



Quiz Django Intro	(/m/83%)
Time Display	(/m/)
Random Word Generator	(/m/)
Survey Form	(/m/)
Ninja Gold	(/m/)
(optional) Basic Regex	(/m/)
Context Objects	(/m/)
Route Parameters	(/m/)
Setup Cheatsheet	(/m/)
Quiz	(/m/)
Disappearing Ninjas	(/m/)
Surprise Me!	
Landscapes	
(optional) Django and AJAX	
(optional) Django Crawler	
Full Stack Django	20% (/m/13/3834)
Django and AJAX	0%

Django Project Cheat Sheet

Here is a crib sheet of the process of creating a Django application from scratch. This is a summary sheet of the commands learned up to this point. It's a good idea to be very familiar with this process, and this sheet is useful for a quick reference or you've gotten a good grasp of the concepts they describe.

Install

Mac:

```
> virtualenv djangoEnv
> source djangoEnv/bin/activate
(djangoEnv)>pip install django
```

PC:

```
> python -m virtualenv djangoEnv
> call djangoEnv/scripts/activate
(djangoEnv)>pip install django
```

Activate virtualenv

Mac:

- source djangoEnv/bin/activate

PC:

- call djangoEnv/scripts/activate

◀ PREVIOUS (/M/13/3832/26216)

Create project

'main' is the project name used below
django-admin startproject main

The project has been created, inspect the file & folders created as a result:

- Folder: main
 - File: manage.py <= used to run the application.
 - Folder: main <= contains the files specific to our app.
 - File: __init__.py
 - File: settings.py
 - File: urls.py
 - File: wsgi.py

A few more config items to complete the setup:

- Navigate into the main directory that you just created.

```
> cd main
```

- Make a new apps directory

```
> mkdir apps
```

- Navigate into apps

```
> cd apps
```

- Create an __init__.py file. This just let's Python know that this folder is viewable by Python so that # other Python modules can look in here and access the code!

Bash: (Mac, Linux, PCs with GitBash) =>

```
touch __init__.py
```

PCs: (Command Prompt) =>

```
nul> __init__.py
```

Start the new app.

```
> python ../manage.py startapp first_app
```

The app 'first_app' has been created, lets look at the updated file/folder structure:

- Folder: main
 - Folder: apps <= created by our steps above
 - File: __init__.py <= created by our steps above
 - Folder: first_app <= created as a result to the call to 'startapp'
 - File: __init__.py
 - File: admin.py
 - File: apps.py
 - File: models.py <= The 'M' in MVC
 - File: tests.py
 - File: views.py <= The CONTROLLER in our MVC pattern
 - File: manage.py <= used to run the application.
 - Folder: main <= contains the files specific to our app.
 - File: __init__.py
 - File: settings.py
 - File: urls.py
 - File: wsgi.py

Open the settings.py file and add a line to your INSTALLED_APPS:

```
INSTALLED_APPS = [  
    'apps.first_app', ### <= added this line!  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
]
```

Open the urls.py file, add the line to import include and add a line to include all of the url routes from our app into the main routes.

```
from django.conf.urls import url, include # Notice we added include  
from django.contrib import admin  
urlpatterns = [  
    url(r'^', include('apps.first_app.urls')), # And now we use include to pull in our first_app.urls...  
]
```

Now, create the urls.py file in our 'first_app' folder so that the final file/folder structure looks like this:

- Folder: main
 - Folder: apps <= created by our steps above
 - File: __init__.py <= created by our steps above
 - Folder: first_app <= created as a result to the call to 'startapp'
 - File: __init__.py
 - File: admin.py
 - File: apps.py
 - File: models.py <= The 'M' in MVC
 - File: tests.py
 - File: views.py <= The CONTROLLER in our MVC pattern
 - urls.py <= the file that will handle the routes of our app

- File: manage.py <= used to run the application.
- Folder: main <= contains the files specific to our app.
 - File: __init__.py
 - File: settings.py
 - File: urls.py
 - File: wsgi.py

Add some routes to our urls.py file!

```
from django.conf.urls import url
from . import views      # So we can call functions from our routes!
urlpatterns = [
    url(r'^$', views.index)    # 'home' route.
]
```

Finally, run the app:

```
python manage.py runserver
```

Adding views & static files

- Folder: main
 - Folder: apps <= created by our steps above
 - File: __init__.py <= created by our steps above
 - Folder: first_app <= created as a result to the call to 'startapp'
 - Folder: templates
 - Folder: first_app
 - Files: *.html
 - Folder: static
 - Folder: first_app
 - Folder: css
 - Folder: js
 - Folder: images
 - File: __init__.py
 - File: admin.py
 - File: apps.py
 - File: models.py <= The 'M' in MVC
 - File: tests.py
 - File: views.py <= The CONTROLLER in our MVC pattern
 - urls.py <= the file that will handle the routes of our app
 - File: manage.py <= used to run the application.
 - Folder: main <= contains the files specific to our app.
 - File: __init__.py
 - File: settings.py
 - File: urls.py
 - File: wsgi.py

Example of a Controller action ('View' in Django:)

```
# views.py
...
def index(request):
    return render(request, "first_app/index.html")
```

Implement sessions in our app:

```
# Need to be in same directory as manage.py file
> python manage.py makemigrations
> python manage.py migrate
```

Context

On the server:

```
from django.shortcuts import render, HttpResponseRedirect, redirect
def show_ninja(request, ninja):
    #ninja got passed in through the url parameter!
    context = {'myninja' : ninja}
    return render(request, '/myproject/showmyninja.html', context)
```

In our templates:

```
<!-- From inside /myproject/showmyninja.html -->
{%myninja%}
```

Route Parameters

In our urls.py file:

```
# Inside apps/appname/urls.py might look like this:
from django.conf.urls import url
from . import views
urlpatterns = [
    url(r'^/en/(?P<djangoversion>[0-9]\.[0-9])/topics/http/urls/$', views.index)
]
```

In our views.py file:

```
from django.shortcuts import render, HttpResponseRedirect, redirect
# Create your views here.
def index(request, djangoversion):
    print(djangoversion) # will result in a string e.g. "1.9"
    print("hello, I am your first request")
    return HttpResponseRedirect('hello')
```