Advanced Data Structures.

When we start designing programs that handle (model) more complex objects we run into a challenge with only having the basic types available.

Reminder: Basic types are: int, float, boolean, String, & List.

For example say we have a csv file that contains data that we want to read into our program and use for an analysis.

What information do we want to maintain about this file? How would we keep track of this data?

Objects allow us to bundle together multiple pieces of information so that we can treat the bundle as a single unit.

Objects allow programmers to create custom data types.

As a beginner data scientist, you will mainly be using objects that other programmers have created, but eventually you will want to create your own objects.

Objects combine two things:

1) State - What an item is.

2) Behavior - What an item can do.

Example: Car

State variables could be:

- Make:
- Model:
- Color:
- Top Speed:
- Current Speed:
- Fuel Level:
- Miles per gallon:

Note: Some of these are fixed for a given car and some of these are variable

Example: Car cont.

Behaviors would include:

Start!
Stop!
Set Speed!
Brake
Turn Left
Turn Right.

One important note:

When you design a car object in code you are creating a model.

The model should only include those features that are necessary for your program

We are not trying to simulate reality.

When we create and program objects we do not build an actual object, but a template that can be used to create any number of instances.

The template is called a **class**

When we "instantiate" a class, we make an instance of that class.

We can have as many instances as we need.

Objects as a recipie.

There is an analogy that I find helpful.

A class is like a recipe - it tells us how to make a dish, but is not actually the dish.

A recipe however can be used over and over again to make the dish

The dish is an instance of the class.

The state and behavior of an object are associated with each instance of that class.

Each state variable — called an attribute —

is associated with a specific instance of that class.

The behaviors - called methods - of a class are also associated with a specific instance of the class.

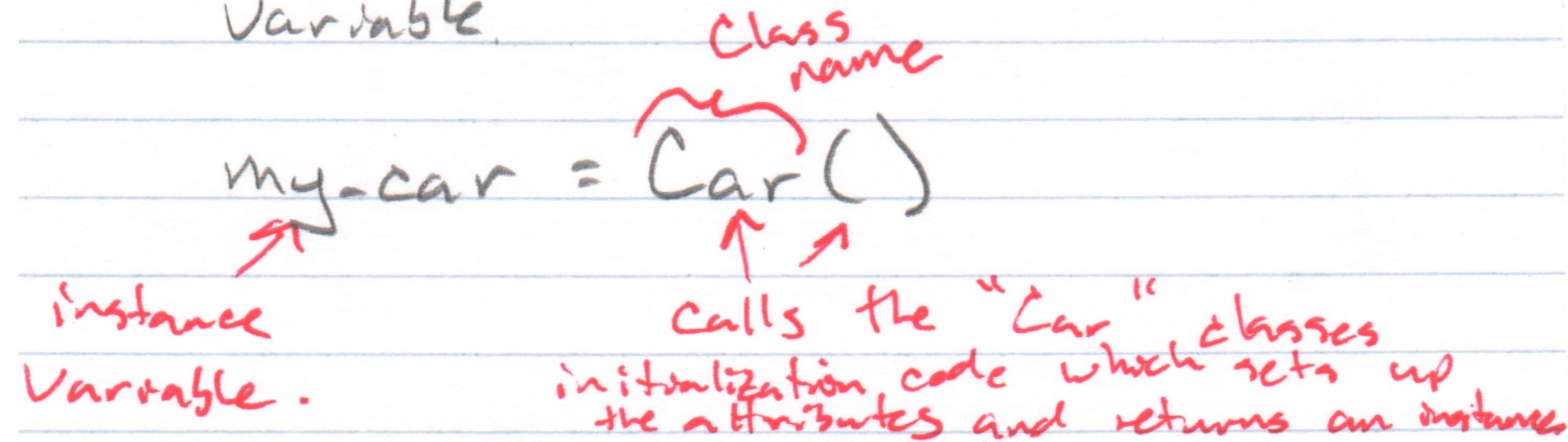
They can only modify or read attributes of their specific instance.

(Can you say "encapsulation"?)

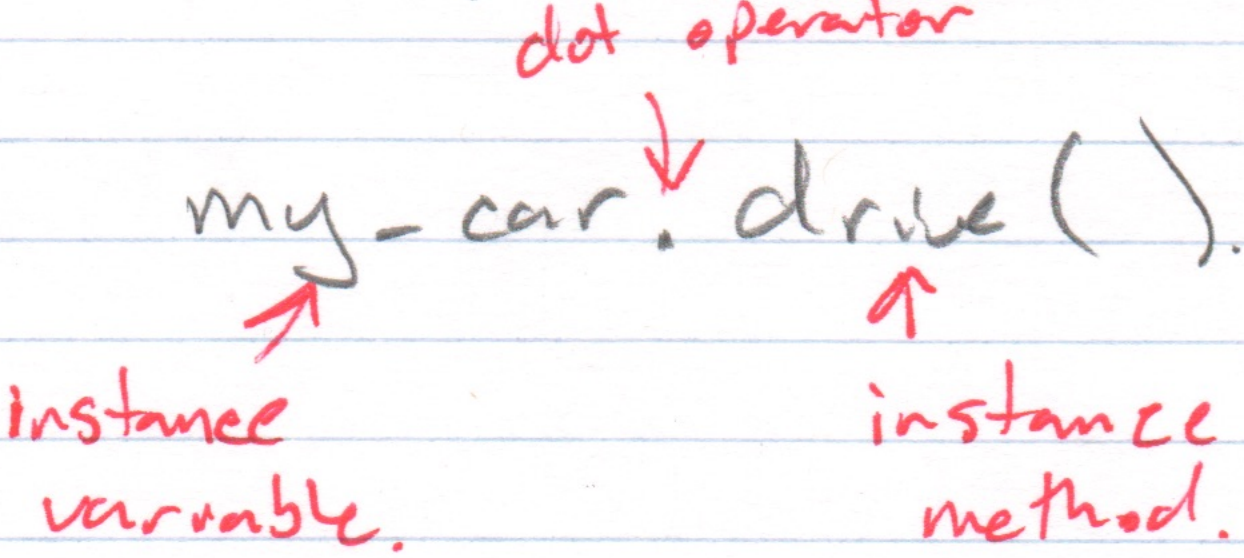
Methods are just functions that are associated with a specific of instance of the class.

They are used to read or modify the classes attributes.

To create an instance of an object you initialize the class and store it to a variable.



You call an instance's method using the dot operator.



Note: I did not call

```
Car.drive()
```

↑

Just like I don't eat a recipe. I don't call methods on the class.

* There are some rare exceptions called class or static methods

Example & Discussion

Turtle Drawing Code

Objects in Python are every where

dot operator

```
my_lst.append()
```

instance variable.

method.

In fact all of Python's basic types are technically objects!

```
greeting = str("Hello")
```

greeting: instance variable

str: initializer of the string class

```
greeting.upper()
```