Django Summaries

Forms

Forms are a very important topic in Django and web development as a whole.

There are two main form models provided by Django. The first is the `ModelForm`. This form is used when you want to reduce code for interacting with your models. The second form is the `Form` which is a basic form that requires you to specify all of the fields you want on the form.

Syntax

ModelForm

```python
from django import forms
from .models import Todo

class TodoForm(forms.ModelForm):
    class Meta:
        model = Todo
        fields = ['name']
```

What this does is create a form that links with our model. The form will only have one field which is for the name of the Todo.

Form

```python
from django import forms

class TodoForm(forms.Form):
    name = forms.CharField()
```

As you can see the normal Form is a lot less code but requires us to specify the exact fields we want. In this case we've added a `CharField` which accepts string values and is also a required field by default.

Working with CBV's

You can use a form in any CBV, but the views you will typically work the most with forms are the `FormView`, `CreateView` and `UpdateView`.

The `FormView` is a generic form view that simply requires the form you want to display. Whereas the `CreateView` and `UpdateView` require both a form aswell as a model or queryset. The choice of which one to use will become more natural over time. Every situation is different but ultimately you'll want the solution that is simplest, even if that is slightly more code than another alternative.

Working with forms

Like Class Based Views, forms come with their own set of logic to understand, such as all the available fields, how to validate inputted data and other methods available on the form. Most of the time you will only be working with different form fields, as well as validating form data, so don't focus on learning everything there is to know about forms. Again only learn concepts as you need them to solve a problem.