Les étapes Site/Appli Django

1. On créé un projet (toute l’architecture du dossier et sous-dossiers)

django-admin startproject monprojet

1. Lancer le serveur de développement

py manage.py runserver

1. Créer les chemins d’urls :

* Fichier urls.py
* Importer les vues *(les fonctions, les noms des pages web)*

1. Créer les vues : *(les fonctions, les rôles, les noms des pages web)*

* Fichier : views.py
* Importer render

1. Créer un dossier templates + sous-dossier contenant les fichiers html
2. Configurer les templates : *(pas besoins lorsqu’on créé les applications)*

* Ficher settings.py
* Importer os.path
* TEMPLATES = [

{

'DIRS': [os.path.join(BASE\_DIR,'monsite/templates'),

'APP\_DIRS': **True**,

},

]

1. Insérer les données dans un templates

* Dans un contexte (clé, valeur) depuis le fichier views.py
* Insérer ce contexte dans le templates, dans des filtres
* Voir documentation

1. Créer une application (plus tard…)
2. Créer les fichiers statiques

* un dossier nommé ‘static’ => un sous-dossier nommé => ‘css’ => un fichier css
* pareil pour les fichiers images, javscript et autres…
* ne pas oublier {% **load** static %}
* et src ou href="{% **static** 'chemin/d’accès' %}" ...
* Faire des réglages dans le fichier settings.py

STATICFILES\_DIRS = [

os.path.join(BASE\_DIR,'monsite/templates'),

]

1. Mettre en forme le site *(après)*

Par exemple :

* Utiliser un contexte, avec des filtres pour insérer des données dans un template
* Dans le dossier css, on peut stocker directement les images du site

1. Créer des modèles (les tables de la base de données)

* Créer une application
* Créer une application : py manage.py startapp monappli
* Configurer l’application :
  + Fichier : settings.py
  + INSTALLED\_APPS = [

'Nom de l’application',

]

* Dans fichier : models.py
* Créer des tables *(des classes)*
* Ouvrir le logiciel TablePlus
  + Ouvrir TablePlus =>
  + Create new connexion (nouvelle connexion) =>
  + Choisir la base de données (ex : Sqlite)
  + Remplir les infos et =>
  + Select file : choisir le fichier db.sqlite3 qui correspond à la base de données
* Lancer la migration : (pour crééer tout ce qui a trait à la base de donnée

py manage.py migrate

* + Sur TablePlus,faire :
  + Connection => reload (pour rafraîchir la base de données et voir les migrations obtenues)

1. Créer des migrations *(les tables correspondantes à nos modèles)*

py manage.py makemigrations monappli

* + C’est une commande qui met à jour la base de données
  + un fichier est créé par défaut contenant les détails des tables
  + c’est un fichier qui contiendra toute l’historique des modifications qu’on fera sur le modèle
  + faire ces migrations chaque fois que vous faites une modification dans nos modèles du fichier models.py

1. Appliquer les migrations créées avec makemigrations

py manage.py migrate monappli

* + cette commande permet de répercuter les changements effectués par makemigrations
  + et permet, grâce à cet historique, de recréer intégralement les tables supprimées
  + récupère les fichiers de migrations précédemment créés et les appliques à la base de données existante.

1. Le paramètre blank dans un formulaire

* blank=true signifie que le champ est requis (obligatoire)

1. La différence entre blank dans une table (un modèle)

* blank=true signifie, dans un formulaire, qu’un champs ne peut être vide,
* null=true signifie, au niveau de la table, qu’un champ (une colonne) peut être vide. *NB : à ne pas utiliser dans les champs (colonnes) de types* CharField ().

1. Faire des requêtes dans la base de données