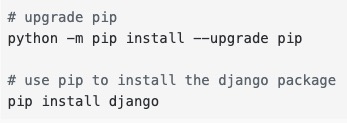
Project Setup:

* Make a working directory: Andy’s Seafood
* cd into Andy’s Seafood
* Create a new virtual environment, and activate it.



* Upgrade pip and install Django



* Capture/freeze the Django independencies (the version of the Django), into the requirements.txt (which needs to be created under the main project folder)



* Create a new Django project: Andy’s Seafood
* Create a new Django app: products



* In products/settings.py, add  to INSTALLED\_APPS, which links the products app to the Andy\_seafood project.
* Run migrate before starting the application



* Run the new Django server, which shows successfully install page, little rocket!



Make apps:

* Under the products app, make 2 new directories, static and templates for css and html files. And make an urls.py fille to link products app to Andy\_Seafood project
* In view.py in products app, define what to see using html to show. Render function takes request as first argument and the path to html as the second argument. (auto assume the current directory is templates)



* Inside the urls.py (empty python file) in products app, import 2 functions: path from Django.urls which is the way to associate a URL path with the function in views.py.



And then import the function in views.py



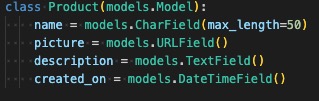
Create a list which names urlpatterns, and put all if the path-function registrations into the list.



* In urls.py in project Andy\_Seafood, link the app urls.py file to the project urls.py file. First, import include function from Django.urls

Then add the products.urls into the urlpatterns list.

* In product.html file, start with , which loads the static tag so we can use css file. Then in link tag, change href into which means using the static function to find the path to “product.css” inside the static folder.
* In models.py file in products app, create a class Product, and create attributes that associate with the Product class. The models package has a Model class



* After create a new Django model class or make changes, make migrations that keep track the changes in models.py 

Apply migrations to let Django actually make those changes in model class. 

* In admin.py in project Andy\_Seafood, admin.py allows us to create, edit, and delete data without writing any new html or css. Create a super user. 

I created a super user, username: evelyn, password: $Sunflower0723.

In order for Django model shows up in admin, I have to create a class that inherits from the ModelAdmin class, which is an admin for models, like Product model class.



Use code below to add Product class into admin page and List\_display shows which attribute will be seen in the admin page.

