


# Exercise 21: Adding Static Files to Django Jokes

 20 to 30 minutes

In this exercise, you will add the following static files to the Django Jokes site:

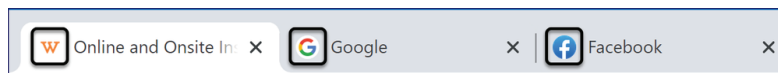
 favicon.png

 logo.png

 main.css

## favicon

A *favicon* is a special file used by browsers, usually on the tab, as a logo-like identifier for the website:



## ❖ E21.1. settings.py

Open `djangojokes.com/djangojokes/settings.py` in your editor. Locate the `STATIC_URL` line at the end of the file and add a `STATICFILES_DIRS` setting below it:

```
STATIC_URL = '/static/'
```

```
STATICFILES_DIRS = [  
    BASE_DIR / 'static',  
]
```

## ❖ E21.2. Images

1. Create a `static` directory in the `djangojokes.com` folder, and create an `images` directory within that.

2. Add `favicon.png` and `logo.png` images to the `images` directory. You can create your own images or use the ones located in the `starter-code/static-files` folder.<sup>37</sup>
3. Open `templates/_base.html` in your editor.

A. At the very top of the document, load the `static` tag:

```
{% load static %}
```

B. Add the following `link` element immediately before the `title` element in the head:

```
<link rel="shortcut icon" type="image/png"
      href="{% static 'images/favicon.png' %}">
```

C. Replace the “Django Jokes” text within the `navbar-brand` link with an image:

```
<a class="navbar-brand" href="{% url 'homepage' %}">
  
</a>
```

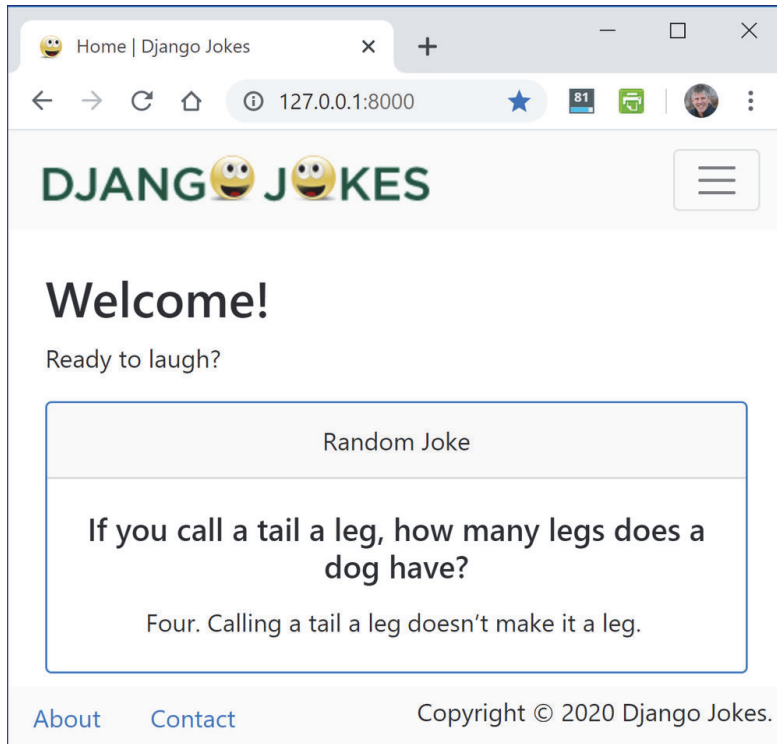
Start the server by opening `djangojokes.com` in the terminal and running:

```
(.venv) .../projects/djangojokes.com> python manage.py runserver
```

Then, point your browser to `http://127.0.0.1:8000/`. The page should look similar to this:

---

37. The [https://commons.wikimedia.org/wiki/File:Gaim\\_balloon-free.svg](https://commons.wikimedia.org/wiki/File:Gaim_balloon-free.svg) image is in the public domain ([https://commons.wikimedia.org/wiki/Public\\_domain](https://commons.wikimedia.org/wiki/Public_domain)).



In addition to the logo in the upper left, notice the favicon image on the tab:



### ❖ E21.3. CSS File

1. Within the `static` folder, create a `css` folder.
2. Create a `main.css` file within the `css` folder with the following content, which you can copy from `starter-code/static-files/main.css`.

## Exercise Code 21.1: static/css/main.css

---

```
1.  body {
2.    padding-bottom: 50px;
3.  }
4.
5.  footer {
6.    border-top: 1px solid darkgreen;
7.  }
8.
9.  header {
10.   border-bottom: 1px solid darkgreen;
11. }
12.
13. h2 {
14.   font-style: italic;
15. }
```

---

If you like, you may add some of your own styles.

3. In `templates/_base.html`, add a `<link>` tag below the existing Bootstrap `<link>` tag to include `main.css`:

```
<link rel="stylesheet" href="{% static 'css/main.css' %}">
```

4. Return to the browser and refresh the page to see the new styles you added.

## Git Commit

Commit your code to Git.



## 4.2. Review of the Settings

The settings for static files can be confusing. Let's review the difference between `STATIC_URL` and `STATICFILES_DIRS`.

### ❖ 4.2.1. `STATIC_URL`

Remember that `STATIC_URL` is set to `'/static/'`. While this also happens to be the name of the folder that holds static files, there is no connection between the `STATIC_URL` setting and the actual

location of static files. Rather, this is the URL pattern that tells Django to look for static files. Let's prove this:

1. With the Django Jokes site running, go to the following URLs in your browser:
  - A. `http://127.0.0.1:8000/static/css/main.css` – The CSS page should load.
  - B. `http://127.0.0.1:8000/static/images/logo.png` – The logo should load.
  - C. `http://127.0.0.1:8000/static/images/favicon.png` – The favicon should load.
2. Return to your editor and modify the `STATIC_URL` setting in `settings.py` as follows:

```
STATIC_URL = '/foo/'
```

3. Back in your browser, visit the following URLs. The files should still load:
  - A. `http://127.0.0.1:8000/foo/css/main.css`
  - B. `http://127.0.0.1:8000/foo/images/logo.png`
  - C. `http://127.0.0.1:8000/foo/images/favicon.png`
4. Visit the home page at `http://127.0.0.1:8000/` and notice that it works just as before.
5. Right-click the home page, select **View (or Show) page source**, and look for these tags:

```
<link rel="stylesheet" href="/foo/css/main.css">

<link rel="shortcut icon" type="image/png" href="/foo/images/favicon.png">


```

Change `STATIC_URL` back to  `'/static/'` as that is the standard practice for Django development:

```
STATIC_URL = '/static/'
```

Again, the `STATIC_URL` setting sets the URL pattern, not the location of static files.

## ❖ 4.2.2. STATICFILES\_DIRS

The `STATICFILES_DIRS` setting, on the other hand, does have to do with where static files are stored. Again, the easiest way to see this is to make a quick change:

1. Modify the `STATICFILES_DIRS` setting in `settings.py` as follows:

```
STATICFILES_DIRS = [  
    BASE_DIR / 'bar',  
]
```

2. Point your browser to `http://127.0.0.1:8000/` and refresh the page. The logo and favicon images and the styles you added in `main.css` should disappear. If they don't, try doing a hard refresh by holding the **Shift** key down and pressing the browser's refresh button.
3. Change the name of the `static` folder to `bar`. In Visual Studio Code, you can do this by right-clicking the folder and selecting **Rename**.
4. Return to your browser and refresh the page (again, you may need to do a hard refresh). The logo and favicon images should now show up and your styles should be back.

This illustrates that the `STATICFILES_DIRS` setting tells Django where to look for static files.

Change the name of the `bar` folder back to `static` and `STATICFILES_DIRS` back to its original setting:

```
STATICFILES_DIRS = [  
    BASE_DIR / 'static',  
]
```

## Conclusion

In this lesson, you have learned to add static files to your Django web pages.