Objectives

The objective of this lesson is to learn how forms work in Django using Django’s form framework.

What your program must have

Build an app that you could use to track inventory (such as items in your pantry, widgets in a warehouse, etc.).

- A category has:
  - name
  - description
  - parent category
- An inventory item has:
  - name
  - description
  - quantity
- Pages
  - Index page with a list of categories
  - Clicking on a category shows you all items for that category
  - Clicking +1 on an item increments the quantity available. Clicking on a -1 decrements the quantity for the item.
- Extra Credit Ideas:
  - A search form
  - Use ajax for form submission. A library like jQuery is completely fine.

Resources

- How to make a model its own parent: https://docs.djangoproject.com/en/1.5/ref/models/fields/#foreignkey “To create a recursive relationship – an object that has a many-to-one relationship with itself – use models.ForeignKey(‘self’).”
- Forms in Django (introductory material): https://docs.djangoproject.com/en/1.8/topics/forms/
- Form widgets reference: https://docs.djangoproject.com/en/1.8/ref/forms/widgets/
- Handling forms with class-based views: https://docs.djangoproject.com/en/1.8/topics/class-based-views/intro/#handling-forms-with-class-based-views
- Shortcuts (cheat codes) to making forms easier in Django using class-based views: https://docs.djangoproject.com/en/1.8/topics/class-based-views/generic-editing/
- Forms API reference: https://docs.djangoproject.com/en/1.8/ref/forms/api/
- Reverse_lazy api: https://docs.djangoproject.com/en/1.8/ref/urisolvers/#reverse-lazy
- jQuery Ajax documentation: http://api.jquery.com/jQuery.ajax/