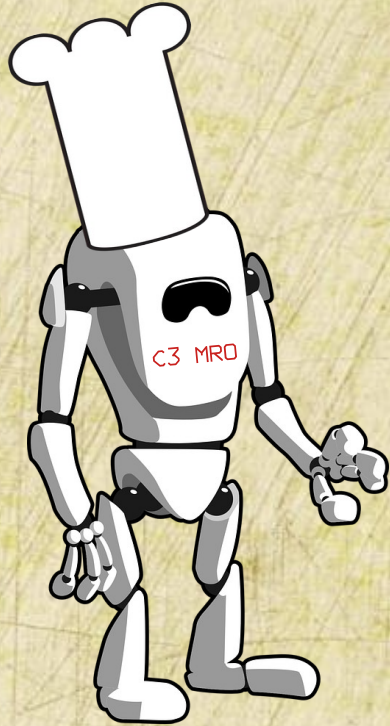
(They don't inherit from one another.)



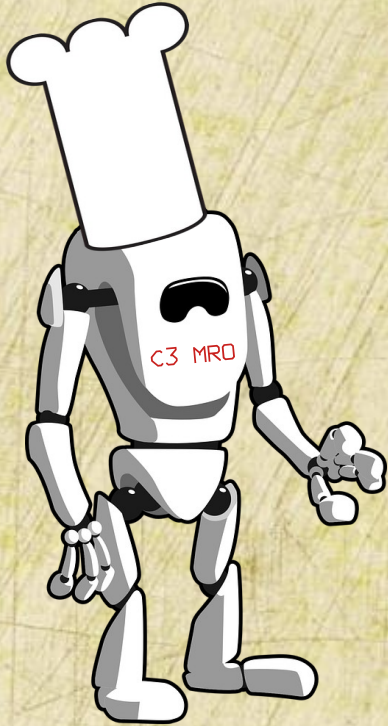


# ShortOrderCook





ShortOrderCook  
Needs restaurant data.

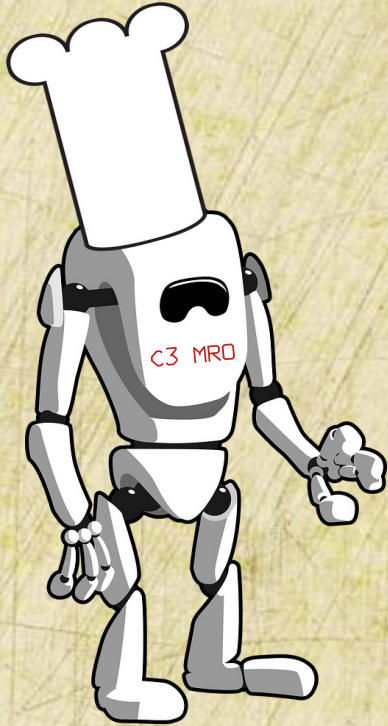


# ShortOrderCook

Needs restaurant data.

Produces food...





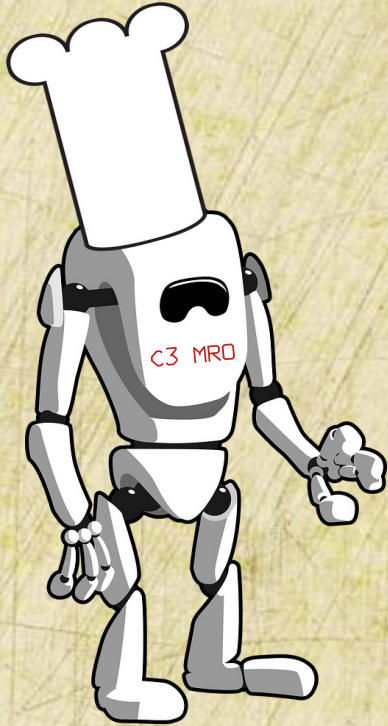
# ShortOrderCook

Needs restaurant data.

Produces food...

...for restaurant.





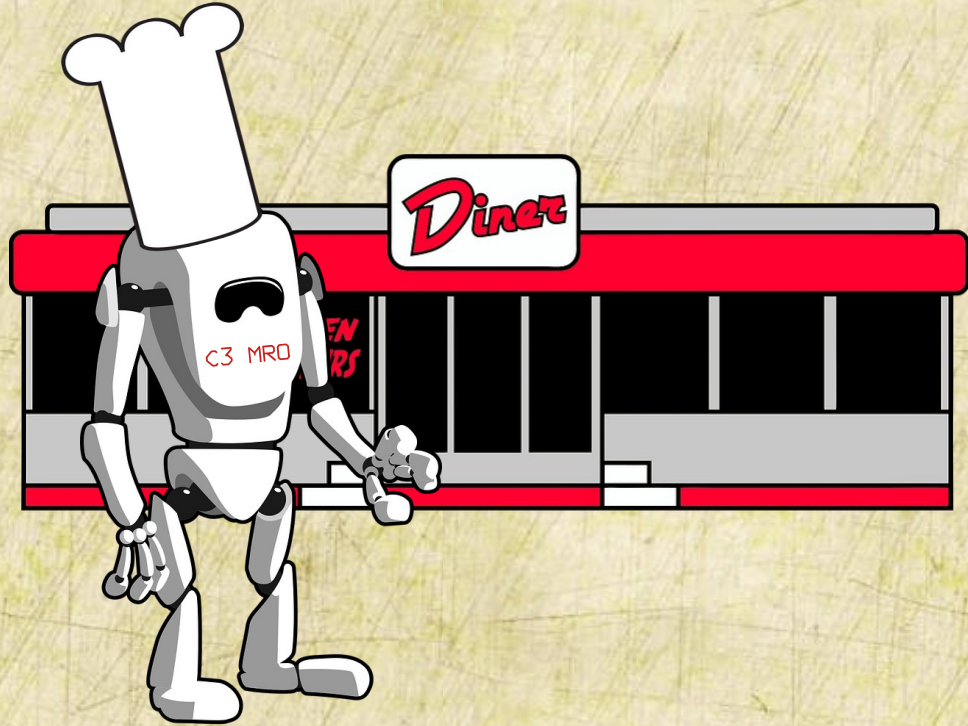
# ShortOrderCook

Needs restaurant data.

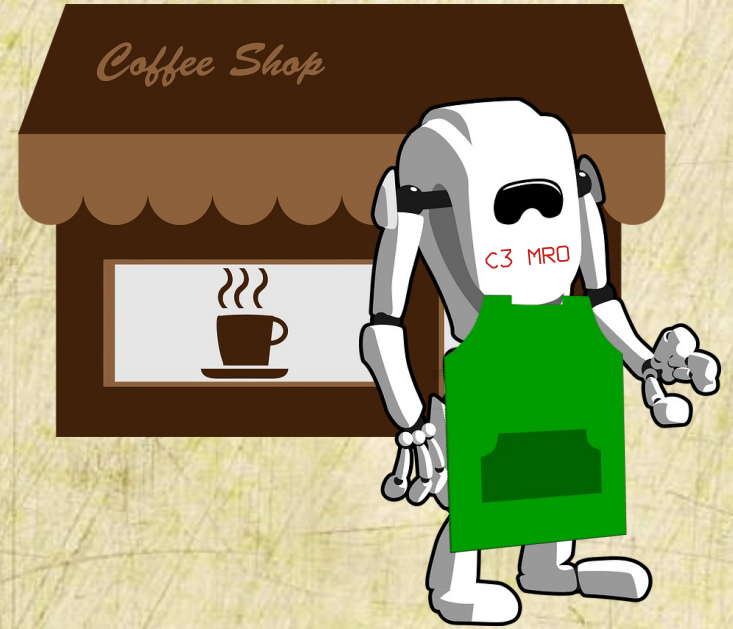
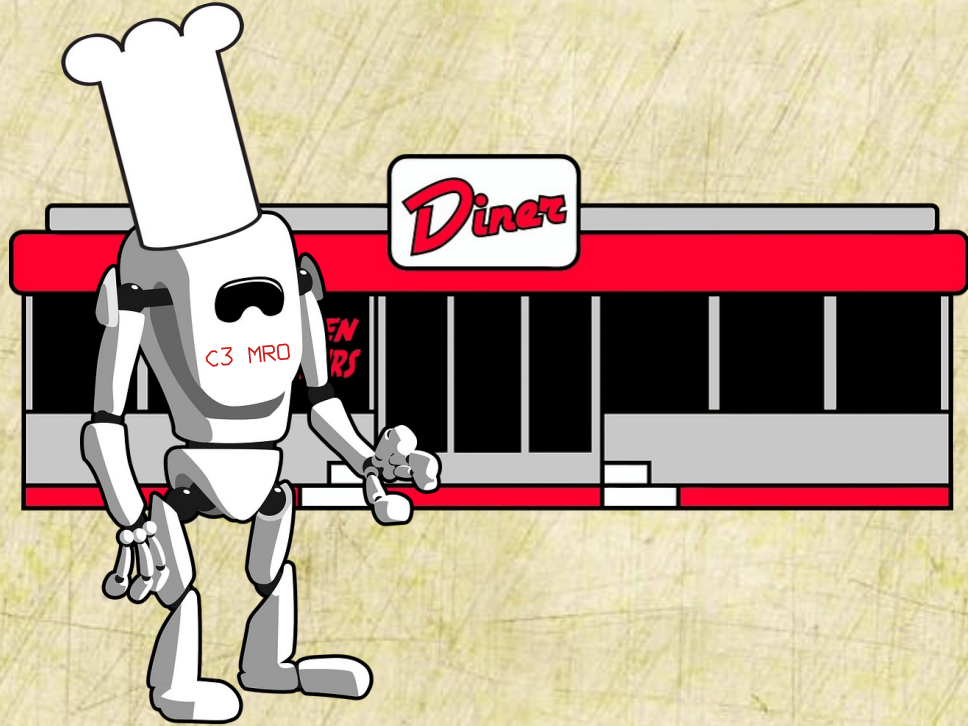
Produces food...

...for restaurant.

(Doesn't make much sense alone!)

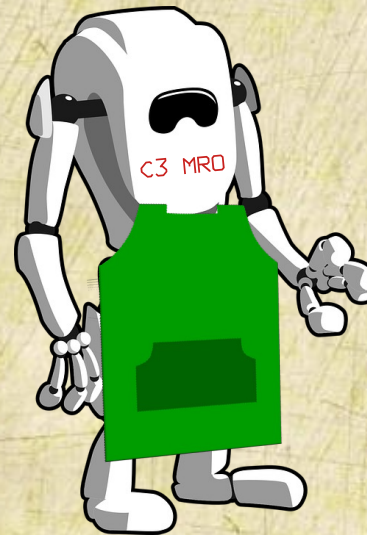






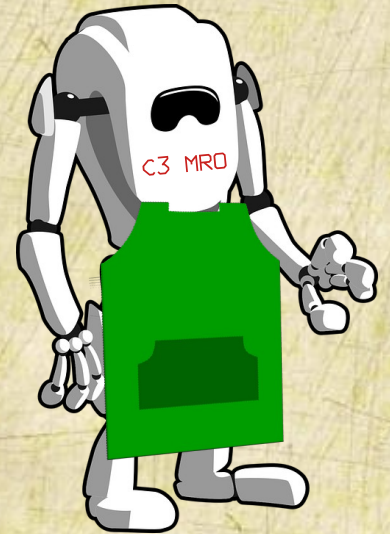


# Barista



# Barista

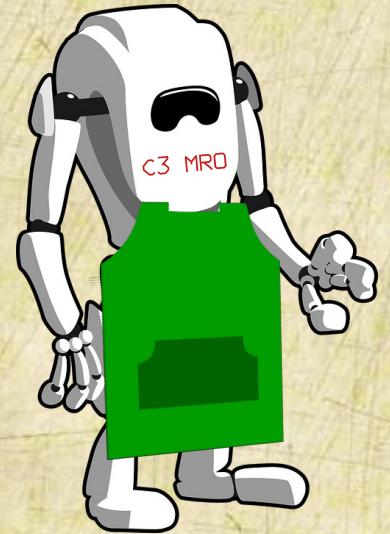
Needs restaurant data.



# Barista

Needs restaurant data.

Produces drinks...



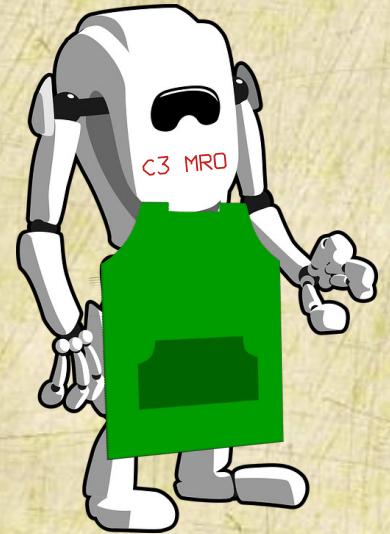


# Barista

Needs restaurant data.

Produces drinks...

...for restaurant.



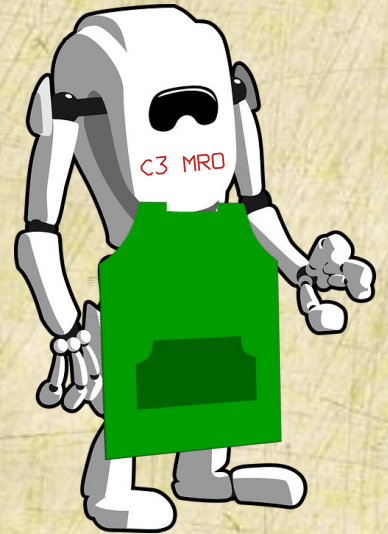
# Barista

Needs restaurant data.

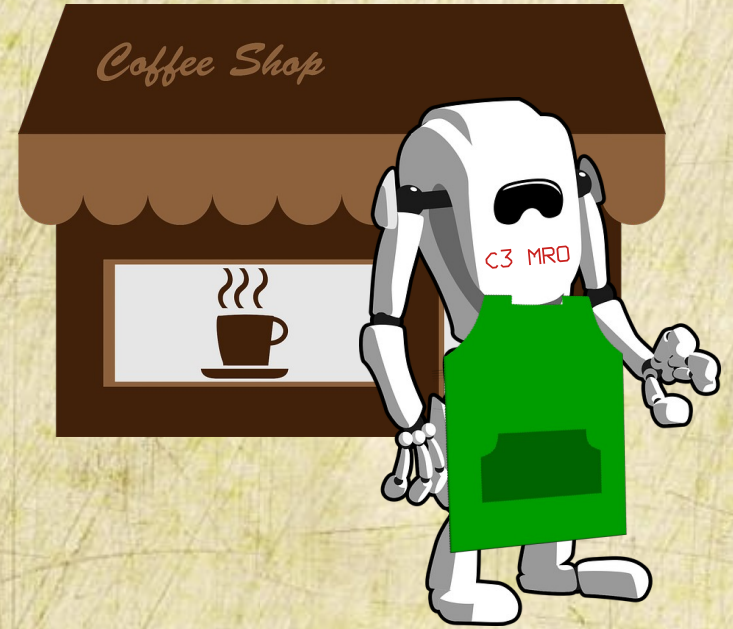
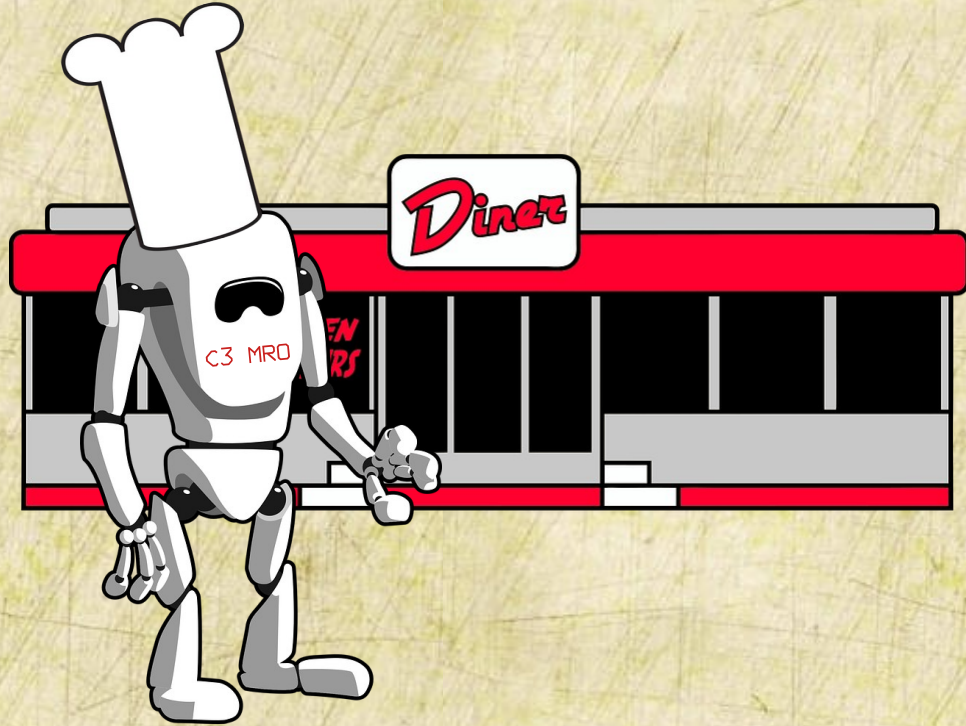
Produces drinks...

...for restaurant.

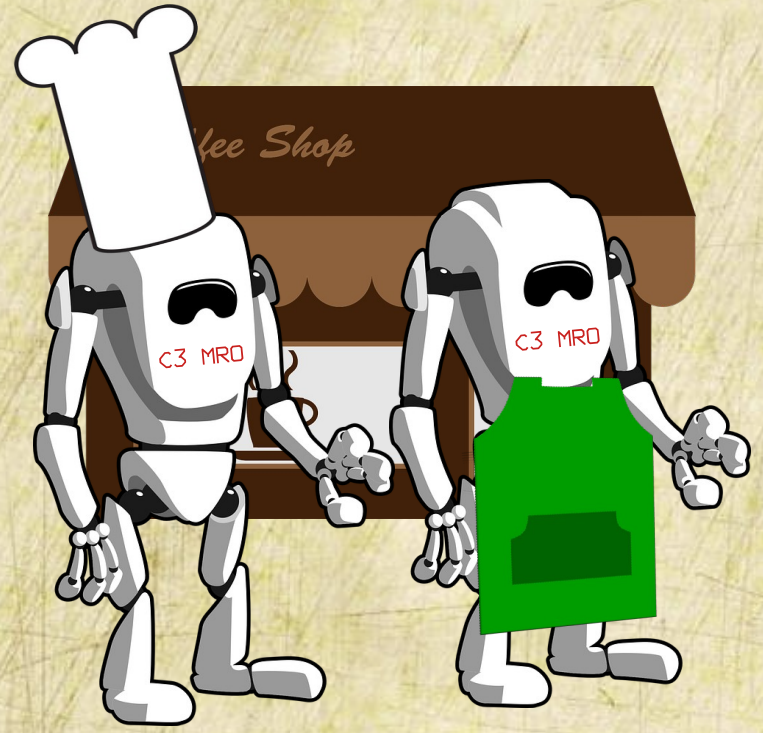
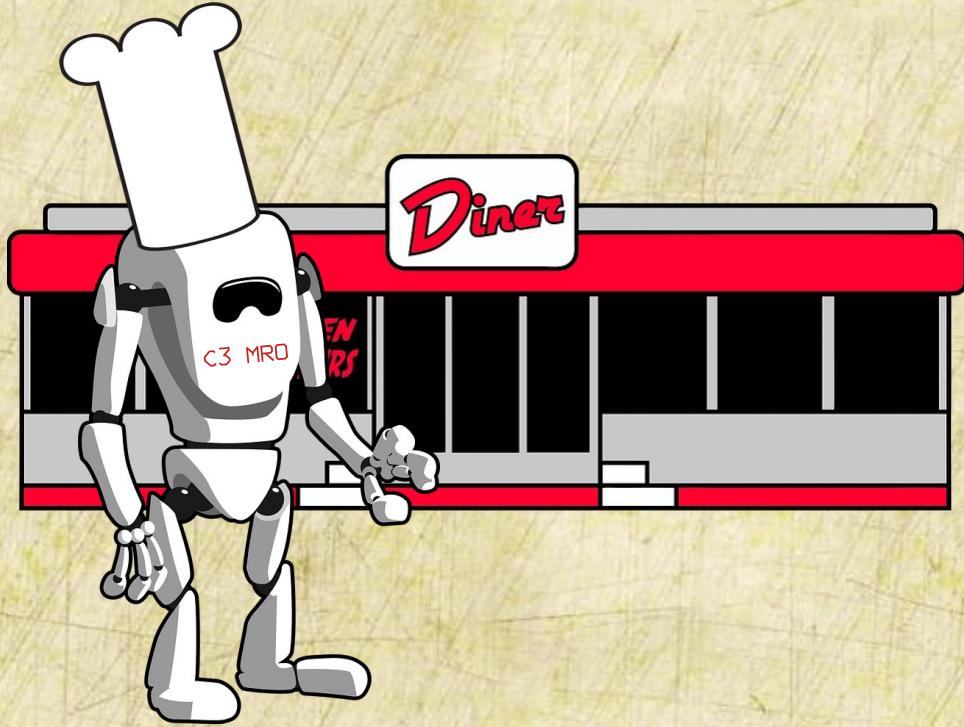
(Also doesn't make much sense alone!)

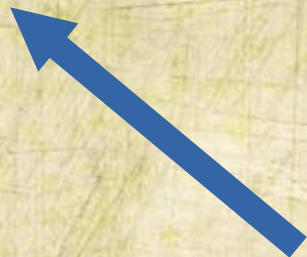
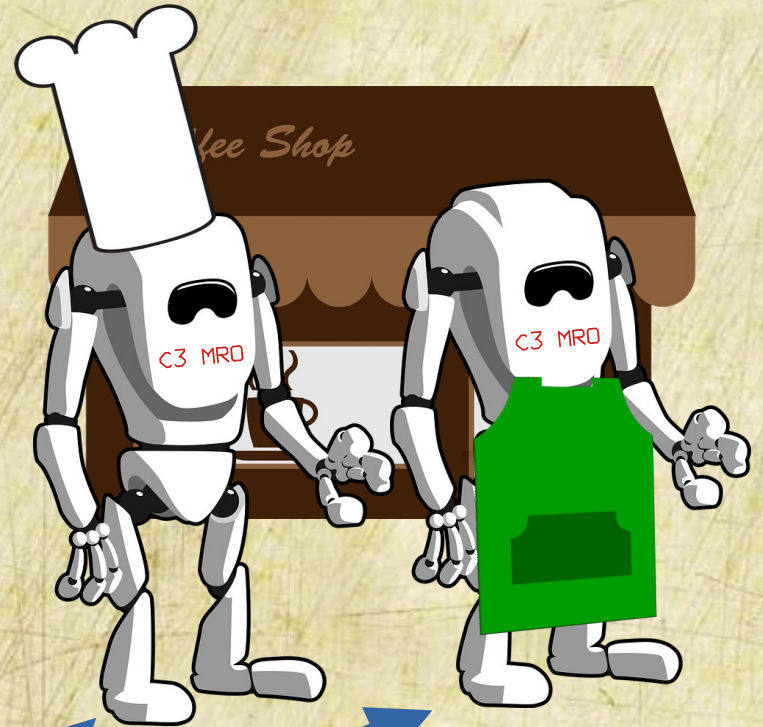
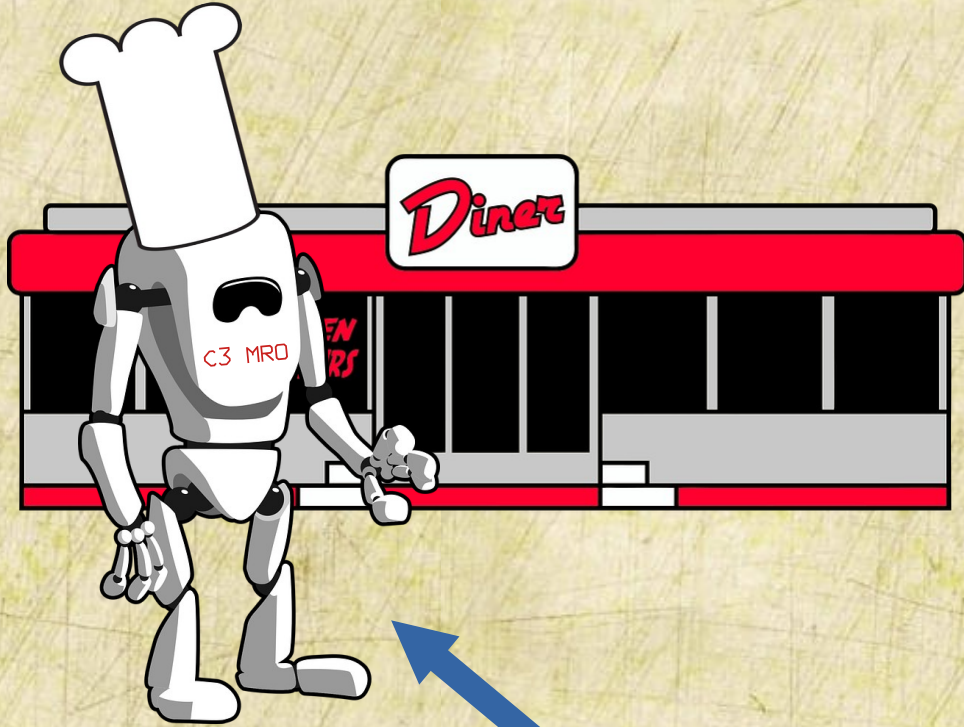












Mixins

**Mixin**



# Mixin

Contributes methods and/or attributes.

# Mixin

Contributes methods and/or attributes.

Can rely on attributes/methods of class using it.

# Mixin

Contributes methods and/or attributes.

Can rely on attributes/methods of class using it.

Not intended to stand alone.



# Mixin

Contributes methods and/or attributes.

Can rely on attributes/methods of class using it.

Not intended to stand alone.

Composition...via inheritance.

# Mixin

Contributes methods and/or attributes.

Can rely on attributes/methods of class using it.

Not intended to stand alone.

Composition...via inheritance.

**Let's have a little code,  
shall we?**



[https://github.com/codemouse92/  
WhoseMethod](https://github.com/codemouse92/WhoseMethod)



Jason C. McDonald

CodeMouse92

[codemouse92.com](http://codemouse92.com)