

# Web Development Paradigms and how django and GAE webapp approach them.

Lakshman Prasad

Agiliq Solutions



September 25, 2010



# Programming paradigm

From Wikipedia, the free encyclopedia



This article **needs attention from an expert on the subject**. See the [talk page](#) for details.  
[WikiProject Computer science](#) or the [Computer science Portal](#) may be able to help recruit an expert.  
(January 2009)

A **programming paradigm** is a fundamental style of [computer programming](#). (Compare with a [methodology](#), which is a style of solving specific [software engineering problems](#).) Paradigms differ in the concepts and abstractions used to represent the elements of a program (such as objects, functions,

## Programming paradigms

- Agent-oriented
- Component-based

## Concepts and abstractions used to represent elements of a program

### Contents [hide]

- 1 Overview
- 2 Multi-paradigm programming language
- 3 History
- 4 See also
- 5 References
- 6 External links

## Overview

[edit]

Programming model: Abstraction of a computer system, for example the "von Neumann model" used in traditional sequential computers. For [parallel computing](#), there are many possible models typically reflecting different ways processors can be interconnected. The most common are based on shared memory, distributed memory with message passing, or a hybrid of the two.

A [programming language](#) can support [multiple paradigms](#). For example programs written in [C++](#) or [Object Pascal](#) can be purely [procedural](#), or purely [object-oriented](#), or contain elements of both paradigms. Software designers and programmers decide how to use those paradigm elements.

In [object-oriented programming](#), programmers can think of a program as a collection of interacting objects, while in [functional programming](#) a program can be thought of as a sequence of stateless function evaluations. When programming computers or systems with many processors, [process-oriented programming](#) allows programmers to think about applications as sets of concurrent processes.

- Concatenative
- Concurrent computing
- Declarative (contrast: Imperative)
  - Functional
  - Dataflow
    - Cell-oriented (spreadsheets)
    - Reactive
- Constraint
- Logic
  - Constraint logic
  - Abductive logic
  - Inductive logic
- Event-driven
  - Service-oriented
  - Time-driven
- Expression-oriented
- Feature-oriented
- Function-level (contrast: Value-level)
- Imperative (contrast: Declarative)
  - Procedural
- Non-structured (contrast: Structured)
  - Array (contrast: Scalar)



Web Development Paradigms: Modeling common web patterns



Why?





Batteries Included

## Other batteries included

- [Admin site](#) | [Admin actions](#)
- [Authentication](#)
- [Cache system](#)
- [Conditional content processing](#)
- [Comments](#) | [Moderation](#) | [Custom comments](#)
- [Content types](#)
- [Cross Site Request Forgery protection](#)
- [Databrowse](#)
- [E-mail \(sending\)](#)
- [Flatpages](#)
- [GeoDjango](#)
- [Humanize](#)
- [Internationalization](#)
- [Jython support](#)
- "Local flavor"
- [Messages](#)
- [Pagination](#)
- [Redirects](#)
- [Serialization](#)
- [Sessions](#)
- [Signals](#)
- [Sitemaps](#)
- [Sites](#)
- [Syndication feeds \(RSS/Atom\)](#)
- [Unicode in Django](#)
- [Web design helpers](#)
- [Validators](#)

docs.djangoproject.com



Webapp: Simple web application framework (with custom APIs)



Lets examine the batteries

# How do I know about web development or django

- Active Djangonaut
- Part of a few popular open source django applications  
[github.com/becomingGuru](https://github.com/becomingGuru), [github.com/agiliq](https://github.com/agiliq)
- Consulting and Development via AGILIQ SOLUTIONS
- Developed several custom proprietary django applications



- @becomingGuru

## Introduction

## Forms

## Authentication

## Generic Views

## Caching

## Others

## Admin

## Forms

Account not verified

**SEND VERIFICATION LINK**



No file selected

**Choose File**

About me :

This is my profile

\* Username :

whitekites

\* E-mail address:

ashok.raavi@yahoo.co.in

First name :

Last name :

Date of birth

1980-06-19

Gender

Male

**CHANGE PASSWORD**

## Save form data to a table

Company name :

clearthoughtsolutions

Address Line 1 :

555 west kinzie st

Company webpage :

http://www.clearthoughtsolutions

Address Line 2 :

apt#E1712

Banner image :

No file selected

**Choose File**

City :

chicago

Phone number :

1234567890

State :

IL

Fax number :

123546789

Zip code :

60654

Country :

USA

Daytime Phone :

5736804656

ext.

Evening Phone :

3127420983

ext.

**SAVE PROFILE**

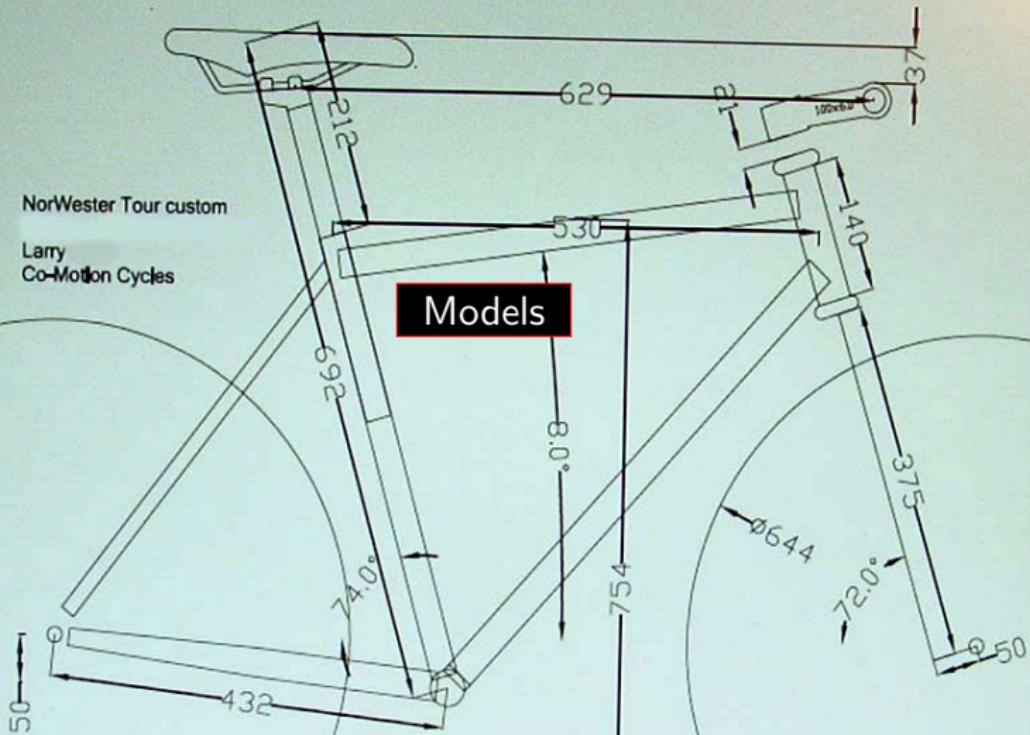
# Use a Model Form

```
>>> from django.forms import ModelForm

# Create the form class.
>>> class ArticleForm(ModelForm):
...     class Meta:
...         model = Article

# Creating a form to add an article.
>>> form = ArticleForm()

# Creating a form to change an existing article.
>>> article = Article.objects.get(pk=1)
>>> form = ArticleForm(instance=article)
```



# 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 blog.models import Post, Comment  
>>>post = Post.objects.all()[0]  
>>>post_comments = post.comment_set.all()
```

# Save Foreign Keys

POLL

Enter your question here

## Use Model Formset

```
from django.forms.models import modelformset_factory
AuthorFormSet = modelformset_factory(Author)
formset = AuthorFormSet()
>>> print formset
<input type="hidden" name="form-TOTAL_FORMS"
value="1" id="id_form-TOTAL_FORMS" />
<input type="hidden" name="form-INITIAL_FORMS"
value="0" id="id_form-INITIAL_FORMS" />
<input type="hidden" name="form-MAX_NUM_FORMS"
id="id_form-MAX_NUM_FORMS" />
<tr><th>
<label for="id_form-0-name">Name:</label>
</th><td>
<input id="id_form-0-name" type="text"
name="form-0-name" maxlength="100" />
</td></tr>
```

# Model Formset options

```
formset = AuthorFormSet(qs=Author.objects.all(),  
                       extra=5)
```

```
formset.is_valid()
```

```
formset.errors  
{'name': 'This field is required'}
```

```
formset.changed_forms
```

```
formset.save()
```

# Form pre populated

POLL

Enter your question here

⊕ This is option one

⊕ This is option two

⊕ This is option three

⊕

⊕

# Forms dont have to save to a Model

```
from django import forms

class ContactForm(forms.Form):
    subject = forms.CharField(max_length=100)
    message = forms.CharField()
    sender = forms.EmailField()
    cc_myself = forms.BooleanField(required=False)

    def save(self):
        #Do anything
        ...
```

Select "Submit" if this information is correct to complete Step 10: Review Electronic Funds Transfer (EFT) Information. After submission, the Review Electronic Funds Transfer (EFT) Verification page will display requesting

## Final Verification

Select "Cancel" to return to the Reconciliation Checklist and Step 10 will remain in an "Incomplete" status.

### Account Information:

Bank Name: My Bank  
Account Type: Savings  
Company Name: Company 1  
Associated with Account:  
Account Number: 1233445566  
Bank Routing Number: 112233448

### Bank Contact

First Name: Joseph  
Middle Initial: R  
Last Name: Smith  
Telephone: (123) 555-9292 ext.:  
Fax: (123) 555-9293

E-mail Address:

### Bank Address

Street Line 1: 123 Main Street  
Street Line 2:  
City: Boulder  
State: COLORADO  
Zip Code: 22222-1111

## Form Preview

**Confirm**

**Cancel**

# Form Preview

```
from django.contrib.formtools.preview import FormPreview
from myapp.models import SomeModel

#Add a url
(r'^post/$', SomeModelFormPreview(SomeModelForm)),

#Define form preview

class SomeModelFormPreview(FormPreview):

    def done(self, request, cleaned_data):
        # Do something with the cleaned_data, then
        # redirect to a "success" page.
        return HttpResponseRedirect('/form/success')
```

1

2

3

4

## STEP 1: CREATE YOUR FORM

It will only take 4 simple steps for you to create and use your new form. To start, enter a name for your form below. If you have an existing form you want to import, select one of the import methods below. Otherwise, start with a pre-built or blank form.

### Form Wizard

\*Form Name

\*Start With

Pre-built form: Contact Form

Copy an existing form from a webpage

Upload an HTML file containing a form

Blank form

Next Step ►

# Form Wizard

```
class ContactWizard(FormWizard):
    def done(self, request, form_list):
        do_something_with_the_form_data(form_list)
        return HttpResponseRedirect('/redirect/')
```

\*\*\*\*\*

# PASSWORD

Authentication

# Point to url pattern

```
urlpatterns += patterns('django.contrib.auth.views',  
    url(r'^login/$', 'login'),  
    url(r'^logout/$', 'logout'),  
    url(r'^register/$', 'register'),  
    url(r'^passreset/$', 'password_reset'),  
    url(r'^passreset2/$', 'password_reset_done'),  
    url(r'^passresetconfirm/(?P<uidb36>[-\w]+)/$',  
        'password_reset_confirm'),  
    url(r'^passresetcomplete/$',  
        'password_reset_complete'),  
)
```

# Login Required

```
from django.contrib.auth.decorators import \
login_required

@login_required(redirect_field_name='redirect_to')
def my_view(request):
    ...
```

# Authentication backends

```
class SettingsBackend:  
    """  
        Authenticate against the settings ADMIN_LOGIN  
        and ADMIN_PASSWORD.  
    """  
  
    def authenticate(self, username=None, password=None):  
        if login_valid and pwd_valid:  
            return user  
        return None  
  
    def get_user(self, user_id):  
        try:  
            return User.objects.get(pk=user_id)  
        except User.DoesNotExist:  
            return None
```

# Dinette

Forum based on dinette. Discuss about web application development using django and the related tech stack.

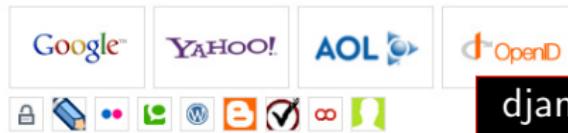
[Home](#) [Search](#) [Blog](#) [Download Dinette](#)

[Login](#)

[Active Topics](#) [Unans](#)

## Sign-in or Create New Account via these openid providers

Please click your account provider:



django-socialauth

## Sign-in or Create New Account via twitter



Copyright © 2010 [Aailig](#)

Power

DEAR Stephen,

THANK YOU SO MUCH FOR THE  
LOVELY knife . IT IS VERY

sharp , AND IT WILL LOOK

Send to template

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

# Generic Views

```
from django.conf.urls.defaults import *
from django.views.generic.simple \
import direct_to_template

urlpatterns = patterns('',
    ('^about/$', direct_to_template, {
        'template': 'about.html'
    })
)
```

# Object based generic views

World Online administration  
LJWorld.com, Lawrence.com, KUsports.com, 6News

Welcome, Jacob.  
Change password / Logout

Home > Photos

Add photo

Select photo to change

Photo	Caption	Creation date
	Deron Lovaas, National Resources Defense Council	April 25, 2006, 11:21 a.m.
	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
- www2.ljworld.com
- www.6newslawrence.com
- www.6productions.com
- www.lawrence.com

By creation date:

- Any date
- Today
- Past 7 days
- This month
- This year

# Display objects and their lists

```
from django.views.generic.list_detail import \
    object_detail, object_list
url((r'^objects/page(?P<page>[0-9]+)/$', \
    object_list,
    dict(info_dict)),
    (r'^objects/(?P<id>[0-9]+)/$', \
    object_detail,
    dict(info_dict)),
)
```

# List of generic views

`django.views.generic.simple.direct_to_template`  
`django.views.generic.simple.redirect_to`

`django.views.generic.list_detail.object_list`  
`django.views.generic.list_detail.object_detail`

`django.views.generic.create_update.create_object`  
`django.views.generic.create_update.update_object`  
`django.views.generic.create_update.delete_object`



"Normalized data is for sissies" - Cal Henderson

**reddit is down. it's turning out to be one of those days, again.**

in the meantime, here's a simulated reddit to help you stave off the withdrawal symptoms:

- 1 We're \*so\* close to achieving our Haiti relief goal. Let's do it right now! ([\(dri.com\)](#))  
↳ no comments these links don't go anywhere
- 2 Usually redditall continues to work during downtimes like this. ([\(redditall.com\)](#))  
↳ no comments these links don't go anywhere
- 3 IRC party in #reddittowntime on Freenode ([\(irc.freenode.org\)](#)) ([\(www.google.com\)](#))  
↳ no comments these links don't go anywhere
- 4 Keep mashing reload; that might help. ([\(reddit.com\)](#))  
↳ no comments these links don't go anywhere
- 5 You know about the red button, right? ([\(www.smbc-comics.com\)](#))

# Different blocks need different caching durations

The screenshot displays a web application interface. On the left, there is a 'Flickr' feed showing a grid of 12 small images. A large pink circular overlay covers the top half of the feed. In the center, there is a text input field containing the word 'Flickr'. To the right of the feed, a blue circular overlay covers the bottom half of a sidebar. The sidebar contains several tweets from various users:

- I changed my religion. I have just started using and loving emacs.  
about 1 hour ago
- @sixhsix Where is the new API for accessing from the python console documented?  
4 hours ago
- @sixhsix how do I authenticate now? Api remains the same?  
14 hours ago
- @pyconindia that's a nice milestone. I can bet there is going to be atleast as many more  
14 hours ago
- Any python library to deal with the google apps federated logins, open id? Or any library to start with, atleast, cc @simonw  
#lazyweb  
22 hours ago

# Generally Memcache Every page

```
CACHE_BACKEND = 'memcached://127.0.0.1:11211/'  
CACHE_BACKEND = 'memcached://172.19.26.240:11211;\\  
172.19.26.242:11212;172.19.26.244:11213/'
```

# Implement a caching decorator

```
from django.core.cache import cache
def cache_for(seconds, fetch=0):
    def cache_it(func):
        def deco_func(self):
            val = cache.get(self.get_cache_key(fetch))
            if not val:
                val = func(self)
                cache.set(self.get_cache_key(fetch),
                          val, seconds)
            return val
        return deco_func
    return cache_it
```

# Apply on the blocks

```
class TwitterBlock(TwitterSearchBlock):  
  
    template_name = 'twitter_user_block.html'  
    ajax_template = 'tweets_user.html'  
  
    @cache_for(60*60,fetch=1)  
    def fetch_data(self):  
        tw = Twitter(email='umoja_com')  
        user_tweets = tw.statuses.user_timeline(  
            screen_name=self.data)  
        return user_tweets,None
```

# Different cache Backends

- Database
- File System
- Local Memory
- Dummy- For Development

# More Paradigms

- Internationalization
- Localization
- Humanization
- CSRF Protection
- Pagination
- Sorting
- Data grid
- Sorl Thumbnail

One more thing...



# Admin by models alone

## Administracja stroną

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

Ostatnie akcje
<b>Moje akcje</b>
<a href="#">LinuxMint 2.1</a> News
<a href="#">Nowe phpSolutions z punBB już jest</a> News
<a href="#">Python full text search engines</a> News
<a href="#">Nowe artykuły</a> News
<a href="#">Nowe artykuły</a> News
<a href="#">Python full text search engines</a> News
<a href="#">Nowe artykuły</a> News
<a href="#">users</a> Group
<a href="#">Zrób sobie Auroxa</a> News
<a href="#">Zrzut Tygodnia</a> News

# 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)
```

[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.
	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.

[List](#) [Page](#)

## 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 &gt; Photos &gt; Add photo

## Add photo

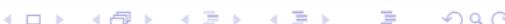
**Photo:**[Browse...](#)**Caption:****Add an Entry****Sites:**

- [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 | 

[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:  