Django – Installation Guide

1. How to install Django on Windows (If you have Python on your system directly go to C step)

This document will guide you through installing Python and Django for basic usage on Windows. This is meant as a beginner's guide for users working on Django projects and does not reflect how Django should be installed when developing patches for Django itself.

The steps in this guide have been tested with Windows 7 and 8. In other versions, the steps would be similar.

a. Install Python

Django is a Python web framework, thus requiring Python to be installed on your machine.

To install Python on your machine go to https://python.org/download/, and download a Windows MSI installer for Python. Once downloaded, run the MSI installer and follow the on-screen instructions.

After installation, open the command prompt and check the Python version by executing **`python --version`**. If you encounter a problem, make sure you have set the **`PATH`** variable correctly. You might need to adjust your **`PATH`** environment variable to include paths to the Python executable and additional scripts. For example, if your Python is installed in **`C:\Python34\`**, the following paths need to be added to **`PATH`**:

b. Install PIP

PIP is a package manager for Python that uses the Python Package Index to install Python packages. PIP will later be used to install Django from PyPI. If you've installed Python 3.4, **`pip`** is included so you may skip this section.

Open a command prompt and execute **`easy_install pip`**. This will install **`pip`** on your system. This command will work if you have successfully installed Setuptools.

Alternatively, go to http://www.pip-installer.org/en/latest/installing.html for installing/upgrading instructions.

c. Install Django

Django can be installed easily using **`pip`**.

In the command prompt, execute the following command: **`pip install django`**. This will download and install Django.

After the installation has completed, you can verify your Django installation by executing **`django-admin --version`** in the command prompt.

2. How to Create a Django Project in Anaconda?

Here are the steps that you need to follow to create a Django project in anaconda:

a. Create a new environment

First of all, you need to create a separate environment for Django development.

By default, anaconda comes up with the base environment. But creating a separate environment will be helpful to manage the projects without any package dependency confliction.

So, let's create a separate environment specifically for web development projects using the Django framework.

Open your CMD or Anaconda Prompt and create a separate environment called *djangoenv* by typing in the following command.

```
conda create -n djangoenv python=3.6 anaconda
```

I just mentioned the version of Python that I'm using as *Python 3.6*. If you want to use other versions of Python, you can do so.

The previous command may take some time to finish its execution. Once it is done, let's go to the next step.

b. Activate the new environment

Activate the newly created environment by typing in the following command.

```
conda activate djangoenv
```

Now, we are inside the newly created environment. But our environment does not have Django in it. So, let's install it.

c. Install Django in the new environment

Let's install Django in the new environment using the following command.

```
conda install -c anaconda django
```

Wait till the installation completes. It may take some time to finish. That's it. Now you can create Django projects in this separate environment.

Let's try to create a sample project to check whether everything is ready. Let's do this using the following commands.

d. Create a new Django project

First of all, let's start a new project called *website*.

```
django-admin startproject website
```

e. Run the Django server

Now, navigate to the project folder by using the cd command. Let's run the project and see whether everything is working fine or not.

```
python manage.py runserver
```

If you are on a Mac or Linux, use *python3 manage.py runserver* instead of the command given above.

Once you do that, a URL will be generated on the command line. Now, copy the URL that is generated and paste it on a new tab in your browser.

Now, your Django environment is set up nicely. After this, you can open Command Prompt or Anaconda Prompt whenever needed, and start doing the project just by activating the environment for Django.

The following command is used whenever we need to activate the Django environment.

```
conda activate djangoenv
```

Do these and create stunning projects using your favorite backend web framework *"Django",* without changing your Anaconda setup.