



django

tutorialspoint
SIMPLY EASY LEARNING

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

Django is a web development framework that assists in building and maintaining quality web applications. Django helps eliminate repetitive tasks making the development process an easy and time saving experience. This tutorial gives a complete understanding of Django.

Audience

This tutorial is designed for developers who want to learn how to develop quality web applications using the smart techniques and tools offered by Django.

Prerequisites

Before you proceed, make sure that you understand the basics of procedural and object-oriented programming: control structures, data structures and variables, classes, objects, etc.

Disclaimer & Copyright

© Copyright 2015 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com.

Table of Contents

About the Tutorial.....	i
Audience	i
Prerequisites	i
Disclaimer & Copyright.....	i
Table of Contents	ii
 1. DJANGO – BASICS	 1
History of Django	1
Django – Design Philosophies.....	1
Advantages of Django	1
 2. DJANGO – OVERVIEW.....	 3
 3. DJANGO – ENVIRONMENT.....	 4
Step 1 – Installing Python.....	4
Step 2 - Installing Django.....	4
Step 3 – Database Setup.....	6
Step 4 – Web Server	6
 4. DJANGO – CREATING A PROJECT	 7
Create a Project.....	7
The Project Structure	7
Setting Up Your Project	8
 5. DJANGO – APPS LIFE CYCLE	 10
 6. DJANGO – ADMIN INTERFACE	 12
 7. DJANGO – CREATING VIEWS.....	 15

8.	DJANGO – URL MAPPING	17
9.	DJANGO – TEMPLATE SYSTEM.....	23
	The Render Function	23
	Django Template Language (DTL)	23
	Filters	24
	Tags.....	24
10.	DJANGO – MODELS	28
	Creating a Model.....	28
	Linking Models	30
11.	DJANGO – PAGE REDIRECTION	33
12.	SENDING E-MAILS.....	36
	Sending a Simple E-mail	36
	Sending Multiple Mails with send_mass_mail.....	37
	Sending HTML E-mail	39
	Sending E-mail with Attachment	41
13.	DJANGO – GENERIC VIEWS.....	42
14.	DJANGO – FORM PROCESSING	46
15.	DJANGO – FILE UPLOADING	51
16.	DJANGO – APACHE SETUP	55
17.	DJANGO – COOKIES HANDLING	57
18.	DJANGO – SESSIONS.....	60

19. DJANGO – CACHING	64
Setting Up Cache in Database.....	64
Setting Up Cache in File System.....	65
Setting Up Cache in Memory	65
Caching the Entire Site	65
Caching a View	66
Caching a Template Fragment	67
20. COMMENTS.....	69
21. DJANGO – RSS	74
22. DJANGO – AJAX	78

1. DJANGO – BASICS

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Django makes it easier to build better web apps quickly and with less code.

Note: Django is a registered trademark of the Django Software Foundation, and is licensed under BSD License.

History of Django

- **2003:** Started by Adrian Holovaty and Simon Willison as an internal project at the Lawrence Journal-World newspaper.
- **2005:** Released July 2005 and named it Django, after the jazz guitarist Django Reinhardt.
- **2005:** Mature enough to handle several high-traffic sites.
- **Current:** Django is now an open source project with contributors across the world.

Django – Design Philosophies

Django comes with the following design philosophies:

- **Loosely Coupled:** Django aims to make each element of its stack independent of the others.
- **Less Coding:** Less code so in turn a quick development.
- **Don't Repeat Yourself (DRY):** Everything should be developed only in exactly one place instead of repeating it again and again.
- **Fast Development:** Django's philosophy is to do all it can to facilitate hyper-fast development.
- **Clean Design:** Django strictly maintains a clean design throughout its own code and makes it easy to follow best web-development practices.

Advantages of Django

Here are few advantages of using Django which can be listed out here:

- **Object-Relational Mapping (ORM) Support:** Django provides a bridge between the data model and the database engine, and supports a large set of database systems including MySQL, Oracle, Postgres, etc. Django also supports NoSQL database through Django-nonrel fork. For now, the only NoSQL databases supported are MongoDB and google app engine.
- **Multilingual Support:** Django supports multilingual websites through its built-in internationalization system. So you can develop your website, which would support multiple languages.
- **Framework Support:** Django has built-in support for Ajax, RSS, Caching and various other frameworks.
- **Administration GUI:** Django provides a nice ready-to-use user interface for administrative activities.
- **Development Environment:** Django comes with a lightweight web server to facilitate end-to-end application development and testing.

2. DJANGO – OVERVIEW

As you already know, Django is a Python web framework. And like most modern framework, Django supports the MVC pattern. First let's see what is the Model-View-Controller (MVC) pattern, and then we will look at Django's specificity for the Model-View-Template (MVT) pattern.

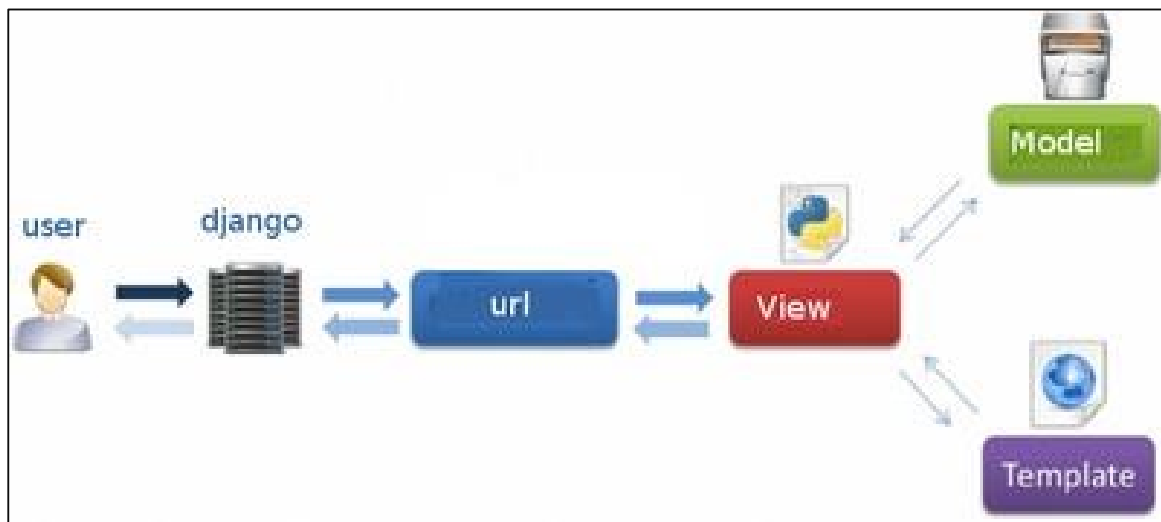
MVC Pattern

When talking about applications that provides UI (web or desktop), we usually talk about MVC architecture. And as the name suggests, MVC pattern is based on three components: Model, View, and Controller. [Check our MVC tutorial here](#) to know more.

DJANGO MVC - MVT Pattern

The Model-View-Template (MVT) is slightly different from MVC. In fact the main difference between the two patterns is that Django itself takes care of the Controller part (Software Code that controls the interactions between the Model and View), leaving us with the template. The template is a HTML file mixed with Django Template Language (DTL).

The following diagram illustrates how each of the components of the MVT pattern interacts with each other to serve a user request:



The developer provides the Model, the view and the template then just maps it to a URL and Django does the magic to serve it to the user.

3. DJANGO – ENVIRONMENT

Django development environment consists of installing and setting up Python, Django, and a Database System. Since Django deals with web application, it's worth mentioning that you would need a web server setup as well.

Step 1 – Installing Python

Django is written in 100% pure Python code, so you'll need to install Python on your system. Latest Django version requires Python 2.6.5 or higher for the 2.6.x branch or higher than 2.7.3 for the 2.7.x branch.

If you're on one of the latest Linux or Mac OS X distribution, you probably already have Python installed. You can verify it by typing *python* command at a command prompt. If you see something like this, then Python is installed.

```
$ python
Python 2.7.5 (default, Jun 17 2014, 18:11:42)
[GCC 4.8.2 20140120 (Red Hat 4.8.2-16)] on linux2
```

Otherwise, you can download and install the latest version of Python from the link <http://www.python.org/download>.

Step 2 - Installing Django

Installing Django is very easy, but the steps required for its installation depends on your operating system. Since Python is a platform-independent language, Django has one package that works everywhere regardless of your operating system.

You can download the latest version of Django from the link <http://www.djangoproject.com/download>.

UNIX/Linux and Mac OS X Installation

You have two ways of installing Django if you are running Linux or Mac OS system:

- You can use the package manager of your OS, or use `easy_install` or `pip` if installed.
- Install it manually using the official archive you downloaded before.

We will cover the second option as the first one depends on your OS distribution. If you have decided to follow the first option, just be careful about the version of Django you are installing.

Let's say you got your archive from the link above, it should be something like Django-x.xx.tar.gz:

Extract and install.

```
$ tar xzvf Django-x.xx.tar.gz
$ cd Django-x.xx
$ sudo python setup.py install
```

You can test your installation by running this command:

```
$ django-admin.py --version
```

If you see the current version of Django printed on the screen, then everything is set.

Note: For some version of Django it will be django-admin the ".py" is removed.

Windows Installation

We assume you have your Django archive and python installed on your computer.

First, PATH verification.

On some version of windows (windows 7) you might need to make sure the Path system variable has the path the following C:\Python27\;C:\Python27\Lib\site-packages\django\bin\ in it, of course depending on your Python version.

Then, extract and install Django.

```
c:\>cd c:\Django-x.xx
```

Next, install Django by running the following command for which you will need administrative privileges in windows shell "cmd":

```
c:\Django-x.xx>python setup.py install
```

To test your installation, open a command prompt and type the following command:

```
c:\>django-admin.py --version
```

If you see the current version of Django printed on screen, then everything is set.

OR

Launch a "cmd" prompt and type python then:

```
c:\> python
>>> import django
>>> print django.get_version()
```

End of ebook preview
If you liked what you saw...
Buy it from our store @ **<https://store.tutorialspoint>**