

Mémento Django

Version du document : 0.1.1

Commandes de base de données

Installation des applications déclarées dans INSTALLED_APPS

```
python manage.py syncdb
```

Affichage des requêtes de création de tables pour une application

```
python manage.py sql monapplication
```

Vérification des modèles d'une application

```
python manage.py validate monapplication
```

Affichage de données initiales requises par une application (modèles + partie administration)

```
python manage.py sqlinitialdata monapplication
```

Affichage des requêtes de suppression de tables pour une application

```
python manage.py sqlclear monapplication
```

Affichage des requêtes de création d'index pour une application

```
python manage.py sqlindexes monapplication
```

Affichage des requêtes de suppression et création des tables de l'application

```
python manage.py sqlreset monapplication
```

Affichage des requêtes de remise à zéro des séquences/auto_increment des tables de l'application

```
python manage.py sqlsequencereset monapplication
```

Combinaison des commandes sql, sqlinitialdata, sqlindexes

```
python manage.py sqlall monapplication
```

Mise à jour de la base de données après modification du modèle

```
python manage.py reset monapplication
```

Commandes diverses

Lancement d'un shell propre au projet

```
python manage.py shell
```

Lancement d'un serveur Django

```
python manage.py runserver <port>
```

Commandes de création

Création d'un projet

```
django-admin.py startproject nomprojet
```

Création d'une application

```
python manage.py startapp nomapplication
```

Templates

block – Définit une zone permettant d'accueillir du texte

```
{% block content %}Contenu par défaut{% endblock %}
```

comment – Définit une zone de commentaire

```
{% comment %}Un commentaire{% endcomment %}
```

cycle – Alterne entre différentes valeurs

```
<tr class="{% cycle rouge,vert as couleurs %}">...</tr>
```

```
<tr class="{% cycle couleurs %}">...</tr>
```

debug – Affiche les informations de debugage{% debug %}

extends – Permet l'héritage d'un autre template{% extends "base.html" %}

filter – Applique un filtre sur un contenu

```
{% filter lower %}Cette Phrase en Minuscule{% endfilter %}
```

firstOf – Retourne la première valeur vraie ou rien si toutes sont fausses.

```
{% firstof var1 var2 var3 %}
```

for – Boucle sur une liste d'objets

```
{% for personne in personne_list %}
```

```
    {{ personne.name }}
```

```
{% endfor %}
```

if – Evalue une variable à Vrai ou Faux

```
{% if personne %}{{ personne }}{% endif %}
```

ifchanged – Indique si une valeur a été modifiée depuis la dernière boucle

```
{% ifchanged %}{{ name }}{% endifchanged %}
```

ifnotequal – Vérifie une égalité

```
{% ifnotequal v1 v2 %}bonjour{% endifnotequal %}
```

load – Charge une bibliothèque spécialisée {% load comments %}

now – Retourne la date courante {% now "jS F Y H:i" %}

regroup – Regroupe des objets en fonction d'un attribut

```
{% regroup personne by genre as personne_groupe %}
```

spaceless – Normalise les espaces dans le code html

```
{% spaceless %} ...code html... {% endspaceless %}
```

ssi - Inclue une page html {% ssi /home/html/fichier.html parsed %}

templatetag – Permet d'afficher des accolades de tag de template

```
{% templatetag openblock %}
```

withratio – Calcule et retourne un ration en fonction d'une valeur courante d'une valeur max et d'une constante

```
{% widthratio this_value max_value 100 %}
```

Type de données

AutoField : Nombre entier avec un auto incrément

BooleanField : Booléen

CharField : Chaîne de caractères

CommaSeparatedIntegerField : Nombres entiers séparés par une virgule

DateField : Date

DateTimeField : Date et heure

EmailField : Chaîne de caractère de type email

FileField : Fichier

FilePathField : Fichier limité à certains répertoires

FloatField : Nombre à virgule

ImageField : Fichier de type image

IntegerField : Nombre entier

IPAddressField : Chaîne de caractères de type adresse IP

NullBooleanField : Booléen avec valeur nulle possible

PhoneNumberField : Chaîne de caractères de type numéro de téléphone

PositiveIntegerField : Nombre entier positif

PositiveSmallIntegerField : Nombre entier positif de petite taille

SlugField : Chaîne de caractères utilisé en tant qu'étiquette

SmallIntegerField : Nombre entier de petite taille

TextField : Texte

TimeField : Heure

URLField : Chaîne de caractère de type URL

USStateField : Abréviation des états des États-Unis

XMLField : Texte de type XML

Options de la classe Meta

db_table : Indique le nom de la table qui sera utilisée pour le modèle en base de données

get_latest_by : Indique le champ Date ou DateTime à utiliser pour la méthode 'latest()' du modèle

order_with_respect_to : Indique que l'objet est triable via le champ indiqué

ordering : Indique l'ordre par défaut des données de la table

permissions : Permet de spécifier des permissions particulières pour le modèle

unique_together : Indique une unicité en fonction de plusieurs champs de l'objet

verbose_name : Nom du modèle humainement lisible

verbose_name_plural : Nom du modèle au pluriel humainement lisible

Options des types de données

null : La valeur peut être nulle si cette option = 'True'

blank : La valeur peut être laissée à vide lors de la saisie dans la partie administration si cette option = 'True'

choices : Indique une liste de valeurs possibles pour le champ

core : Indique que le champ est obligatoire sous peine de suppression de l'objet

db_column : Indique le nom du champ qui sera utilisé en base de données

db_index : Indique que ce champ est indexé en base de données

default : Indique la valeur par défaut du champ

editable : Si cette option = 'False' alors ce champ ne sera pas modifiable dans la partie administration ou via un formulaire utilisant les objets des classes AddManipulator ou ChangeManipulator

help_text : Étiquette d'aide dans la partie administration

primary_key : Le champ est clé primaire si cette option = 'True'

radio_admin : Indique qu'il sera utilisé des boutons radio à la place d'une liste dans la partie administration pour ce champ

unique : Le champ doit posséder des valeurs uniques si cette option = 'True'

unique_for_date : Le champ doit posséder des valeurs uniques pour chaque date renseignées dans un champ DateTime de la même table

unique_for_month : Le champ doit posséder des valeurs uniques pour chaque mois renseignées dans un champ DateTime de la même table

unique_for_year : Le champ doit posséder des valeurs uniques pour chaque années renseignées dans un champ DateTime de la même table

validator_list : Indique une liste de fonctions qui permettent de valider la valeur du champ

Options de la classe Admin

date_hierarchy : Permet la mise en place d'un calendrier dans la liste des objets sur le champ indiqué

fields : Tuple des champs à afficher

js : Indique une liste de fichiers javascript à inclure

list_display : Liste des champs visibles dans la liste des objets de ce modèle

list_display_links : Liste des champs « cliquables » pour la modification d'un objet

list_filter : Permet le filtrage à l'aide d'un menu qui s'affiche sur la droite de l'écran de la liste des objets

list_per_page : Nombre d'objet par page (par défaut 100)

list_select_related : Permet d'afficher les objets liés

ordering : Indique l'ordre d'affichage des objets dans l'administration

save_as : Propose un bouton de sauvegarde qui permet d'enregistrer un nouvel objet

save_on_top : Indique que les boutons de sauvegarde apparaîtront en haut de la page

search_fields : Indique sur quel champ on peut effectuer une recherche dans l'administration

Informations générales :

La plupart des informations contenues sur ce document ont été trouvées sur les sites internet suivants :

- www.djangoproject.com
- www.django-fr.org

Version : 0.1.1

Ce document provient du site Web www.skitoo.org/blog/

Licence :

Creative Commons Attribution-NonCommercial-ShareAlike 2.5 License

<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>