INSTITUT SUPERIEUR DES ETUDES TECHNOLOGIQUES DE NABEUL

DEPARTEMENT TECHNOLOGIE DE L'INFORMATIQUE

TP1 : INSTALLATION DE L'ENVIRONNEMENT DE DEVELOPPEMENT POUR DJANGO

Objectif

Le but de ce TP est de présenter aux étudiants le framework Django de Python et leur apprendre l'installation de l'environnement nécessaire au développement.

1- Présentation de Django

Django est **un framework** pour Python qui permet de faciliter et accélérer le développement d'applications Web sécurisées et maintenables.
Gratuit et libre, Django a été publié en 2005 et possède déjà une communauté très active et une documentation assez riche.
Django met l'accent sur la réutilisation des composants (Don't Repeat Yourself), et fournit des fonctionnalités prêtes à l'emploi telles que l'authentification, la connexion à une base de données et les opérations CRUD (Create Read Update Delete). Devenu très populaire, Django est utilisé par plusieurs sociétés connues tels qu'Instagram, Pinterest, Mozilla, Spotify et même la NASA.

2- Qu'est-ce qu'un framework ?

Un framework est un ensemble d'outils qui simplifie le travail d'un développeur. Traduit littéralement de l'anglais, un framework est un « cadre de travail ». Il apporte les bases communes à la majorité des programmes ou des sites web. Celles-ci étant souvent identiques (le fonctionnement d'un espace membres est commun à une très grande majorité de sites web de nos jours), un développeur peut les réutiliser simplement et se concentrer sur les particularités de son projet.

Il s'agit donc d'un ensemble de bibliothèques coordonnées, qui permettent à un développeur d'éviter de réécrire plusieurs fois une même fonctionnalité, et donc d'éviter de réinventer constamment la roue. Inutile de dire que le gain en énergie et en temps est

considérable .

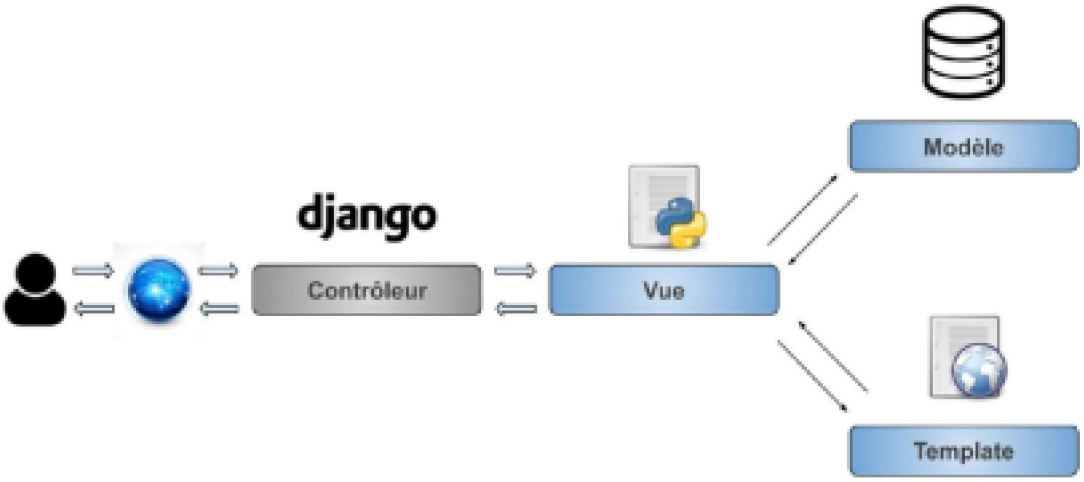
3- Fonctionnement de django

Django fonctionne avec le modèle de conception MVT (Model View Template).

– Modèle : Il permet d'accéder aux données à utiliser, généralement à partir d'une base de données.

– Vue : Elle permet la visualisation des données selon les actions de l'utilisateur.

– Template : Un fichier HTML (appelé « gabarit ») récupéré par la vue, il sera analysé et exécuté par le framework, comme s'il s'agissait d'un fichier avec du code. Django fournit un moteur de templates très utile qui permet, dans le code HTML, d'afficher des variables, d'utiliser des structures conditionnelles (if/else) ou encore des boucles (for), etc.



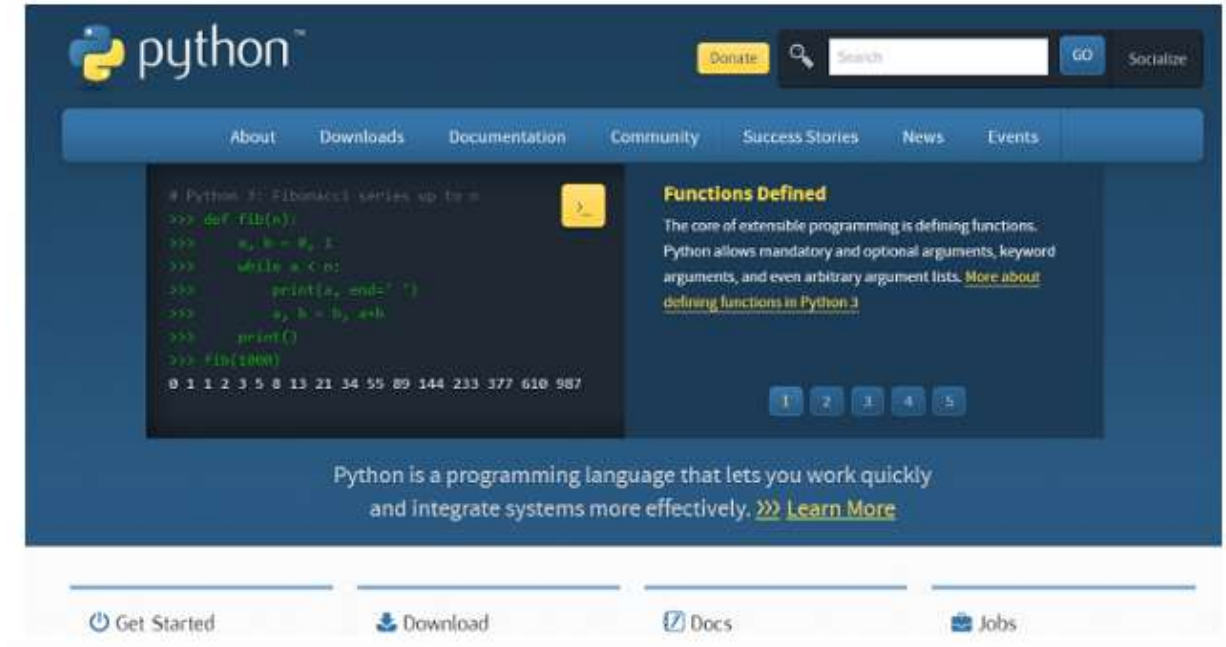
NB : Dans une architecture selon le modèle MVC (Model View Controller), le contrôleur prend en charge tous les événements de l'utilisateur (accès à une page, soumission d'un formulaire, etc.). Il récupère les données demandées dans le modèle, effectue les traitements éventuels sur ces données, puis les transmet à la vue pour l'affichage. Tandis que Django gère lui-même la partie contrôleur.

3- Installation de l'environnement

Pour travailler avec Django, il convient d'abord d'installer Python.

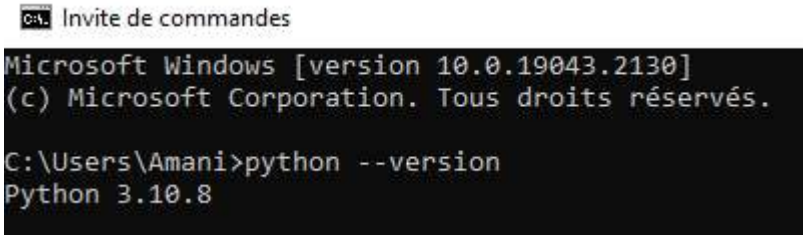
Etape 1 : Téléchargement et installation de Python

On commence par télécharger la dernière version en date de Python, gratuitement sur le site officiel : https://python.org/download/.



Après avoir suivi l'assistant d'installation, on vérifie que Python est bien installé grâce à

la commande : `python –version.`



Etape 2 : Installer Django

Django peut être installé facilement en utilisant pip (Package Installer for Python) ; un système de gestion de paquets utilisé pour installer et gérer des librairies écrites en Python.

Dans l'invite de commande, exécutez la commande suivante : **`pip install django.`**

Cela téléchargera et installera Django.

```
C:\Users\Amani>pip install django
Collecting django
  Downloading Django-4.1.2-py3-none-any.whl (8.1 MB)
     ---------------------------------------- 8.1/8.1 MB 741.3 kB/s eta 0:00:00
Collecting sqlparse>=0.2.2
  Downloading sqlparse-0.4.3-py3-none-any.whl (42 kB)
     ---------------------------------------- 42.8/42.8 kB 349.8 kB/s eta 0:00:00
Collecting tzdata
  Downloading tzdata-2022.5-py2.py3-none-any.whl (336 kB)
     ---------------------------------------- 336.7/336.7 kB 563.8 kB/s eta 0:00:00
Collecting asgiref<4,>=3.5.2
  Downloading asgiref-3.5.2-py3-none-any.whl (22 kB)
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.5.2 django-4.1.2 sqlparse-0.4.3 tzdata-2022.5

[notice] A new release of pip available: 22.2.2 -> 22.3
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Une fois l'installation terminée, vous pouvez vérifier que Django est bien installé en exécutant `django-admin --version`.

```
C:\Users\Amani>django-admin --version
4.1.2
```

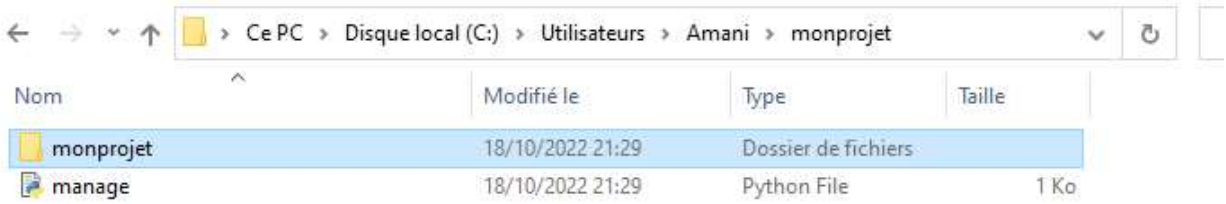
NB :
En tant que débutant, on peut installer Django directement sur notre PC. Mais, plus tard, il est recommandé d'avoir un environnement virtuel dédié pour chaque projet Django afin de garantir leurs indépendances. Il faut donc créer un environnement virtuel avant d'installer Django. L'un des outils permettant de gérer un environnement virtuel est venv, qui est inclus dans Python.
Avec venv, il suffit d'accéder au dossier où créer un projet, puis créer un environnement virtuel en tapant la commande : `py -m venv monprojet.`

4- Structure d'un projet avec Django

En plus de l'architecture MVT, Django introduit le développement d'un site sous forme de projet. Chaque site web conçu avec Django est considéré comme un projet, composé de plusieurs applications.

Une fois créé à l'aide de la commande `django-admin startproject nom_projet`,
Dans le dossier principal `monprojet`, nous retrouvons deux éléments :

- un fichier `manage.py`
- un autre sous-dossier nommé également `monprojet`



telle que la structure est suivante :

```
monProjet/
 manage.py
 monProjet/
 __init__.py
 settings.py
 urls.py
 asgi.py
 wsgi.py
```

Les fichiers créés dans le projet sont :

– manage.py : un utilitaire en ligne de commande pour interagir avec ce projet.
__init__.py : un fichier vide pour indiquer à Python que ce répertoire doit être considéré comme un paquet.
– settings.py : contient les réglages et configuration de ce projet Django. – urls.py : contient les déclarations des URL de ce projet.
– asgi.py et wsgi.py : points d'entrée pour les serveurs Web compatibles ASGI et WSGI pour déployer le projet.

5- Structure d'une application avec Django

Une application consiste en un dossier (dans un projet) contenant plusieurs fichiers de code, chacun étant relatif à une tâche du modèle MVT que nous avons vu. En effet, chaque bloc du site web est isolé dans un dossier avec ses vues, ses modèles et ses schémas d'URL.

Pour créer une application, il faut accéder au projet existant et exécuter la commande : `py manage.py startapp nom_app`. Cela créera un autre répertoire dans le projet telle que la structure est suivante :

```
monProjet/
 manage.py
 monProjet/
 monApp/
 migrations/
                __init__.py
 __init__.py
 admin.py
 apps.py
```

```
models.py
tests.py
views.py
```

Les fichiers créés dans le répertoire de l'application sont :

– __init__.py indique à Python de traiter le répertoire comme un package Python. – admin.py contient les paramètres des pages d'administration de Django.

– apps.py contient des paramètres pour la configuration de l'application.

– models.py contient des classes que l'ORM de Django convertit en tables de BD. – tests.py contient des classes de test.

– views.py contient des fonctions et des classes qui gèrent les données affichées dans les templates HTML.

6- Travail demandé

Dans la suite, on utilisera Visual Studio Code pour réaliser les exercices.

Exercice 1

1) Ouvrez une fenêtre d'éditeur de commande et créez un projet Django intitulé DSI22.

2) Accédez au dossier dsi22 et exécutez la commande `py manage.py runserver`.

3) Ouvrez un navigateur et accédez à l'adresse suivante : `127.0.0.1:8000`.

4) Assurez-vous que la page de test Django s'affiche avec succès.

Exercice 2

1) A l'aide de VS Code, ouvrez le projet DSI22 précédent, puis, ouvrez un terminal et créez une application « hello » dans le projet DSI22.

2) Dans le fichier « views.py » copiez le code suivant :

```
from django.shortcuts import render
from django.http import HttpResponse

def index(request):
 return HttpResponse("Voici mon premier exemple Django")
```

3) Créez un nouveau fichier nommé « urls.py » dans le dossier « hello » et écrivez ce code dedans :

```
from django.urls import path
from . import views

urlpatterns = [
 path('', views.index, name='index'),
]
```

4) A présent, ouvrez le fichier « urls.py » dans le dossier du projet « DSI22 » et remplacez son contenu par le code suivant :

```
from django.contrib import admin
from django.urls import include, path

urlpatterns = [
 path('hello/', include('hello.urls')),
 path('admin/', admin.site.urls),
]
```

5) Dans le terminal, remontez au niveau du projet et lancez le serveur.
6) Ouvrez un navigateur et accédez à l'adresse suivante : `127.0.0.1:8000/hello`.
7) Assurez-vous que le message du HttpResponse (question 2) s'affiche avec succès.

Exercice 3

1) A l'aide de VS Code, créez sous « hello », un dossier « templates » dans lequel ajoutez un fichier « first.html » qui contient le code suivant :

```
<!DOCTYPE html>
<html>
<body>

<h1>Bonjour name={{name}}!</h1>
<p>Voici mon premier template avec Django</p>

</body>
```

2) Afin de prendre en considération le nouveau template, modifiez le fichier « views.py » comme suit :

```
from django.http import HttpResponse
from django.template import loader

def index(request):
 context = {
  'nom': 'Foulen', # écrire votre nom ici
  }
 template = loader.get_template('first.html')
 return HttpResponse(template.render(context,request))
```

3) A présent, on indique à Django qu'une nouvelle application a été créée en modifiant le paramètre « INSTALLED_APPS » dans le fichier « settings.py» sous « DSI22 » en ajoutant la dernière ligne :

```
INSTALLED_APPS = [
 'django.contrib.admin',
 'django.contrib.auth',
 'django.contrib.contenttypes',
 'django.contrib.sessions',
 'django.contrib.messages',
 'django.contrib.staticfiles',
 'hello.apps.HelloConfig'
]
```

4) Maintenant, exécutez la commande `py manage.py migrate`.

5) Enfin, lancez le serveur sous le répertoire « DSI22 » et réaffichez la page web `127.0.0.1:8000/hello`.