

# Formation Django,développement Web avec Python

## Correction des travaux pratiques

Matthieu Falce

Décembre 2022

## 1 Programmation orientée objet

### 1.1 Formation

Voir le code de correction situé : `corrections/poo/formation.py`

### 1.2 Convertisseur de température

Réponses :

1. c'est une mauvaise idée de dupliquer les informations, quelle est la source de vérité? En plus, en python, le manque d'encapsulation permet de modifier les deux attributs indépendamment, ce qui peut complètement gêner le code
2. `property` c'est la vie 😊. On peut simuler des getters et setters en les manipulant comme des attributs

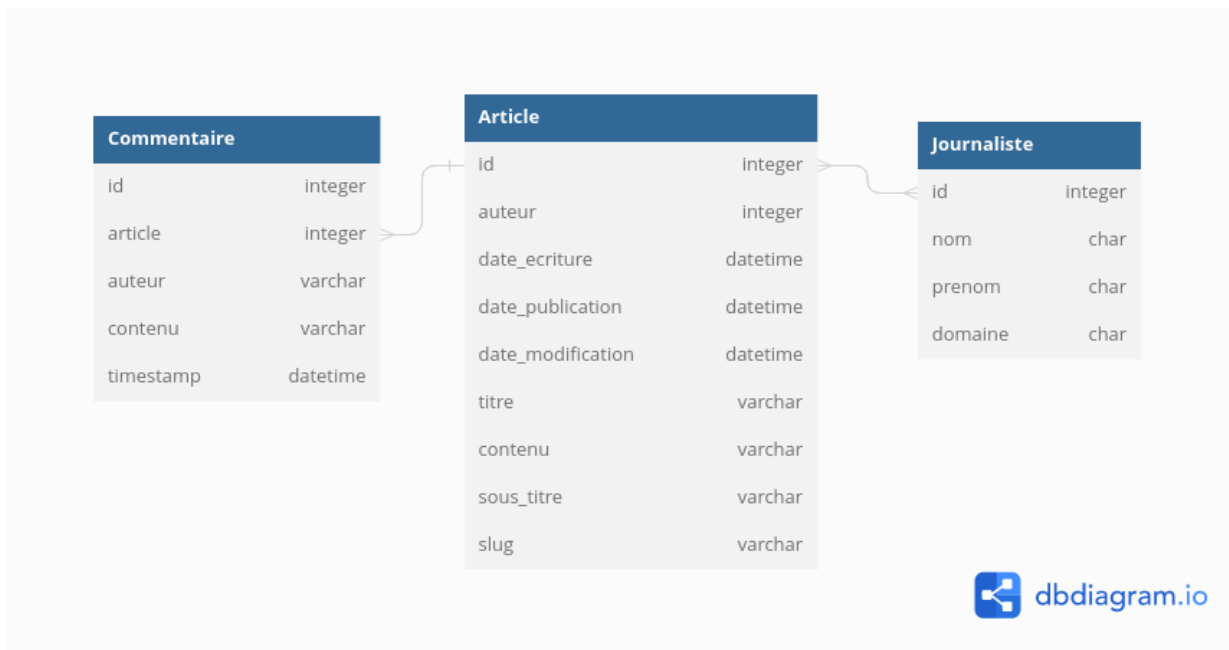
Voir le code de correction situé : `corrections/poo/convertisseur.py`

## 2 Django

### 2.1 Modélisation SQL

Réponses :

1. 3 tables (journalistes, article et commentaire)
2. voir dessin ci-dessous
3. N article peuvent être écrits par N journalistes, N commentaires peuvent être affichés pour 1 article



## 2.2 Création d'un projet

Réponses :

1. la première dans le dossier où l'on veut avoir le projet, pour les autres, il faut aller dans ce dossier / projet pour accéder à `manage.py`
2. `startproject` crée un projet, `startapp` crée une application et `runserver` lance un serveur web de développement
3. on pourrait le faire à la main, mais les commandes permettent de le faire plus facilement
4. l'architecture d'un projet django est toujours la même, `mysite` est le parent, il contient 2 dossiers `mysite` avec les paramètres du projet et `myapp` avec les paramètres de l'application
5. globalement, une fois la configuration faite dans `mysite`, nous ne toucherons plus qu'au dossier `myapp`
6. on supprime le dossier du projet tout simplement

## 2.3 Tutoriel

Voir le code du tutoriel.

## 2.4 Différentes vues

Voir code de correction ...

## 2.5 Cynisme

Réponses :

1. Ce n'est pas évident. Le serveur django tourne en théorie tout le temps, il peut donc gérer l'envoi des mails de façon autonome. Cependant, on ne peut pas facilement appeler une fonction à un moment précis avec django nativement, je vois 3 solutions :
  - mettre en place une commande de management qui va déterminer quels emails envoyer et un cron qui va être appelé toutes les N mins pour envoyer les mails (solution niveau OS)
  - utiliser un service comme <https://cronhooks.io> pour avoir une callback sur une vue précise et donc envoyer un email. On configurera les hooks avec leur API juste après avoir sauvegardé un nouvel anniversaire. Remarque : dans notre cas, tant que l'application n'est pas déployée, cronhook ne pourra pas accéder à notre instance django...

- la solution la plus complexe / pérenne : utiliser un outil fait pour ça comme `celery` :  
<https://docs.celeryq.dev/en/stable/userguide/periodic-tasks.html>

Voir code situé ...

## 2.6 Raccourcisseur d'URL

Réponses :

1. il existe 2 grandes catégories de redirection, il faut faire attention quand on en utilise car le navigateur peut les mettre en cache et aller directement à l'URL finale<sup>1</sup> :
  - les 301 : redirection permanent
  - les 302 : redirection temporaire (ce que nous voulons utiliser pour ce projet)

Voir le code ...

## 2.7 Fil rouge

Voir code ...

## 2.8 Auberge espagnole

Les applications django sont indépendantes. Il ne faut pas oublier de les intégrer au `INSTALLED_APPS` des `settings.py` et tout devrait bien fonctionner. Réponses :

1. il y a un conflit, mais pas de panique, vous pouvez préfixer les applications avec un motif unique (`/app1/vue1`)
2. ils servent à nommer les URL. Ce nom est prévu pour rester unique et permanent, contrairement à l'URL qui pourrait changer. N'hésitez pas à appeler vos urls par leur nom plutôt que leur motif, c'est la bonne pratique django
3. le namespace sert à définir un préfixe aux apps, pour éviter des conflits au niveaux des noms dans les URL, cela est utile quand vous avez beaucoup de routes dans des applications différentes

---

1. <https://realpython.com/django-redirects/#temporary-vs-permanent-redirects>