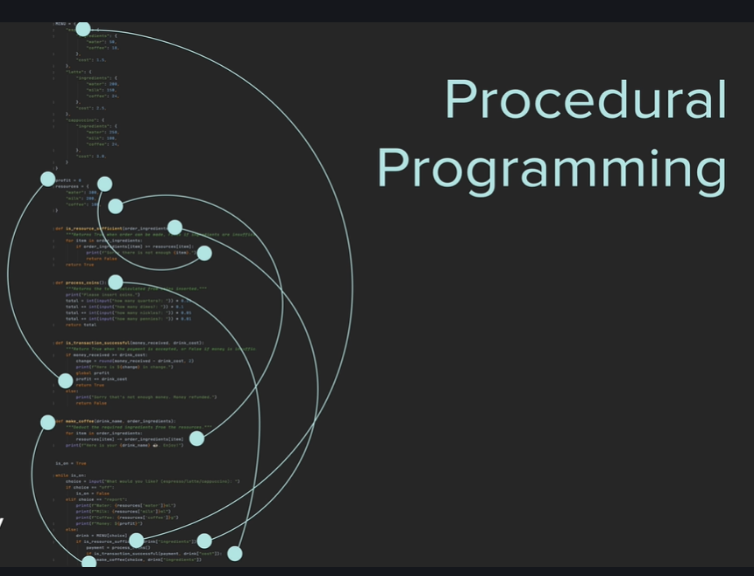
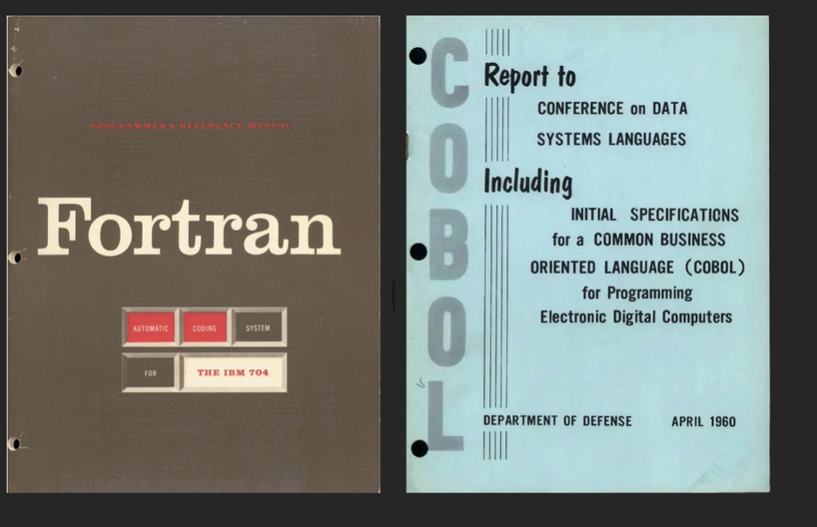
Day 16 intermediate object oriented



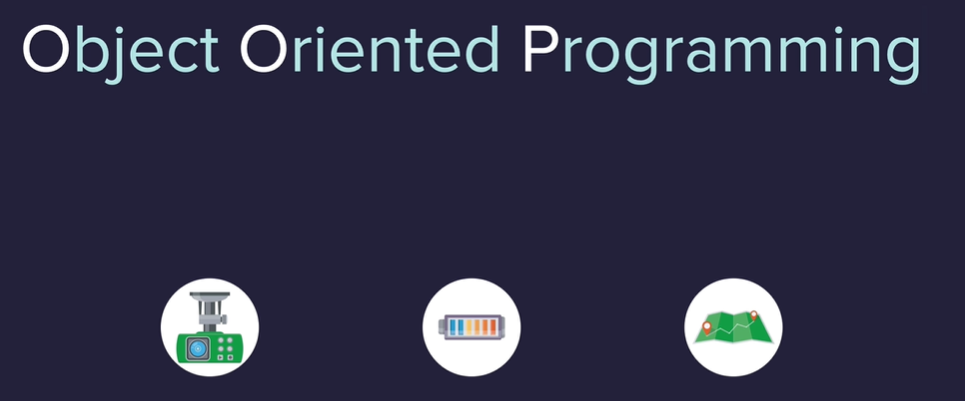
Top to bottom

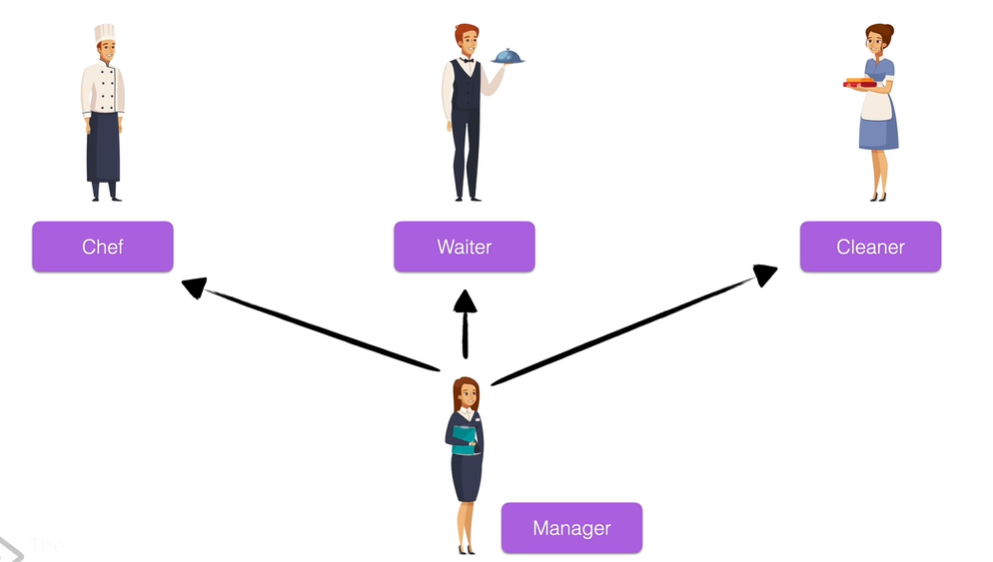


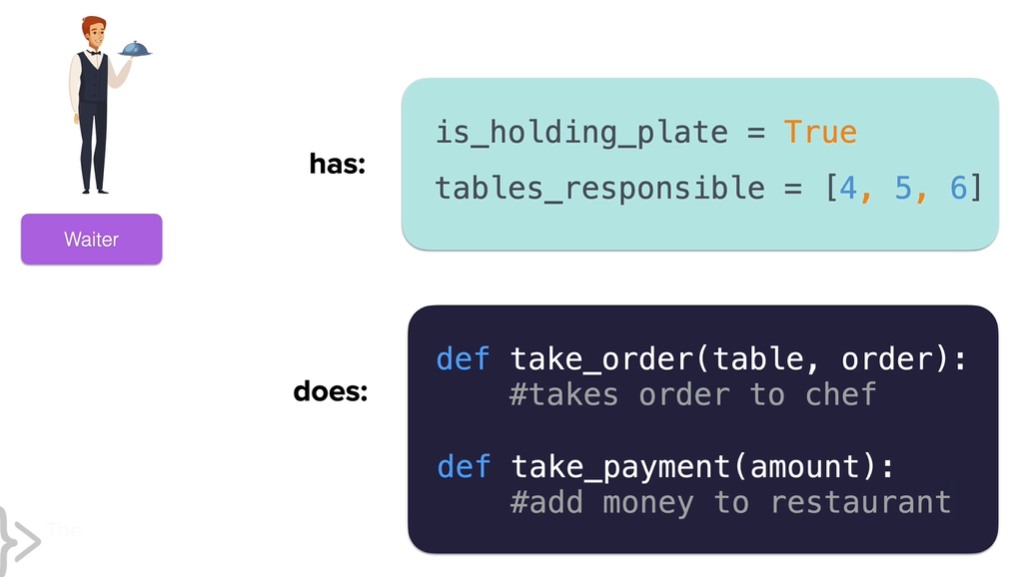
Old School

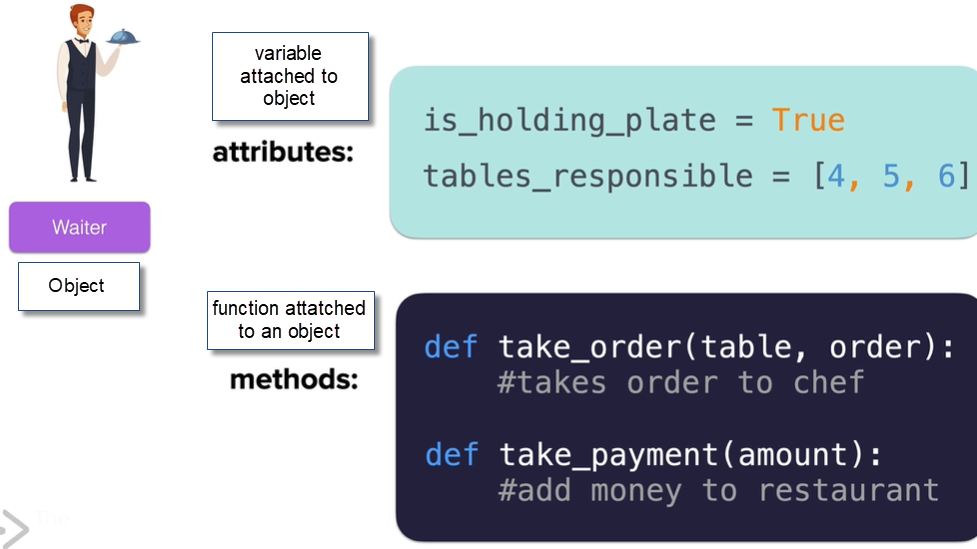


Split up project Making modules, many re useable – scalable for larger projects

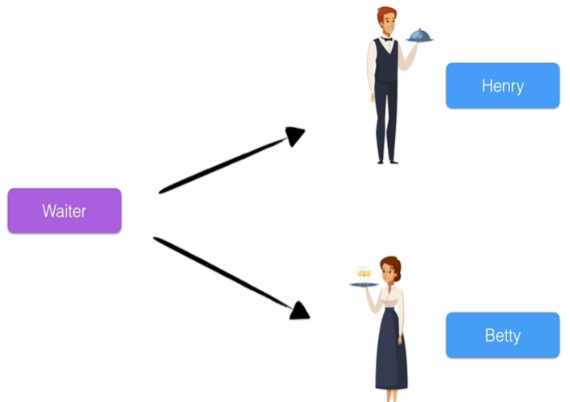


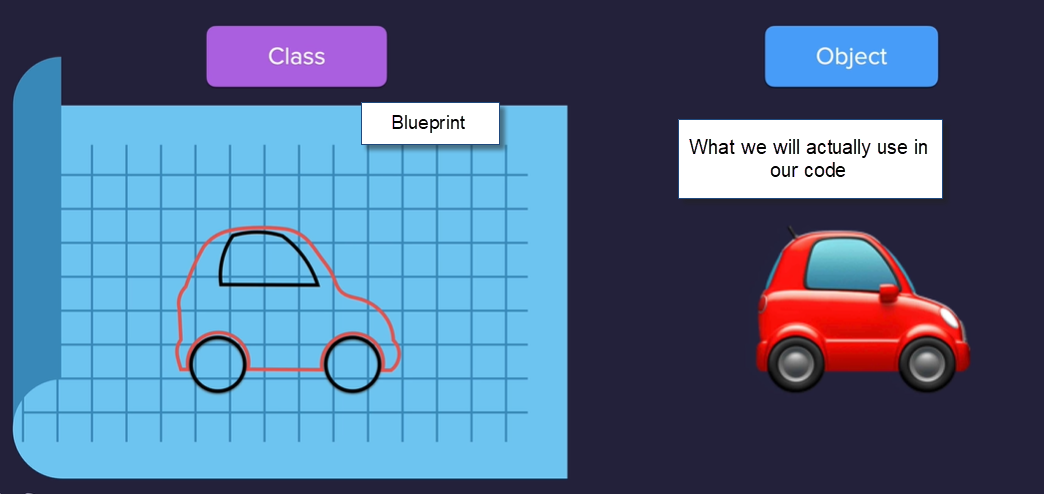
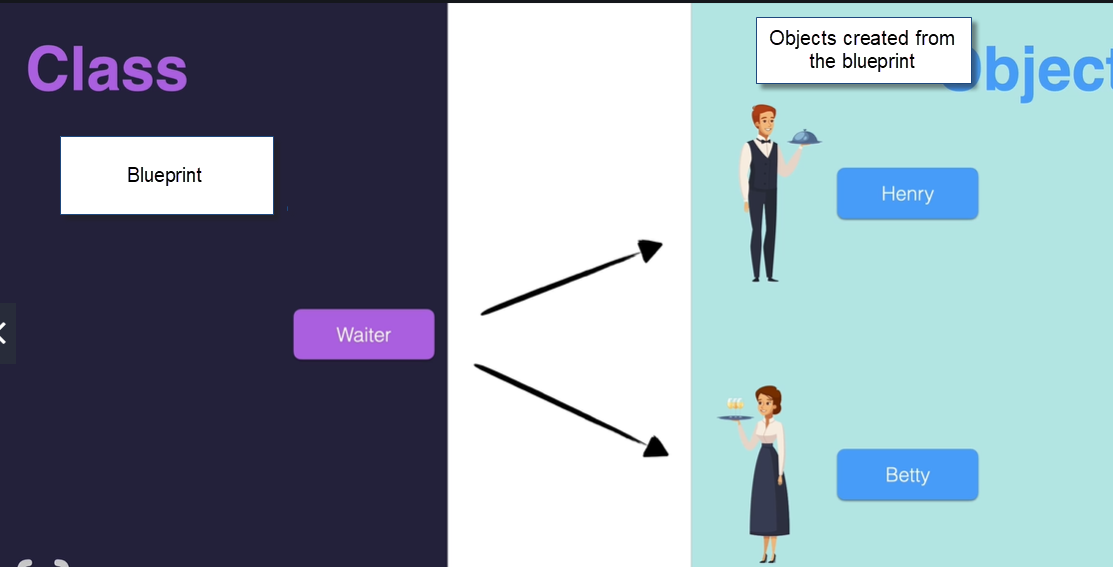


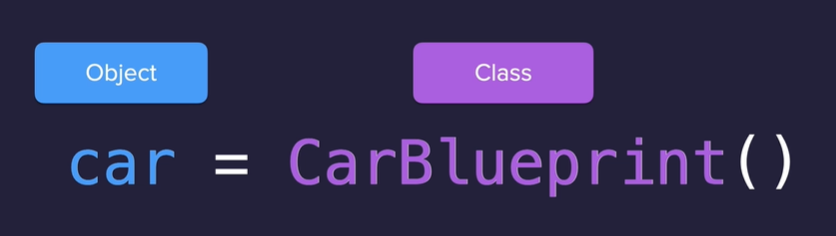
Using OOP – model a real world project – what it has and what it does – IE attributes and methods 



Data to functionality

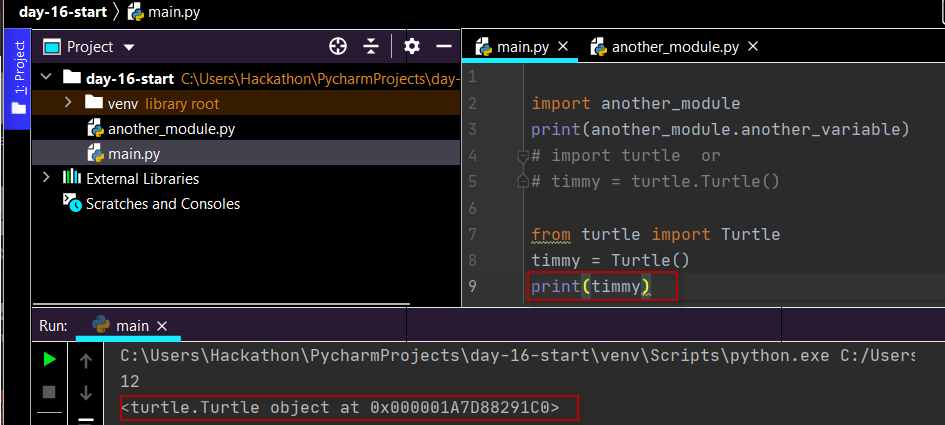
Can have multiple objects generated from the same type/blueprint – like multiple versions of a waiter

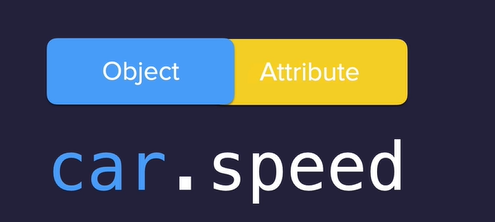
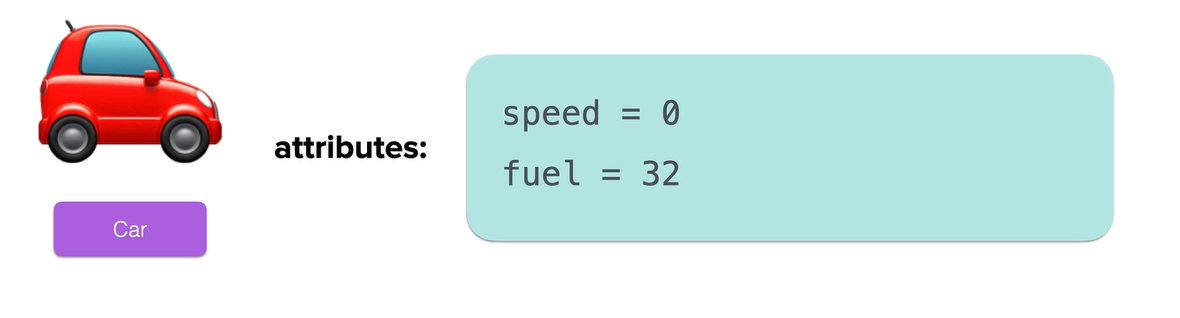


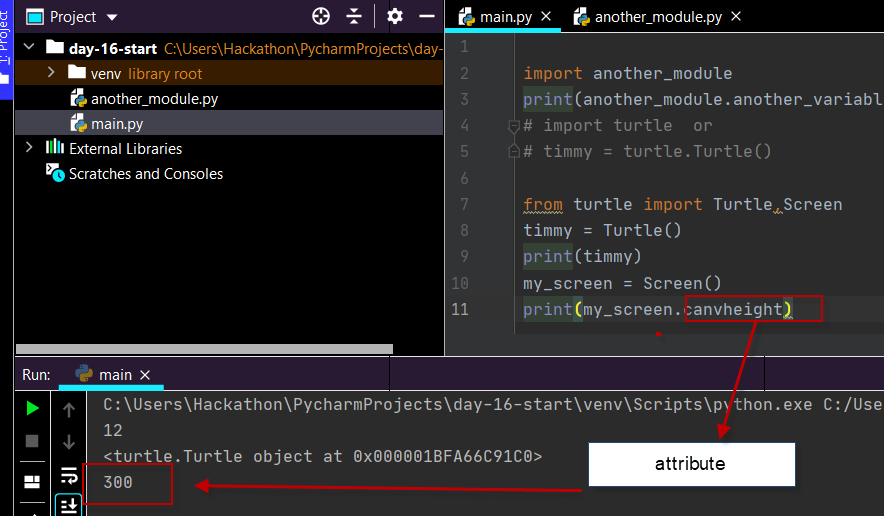
Syntax in Python 

Graphics library – Turtle Graphics python has this library out of the box



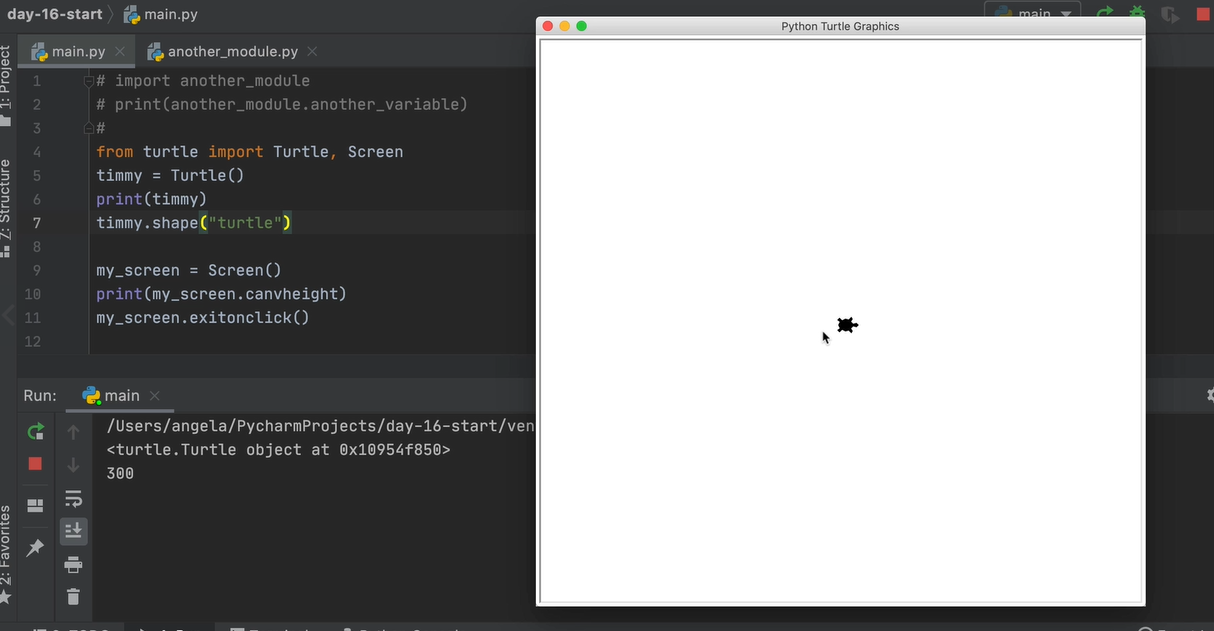






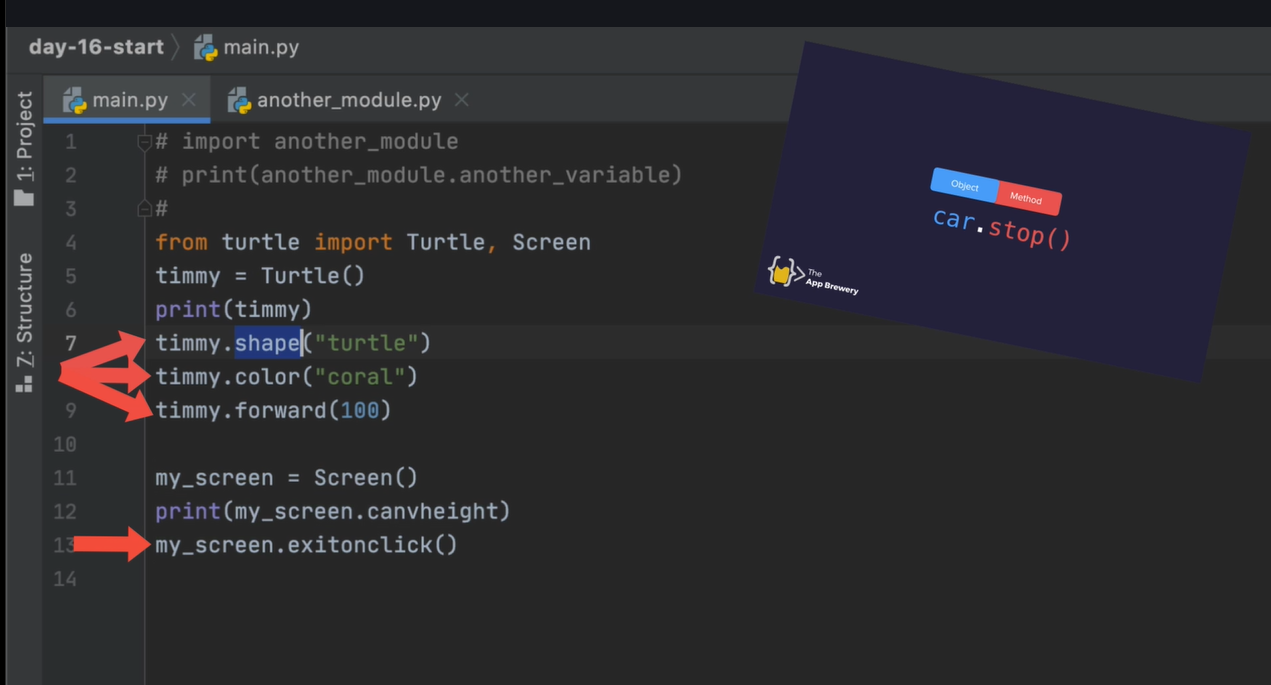
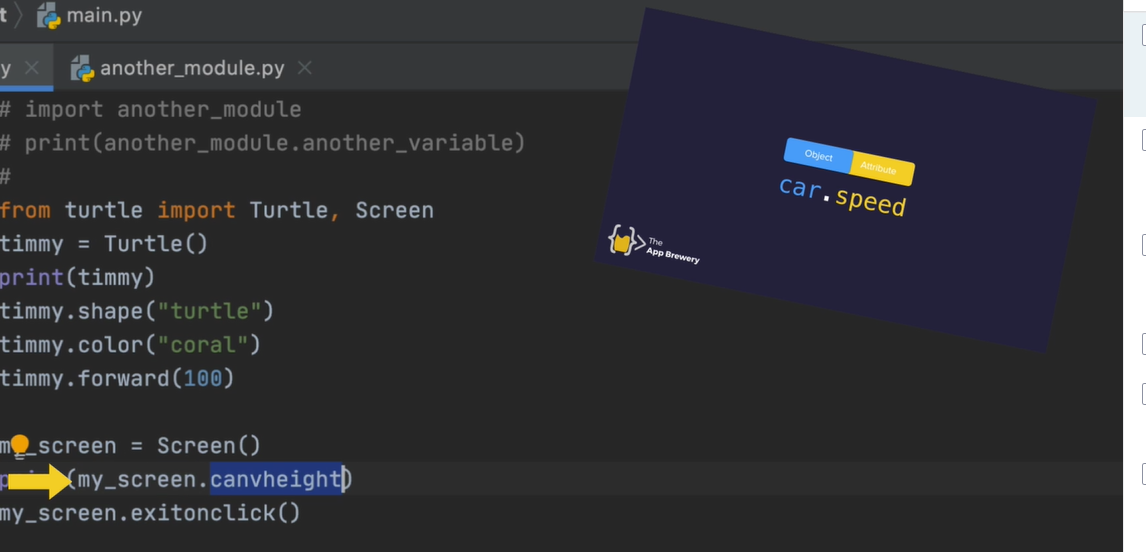
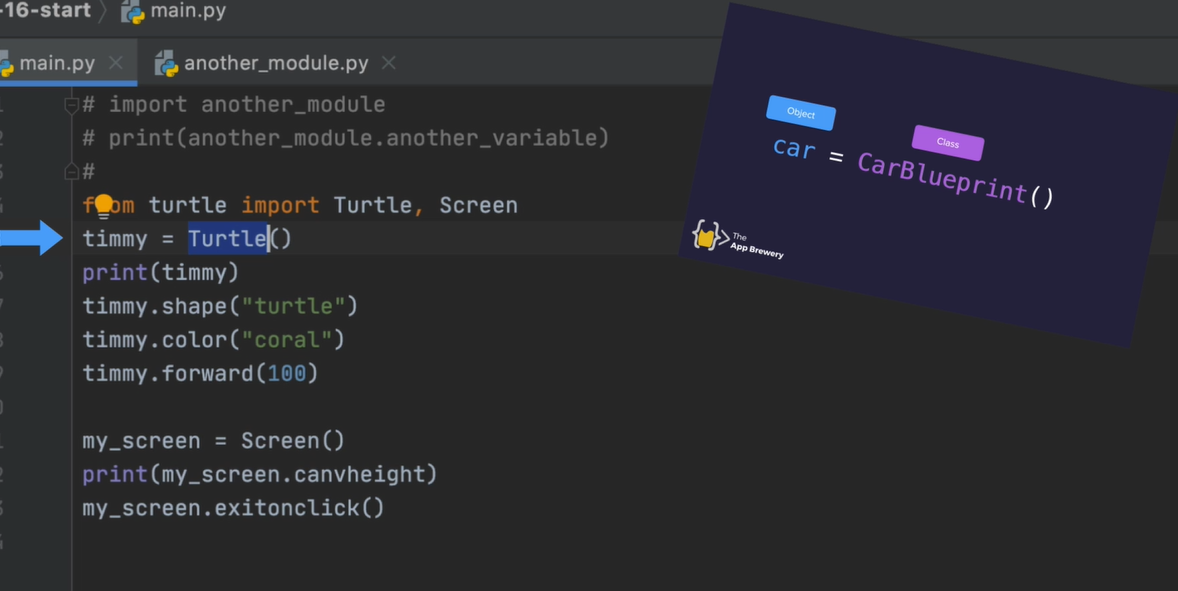
Functions





<https://cs111.wellesley.edu/labs/lab01/colors>

<https://docs.python.org/3/library/turtle.html>



C:\Users\Hackathon\PycharmProjects\day-16-start

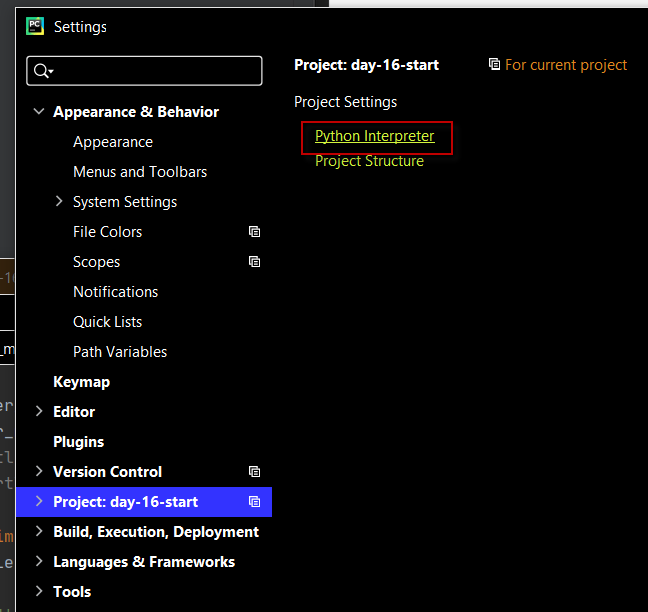
Get them at pypi.org

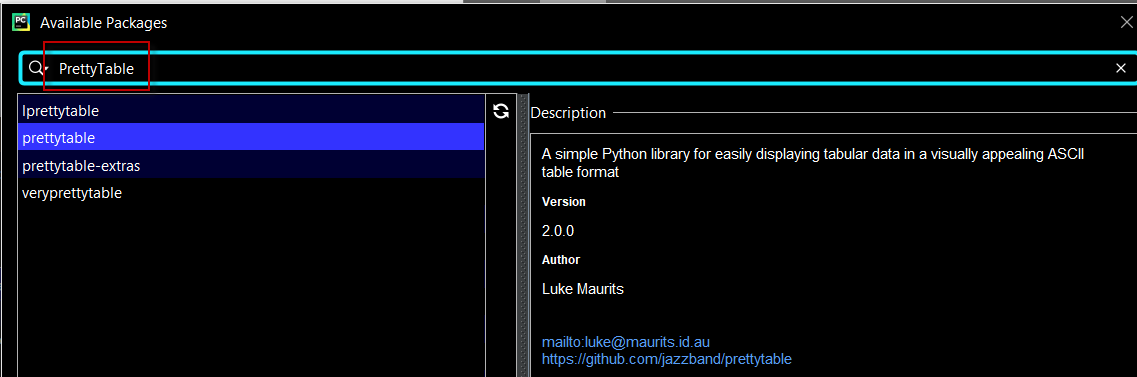
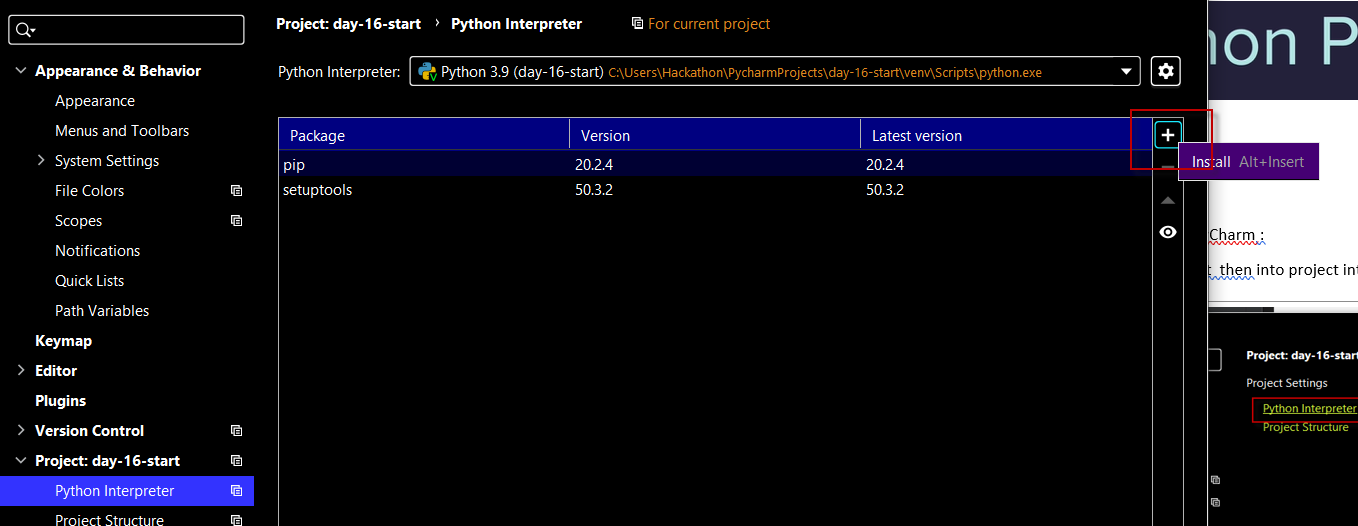
<https://pypi.org/>

PrettyTable package

Need to install to use in MyCharm :

settings >find project on list then into project interpreter





Project OOP Coffee machine

<https://docs.google.com/document/d/e/2PACX-1vTragRHILyj76AvVgpWeOlEaLBXoxPM_43SdEyffIKtOgarj42SoSAsK6LwLAdHQs2qFLGthRZds6ok/pub>

**MenuItem Class**

Attributes:

* **name**

(str) The name of the drink.

e.g. “latte”

* **cost**

(float) The price of the drink.

e.g 1.5

* **ingredients**

(dictionary) The ingredients and amounts required to make the drink.

e.g. {“water”: 100, “coffee”: 16}

**Menu Class**

Methods:

* **get\_items()**

Returns all the names of the available menu items as a concatenated string.

e.g. “latte/espresso/cappuccino”

* **find\_drink(order\_name)**

Parameter order\_name: (str) The name of the drinks order.

Searches the menu for a particular drink by name. Returns a MenuItem object if it exists, otherwise returns None.

**CoffeeMaker Class**

Methods:

* **report()**

Prints a report of all resources.

e.g.

Water: 300ml

Milk: 200ml

Coffee: 100g

* **is\_resource\_sufficient(drink)**

Parameter drink: (MenuItem) The MenuItem object to make.

Returns True when the drink order can be made, False if ingredients are insufficient.

e.g.

True

* **make\_coffee(order)**

Parameter order: (MenuItem) The MenuItem object to make.

Deducts the required ingredients from the resources.

**MoneyMachine Class**

Methods:

* **report()**

Prints the current profit

e.g.

Money: $0

* **make\_payment(cost)**

Parameter cost: (float) The cost of the drink.

Returns True when payment is accepted, or False if insufficient.

e.g. False