

Cours Django #1

Aymeric Bringard

FIXME Hackerspace Lausanne

7 mars 2013



The Web framework for perfectionists with deadlines.

- Framework de développement d'applications web libre et gratuit (Licence BSD depuis juillet 2005).

- Framework de développement d'applications web libre et gratuit (Licence BSD depuis juillet 2005).
- Ecrit en Python.

- Framework de développement d'applications web libre et gratuit (Licence BSD depuis juillet 2005).
- Ecrit en Python.
- Suit le principe d'architecture modèle-vue-contrôleur.

- Framework de développement d'applications web libre et gratuit (Licence BSD depuis juillet 2005).
- Ecrit en Python.
- Suit le principe d'architecture modèle-vue-contrôleur.
- Maintenu par la Django Software Foundation (DSF) depuis 2008.

Objectifs

- Faciliter la création de sites complexes.

Objectifs

- Faciliter la création de sites complexes.
- Projets axés autour d'une base de données.

Objectifs

- Faciliter la création de sites complexes.
- Projets axés autour d'une base de données.
- Développement rapide.

Objectifs

- Faciliter la création de sites complexes.
- Projets axés autour d'une base de données.
- Développement rapide.
- Modularité, réutilisabilité.

- Faciliter la création de sites complexes.
- Projets axés autour d'une base de données.
- Développement rapide.
- Modularité, réutilisabilité.
- Principe DRY (Don't Repeat Yourself).

Composants

- Serveur web allégé indépendant pour le dev et les tests.

Composants

- Serveur web allégé indépendant pour le dev et les tests.
- Système de traitement de formulaires muni de *widgets*: HTML – Base de données.

Composants

- Serveur web allégé indépendant pour le dev et les tests.
- Système de traitement de formulaires muni de *widgets*: HTML – Base de données.
- Framework pour la mise en cache (différentes solutions utilisables).

Composants

- Serveur web allégé indépendant pour le dev et les tests.
- Système de traitement de formulaires muni de *widgets*: HTML – Base de données.
- Framework pour la mise en cache (différentes solutions utilisables).
- Support de classes intermédiaires *middleware*.

- Serveur web allégé indépendant pour le dev et les tests.
- Système de traitement de formulaires muni de *widgets*: HTML – Base de données.
- Framework pour la mise en cache (différentes solutions utilisables).
- Support de classes intermédiaires *middleware*.
- Support complet d'Unicode.

- Serveur web allégé indépendant pour le dev et les tests.
- Système de traitement de formulaires muni de *widgets*: HTML – Base de données.
- Framework pour la mise en cache (différentes solutions utilisables).
- Support de classes intermédiaires *middleware*.
- Support complet d'Unicode.
- Moteur de templates extensible.

- Serveur web allégé indépendant pour le dev et les tests.
- Système de traitement de formulaires muni de *widgets*: HTML – Base de données.
- Framework pour la mise en cache (différentes solutions utilisables).
- Support de classes intermédiaires *middleware*.
- Support complet d'Unicode.
- Moteur de templates extensible.
- Système d'internationalisation.

- Serveur web allégé indépendant pour le dev et les tests.
- Système de traitement de formulaires muni de *widgets*: HTML – Base de données.
- Framework pour la mise en cache (différentes solutions utilisables).
- Support de classes intermédiaires *middleware*.
- Support complet d'Unicode.
- Moteur de templates extensible.
- Système d'internationalisation.
- Système de test unitaires Python intégré.

Composants (suite)

- Système d'authentification extensible.

Composants (suite)

- Système d'authentification extensible.
- Interface d'administration dynamique.

Composants (suite)

- Système d'authentification extensible.
- Interface d'administration dynamique.
- Outil pour générer des flux RSS et Atom.

Composants (suite)

- Système d'authentification extensible.
- Interface d'administration dynamique.
- Outil pour générer des flux RSS et Atom.
- Système de commentaires flexible.

Composants (suite)

- Système d'authentification extensible.
- Interface d'administration dynamique.
- Outil pour générer des flux RSS et Atom.
- Système de commentaires flexible.
- Framework de sites pour exécuter plusieurs sites sur une même installation Django.

Composants (suite)

- Système d'authentification extensible.
- Interface d'administration dynamique.
- Outil pour générer des flux RSS et Atom.
- Système de commentaires flexible.
- Framework de sites pour exécuter plusieurs sites sur une même installation Django.
- Outil pour générer des Sitemaps Google.

Composants (suite)

- Système d'authentification extensible.
- Interface d'administration dynamique.
- Outil pour générer des flux RSS et Atom.
- Système de commentaires flexible.
- Framework de sites pour exécuter plusieurs sites sur une même installation Django.
- Outil pour générer des Sitemaps Google.
- Outil pour prévenir des attaques CSRF.

Développement en bref

- Créer les modèles, les classes objet.

Développement en bref

- Créer les modèles, les classes objet.
- Installation automatique en base de données.

Développement en bref

- Créer les modèles, les classes objet.
- Installation automatique en base de données.
- Jouer avec l'API.

Développement en bref

- Créer les modèles, les classes objet.
- Installation automatique en base de données.
- Jouer avec l'API.
- L'interface d'administration.

Développement en bref

- Créer les modèles, les classes objet.
- Installation automatique en base de données.
- Jouer avec l'API.
- L'interface d'administration.
- Définir les urls.

Développement en bref

- Créer les modèles, les classes objet.
- Installation automatique en base de données.
- Jouer avec l'API.
- L'interface d'administration.
- Définir les urls.
- Ecrire les vues.

Développement en bref

- Créer les modèles, les classes objet.
- Installation automatique en base de données.
- Jouer avec l'API.
- L'interface d'administration.
- Définir les urls.
- Ecrire les vues.
- Définir les templates.

Développement en bref

- Créer les modèles, les classes objet.
- Installation automatique en base de données.
- Jouer avec l'API.
- L'interface d'administration.
- Définir les urls.
- Ecrire les vues.
- Définir les templates.
- Ajuster et optimiser.

Système de sondage

- Créer le projet.

Système de sondage

- Créer le projet.
- Création des modèles: Question, Choix.

Système de sondage

- Créer le projet.
- Création des modèles: Question, Choix.
- Jouer avec l'API.

Système de sondage

- Créer le projet.
- Création des modèles: Question, Choix.
- Jouer avec l'API.
- L'interface d'administration.

Systeme de sondage

- Créer le projet.
- Création des modèles: Question, Choix.
- Jouer avec l'API.
- L'interface d'administration.
- Définir les urls.

Système de sondage

- Créer le projet.
- Création des modèles: Question, Choix.
- Jouer avec l'API.
- L'interface d'administration.
- Définir les urls.
- Ecrire les vues.

Système de sondage

- Créer le projet.
- Création des modèles: Question, Choix.
- Jouer avec l'API.
- L'interface d'administration.
- Définir les urls.
- Ecrire les vues.
- Définir les templates.

Systeme de sondage

- Créer le projet.
- Création des modèles: Question, Choix.
- Jouer avec l'API.
- L'interface d'administration.
- Définir les urls.
- Ecrire les vues.
- Définir les templates.
- Ajuster et optimiser.

Système de sondage

- Créer le projet.

Créer le projet

- Commande "django-admin.py startproject".

Créer le projet

- Commande "django-admin.py startproject".
- Lancement du serveur.

Créer le projet

- Commande "django-admin.py startproject".
- Lancement du serveur.
- Configuration de la base de données.

Créer le projet

- Commande "django-admin.py startproject".
- Lancement du serveur.
- Configuration de la base de données.
- Test de la connexion à la base de données.

Système de sondage

- Créer le projet.
- Création des modèles: Question, Choix.

- Commande "`django-admin.py startapp`".

Création des modèles

- Commande "django-admin.py startapp".
- Définition des modèles: Question, Choix.

- Commande "django-admin.py startapp".
- Définition des modèles: Question, Choix.
- Activation des modèles.

Création des modèles

- Commande "django-admin.py startapp".
- Définition des modèles: Question, Choix.
- Activation des modèles.
- Synchronisation de la base de données.

Système de sondage

- Créer le projet.
- Création des modèles: Question, Choix.
- Jouer avec l'API.

Jouer avec l'API

- Dans le shell.

Jouer avec l'API

- Dans le shell.
- Etoffer les modèles.

Jouer avec l'API

- Dans le shell.
- Etoffer les modèles.
- Activation des modèles.

Jouer avec l'API

- Dans le shell.
- Etoffer les modèles.
- Activation des modèles.
- Synchronisation de la base de données.

Système de sondage

- Créer le projet.
- Création des modèles: Question, Choix.
- Jouer avec l'API.
- L'interface d'administration.

Interface d'administration

- Activation de l'admin.

Interface d'administration

- Activation de l'admin.
- Entrer dans l'admin.

Interface d'administration

- Activation de l'admin.
- Entrer dans l'admin.
- Rendre les Questions modifiables dans l'admin.

Interface d'administration

- Activation de l'admin.
- Entrer dans l'admin.
- Rendre les Questions modifiables dans l'admin.
- Personnalisation des formulaires d'admin.

Interface d'administration

- Activation de l'admin.
- Entrer dans l'admin.
- Rendre les Questions modifiables dans l'admin.
- Personnalisation des formulaires d'admin.
- Ajouter les objets en rapport.

Interface d'administration

- Activation de l'admin.
- Entrer dans l'admin.
- Rendre les Questions modifiables dans l'admin.
- Personnalisation des formulaires d'admin.
- Ajouter les objets en rapport.
- Personnalisation de la liste des objets de l'admin.

Système de sondage

- Créer le projet.
- Création des modèles: Question, Choix.
- Jouer avec l'API.
- L'interface d'administration.
- Définir les urls.

Définir les urls

- Définition des urls dans urls.py.

Systeme de sondage

- Créer le projet.
- Création des modèles: Question, Choix.
- Jouer avec l'API.
- L'interface d'administration.
- Définir les urls.
- Ecrire les vues.

- Premières vues.

Ecrire les vues

- Premières vues.
- Vues qui font quelque chose.

- Premières vues.
- Vues qui font quelque chose.
- Introduction au raccourci *shortcut*.

Ecrire les vues

- Premières vues.
- Vues qui font quelque chose.
- Introduction au raccourci *shortcut*.
- Gestion d'erreur 404.

Système de sondage

- Créer le projet.
- Création des modèles: Question, Choix.
- Jouer avec l'API.
- L'interface d'administration.
- Définir les urls.
- Ecrire les vues.
- Définir les templates.

Définir les templates

- Utiliser le système de templates.

Définir les templates

- Utiliser le système de templates.
- Utilisation des *templatetags*.

Définir les templates

- Utiliser le système de templates.
- Utilisation des *templatetags*.
- Introduction au raccourci *shortcut*.

Définir les templates

- Utiliser le système de templates.
- Utilisation des *templatetags*.
- Introduction au raccourci *shortcut*.
- Gestion d'erreur 404.

Système de sondage

- Créer le projet.
- Création des modèles: Question, Choix.
- Jouer avec l'API.
- L'interface d'administration.
- Définir les urls.
- Ecrire les vues.
- Définir les templates.
- Création de formulaires.

- Ecrire un formulaire simple.

Systeme de sondage

- Créer le projet.
- Création des modèles: Question, Choix.
- Jouer avec l'API.
- L'interface d'administration.
- Définir les urls.
- Ecrire les vues.
- Définir les templates.
- Création de formulaires.
- Ajustements et optimisations.

Ajustements et optimisations

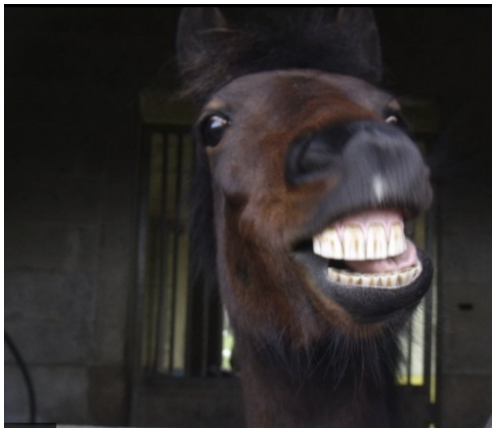
- Séparer les urls des applications.

Ajustements et optimisations

- Séparer les urls des applications.
- Utiliser les vues génériques.

Fin

Merci!



Avez-vous des questions?