

Formation Django,développement Web avec Python

Sujet des travaux pratiques

Matthieu Falce

Décembre 2022

1 Programmation orientée objet

1.1 Formation

Dans cette section vous allez améliorer le même code petit à petit afin d'avoir une application permettant de gérer les formations.

1.1.1 Tout commence par des humains

Concepts : 1) création de classes 2) méthodes et attributs 3) méthodes magiques

Créez une classe **Personne**. Elle va contenir les informations d'une personne (en gros les informations d'une carte d'identité).

Une personne doit pouvoir dire si elle est majeure. Attention, la limite d'âge de la majorité peut changer et doit donc être dynamique.

```
p1 = Personne(naissance=1990, nom="Matthieu")
p2 = Personne(naissance=2015, nom="Paul")
```

```
assert p1.est_majeur()
assert not p2.est_majeur()
```

```
Personne.AGE_MAJORITE =40
```

```
print(p1.est_majeur() == False)
print(p2.est_majeur() == False)
```

La personne doit pouvoir se présenter en faisant :

```
p1 = Personne(naissance=annee_naissance, nom=nom)
print(p1)
```

1.1.2 Attributs, Getters, Setters

Concepts : 1) **properties**

L'âge d'un Eleve ne se recalcule pas. Comment régler ce problème de la façon la moins intrusive possible ? On veut pouvoir faire

```
p1 = Personne("1", 1950)
print(p1.age == 68)
```

```
p1.age = 10
print(p1.annee_naissance == 2008)
```

```
p1.annee_naissance = 0
print(p1.age == 2018)
```

1.1.3 Puis des élèves et des profs

Concepts : 1) héritage 2) surcharge

Dessinez les diagrammes de classes correspondant à ce qui est demandé.

Créez **Eleve** et **Prof** qui sont des **Personnes**.

Un **Prof** doit pouvoir dire si un élève a payé ses frais de scolarité.

1.1.4 Et enfin, une formation

Concepts : 1) composition

Une **Formation**, c'est des élèves et des profs en même temps.

Un élève doit être majeur et avoir payé ses frais de scolarité pour participer. Modifiez le prof pour qu'il sache si un élève peut s'inscrire ou pas.

1.2 Convertisseur de température

Concepts : 1) **property** 2) accès aux attributs

Comment implémenter un convertisseur de température Celsius / Fahrenheit ?

Pour rappel :

$$T_{[^\circ C]} = (T_{[^\circ F]} - 32) * 5/9$$

$$T_{[^\circ F]} = T_{[^\circ C]} * 9/5 + 32$$

Questions :

1. essayer de gérer 2 attributs simultanément. Quels problèmes rencontre-t-on ?
2. essayer la même chose en utilisant les **properties**. Quel est l'intérêt de cette solution ?

2 Django

2.1 Modélisation SQL

Concepts : 1) réflexion de base sur la modélisation 2) sélectionner des informations importantes à garder

Vous travailler dans le département IT d'un journal qui a décidé de se moderniser. Vous devez faire la modélisation des bases de données.

Vous avez à stocker :

- des journalistes
- des articles potentiellement écrits par plusieurs journalistes
- des commentaires sur les articles

En utilisant votre bon sens, modélisez la base de données à l'aide de ces informations.

Questions :

1. quelles seront les tables à créer
2. quelles seront les informations à stocker dans chaque table ? Et quel type ?
3. quelles sont les différents types d'associations entre les tables ?
4. faire un schéma

2.2 Création d'un projet

Concepts : 1) gestion d'un projet python 2) commande de base de django 3) création d'une application

Vous allez créer un projet django pour comprendre l'architecture des dossiers.

Étapes :

1. installer python et django
2. `django-admin startproject mysite`
3. `python manage.py startapp myapp`
4. `python manage.py runserver`

Questions :

1. où faut-il exécuter ces commandes ?
2. que font ces différentes commandes ?
3. pourquoi les utiliser plutôt que créer les fichiers et dossier à la main ?
4. quel est l'impact des différents noms de dossiers ?
5. à quel endroit allons nous écrire nos routes, modèles et vues ?
6. comment supprimer un projet django ?

2.3 Tutoriel

Concepts : 1) découvrir django

La documentation de django est *extrêmement* bien faite, suivez le tutoriel officiel (parties 1 à 4) pour bien démarrer : <https://docs.djangoproject.com/fr/4.1/intro/tutorial01/>

2.4 Différentes vues

Concepts : 1) routage d'URL 2) FBV 3) CBV 4) templates

Créez un nouveau projet `demo_vues` et créez une application permettant de router 3 vues.

Vous allez générer un document HTML qui va afficher une liste de noms de personnes (stockés comme un liste de chaîne de caractères en python : `["Matthieu", "Pierre", "Ernest"]`)

Questions :

1. faites une fonction de vue FBV qui fait le rendu en pure python et retourne une `HttpResponse`
2. faites une fonction de vue FBV qui fait le rendu d'un template sans utiliser de raccourcis django
3. faites une fonction de vue FBV qui fait le rendu d'un template en utilisant le raccourcis django
4. faites une classe de vue CBV / vue générique (`TemplateView`)

2.5 Cynisme

Concepts : 1) gestion d'un projet python 2) commandes de management 3) tâches planifiées 4) personnalisation de l'admin

Vous en avez marre d'oublier les anniversaires de vos amis. Créez un site permettant de leur envoyer un mail automatiquement le jour de leur anniversaire sans ne plus avoir à y penser.

Étapes :

1. créer un nouveau projet / application

2. modéliser les données (nom / date / email / ...)
3. activez l'admin et enregistrez quelques anniversaires
4. faites un template pour un mail d'anniversaire et rajoutez une action dans l'admin pour envoyer ce mail à votre "ami"
5. faites un formulaire pour ajouter un nouvel anniversaire sans passer par l'admin

Questions :

1. comment faire pour automatiser ce processus d'envoi de mails ?

2.6 Raccourcisseur d'URL

Concepts : 1) gestion d'un projet python 2) création d'une application 3) gestion des redirection HTTP 4) gestion des migrations

Vous allez créer un site permettant de raccourcir les URL, un peu comme `bit.ly`. En gros : vous accédez à une page `http://localhost:8000/a` et vous êtes redirigé vers `https://google.com`, `http://localhost:8000/b` vous renvoie vers `https://bing.com`, ... L'administrateur peut gérer les URLs depuis le backoffice.

Étapes :

1. testez un raccourcisseur d'URL si vous ne connaissez pas le concept
2. faire un schéma permettant de comprendre le parcours des requêtes
3. créer un nouveau projet django
4. modélisez les informations nécessaires
5. créez les urls / vues pour avoir un prototype
6. le service marketing est intéressé par le projet et veut s'en servir pour avoir des statistiques d'accès aux liens, rajouter les : nombre d'accès, date, heure, `user agent`, ...

Questions :

1. quels sont les différents types de redirection HTTP ? quelles sont leurs différences ?

2.7 Fil rouge

Concepts : 1) gestion d'un projet python 2) commande de base de django 3) création d'une application complète 4) utilisation des ressources statiques 5) création automatisée de fausses données 6) utilisation d'applications django

Vous allez reprendre le projet de modélisation SQL (le journal) et l'implémenter en django.

Étapes :

1. créer le projet / l'app
2. implémentez la modélisation, tester d'enregistrer des articles / journalistes / commentaires dans un shell et dans l'admin
3. customisez l'admin pour permettre de filtrer par année / journaliste / nombre de commentaires
4. faite le site du journal (réfléchissez bien à l'organisation de vos templates) :
 - une page d'accueil listant les articles par ordre antéchronologique
 - une page de détail permettant de lire un article
 - des pages d'archive annuelle / mensuelle
 - des pages représentant tous les articles écrits par chaque journalistes
5. faite un backoffice permettant à un journaliste
 - de se loguer
 - de voir ses articles
 - d'écrire un article

6. maintenant que le site est fonctionnel, customiser le avec du CSS / bootstrap...
7. rajouter un système de commentaires en base des articles (listing + formulaire de soumission)
8. mettez en place des requêtes AJAX pour ne pas recharger la page en cas de soumission de commentaires
9. créez une multitude de faux articles en utilisant la bibliothèque avec **factoryboy** (<https://factoryboy.readthedocs.io/en/stable/orms.html#the-djangomodelfactory-subclass>) et ajouter un infinite scrolling (<https://simpleisbetterthancomplex.com/tutorial/2017/03/13/how-to-create-infinite-scroll-with-django.html>)

2.8 Auberge espagnole

Concepts : 1) comprendre la notion d'application django et leur indépendance

Étapes :

1. vous allez créer un nouveau projet
2. copiez toutes les applications que vous avez codé jusqu'à présent dedans
3. essayez d'appeler les vues de chacune
4. faites une page d'accueil permettant d'accéder à chacune des vues

Questions :

1. que se passe-t-il si plusieurs vues de différentes apps ont le même schéma d'URL
2. à quoi servent les names les urls de `url.py`
3. à quoi sert le namespace dans `url.py`