Django Summaries

The ORM

Using Django's model allows us to use the ORM (Object Relational Mapper). The Django ORM is a powerful tool for interacting with the database. It allows us to perform CRUD (Create, Retrieve, Update, Delete) operations on the database, but by using our models as an interface to it.

Syntax

By default, Django's model has a property called `objects` which uses Django's `Manager` class. It is specified like this:

```
from django.db import models


class Todo(models.Model):
    name = models.CharField(max_length=100)

    objects = models.Manager()
```

That is why we are able to call `Todo.objects`

The model manager gives the functionality to perform CRUD operations on a model instance

Creating a Todo

```
Todo.objects.create(name='My todo')
```

Retrieving a Todo

```
Todo.objects.get(name='My todo')
```

Updating a Todo

```
todo = Todo.objects.get(name='My todo')
todo.name = 'A new todo'
todo.save()
```

Deleting a Todo

```
todo = Todo.objects.get(name='A new todo')
todo.delete()
```

Querysets

When we want to return multiple instances by filtering our database, Django returns a `Queryset` of data back to us. A Queryset is a Python class provided by Django. It is an iterable object which means it works the same way as a list or tuple.

The best way to think of a queryset is that it is just a list of instances of a Django model. And the list has some special properties that allow us to manipulate and extract data from it.

Here are some very basic filtering commands:

```
from .models import Todo

# get all the posts
all_todos = Todo.objects.all()  # returns a Queryset

# filter by name
query = Todo.objects.filter(name='This specific name')  # returns a
Queryset

# filter by name case insensitive
query = Todo.objects.filter(name__iexact='ThIS SpEcIfIc NAme')  # returns
a Queryset

# filter by name containing
query = Todo.objects.filter(name__icontains='specific')  # returns a
Queryset
```

Of course this is only filtering on one field. There are ways to filter every field, as well as spanning across relationships like foreignkeys.