PythonAnywhere and Django Tutorials

Running Django Project Tutorials on PythonAnywhere

The recommendation is to read this document completely before starting and also read the text from the tutorial completely before starting to write code.

You will encounter a lot of problems if you jump right in and start cutting and pasting.

The rest of theis document is here to help you with problems you might encounter while building the tutorial on PythonAnywhere.

- What happens when you don't activate your virtual environment (django42)
- Your line 16 seems to have a syntax error in the text editor in `manage.py`
- What to do when you see 'SyntaxError' when running `manage.py`
- What to do when the tutorial tells you to do a `python manage.py runserver`
- What to do when the tutorial tells you to access 'localhost:8000'
- How and when you exit the Django shell (>>> prompt)
- When everything works but your application says 'Something went wrong :-('

You *must* be in your virtual environment

If you are not in your virtual environment in a bash shell on PythonAnywhere lots of things will fail. You will not have access to the correct version of Python and you will not have a proper installation of Django. You can always check which python you are running using the `--version` option.

In the example below we are running Python 2.7.12 (bad) without the virtual environment and once we activate into the virtual environment we are using Python 3.8 and Django 3.1.

```
17:33 ~ $ python --version
Python 2.7.12
17:33 ~ $ python -m django --version
1.11.26
17:33 ~ $ workon django42
(django42) 17:33 ~ $ python --version
Python 3.9.5
(django42) 17:36 ~ $ python -m django --version
4.2.7
(django42) 17:33 ~ $
```

Each time you start a new bash shell, you need to type `workon django42`. If you leave and come back to a shell that is still running, if you see the '(django42)' in your prompt - you do not have to re-run the `workon` command. It just needs to be done once per shell.

There are several errors that you might get if your virtual environment is not activated:

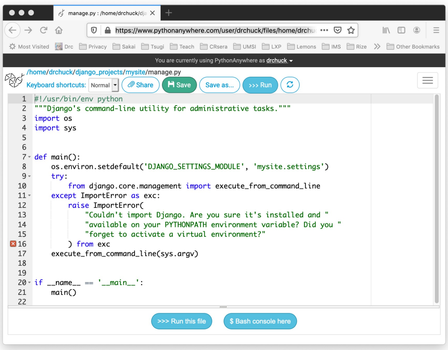
- TypeError: argument of type 'PosixPath' is not iterable

The problem with these errors is that you think you are supposed to edit files like `manage.py` or `settings.py` but the files are already just fine, you are just using the wrong versions of software.

Your manage.py looks incorrect in the PythonAnywhere editor

First and most important - you should not ever change the contents of the `manage.py` file. The file is built by Django for you and does not need to be changed.

However, if you open `manage.py` in the PythonAnywhere text editor even to look at it, it will show a little red "X" on line 16 or 17 indicating a syntax error.



The problem (much like the virtual environment problem) is the wrong version of Python. The text editor is looking at the file as a Python 2.x file and not as a Python 3.x file. So the file shows a syntax error.

Do not fix this error.

If the error really bothers you, edit the first line of the file and change `python` to `python3` and save the file. The error will magically disappear. Do not make any other changes to the file.

If you came here after you tried to fix the syntax error and made it worse, simply look closely at every line of the correct file and fix your file. Usually the problem is indentation, a line has been deleted or something was moved around.

If you see a `SyntaxError` in 'manage.py' in the shell

If the `check` identifies errors, do not go on to the rest of the assignment once you can run `check` and there are no errors. If you see this error:

```
python manage.py check
  File "manage.py", line 17
    ) from exc
      ^
SyntaxError: invalid syntax
```

Do *not* edit your `manage.py` file - the problem is never in that file.

There are several possible reasons for this:

- Check your virtual environment (above)
- Make sure you have not edited you `manage.py` (above)

Don't use `runserver` on PythonAnywhere

Just as a note, you *never* run the `runserver` command on PythonAnywhere. Often the tutorials have you make a bunch of changes to yout files and then tell you to do:

```
python manage.py runserver
```

This functionality is replaced by the "reload" button on your Web tab. So if you are reading any Django instructions that say to do a `runserver`, instead do a

```
python manage.py check
```

And then reload the application in the PythonAnywhere Web tab. There is a short cut to reload your application on the PythonAnywhere file editor. It is a little icon in the upper right of the editor that reloads the application. Usually you save the file, press reload and then check if your application worked.

Don't Use `localhost` URLs on PythonAnywhere

Usually, right after the tutorial tells you to `python manage.py runserver` it tells you to navigate to a url that looks like:

```
http://127.0.0.1:8000/
```

Once you have reloaded your application you need to go to the URL that PythonAnywhere has assigned to your application.

```
http://(your-account).pythonanywhere.com/
http://drchuck.pythonanywhere.com/
```

If the tutorial tells you to go to a URL like

```
http://127.0.0.1:8000/polls
```

Add the `polls` to your URL.

```
http://drchuck.pythonanywhere.com/polls
```

You do not need to add `django_projects` or `mysite` to your URL - this is all captured in the settings under the 'Web' tab in the PythonAnywhere user interface.

How and when you exit the Django shell

In tutorial 2, you edit `models.py` and run the Django Shell, then you edit the `models.py` file again and then run the shell again. What the tutorial does not mention is the need to exit and restart the shell any time you change `models.py`. The tutorial tells you to run the shell again but it does not tell you to exit the existing shell first - so you might see an *error* like this:

```
(django42) 17:16 ~/django_projects/mysite (master)$ python manage.py shell
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> # Do some django shell stuff

>>> python manage.py shell
File "<console>", line 1
    python manage.py shell
           ^
SyntaxError: invalid syntax
>>>
```

The correct way is to exit the shell and restart it.

```
(django42) 17:20 ~/django_projects/mysite (master)$ python manage.py shell
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> # Do some django shell stuff

>>> quit()
(django42) 17:20 ~/django_projects/mysite (master)$
```

Then you edit your `models.py` and *re-start* the Django shell from the `bash` console:

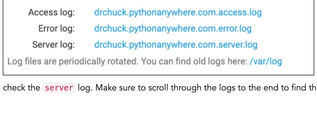
```
(django42) 17:24 ~/django_projects/mysite (master)$ python manage.py shell
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> # Do some more django shell stuff

>>> quit()
(django42) 17:20 ~/django_projects/mysite (master)$
```

After a while you will understand that you need to be in `bash` (dollar sign prompt) to run bash commands and be in the Django shell (>>> prompt) to run Django commands.

When everything works but your application says 'Something went wrong :-('

If your application passed a `check` but 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:



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