Quel outil pour quel projet : Le framework django

Min2rien, jeudi 29 novembre 2012

Université Lille1
Sciences et Technologies

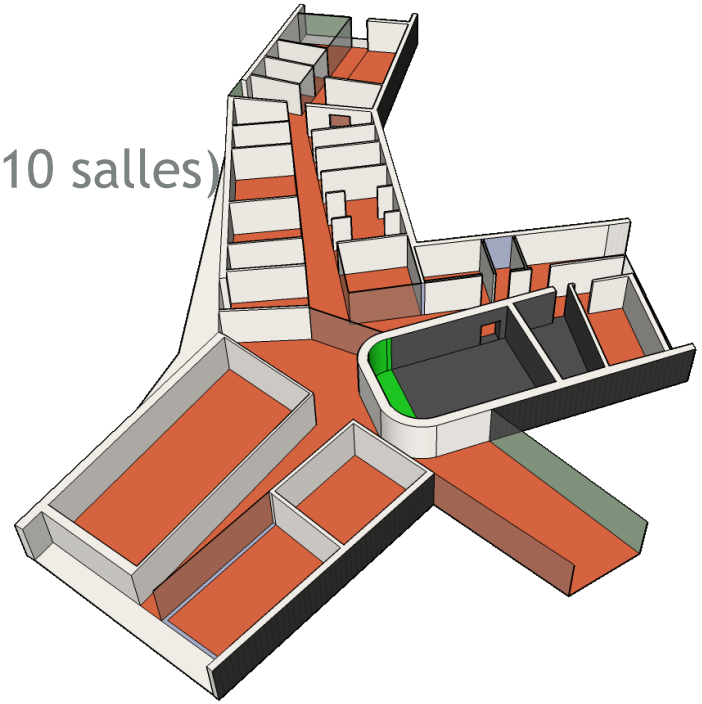
SEMM
Service Enseignement et Multimédia

Bvd Paul Langevin , 59655 Villeneuve d'Ascq
Tél. +33 (0) 20 33 64 41 | Fax. +33 (0) 20 33 63 95
Mail : semm@univ-lille1.fr| Site : semm.univ-lille1.fr

Le SEMM

Service Enseignement et Multimédia

- **En chiffre :**
  - Une équipe de 20 personnes
  - Un studio de 80m2
  - Un parc de 300 PC Multimédia (10 salles)
- **En activité :**
  - Pédagogie
  - Vidéo
  - Développement, infographie et intégration Multimédia



Quels langages pour quels usages...

- **Php :**
  - Blog Video Lille1 (Zend)
  - Demain l'universite (Zend)
  - Blogs Lille1 (Zend)
  - Moodle / Mahara
  - Carte de vœux
  - ...
- **Aspx :**
  - Lille1TV et Lille1TV.POD
  - Pédagothèque
  - Blogs Lille1
  - Phymuse
  - Sysbio (référencement espèce vivante)
  - ...
- **Python :**
  - Photothèque
  - Lille1Pod (A venir en remplacement de l'actuel)
  - ...
- **Scenari**
  - Création de source de contenu au format XML et publication multisupport.(Biologie Animale, C2i MEAD, UEL, ...)
- **Java :**
  - Applets (animations et simulations)
- **Javascript / HTML5 / CSS3**
- **Flash (AS3) :**
  - Animations
  - Players

Quels langages pour quels usages : Exemple

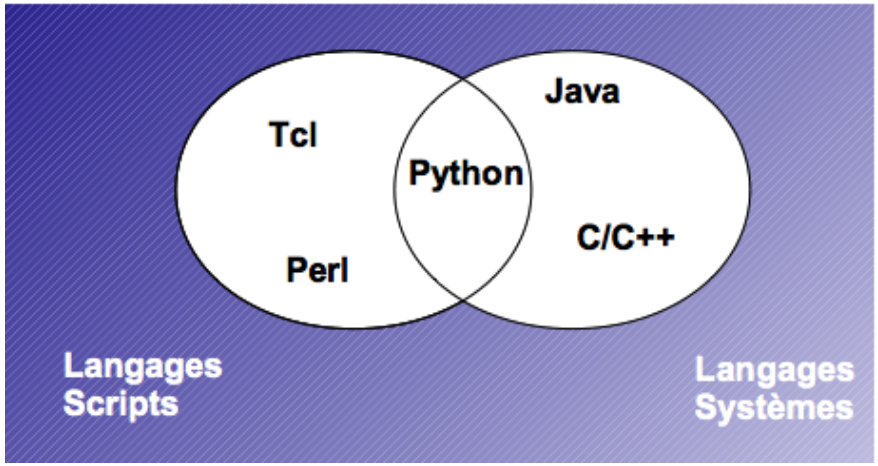
- Lille1tv.univ-lille1.fr (Aspx)
- Voeux (php)
- webconf.univ-lille1.fr (python, As3, java)
- Etc.

Le python...

- **Présentation :**
  - Langage de programmation simple, facile à apprendre et très puissant
  - Orienté objet, disponible sur toutes les plates-formes principales, facilement extensible
  - Logiciel libre : www.python.org
- **Utilisation:**
  - Script, administration système
  - IHM via TkInter, GTK, QT, wxPython
  - Programmation scientifique
  - Internet/Web:CGI,HTML,FTP,XML
- **Execution:**
  - Interpréteur / Script Unix
  - Embarqué dans du code C/C++
  - Module (script, fonctions, classes) dans fichier

Le python...

- **Comparaison:**



http://wiki.python.org/moin/LanguageComparisons

http://twistedmatrix.com/users/glyph/rant/python-vs-java.html

- **Concepts:**

Interprété, mémoire (ramasse miette), typage, nombres / chaînes de caractère, listes / dictionnaires, espaces ou tabulations...

Le framework …

- **Pourquoi :**
  - Parce qu'ils le vendent bien !
  - Ni moins bon ni meilleur qu'un autre mais correspond à ma manière de développer !

http://red-bean.com/~adrian/django_pronunciation.mp3

- **Concepts :**
  - DRY not CRY
  - MVT not MVC

MVC	MTV
Modèle	
Vue	*Template*
	Vue
Contrôleur	*Django*

- **Principes :**
  - Mapping relationnel-objet
  - Interface d'administration automatisée
  - Design élégant des URLs
  - Système de template
  - Système de cache
  - Internationalisation

Dernière version : 1.4.1

Les modèles

- **ORM :**
  - Object-Relationnal Mapping ou couplage objet relationnel

Modèle Django	SGBD
classe	table
objet	ligne d'une table
champ	colonne d'une table

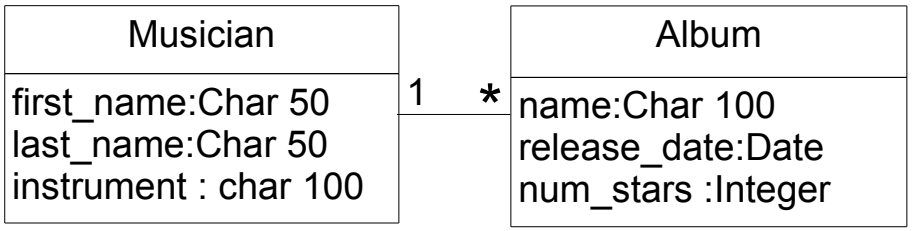
- **No SQL...**
  - Manage sql and syncdb (dump data, south)
  - Fields
  - Relation (one-to-one, ForeignKey, Many-to-many)

Les modèles

- **Exemple :**

```
from django.db import models
class Person(models.Model):
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
```

```
CREATE TABLE myapp_person (
    "id" serial NOT NULL PRIMARY KEY,
    "first_name" varchar(30) NOT NULL,
    "last_name" varchar(30) NOT NULL
);
```




```
class Musician(models.Model):
    first_name = models.CharField(max_length=50)
    last_name = models.CharField(max_length=50)
    instrument = models.CharField(max_length=100)

class Album(models.Model):
    artist = models.ForeignKey(Musician)
    name = models.CharField(max_length=100)
    release_date = models.DateField()
    num_stars = models.IntegerField()
```

Les vues

- **Gestion des URLs :**
  - URL patterns ou dispatcher
  - (regular expression, Python callback function [, optional dictionnary [, optional name]])
  - Nommage des urls (reverse() ou {% url %})
  - Inclusion d'urls
- **Creation de vues « from scratch »**
  - Def, HttpRequest et HttpResponse
  - Les fonctions raccourcies(object_or_404, list_or_404, redirect, render...)
  - Les décorateurs
- **Les vues génériques**
  - Les « CRUD » View

Les vues : Exemple

```
urlpatterns = patterns('',
    url(r'^polls/$', 'polls.views.index'),
    url(r'^polls/(?P<poll_id>\d+)/$',
'polls.views.detail'),
    url(r'^polls/(?P<poll_id>\d+)/results/
$', 'polls.views.results'),
    url(r'^polls/(?P<poll_id>\d+)/vote/$',
'polls.views.vote'),
)
#.. OU ...#
urlpatterns = patterns('polls.views',
    url(r'^polls/$', 'index',
name='accueil'),
    url(r'^polls/(?P<poll_id>\d+)/$',
'detail', name='detail'),
    url(r'^polls/(?P<poll_id>\d+)/results/
$', 'results', name='result'),
    url(r'^polls/(?P<poll_id>\d+)/vote/$',
'vote', name='vote'),
)
```

```
from django.template import Context, loader
from polls.models import Poll
from django.http import HttpResponse
from django.shortcuts import render_to_response, get_object_or_404
#...
def index(request):
    latest_poll_list = Poll.objects.all().order_by('-pub_date')[:5]
    t = loader.get_template('polls/index.html')
    c = Context({
        'latest_poll_list': latest_poll_list,
    })
    return HttpResponse(t.render(c))
#.. OU ...#
def index(request):
    latest_poll_list = Poll.objects.all().order_by('-pub_date')[:5]
    return render_to_response('polls/index.html',
                        {'latest_poll_list': latest_poll_list})
# …
# AUTRE EXEMPLE
#...
def detail(request, poll_id):
    p = get_object_or_404(Poll, pk=poll_id)
    return render_to_response('polls/detail.html', {'poll': p})
```

Les templates

- **Fichiers textes :**
  - HTML, XML, CSV, etc.
- **Variables**
  - {{variable}}
  - Evaluées par le template qui affiche le résultat
  - Le point (.) permet d'accéder aux attributs de la variable (instance d'objet)
- **Filtres**
  - {{name|lower}} Ils modifient l'affichage des variables
  - Peuvent être chainés et avoir des arguments {{ text|escape|truncatewords:30 }}
  - Exemple : add, capfirst, date, length , lower, safe etc...
- **Tags**
  - {% … %} Plus complexe, ils contrôlent la logique du template.
  - Environ 24 fournis par Django (for, if, load, url, with, extends, includ, block...)

On peut créer ses propres filtres et Tags !

Les templates : Exemple

base_generic.html

```
<html>
<head>
    <title>{% block title %}{% endblock %}</title>
</head>
<body>
    <img src="sitelogo.gif" alt="Logo" />
    {% block content %}{% endblock %}
</body>
</html>
```

section.html

```
{% extends "base_generic.html" %}
{% block title %}{{ section.title }}{% endblock %}
{% block content %}
<h1>{{ section.title }}</h1>
{% for story in story_list %}
<h2>
  <a href="{{ story.get_absolute_url }}">
    {{ story.headline|upper }}
  </a>
</h2>
<p>{{ story.tease|truncatewords:"100" }}</p>
{% endfor %}
{% endblock %}
```

https://docs.djangoproject.com/en/1.4/ref/templates/builtins/

Les forms

- **Librairie de manipulation de formulaire :**
  - Génération automatique du formulaire HTML avec ses widgets (composants) pour affichage
  - Validation des données soumises en fonction d'un certain nombre de règles données
  - Réaffichage du formulaire avec ses données dans le cas d'erreur de validation
  - Conversion des données soumises en model pour sauvegarde en base
- **Avec seulement 4 concepts :**
  - Les Forms : collection de fields qui sait se valider et s'afficher en html
  - Les Fields : récupérer les données saisies et les valider
  - Les Widgets : pour générer les représentations html des fields
  - Les Forms Media : ensemble de css et js nécessaires au rendu du formulaire

```
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)
```

```
from django.shortcuts import render
def contact(request):
    form = ContactForm() # An unbound form
    return render(request, 'contact.html', {'form':form,})
```

```
<form action="/contact/" method="post">{% csrf_token %}
{{ form.as_p }}
<input type="submit" value="Submit" />
</form>
```

Les briques

- **Les applications :**
  - L'interface d'administration !
  - Authentification et permissions
  - Internationalisation
  - Et un ensemble d'application mise à disposition par la communauté...
- **Les middlewares :**
  - Traitement effectué entre les requêtes et les réponses HTTP dans Django
  - Mécanisme légé et bas niveau pour altérer les entrées/sorties
  - Exemple : CSRF, authentification, message, session etc.
- **Les context processors :**
  - Dictionnaire de données injectées dans le context des vues (CSRF, media , request etc.)

+ la pagination, la syndication, les commentaires et toutes les librairies du langage Python sans aucune restriction !

Les exemples

- Webinaires : webinaires.unisciel.fr
- Dépôt de projet : projets.unisciel.fr
- Photothèque : phototheque.unisciel.fr

A venir :
- Lille1pod

Lille1pod


Université Lille1
Accéder aux autres sites de l'Université: Inclure le select dynamique ok
lille1.pod | La Plateforme Vidéo de Lille 1
Lille1.TV Lille1.POD
ACCUEIL | CHAÎNES | AUTEURS |
Se connecter Se déconnecter
Dentitions & Soins
page ▸ page ▸ page ▸ current page
L'université, la démocratie, la science et le débat public - Conférence inaugurale des Universités Lille 1 et Lille 3
Mise en ligne le 16/10/2012 à 17:00 | Durée 01:58:19
Favori Télécharger Partager
partager En savoir plus
Chapitres
» Porttitor sit amet bibendum quis
» Risus fermentum
» Sed commodo, magna at dignissim pharetra, dolor risus tristique ipsum
» Conclusion
Phasellus est risus, porttitor sit amet bibendum quis, gravida et orci. Sed lacus odio, porttitor et facilisis eget, sodales ac urna. Sed ac consectetur nulla. Quisque faucibus, neque euismod semper ultrices, elit nisl tempus mauris, vel vestibulum risus metus eu tellus. Integer id quam purus. Sed commodo, magna at dignissim pharetra, dolor risus tristique ipsum, ac fringilla orci arcu et dui. Donec pulvinar turpis sit amet nulla porttitor in imperdiet mauris dictum. Ut blandit risus fermentum mauris euismod quis feugiat leo mollis. Cras sed orci eu
+ AJOUTER UNE VIDEO
Rechercher
recherche avancée
Mon compte
Mes vidéos
Mes favoris
Mes chaînes
Types de documents
Cours (8)
Conférence (11)
Documentaire (13)
Document pédagogique (4)
Film (6)
Entretien / Interview (2)
Série (7)
Spectacle (6)
Autre (24)

Lille1pod


Université Lille1
Accéder aux autres sites de l'Université:
inclure le select dynamique
ok
lille1.pod | La Plateforme Vidéo de Lille 1
Lille1.TV
Lille1.POD
ACCUEIL | CHAÎNES | AUTEURS |
Se connecter
Se déconnecter
Dentitions & Soins
page ▸ page ▸ page ▸ current page
L'université, la démocratie, la science et le débat public - Conférence inaugurale des Universités Lille 1 et Lille 3
Mise en ligne le 16/10/2012 à 17:00 | Durée 01:58:19
Favori
Télécharger
Partager
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras ut ante vitae dui viverra consequat. Duis porttitor venenatis mauris vel pharetra. Curabitur varius sollicitudin augue quis hendrerit. Aenean sed rutrum ipsum. Sed magna orci, interdum quis molestie vitae, blandit consectetur nisl. Nulla vestibulum molestie nibh in scelerisque. Praesent vitae purus odio. Quisque pharetra, mauris sed volutpat accumsan, felis orci suscipit ligula, euismod cursus nisl ipsum at est. Nulla non placerat lacus. Morbi enim dolor, fringilla nec mollis eget, ultrices vel felis. Suspendisse lacus quam, consectetur eleifend dictum id, malesuada vel nunc. Proin vitae metus et augue congue molestie. Nulla faucibus volutpat tortor ac aliquam.
Chapitres
Résumé
Metadata
Documents
chap 1
chap 2
chap 3
chap 4
options
0:00/0:00
+ AJOUTER UNE VIDEO
Rechercher
recherche avancée
Mon compte
Mes vidéos
Mes favoris
Mes chaînes
Types de documents
Cours (8)
Conférence (11)
Documentaire (13)
Document pédagogique (4)
Film (6)
Entretien / Interview (2)
Série (7)
Spectacle (6)
Autre (24)
Université Lille1
Sciences et Technologies
Service Enseignement & Multimédia

Merci de votre attention

Université Lille1
Sciences et Technologies

SEMM
Service Enseignement et Multimédia

Bvd Paul Langevin , 59655 Villeneuve d'Ascq
Tél. +33 (0) 20 33 64 41 | Fax. +33 (0) 20 33 63 95
Mail : semm@univ-lille1.fr| Site : semm.univ-lille1.fr