Postgres Database setup

For mac only! (Postgres is not for mac only, but this setup is for mac only.)

Get homebrew:
Go to this link: https://brew.sh/
Copy the one line of code that they have on that page, which is this:

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Open your Terminal and paste that entire line in. It will run for a while.
It may prompt you for your computer password (or it may not).

Yay!

OK.

Database setup: Postgres — use the Homebrew package manager to install the postgresql database.

```
brew install postgres
```

Then run this to start the database server: *(note: you will also have to run this any time you get an error that says your server isn't running. That's OK! Just run this again at the command prompt)*

```
pg_ctl -D /usr/local/var/postgres start
```

after you do this, you may need to open a new Terminal window. But all is well! This server will run in the background.

You may need to run that command again in the future — any time you restart your computer and want to use Postgres again (*or sometimes the server seems to randomly shut down. No harm done — but need to start it again before relying on a Postgres database! So you'll want to keep that line close. I always have to look it up myself*).

You should run the following command @ any command prompt, which will let you set up the database so you as a user can do stuff in the interface

```
createdb
```

Literally just that! This creates a database associated with *you* as the user of your computer, so that you are able to create more databases and use the command line interface for postgres. You only have to do this *once*, during the setup process.

Useful commands for starting/stopping psql db server (in general) — you may want to refer to these in future:

```
pg_ctl -D /usr/local/var/postgres status # to check status
pg_ctl -D /usr/local/var/postgres start # to start the server
pg_ctl -D /usr/local/var/postgres stop # to stop the server...
```

If you restart your computer, for example, you'll need to run that ^ **start** command to start up the DB server (this does what pressing the Servers On button on MAMP, as shown in the other instructions, does).

You generally won't need to stop the server yourself, it runs in the background, but you can by running that respective command.

Useful if you have used an unusual way of installing postgresql, or for some reason already have the APPLICATION installed (a different method of installing PostgreSQL than we are recommending here):
If you are using the Postgres APP, it is a button to start the server / stop the server, not a command on the command line. (I generally do not recommend the postgres application as a way of using this database, but it can be used if it is working for you. It sometimes introduces annoying complications.)

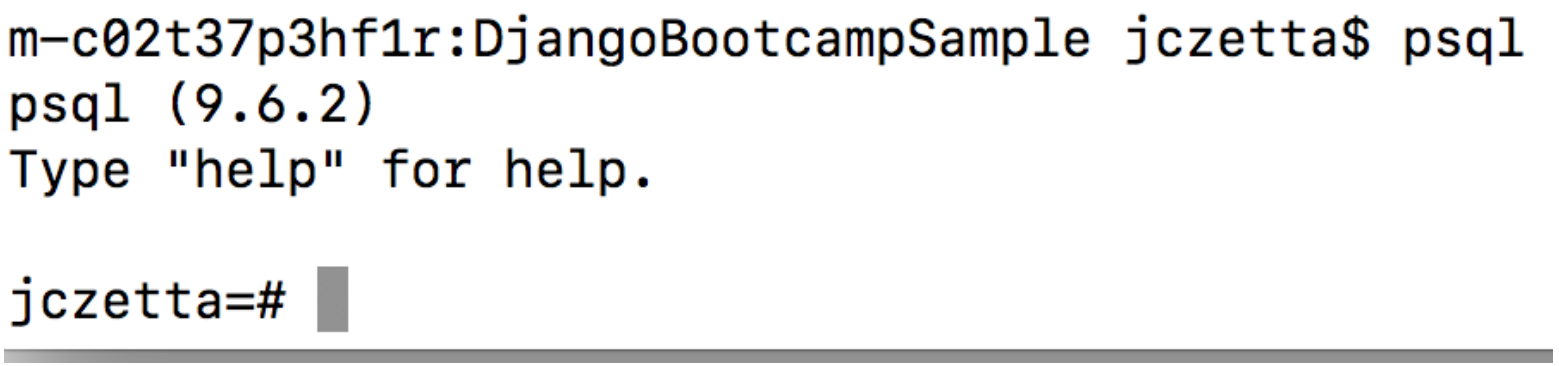
After you run one of those commands in a Terminal window/tab, you'll need to open a new Terminal window/tab to do something different in the command prompt.

Commands to interact with databases in Postgres via the command prompt:

@ command prompt to start command line interaction with postgresql

```
psql
```

If you do that, you'll now see a different prompt, something like this:

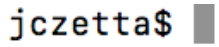


https://www.dropbox.com/s/0a0xsa99fnz7g7w/Screenshot%202017-04-30%2019.51.16.png?dl=0
(The name will be your computer user name, which is the default — for postgres to open with access to the main user db, which you created above when you ran `createdb` .)

You can always type `\q` to quit out of this and get back to the command prompt you are used to.

Now, create a database for your project — whatever you want to call the database, I recommend something simple.

You should quit out, as described above, of the psql-interface and should just be able to interact directly with the command line as usual:



Plain Terminal window ready for me to type something

https://www.dropbox.com/s/izwafcxikbfpeef/Screenshot%202018-02-01%2016.00.15.png?dl=0

Here is where Windows users may jump in, too:

Type this directly at the command prompt as usual — *not* what you get after typing psql. Just at the `$` prompt:

```
createdb sampledb_name_here
```

That creates a brand new database, stored in the Postgres Server place (so you can't see it the way you could e.g. a .txt file), called *sampledb_name_here*, in this case.

To access this new database and deal with it in the command line if you should want to

```
psql sample_db_name_here
```

Depending on what you already have installed, you also may need to

```
conda install psycopg2
```

or

```
pip install psycopg2 / pip3 install psychopg2
```

That is the Python library that allows you to interact with PostgreSQL databases in a Python script.

Everything else (probably) will come with the web development setup you're using...