

**django**

# Django

The Web framework for perfectionists with deadlines.

- ✦ Written in Python (<http://python.org>).
- ✦ Developed by a newspaper company.

## KU unplugs steam whistle to cut costs

September 3, 2008 · 50 Comments

Don Steeples embraces tradition, understands the importance of getting to class on time, and respects the public's communal reliance on familiar on-campus rituals. He also knows that tooting KU's familiar steam whistle costs about \$3,000 a year. So he's pulled the plug.

[Chat: KU Provost Richard Lariviere discusses upcoming issues facing the university](#)

[Latest news](#)

[Most discussed](#)

[Blog - Lawrence Weather Watch: Cape Verde-type Hurricanes](#) 8:58 p.m.

[Fields to miss Louisiana Tech game](#) 6:38 p.m.

[Mistrial declared in kidnapping case](#) 5:05 p.m.

[Police Web site provides safety tips for students](#) 4:19 p.m.



Photo by Scott McClurg

Kansas University's Facilities Operations workers, from left, Jack Bame, Floyd Grant and Bob Sieber affix the new steam whistle as



**Marketplace**

**Shelter Insurance - Lance Wollesen, Agent**  
785-865-4040

*Find more businesses on Marketplace*

[Arts & Entertainment](#) · [Business](#)  
[Coffeeshouses](#) · [Libraries](#)

- Released publicly under a BSD license in July 2005

3 Sept., 2008 Django 1.0  
was released!

## Django 1.0 released!

No, you're not hallucinating, it's really here.

Around three years ago, Adrian, Simon, Wilson and I released some code to the world. Our plan was to hack quietly on it for a bit, release a solid 1.0 release, and then *really* get the ball rolling.

Well.

What happened, of course, was that an amazing community sprung up literally overnight — our IRC channel had over a hundred people in it the day after release, and it's never been that "empty" since.

I really can't stress enough how amazing our community of users and developers are. About half of the code that's gone into Django over the past three years has been contributed by someone *other* than a core committer. Since our last stable release, we've made over 4,000 code commits, fixed more than 2,000 bugs, and edited, added, or removed around 350,000 lines of code. We've also added 40,000 lines of new documentation, and greatly improved what was already there.

Django 1.0 represents a the largest milestone in Django's development to date: a web framework that a group of perfectionists can truly be proud of. Without this amazing community, though, it would have never happened.

You can download Django 1.0 on the [Django downloads page](#), and read the [complete release notes](#).

For distributors and for verification purposes, a file containing the MD5 and SHA1 checksums of the 1.0 package has been placed on [the django project.com server](#). This

### Archives

[September 2008](#)

[August 2008](#)

[July 2008](#)

[June 2008](#)

[May 2008](#)

[April 2008](#)

[January 2008](#)

[December 2007](#)

[November 2007](#)

[October 2007](#)

[September 2007](#)

[August 2007](#)

[July 2007](#)

[June 2007](#)

[May 2007](#)

[April 2007](#)

[March 2007](#)

[February 2007](#)

[January 2007](#)

[December 2006](#)

[November 2006](#)

# Django's Goal

- ✦ To create complex database driven websites.
- ✦ Reusability of Code or *Pluggability*.
- ✦ Rapid development.
- ✦ DRY (Do not Repeat Yourself).



# What does Django's Core Include

- ✦ Database Abstraction Layer
- ✦ Views
- ✦ Templates
- ✦ URL Schema

# MTV

- ✦ **M**odel
- ✦ **T**emplate
- ✦ **V**iew



Django	MVC
Model	Model
Views and Templates	View
URL	Controller

# Model

- Takes Python Code

```
class Post(models.Model):
    title          = models.CharField(_('title'), max_length=200)
    slug           = models.SlugField(_('slug'))
    author         = models.ForeignKey(User, blank=True, null=True)
    body           = models.TextField(_('body'))
    tease         = models.TextField(_('tease'), blank=True)
    publish        = models.DateTimeField(_('publish'))
```

# Model

- and Coverts it to SQL

```
CREATE TABLE "blog_posts" (  
  "id" integer NOT NULL PRIMARY KEY,  
  "title" varchar(200) NOT NULL,  
  "slug" varchar(50) NOT NULL,  
  "author_id" integer NULL REFERENCES "auth_user" ("id"),  
  "body" text NOT NULL,  
  "tease" text NOT NULL,  
  "publish" datetime NOT NULL,  
);
```

- ✦ Basic Meta

- ✦ URL's



```
class Meta:
    verbose_name           = _('post')
    verbose_name_plural   = _('posts')
    db_table               = 'blog_posts'
    ordering                = ('-publish',)
    get_latest_by         = 'publish'

def __unicode__(self):
    return u"%s" % self.title

@permalink
def get_absolute_url(self):
    return ('blog_detail', None, {
        'slug' : self.slug
    })
```

- ✦ Also does business logic

```
@property  
def digital_fingerprint(self):  
    return md5.new(self.title).hexdigest()
```

```
@property
def lunch(self):
    if self.lunch_in and self.lunch_out:
        lunch = relativedelta(
            datetime.combine(self.date, self.lunch_in),
            datetime.combine(self.date, self.lunch_out)
        )
        return lunch
    else:
        return relativedelta(hours=0, minutes=0)
```

# Managers

- ✦ `Post.objects.all()` - Return all posts
- ✦ `Post.objects.get(pk=1)` - Return only the post with the private key of 1
- ✦ `Post.objects.filter(title__contains='TLUG')` - Return the posts with 'TLUG' in the title.

# Custom Managers

```
from django.db.models import Manager
import datetime

class ManagerWithPublished(Manager):
    def published(self):
        return self.get_query_set().filter(
            publish__lte=datetime.datetime.now()
        )
```

# View

- ✦ Contains the business logic for the page

```
from django.http import Http404
from django.shortcuts import render_to_response
from django.template import RequestContext

from blog.models import Post

def index(request):
    posts = Post.objects.all()[10]

    return render_to_response('blog/index.html', {
        'posts': posts
    }, context_instance=RequestContext(request))
```



```
def detail(request, slug):
    try:
        post = Post.objects.get(slug__iexact=slug)
    except Post.DoesNotExist:
        raise Http404

    return render_to_response('blog/detail.html', {
        'post': post
    }, context_instance=RequestContext(request))
```

# Generic Views

- ✦ Paginated Results
- ✦ Handles HTTP 404 for us

```
from django.views.generic import list_detail

from blog.models import Post

def index(request, page=0):
    return list_detail.object_list(
        request,
        queryset = Post.objects.published(),
        paginate_by = 10,
        page = page,
    )
```

```
def detail(request, slug):  
    return list_detail.object_detail(  
        slug = slug,  
        queryset = Post.objects.published(),  
    )
```

# URL

- ✦ Controller

```
from django.conf.urls.defaults import *  
  
urlpatterns = patterns('blog.views',  
    url(r'^$',  
        view      = 'index',  
        name      = 'blog_index',  
    )  
)
```

- Regular Expressions

```
from django.conf.urls.defaults import *

urlpatterns = patterns('blog.views',
    url(r'^(?P<slug>[-\w]+)/$',
        view      = 'detail',
        name      = 'blog_detail',
    ),
    url(r'^$',
        view      = 'index',
        name      = 'blog_index',
    )
)
```



```
@permalink
```

```
def get_absolute_url(self):  
    return ('blog_detail', None, {  
        'slug' : self.slug  
    })
```

Template

```
{% extends "base.html" %}
```

```
{% block title %}
```

```
    Blog Index
```

```
{% endblock %}
```

```
{% block content %}
```

```
    {% for post in posts %}
```

```
        <h2>{{ post.title }}</h2>
```

```
        {{ post.body|linebreaks }}
```

```
    {% endfor %}
```

```
{% endblock %}
```

# Django's Contribution

- ✦ Administration Interface
- ✦ Authentication
- ✦ Comments
- ✦ GIS (Geographic Information System)
- ✦ Multiple Sites
- ✦ Syndication Built in - RDF 0.9, RSS 2.0, and Atom
- ✦ Google Sitemap
- ✦ and more

Demo Time

# Pluggable

- ✦ Separate applications for everything.
  - ✦ Blog
  - ✦ Tumblelog
  - ✦ Photos
  - ✦ Wiki

# Search Django on Google Code

Myles.Braithwaite@gmail.com | [What's new?](#) | [Profile](#) | [Settings](#) | [Help](#) | [Sign out](#)



## Projects on Google Code

Results 1 - 10 of 847

### [django-profile - Generic User Profile Control Panel developed for django](#)

== Introduction == This is a [http://djangopluggables.com Django pluggable] user profile zone which can be used and customized easily in your social application web platform developed in django. It's developed with [http://www.djangoproject.com Django] for the backend framework and uses [http://...]

[django](#), [profile](#), [avatar](#), [registration](#), [login](#), [maps](#), [jquery](#), [javascript](#)

### [django-command-extensions - Django Custom Management Command Extensions](#)

This is a repository for collecting global custom management extensions for the Django Framework. =Name Change= **\*IMPORTANT\*** django-command-extensions has become django-extensions – We have renamed the project and the import point from extensions to django\_extensions. **\*IMPORTANT\*** You wil...

[django](#), [management](#), [command](#), [extension](#)

### [django-tagging - A generic tagging application for Django projects](#)

A generic tagging application for Django projects, which allows association of a number of tags with any 'Model' instance and makes retrieval of tags simple. The [http://django-tagging.googlecode.com/svn/trunk/CHANGELOG.txt CHANGELOG] is kept up to date with notable commits and the latest version

# Deployment

- ✦ mod\_python
- ✦ FastCGI
- ✦ wsgi compatible web server
  - ✦ mod\_wsgi
  - ✦ CherryPy



# mod\_python

```
<Location />  
    SetHandler python-program  
    PythonPath "[ '/path/to/project' ] + sys.path"  
    PythonHandler django.core.handlers.modpython  
    SetEnv DJANGO_SETTINGS_MODULE blog.settings  
    PythonDebug Off  
</Location>
```

Scaling