# ----- DJANGO TUTORIAL 4 -----

```
# We Cover CSS Styling, Static Files, Adding Detail Pages and Linking to Them, Updating
Models, Working with Images and More

# 1. Create directory for css files
mkdir project_3/static/css

# 2. Add base.css to static/css
* {
  margin: 0;
  padding: 0;
}
body {
  background-color: gray;
  font-size: 62.5%;
  padding: 0;
  margin: 0;
}
.wrapper {
  width: 900px;
  font-size: 1.6em;
  margin: 0 auto;
  background-color: white;
}
p {
  padding: 0 0 1em 0;
}
header {
  text-align: right;
  background-color: brown;
  color: white;
  padding: 2em;
}
main {
  padding: 2em;
  position: relative;
}
.art-img {
  float: left;
  margin: 15px;
  height: 100px;
  width: 100px;
}
h3{
  margin-bottom: 1em;
}

# 3. Create base.html in templates
{% load static %}
<!DOCTYPE html>
<html lang="en">
  <head>
  <title>Django Blog</title>
  <link href="{% static 'css/base.css' %}" rel="stylesheet">
  </head>
  <body>
    <div class="wrapper">
  {% block content %}
  {% endblock content %}
    </div> <!-- End of wrapper -->
  </body>
</html>
```

```
# 4. Update home.html
{% extends 'base.html' %}
{% block content %}
<main>
  {% for article in all_articles_list %}
  <section>
    <h3>{{ article.title }}</h3>
    <article>
      <p>{{ article.text|linebreaks }}</p>
    </article>
  </section>
{% endfor %}
</main>
{% endblock content %}
```

```
# 5. Update settings with static info
STATIC_ROOT = ''
STATIC_URL = '/static/'
STATICFILES_DIRS = (os.path.join('static'),)
```

```
# 6. Add individual article pages by adding DetailView to articles/views.py
from django.views.generic import ListView, DetailView
from .models import Article

class HomePageView(ListView):
    model = Article
    template_name = 'home.html'
    context_object_name = 'all_articles_list'

class ArticleDetailView(DetailView):
    model = Article
    template_name = 'article_detail.html'
```

```
# 7. Create article_detail.html to templates
{% extends 'base.html' %}

{% block content %}
<main>
  <section>
    <h3>{{ article.title }}</h3>
    <article>
      <p>{{ article.text|linebreaks }}</p>
    </article>
  </section>
</main>
{% endblock content %}

# 8. Add the article detail url to articles/urls
from django.urls import path
from .views import HomePageView, ArticleDetailView
urlpatterns = [
path('article/<int:pk>/', ArticleDetailView.as_view(), name='article_detail'),
path('', HomePageView.as_view(), name='home'),
]

# 9. Update home.html so you can click on article titles and go to the detail page
<main>
  {% for article in all_articles_list %}
  <section>
    <h3><a href="{% url 'article_detail' article.pk %}">{{ article.title }}</a></h3>
    <article>



# 10. Update models.py to have a author and foreign key
from django.db import models

class Article(models.Model):
    title = models.CharField(max_length=200)
    text = models.TextField()
    # Define a foreign key so we can define that
    # an author can have many posts
    # We'll identify users using the built in
    # authentication system
    # When defining a many to one relationship we
    # must define on_delete
    author = models.ForeignKey(
    'auth.User',
    # When this object is deleted also delete all
    # references to it
    on_delete = models.CASCADE,
    default = 1,
    )
    # Returns the article title when the object
    # is printed
    def __str__(self):
        return self.title

# 11. Migrate our model to the DB
Ctrl+C
python manage.py makemigrations articles

# a. When you get 1) Provide a one-off default now provide 1

python manage.py migrate articles
```

```
# 12. Create a media folder, put images in it and update settings
mkdir /media
Add images to it

# Add this to settings
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
MEDIA_URL = '/media/'

# 13. Update project_3/urls.py
from django.contrib import admin
from django.urls import path, include
from . import settings
from django.contrib.staticfiles.urls import static
from django.contrib.staticfiles.urls import staticfiles_urlpatterns

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('articles.urls')),
]

urlpatterns += staticfiles_urlpatterns()
urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
# 14. Install Pillow so we can use images in our model
Ctrl + C
exit
pipenv install pillow==7.0.0
pipenv shell

# 15. Add image to the model.py

# Will hold photos for articles
photo = models.ImageField(upload_to="gallery", default = 'drop-bear.png')

# 16. Create migrations
python manage.py makemigrations articles
python manage.py migrate articles

# 17. Update home.html
<main>
  {% for article in all_articles_list %}
  <section>
    <h3><a href="{% url 'article_detail' article.pk %}">{{ article.title }}</a></h3>
    <article>
      <img src="{{ article.photo.url}}" alt="{{ article.title }}"
      class="art-img">
      <p>{{ article.text|linebreaks }}</p>
    </article>
  </section>
{% endfor %}
</main>

# 18. Resize images in base.css
.art-img {
  float: left;
  margin: 15px;
  height: 100px;
  width: 100px;
}
```

# 19. Go into admin and change the jackalope image

# 20. Update article_detail.html

```
<main>
  <section>
    <h3>{{ article.title }}</h3>
    <article>
      <img src="{{ article.photo.url}}" alt="{{ article.title }}" class="art-img">
      <p>{{ article.text|linebreaks }}</p>
    </article>
  </section>
</main>
```