

Introduction

○
○○○○○
○○○○
○○○○

Using django
○○○○○○○○○○

Features of django
○○○

Reuseable apps
○○○
○○○○

End Notes

○○○
○○○
○○○

Building Pluggable Web Applications using **django**

Lakshman Prasad

Agiliq Solutions



April 21, 2010



django



- For building **database driven** web applications

django



- For building **database driven** web applications
- Emphasis on Reuse, DRY and simplicity

Introduction

○
●○○○○
○○○○
○○○○

Using django

○○○○○○○○○○

Features of django

○○○

Reusable apps

○○○○
○○○○○

End Notes

○○○
○○○
○○○

Hypothetical Application



Install a pluggable app **django-star-rating**

```
$pip install django-star-rating
```

```
INSTALLED_APPS = (  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.sites',  
    'django.contrib.admin',  
    'starrating',  
)
```

Point to an url pattern

```
urlpatterns = patterns('',
    (r'^admin/(.*)', admin.site.root),
    (r'^$', 'blog.views.index'),
    (r'staring/(?P<post_id>\d+)/$', 
     'starrating.views.star'),
)
```

Put the template where you want!

```
{% block ratings %}  
  {% if not user.is_authenticated %}  
    {% endif %}  
    {% starrating %}  
{% endblock %}
```

Introduction

○
○○○●
○○○○
○○○○

Using django

○○○○○○○○○○

Features of django

○○○

Reuseable apps

○○○
○○○○

End Notes

○○○
○○○
○○○

Introduction

Definition

A Case Study: Youtube Stars

Overview

Philosophy

Using django

MTV

Features of django

Features

Reuseable apps

Writing reuseable apps

Community Applications

End Notes

Django Stats

Common Enterprise Hurdles

Other

Introduction



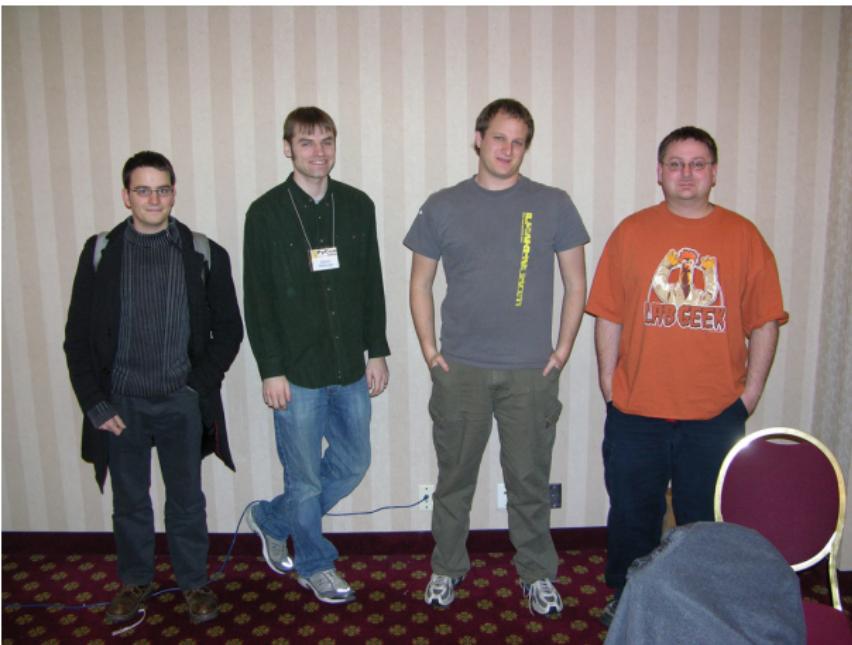
Using django
oooooooooooo

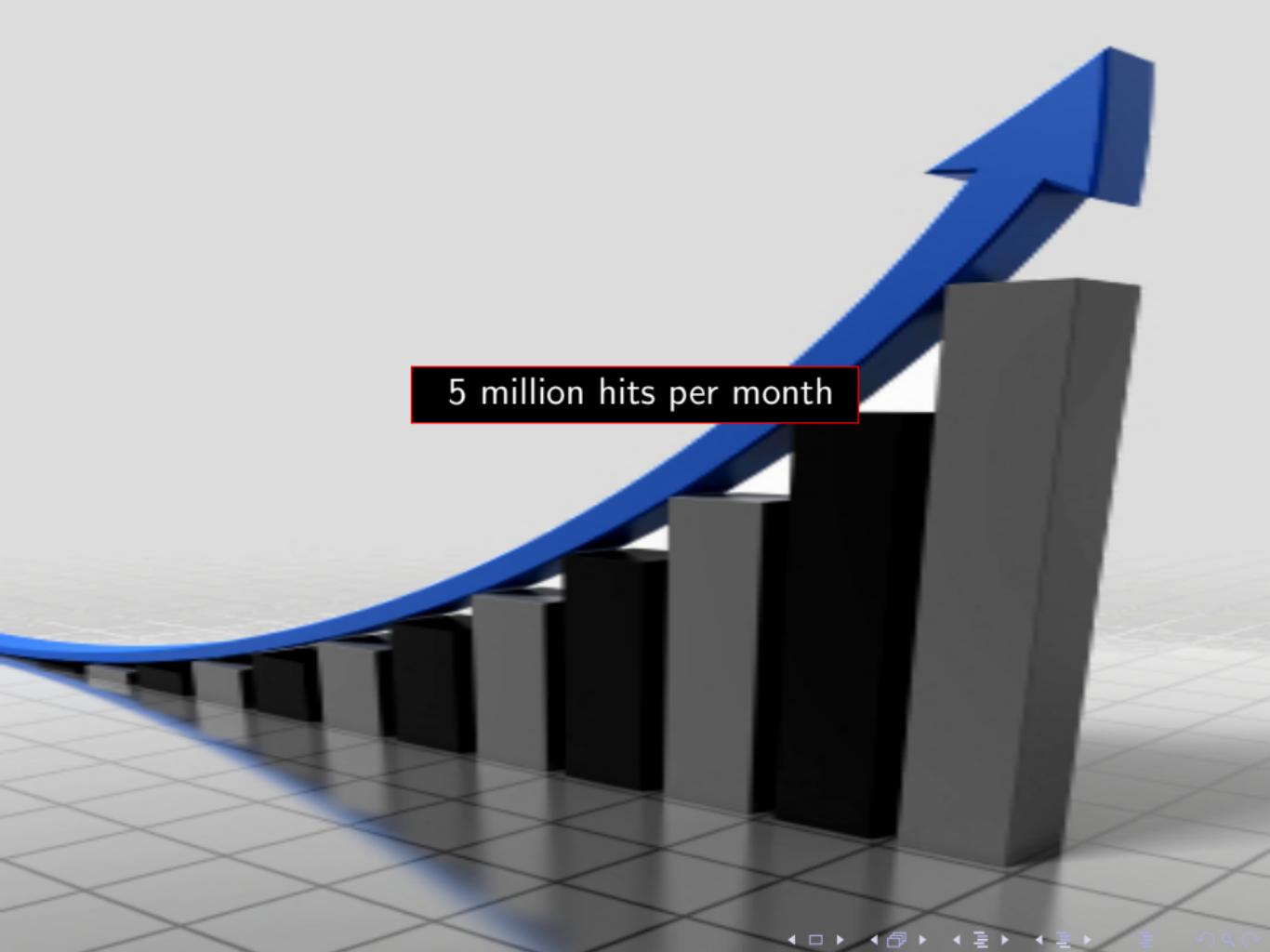
Features of django
ooo

Reuseable apps
oooo
oooo

End Notes
ooo
ooo
ooo

Developed at Lawrence-Journal World





A 3D bar chart illustrating growth. A blue arrow points upwards from the left side of the chart towards the top right. A black rectangular callout box with a red border is positioned in the middle-left area of the chart, containing the text "5 million hits per month". The chart features several dark grey bars of increasing height from left to right, set against a light grey grid background.

5 million hits per month

A wide-angle photograph of a massive outdoor crowd, likely at a festival or concert. The people are packed closely together, filling the frame from the foreground to the middle ground. In the background, there's a dense line of green trees under a bright, slightly cloudy sky. A few small white structures, possibly port-a-potties, are visible on a slight incline above the crowd. On the right side, there are some colorful tents and flags. A prominent black rectangular box with a red border is overlaid on the image, containing the text "600 Contributors".

600 Contributors



Majority of the features community contributed



5 years since Open Source

Introduction

○
○○○○○
○○○○
●○○○

Using django

○○○○○○○○○○

Features of django

○○○

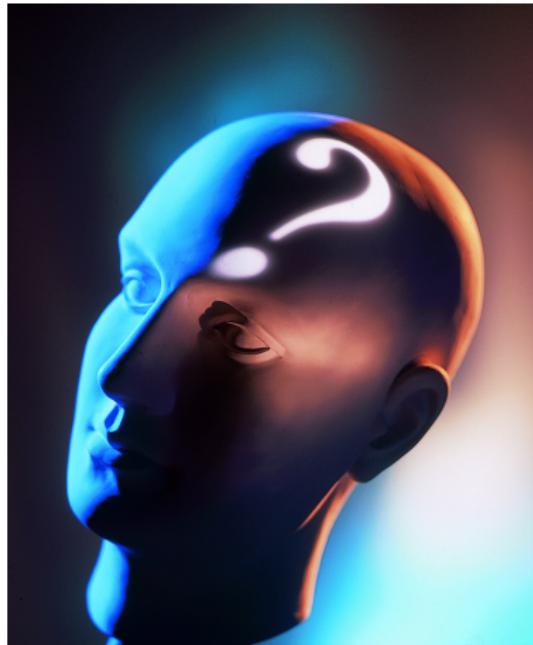
Reusable apps

○○○○
○○○○○

End Notes

○○○
○○○
○○○

Philosophy



I will not talk in class
not talk in cl

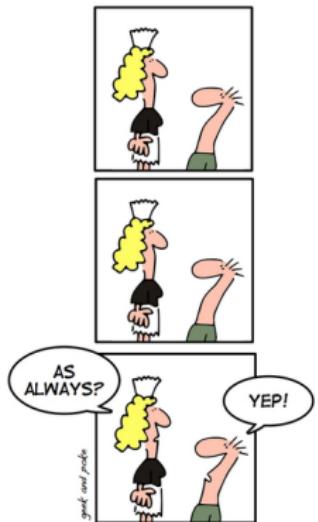
A photograph of a classroom. In the center, a student with long blonde hair, wearing a light blue shirt and dark jeans, stands at a whiteboard. They are holding a red book in one hand and a marker in the other, writing the sentence "I will not talk in class" repeatedly. The whiteboard is otherwise blank. In the foreground, the backs of several wooden desks and blue chairs are visible, suggesting a lecture hall or classroom setting.

Automate repetitive tasks



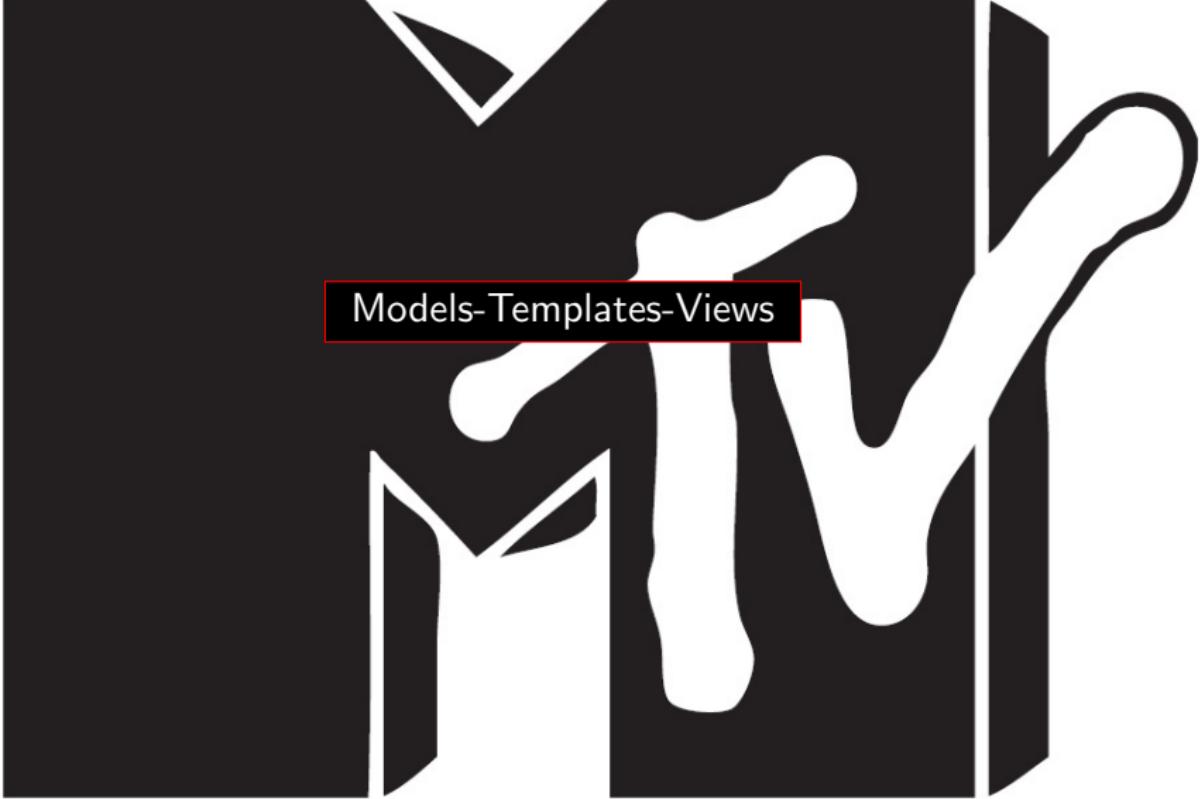
Make Development Fast

Convention Over Configuration

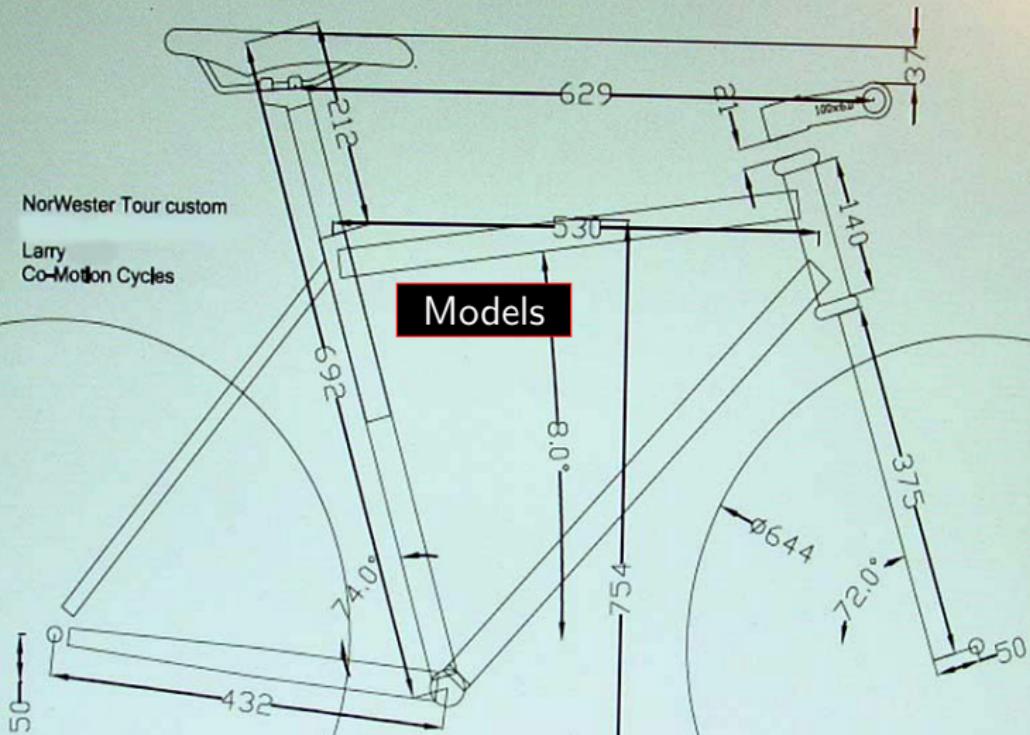


SIMPLY EXPLAINED - PART 18: CONVENTION OVER CONFIGURATION





Models-Templates-Views



Model Syntax

```
from django.db import models

class Post(models.Model):
    title = models.CharField(max_length=100)
    text = models.TextField()
    datetime = models.DateTimeField()

    class Meta:
        ordering = ('-datetime',)

    def __unicode__(self):
        return self.title

class Comment(models.Model):
    post = models.ForeignKey(Post)
    text = models.TextField()
```

Model API

```
from django.db import models

class Post(models.Model):
    title = models.CharField(max_length=100)
    text = models.TextField()
    datetime = models.DateTimeField()

    class Meta:
        ordering = ('-datetime',)

    def __unicode__(self):
        return self.title

class Comment(models.Model):
    post = models.ForeignKey(Post)
    text = models.TextField()
```

Admin Syntax

```
from django.contrib import admin
from models import Post, Comment

class PostAdmin(admin.ModelAdmin):
    list_display = ('title', 'datetime')

class CommentAdmin(admin.ModelAdmin):
    list_display = ('text',)

admin.site.register(Post, PostAdmin)
admin.site.register(Comment, CommentAdmin)
```

Admin by models alone

Administracja stroną

Auth		
Grupy		Dodaj Zmień
Użytkownicy		Dodaj Zmień
Sites		
Strony		Dodaj Zmień
Tasks		
Zadania		Dodaj Zmień
News		
Słowa Kluczowe		Dodaj Zmień
Wiadomości		Dodaj Zmień
Myghtyboard		
Categories		Dodaj Zmień
Forums		Dodaj Zmień

Ostatnie akcje
Moje akcje
LinuxMint 2.1 News
Nowe phpSolutions z punBB już jest News
Python full text search engines News
Nowe artykuły News
Nowe artykuły News
Python full text search engines News
Nowe artykuły News
users Group
Zrób sobie Auroxa News
Zrzut Tygodnia News

[Home](#) > [Photos](#)

Select photo to change

[Add photo](#)  Go[2000](#) [2001](#) [2002](#) [2003](#) [2004](#)
[2005](#) [2006](#)

Photo	Caption	Creation date
	Deron Lovaas, National Resources Defense Council	April 25, 2006, 11:21 a.m. List Page
	Ethan Nuss, KU senior and organizer of the "Powershift" energy security conference	April 25, 2006, 11:15 a.m.
	KU Enviros president Emelia LaFortune	April 25, 2006, 11:13 a.m.

Filter

By site:

All

[blogs.ljworld.com](#)[internal.ljworld.com](#)[www2.kusports.com](#)[www.ljworld.com](#)[www.6newslawrence.com](#)[www.6productions.com](#)[www.lawrence.com](#)

By creation date:

Any date

Today

Past 7 days

This month

This year

[Home](#) > [Photos](#) > Add photo

Add photo

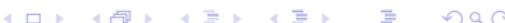
Photo:[Browse...](#)**Caption:****Add an Entry****Sites:**

- [blogs.ljworld.com](#)
- [internal.ljworld.com](#)
- [www2.kusports.com](#)
- [www2.ljworld.com](#)
- [www.6newslawrence.com](#)
- [www.6productions.com](#)
- [www.lawrence.com](#)



Hold down "Control", or "Command" on a Mac, to select more than one.

Credit

Staff member: -----**One-off
photographer:****Credit:****Creation date:****Date:** Today | 

World Online administration

World Online administration

LJWorld.com, Lawrence.com, KUsports.com, 6News

Home > Photos > Add photo

Add photo

 Please correct the errors below.

 This field is required.

Photo: [Browse...](#)

Caption:

Auto Validation

 This field is required.

Sites:

Hold down "Control", or "Command" on a Mac, to select more than one.

Credit

Staff member:  

Ugly urls

`http://art.com/artgallery/default.asp?
sid=9DF4BC0580DF11D3ACB60090271E26A8
&command=freelist`

`http://preview.ynot.com/cgi-bin/
nd_CGI_50.cgi/YnotPhoenix/CFsMain.`

Good urls

<http://www.wired.com/apple/MacbookPro/>

<http://devmarch.com/developersummit/speakers.html>

django url pattern

```
urlpatterns = patterns('',
    (r'^admin/(.*)', admin.site.root),
    (r'^$', 'blog.views.index'),
    (r'staring/(?P<post_id>\d+)/$', 
     'starrating.views.star'),
)
```



Views

django views

```
from django.http import HttpResponse

def post(request):
    return HttpResponse('Hello _World! ')
```

django views

```
def post(request, post_id):
    post = Post.objects.get(pk=post_id)
    if req.method == 'POST':
        comment_form = CommentForm(request.POST)
        comment = comment_form.save()
    payload = {'post':post,
               'comments':
                   Comment.objects.filter(post_id=post_id),
               'comment_form':CommentForm()}
    return render_to_response('post.html',
                             payload,
                             RequestContext(req))
```

DEAR Stephen,

THANK YOU SO MUCH FOR THE
LOVELY knife . IT IS VERY
sharp , AND IT WILL LOOK

Templates

sharp , IN OUR kitchen.

I AM SURE IT WILL GIVE US
MANY HAPPY HOURS cutting things.
IT WAS nice TO SEE YOU AT
THE WEDDING, AND WE SHOULD
DEFINITELY MEET UP NEXT TIME WE
ARE IN Canada .

MUCH LOVE, PETER +
Victoria

django template

```
{% for post in posts %}  
<h2><a href="/node/{{ post.id }}">    {{ post.title }}  
</a></h2>  
<p>    {{ post.datetime }} </p>  
<p>    {{ post.text }} </p>  
{% endfor %}
```



Introduction

○
○○○○○
○○○○
○○○

Using django

○○○○○○○○○○

Features of django

○●○

Reuseable apps

○○○
○○○○

End Notes

○○○
○○○
○○○

- Admin Interface

Introduction

○
○○○○○
○○○○
○○○

Using django

○○○○○○○○○○

Features of django

○●○

Reuseable apps

○○○
○○○○

End Notes

○○○
○○○
○○○

- Admin Interface
- Generic Views

Introduction

○
○○○○○
○○○○
○○○○

Using django

○○○○○○○○○○

Features of django

○●○

Reuseable apps

○○○
○○○○

End Notes

○○○
○○○
○○○

- Admin Interface
- Generic Views
- Testing Tools

Introduction

A 4x3 grid of 12 small circles, arranged in four rows and three columns.

Using django

• • • • •

Features of django

3

Reusable apps

10

End Notes

10

- Admin Interface
 - Generic Views
 - Testing Tools
 - Sessions

Introduction

○
○○○○○
○○○○
○○○○

Using django

○○○○○○○○○○

Features of django

○●○

Reuseable apps

○○○
○○○○

End Notes

○○○
○○○
○○○

- Admin Interface
- Generic Views
- Testing Tools
- Sessions
- Authentication

Introduction

○
○○○○○
○○○○
○○○○

Using django

○○○○○○○○○○

Features of django

○●○

Reuseable apps

○○○
○○○○

End Notes

○○○
○○○
○○○

- Admin Interface
- Generic Views
- Testing Tools
- Sessions
- Authentication
- Caching

Introduction

A 4x3 grid of 12 small circles, arranged in four rows and three columns.

Using django



○○○○○○○○○○

Features of django

3

Reusable apps

10

End Notes

1

- Admin Interface
 - Generic Views
 - Testing Tools
 - Sessions
 - Authentication
 - Caching
 - Internationalization

Introduction

A 4x3 grid of 12 small circles, arranged in four rows and three columns.

Using django

Features of django

Reusable apps

oooo
ooooo

End Notes

- Admin Interface
 - Generic Views
 - Testing Tools
 - Sessions
 - Authentication
 - Caching
 - Internationalization
 - RSS

Introduction

○
○○○○○
○○○○
○○○○

Using django

○○○○○○○○○○

Features of django

○●○

Reuseable apps

○○○
○○○○

End Notes

○○○
○○○
○○○

- Admin Interface
- Generic Views
- Testing Tools
- Sessions
- Authentication
- Caching
- Internationalization
- RSS
- **CSRF protection**

- Admin Interface
- Generic Views
- Testing Tools
- Sessions
- Authentication
- Caching
- Internationalization
- RSS
- CSRF protection
- File Storage

Django documentation

Everything you need to know about Django (and then some).

First steps

- **From scratch:** [Overview](#) | [Installation](#)
- **Tutorial:** [Part 1](#) | [Part 2](#) | [Part 3](#) | [Part 4](#)

The model layer

- **Models:** [Model syntax](#) | [Field types](#) | [Meta options](#)
- **QuerySets:** [Executing queries](#) | [QuerySet method reference](#)
- **Model Instances:** [Instance methods](#) | [Accessing related objects](#)
- **Advanced:** [Managers](#) | [Raw SQL](#) | [Transactions](#) | [Aggregation](#) | [Custom fields](#) | [Multiple databases](#)
- **Other:** [Supported databases](#) | [Legacy databases](#) | [Providing initial data](#) | [Optimize database access](#)

The template layer

- **For designers:** [Syntax overview](#) | [Built-in tags and filters](#)
- **For programmers:** [Template API](#) | [Custom tags and filters](#)

The view layer

- **The basics:** [URLconfs](#) | [View functions](#) | [Shortcuts](#)
- **Reference:** [Request/response objects](#)
- **File uploads:** [Overview](#) | [File objects](#) | [Storage API](#) | [Managing files](#) | [Custom storage](#)
- **Generic views:** [Overview](#) | [Built-in generic views](#)
- **Advanced:** [Generating CSV](#) | [Generating PDF](#)
- **Middleware:** [Overview](#) | [Built-in middleware classes](#)

Forms

- **The basics:** [Overview](#) | [Form API](#) | [Built-in fields](#) | [Built-in widgets](#)
- **Advanced:** [Forms for models](#) | [Integrating media](#) | [Formsets](#) | [Customizing validation](#)
- **Extras:** [Form preview](#) | [Form wizard](#)

The development process

- **Settings:** [Overview](#) | [Full list of settings](#)
- **django-admin.py and manage.py:** [Overview](#) | [Adding custom commands](#)
- **Testing:** [Overview](#)
- **Deployment:** [Overview](#) | [Apache/mod_wsgi](#) | [Apache/mod_python](#) | [FastCGI/SCGI/AJP](#) | [Apache authentication](#) | [Serving static files](#) | [Tracking code errors by e-mail](#)

Latest • 1.0 • 0.96 • All

Browse

- [Table of contents](#)
- [General Index](#)
- [Global Module Index](#)

You are here:

- [Django dev documentation](#)
- [Django documentation](#)

Last update:

April 13, 2010, 5:30 a.m. (CDT)

Getting help

Having trouble? We'd like to help!

- Try the [FAQ](#) – it's got answers to many common questions.
- Looking for specific information? Try the [Index](#), [Module Index](#) or the [detailed table of contents](#).
- Search for information in the archives of the [django-users mailing list](#), or post a question.
- Ask a question in the [#django IRC channel](#), or search the [IRC logs](#) to see if its been asked before.
- Report bugs with Django in our [ticket tracker](#).

Django Documentation

Introduction

○
○○○○○
○○○○
○○○

Using django

○○○○○○○○○○

Features of django

○○○

Reusable apps

●○○○
○○○○○

End Notes

○○○
○○○
○○○

General conventions adopted by the community

Introduction

○
○○○○○
○○○○
○○○

Using django

○○○○○○○○○○

Features of django

○○○

Reuseable apps

○●○○
○○○○○

End Notes

○○○
○○○
○○○

High Level

- Use template tags

Introduction

○
○○○○○
○○○○
○○○

Using django

○○○○○○○○○○

Features of django

○○○

Reuseable apps

○●○○
○○○○○

End Notes

○○○
○○○
○○○

High Level

- Use template tags
- Use signals

Basics

- Take `template_name` and `extra_context` every where

Basics

- Take `template_name` and `extra_context` every where
- Write `urls` in `applications`

Basics

- Take `template_name` and `extra_context` every where
- Write urls in applications
- Import from the application level

Basics

- Take `template_name` and `extra_context` every where
- Write urls in applications
- Import from the application level
- Prefix template name with directory

Basics

- Take `template_name` and `extra_context` every where
- Write urls in applications
- Import from the application level
- Prefix template name with directory
- Use `MEDIA_URL` in templates

Basics

- Take `template_name` and `extra_context` every where
- Write urls in applications
- Import from the application level
- Prefix template name with directory
- Use `MEDIA_URL` in templates
- Reverse url patterns

Introduction

○
○○○○○
○○○○
○○○

Using django

○○○○○○○○○○

Features of django

○○○

Reuseable apps

○○○●
○○○○○

End Notes

○○○
○○○
○○○

Advanced

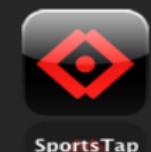
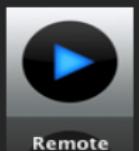
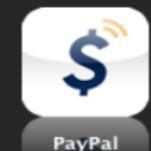
- Use Template Response

Advanced

- Use Template Response
 - Write Views as Classes



There is an app for that



A woman with blonde hair tied back is standing in front of a wall of wooden shelves displaying various styles of women's high-heeled shoes and boots. The shelves are illuminated from behind, creating a warm glow. A circular logo for "STEVE MADDEN" is centered above the middle row of shoes. A red-bordered text box containing the slogan "For every size and style" is overlaid in the center-left area.

For every size and style

Introduction

○
○○○○○
○○○○
○○○○

Using django
○○○○○○○○○○

Features of django
○○○

Reuseable apps
○○○○
○○●○○

End Notes
○○○
○○○
○○○

Github Search "django"

3000 repositories!

Introduction

○
○○○○○
○○○○
○○○○

Using django
○○○○○○○○○○

Features of django
○○○

Reuseable apps

○○○○
○○○●○

End Notes

○○○
○○○
○○○

Pinax

The Pinax logo consists of the word "pinax" in a bold, lowercase, sans-serif font. The letters are a vibrant lime green color, set against a dark blue background.

a platform for rapidly developing websites

"I'm simply astounded by how far Pinax has come in the short time it's been around. The list of features is huge; I think with a little more work Pinax could become a serious competitor to Plone or Drupal." — Jacob Kaplan-Moss, Django Co-BDFL

[Download](#)

or

[browse the documentation](#)

django-mingus

django_extensions
basic
flatblocks
disqus
navbar
djangodblog
sorl
oembeded
template_utils
django_proxy
compressor
django_markup
google_analytics
view_cache_utils
contact_form
honeypot
sugar
quoteme

Introduction

○
○○○○○
○○○○
○○○○

Using django

○○○○○○○○○○

Features of django

○○○

Reusable apps

○○○○
○○○○○

End Notes

●○○
○○○
○○○

Django users

Introduction

○
○○○○○
○○○○
○○○○

Using django

○○○○○○○○○○

Features of django

○○○

Reusable apps

○○○○
○○○○○

End Notes

○●○
○○○
○○○

Popular Users

- Media

Introduction

A 4x5 grid of 20 small circles, arranged in four rows and five columns.

Using django

Features of django

Reusable apps

oooo
ooooo

End Notes

Popular Users

- Media
 - LA Times

Popular Users

- Media
 - LA Times
 - NY Times

Popular Users

- Media
 - LA Times
 - NY Times
 - Washington Post

Popular Users

- Media
 - LA Times
 - NY Times
 - Washington Post
 - **Guardian.**

Popular Users

- Media
 - LA Times
 - NY Times
 - Washington Post
 - Guardian.
 - Web2.0

Popular Users

- Media
 - LA Times
 - NY Times
 - Washington Post
 - Guardian.
- Web2.0
 - Mahalo: 10 million Page views

Popular Users

- Media
 - LA Times
 - NY Times
 - Washington Post
 - Guardian.
- Web2.0
 - Mahalo: 10 million Page views
 - Pownce

Popular Users

- Media
 - LA Times
 - NY Times
 - Washington Post
 - Guardian.
- Web2.0
 - Mahalo: 10 million Page views
 - Pownce
- Full List: djangosites.com

NASA

After an extensive trade study, we selected Django as the first and primary application environment for the Nebula Cloud.

Introduction

A 4x5 grid of 20 small circles, arranged in four rows and five columns.

Using django

Features of django

Reusable apps

End Notes

○○○

Multiple Databases

Introduction



Using django



Features of django



Reusable apps



End Notes



Dynamic settings infrastructure

Introduction

A 4x3 grid of 12 small circles, arranged in four rows and three columns.

Using django



Features of django

Reusable apps

End Notes

Tools

About Me

- lakshman@agiliq.com Agiliq Solutions <http://www.agiliq.com/>
Team of Expert Django Developers @becomingGuru @agiliq

Introduction

○
○○○○○
○○○○
○○○

Using django

○○○○○○○○○○

Features of django

○○○

Reusable apps

○○○
○○○○

End Notes

○○○
○○○
○●○

Image Attributions

Introduction

○
○○○○○
○○○○
○○○

Using django

○○○○○○○○○○

Features of django

○○○

Reusable apps

○○○
○○○○

End Notes

○○○
○○○
○○●

?