

Adding HTML Content to Django

 dj4e.com/assn/dj4e_html.md

Serving HTML Content

Make two folders

```
mkdir ~/django_projects/mysite/site
mkdir ~/django_projects/mysite/site/subfolder
```

Create a file at `~/django_projects/mysite/site/hello.txt` with the text "Hello World".

Create a file at `~/django_projects/mysite/site/subfolder/hello.html` with this text:

```
<h1>Hello World</h1>
```

Change your `~/django_projects/mysite/mysite/urls.py` to be:

```
import os
from django.contrib import admin
from django.urls import include, path
from django.conf.urls import url
from django.views.static import serve

# Up two folders to serve "site" content
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
SITE_ROOT = os.path.join(BASE_DIR, 'site')

urlpatterns = [
    path('admin/', admin.site.urls),
    path('polls/', include('polls.urls')),
    url(r'^site/(?P<path>.*)$', serve,
        {'document_root': SITE_ROOT, 'show_indexes': True},
        name='site_path'
    ),
]
```

Going forward we will be adding entries to this `urlpatterns` variable as we add new features. As you do upcoming assignments, do not remove these entries from your `urls.py`. Just add the new url pattern entries as required by the upcoming assignments.

Once you have made the changes, you should check for errors using:

```
cd ~/django_projects/mysite
python3 manage.py check
```

If the `check` fails, stop and fix any and all errors before continuing.

Once `check` succeeds, you can go to the `Web` tab on PythonAnywhere, reload your application and then check your application by navigating to:

`(your-account).pythonanywhere.com`

Viewing Your New Files

Navigate to your top level page **`(your-account).pythonanywhere.com`** with no path and you should see an error page like [this](#). This is Django's way of letting you know that you have requested a url that has no route and so it is returning a `404 Not found` error. But since you have `DEBUG = True` in your `settings.py` it is giving you some additional detail which will prove very helpful to you as a developer trying to figure out why your site is not working as you expect.

You will see the same error if you go to some random URL that does not exist like **`(your-account).pythonanywhere.com/xyzzzy`** and it should look like [this](#)

In a later assignment, we will add a route for the main path (i.e. no path) so users can visit your site at the top level.

Next test the ability to serve the `site` content.

Go to **`(your-account).pythonanywhere.com/site`** - you should see a list of files including your `hello.txt` (like [this](#)). Click on `hello.txt` on your site and you should see "Hello world".

Go to **`(your-account).pythonanywhere.com/site/subfolder/hello.html`** - you should see "Hello World" styled using a HTML header tag (like [this](#))

Building Some Validated HTML

Create a web page in a file named `dj4e.htm` and store it in the `site` folder according to these specifications.

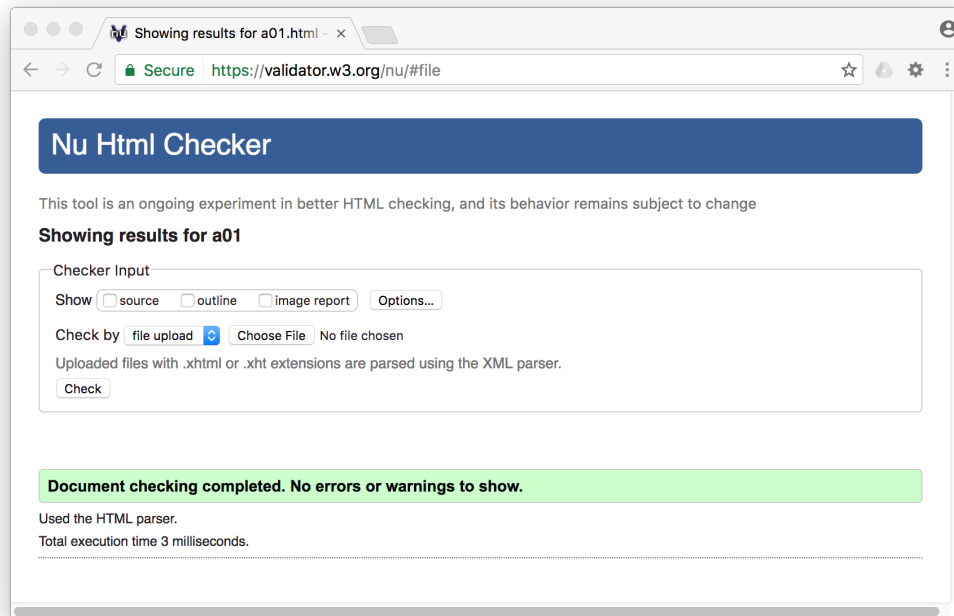
Your page will be well-formed HTML5, and indicate that it is in the UTF-8 character set.

```
<!DOCTYPE html>
<html>
<head>
  <title>Jane Instructor 4c56ff</title>
  <meta charset="UTF-8">
</head>
<body>
  ...
</body>
</html>
```

Your name and code will be different from this example.

Your HTML must pass the validator at:

<https://validator.w3.org/nu/>



The **title** tag must contain your name or the string indicated by the autograder.

Welcome Sue Student from Introduction to Programming

Make sure that your code passes the [W3C HTML Validator](#) before submitting this assignment.

Make sure your title tag in the HTML contains this:

```
<title> ... Sue Student ... </title>
```

Please upload your file containing the HTML.

No file chosen

You must have at least one example of each of the following tags: **span**, **p**, **div**, **h1**, and **ul**

You must have three **a** tags and three **li** tags.

You must have some bold text in the document and some italics text in the document but you must use the correct tags (i.e. neither **b** nor **i** are allowed).

There is a lot of flexibility within those parameters.

Common Errors

If your application fails to load or reload, you might get an error message that looks like [this](#).

If you get an error, you will need to look through the error logs under the `web` tab on PythonAnywhere:

Log files:

The first place to look if something goes wrong.

Access log: drchuck.pythonanywhere.com.access.log

Error log: drchuck.pythonanywhere.com.error.log

Server log: drchuck.pythonanywhere.com.server.log

Log files are periodically rotated. You can find old logs here: [/var/log](#)

First check the `error` log and then check the `server` log. Make sure to scroll through the logs to the end to find the latest error.

Copyright Creative Commons Attribution 3.0