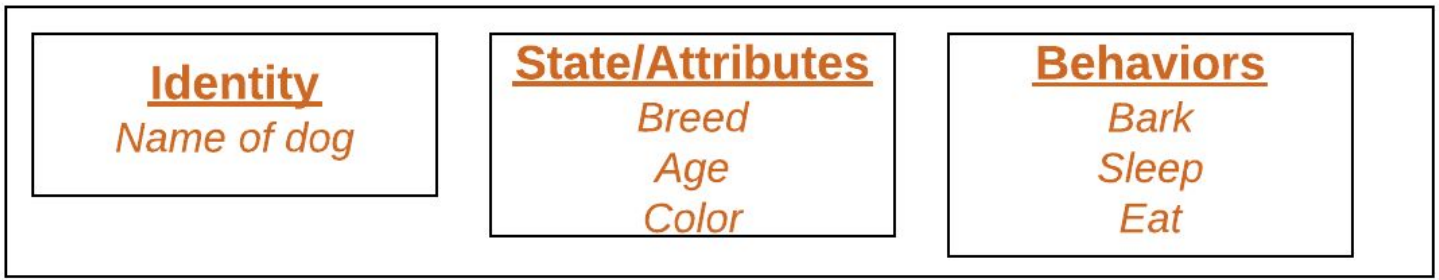
Classes - A review or Introduction

Cal Poly CSC Tutoring Center

Classes can seem hard at first

Classes just group stuff together

- Simple variables can just store numbers, strings, etc.
- Classes are just a way to group together multiple variables and functions together
- A Dog class for example might need the following:



Classes just group stuff together

```
Class Dog:
    name = None
    breed = None
    age = 0
```

Objects: Variables made using classes

```
Class Dog:
    name = None
    breed = None
    age = 0

Jeffery = Dog()
```

We call the class name to create an object of type “Dog”

Methods: functions called on objects

```
Class Dog:
    name = None
    breed = None
    age = 0

    def speak(self):
        print("Arf my name is {}".format(self.name))

Jeffery = Dog()
Jeffery.name = "Jeff"
Jeffery.speak()
```

Will print:

"Arf my name is Jeff"

Self: Refers to the object itself

```
Class Dog:
    name = None
    breed = None
    age = 0

    def speak(self):
        print("Arf my name is {}".format(self.name))

Jeffery = Dog()
Jeffery.name = "Jeff"
Jeffery.speak()
```

Print the objects own name with self

We don't need to actually include self as a parameter

Magic methods! (__ __)

- Special methods that don't get called directly
- __init__ tells us how objects are created from a class
- __repr__ tells us how our class should get printed
- __eq__ tells us how to know when two objects are equal

They are called automatically!

DON'T:

```
Fido = Dog()
print(Fido.__repr__())
```

DO:

```
Fido = Dog()
print(Fido)
```

Full Class Example

```
Class Dog:
    name = None
    age = 0

    def __init__(self, name, age):
        Self.name = name

    def __repr__(self):
        Return (“A dog named {0}”.format(name))

    def increase_age(self):
        self.age = self.age + 1
```

Let's write an eq for our Dog class:

Other represents an object that may or may not be another Dog. How do we want to compare them?

```
Class Dog:
    name = None
    age = None
    breed = None
    ...

    def __eq__(self, other):

        # put stuff here
```

```
def __eq__(self, other):
    if not isinstance(other, Dog):
        return false
    if self.name != other.name:
        return false
    elif self.breed != other.breed:
        return false
    elif self.age != other.age:
        return false
    else:
        return true
```

Let's Create a HumaneSociety class

It needs a list of all dogs currently contained. What else does it need? And how can we add new dogs?

```
class HumaneSociety:
    all_dogs = []
    ...

    def __init__(self, …?):


    # put stuff here
```

Questions?