These are the commands for the follow-along exercises in Lesson 3.4.

3.402: Starting with Django forms

index.html

```
    <h2>Create EC Entry</h2>
    <a href="/create_ec/">Add a EC Entry</a>
    <br />
```

urls.html

```
     path('create_ec/', views.create_ec, name='create'),
```

views.py

```
def create_ec(request):
    Pass
```

forms.py

```
from django import forms


class ECForm(forms.Form):
    ec_name = forms.CharField(label='EC Name', max_length=100)
```

views.py

```
  from .forms import *

  def create_ec(request):
      master_genes = Gene.objects.all()
      if request.method == 'POST':
          form = ECForm(request.POST)
          if form.is_valid():
              ec = EC()
              ec.ec_name = form.cleaned_data['ec_name']
              ec.save()
              return HttpResponseRedirect('/create_ec/')
      else:
          ecs = EC.objects.all()
          form = ECForm()
      return render(request, 'genedata/ec.html', {'form': form,
'ecs': ecs, 'master_genes': master_genes})
```

ec.html

```
{% extends "./base.html" %}
{% load bootstrap4 %}

{% block content %}
<h2>Current EC Names</h2>
<table>
```

```
  <tr><th>EC Name</th></th>
    {% for ec in ecs %}
      <tr><td>{{ ec }}</td></tr>
    {% endfor %}
</table>
<br />
<h2>Add New EC Name</h2>
<form action="/create_ec/" method="post"  class="form">
    {% csrf_token %}
    {% bootstrap_form form %}
    <input type="submit" value="Submit">
</form>
{% endblock%}
```

index.html

```
    <h2>Create Gene Entry</h2>
    <a href="/create_gene/">Add Gene Entry To DB</a>
    <br />
```

3.404: More on Django forms

urls.py

```
    path('create_gene/', views.create_gene, name='create_gene'),
```

views.py

```
def create_gene(request):
    if request.method == 'POST':
        form = GeneForm(request.POST)
    else:
        master_genes = Gene.objects.all()
        form = GeneForm()
    return render(request, 'genedata/create_gene.html', {'form':
form, 'master_genes': master_genes})
```

forms.py

```
from django.forms import ModelForm
from .models import *
class GeneForm(ModelForm):
    class Meta:
        model = Gene
        fields = ['gene_id', 'entity', 'start', 'stop', 'sense',
                  'start_codon', 'sequencing', 'ec']
```

create_gene.html

```
{% extends "./base.html" %}
{% load bootstrap4 %}

{% block content %}
<h2>Add New Gene </h2>
<form action="/create_gene/" method="post" class="form">
    {% csrf_token %}
    {% bootstrap_form form %}
    <input type="submit" value="Submit">
</form>
{% endblock%}
```

views.py

```
def create_gene(request):
    if request.method == 'POST':
        form = GeneForm(request.POST)
        if form.is_valid():
            gene = form.save()
            return HttpResponseRedirect('/create_ec/')
    else:
        form = GeneForm()
        master_genes = Gene.objects.all()
    return render(request, 'genedata/create_gene.html', {'form':
form, 'master_genes': master_genes})
```

3.406: Django validators

forms.py

```
def clean(self):
    cleaned_data = super(GeneForm, self).clean()
    entity = cleaned_data.get("entity")
    sense = cleaned_data.get("sense")

    if not entity == "Chromosome" and not entity == "Plasmid":
        raise forms.ValidationError("Entity must be 'Chromosome'
or 'Plasmid'")
    if not sense == "+" and not sense == "-":
        raise forms.ValidationError("Sense must be '+' or '-'")
    return(cleaned_data)
```

views.py

```
else:
    return render(request, 'genedata/create_gene.html', {'error':
"failed", 'master_genes': master_genes, 'form': form})
```