

Note: Class Methods

- Most methods are “instance methods”
 - These are called on an instance of a class (ie, a single cat)
 - They can refer to/change attributes of that instance
- Some methods are “class methods”
 - They are called on the **class itself**
 - They can't refer to instance attributes
 - Often used as “factories” to return instances

```
class Pet(db.Model):  
  
    @classmethod  
    def get_by_species(cls, species):  
        """Get all pets matching that color."""  
  
        return cls.query.filter(Pet.species == species).all()
```

```
>>> Pet.get_by_species("dog")  
[<Pet ...>, <Pet ...>]
```

Better Representation

```
>>> Pet.query.filter(Pet.species == 'dog').all()  
[<__main__.Pet object at ...>, <__main__.Pet object at ...>]
```

Yeah, but *which* pet is that?

demo/models.py

```
class Pet(db.Model):    # ...  
    def __repr__(self):  
        """Show info about pet."""  
  
        p = self  
        return f"<Pet {p.id} {p.name} {p.species} {p.hunger}>"
```

```
>>> Pet.query.get(1)  
<Pet 1 Whiskey dog 10>
```

SQLAlchemy with Flask