

# Présentation et utilisation des classes

---

## Les modules à télécharger et/ou importer

---

- `os` : permet d'interagir avec le système d'exploitation
- `requests` : permet d'envoyer une requête HTTP pour faciliter l'interaction
- `reportlab` : contient `canvas` qui permet de créer un fichier pdf
- `ebooklib` : permet d'écrire et de lire un fichier epub
- `PyPDF2` : permet de travailler avec des pdf, notamment avec `PdfReader` pour le lire
- `langdetect` : permet de détecter la langue d'un texte
- `shutil` : permet la gestion de fichiers et de répertoire
- `bs4` : contient `BeautifulSoup` pour analyser le contenu de l'url
- `urllib` : permet de lier des chemins pour une url fiable

## Base\_livre & Base\_bibli

---

Base\_livre :

- sert de base pour la classe EPUB et la classe PDF
- aucun objet ne doit être une instance directe de cette classe ( rien ne se passerait )
- renvoie un message d'erreur si l'une des fonctions nécessaires ( `'__init__'`, `'type'`, `'titre'`, `'auteur'`, `'langue'`, `'sujet'` et `'date'` ) n'as pas été définie dans ses sous-classes

Base\_bibli :

- sert de base pour la classe `simple_bibli` et ses sous-classes
- aucun objet ne doit être une instance directe de cette classe ( rien ne se passerait )
- renvoie un message d'erreur si l'une des fonctions nécessaires ( `'__init__'`, `'ajouter'`, `'rapport_livres'` et `'rapport_auteur'` ) n'as pas été implémentée dans ses sous-classes

## Fonctions

---

- `telecharger_url` : télécharge le contenu de l'url dans un dossier défini en argument et renvoie le chemin du fichier téléchargé
- `_generer_pdf` : génère un fichier pdf
- `_generer_epub` : génère un fichier epub

Toutes ces fonctions sont utilisées dans les classes, elles n'ont donc pas besoin d'être appelées par l'utilisateur.

## EPUB & PDF

---

- sont des sous-classes de `base_livre` , elles héritent donc de ses fonctions qui ont été redéfinies
- prennent en argument un fichier local ou une url
- si une url est en argument, elles téléchargent le fichier correspondant à l'url
- définissent pour le livre : son type ( ici PDF ou EPUB ), son titre, son auteur, sa langue, son sujet, sa date, avec 'inconnu' si l'information n'a pas été trouvée
- pour chacune des données du livre, une fonction existe et peut être appelée pour donner l'information demandée

Pour qu'une instance de PDF ou EPUB soit créée, la définir comme :

```
Livre1 = PDF(source) / Livre1 = EPUB(source)
```

Avec 'Livre1' le nom souhaité et 'source' le chemin du fichier local ou une url

Pour obtenir une des données sur l'instance, utiliser :

```
Livre1.auteur() / Livre1.date() / Livre1.type() / Livre1.titre() / Livre1.langue() ou Livre1.sujet()
```

## simple\_bibli

---

- est une sous-classe de `base_bibli`, elle hérite donc de ses fonctions qui ont été redéfinies
- prend en argument un dossier
- crée le dossier si il est inexistant
- définit une liste des livres du dossier et y ajoute les livres déjà présents dans le dossier en créant une instance de PDF pour les fichiers .pdf, et une instance de EPUB pour les fichiers .epub ( attention à ce que vos fichiers finissant par .pdf ou .epub soient bien des pdf ou epub, vos fichiers doivent être fiables )
- les fonctions ' \_\_init\_\_ ', 'ajouter', 'rapport\_livres' et 'rapport\_auteurs' ont été définies

### **Comment l'utiliser ?**

```
Bibliothèque1 = simple_bibli('dossier1')
```

ajouter : Elle prend en argument un livre et l'ajoute au dossier et à sa liste si jamais il n'y est pas déjà.

```
Bibliothèque1.ajouter(Livre1)
```

rapport\_livres: Elle prend en argument le format et le nom du rapport. Elle crée au format demandé ( pdf ou epub seulement) un rapport des livres contenus dans le dossier, avec pour titre le nom donné en argument, et pour chacun des livres : son titre, son auteur, son format (pdf ou epub) et sa ressource.

```
Bibliothèque1.rapport_livres("PDF", 'liste_livres')
```

rapport\_auteurs: La fonction prend en argument le format et le nom du rapport. Elle crée au format demandé (pdf ou epub seulement) un rapport des livres contenus dans le dossier par auteur, avec pour titre le nom donné en argument, et pour chacun des auteurs : la liste de ses livres présents dans la bibliothèque avec pour chacun des livres : son titre, sa ressource et son type (pdf ou epub).

```
Bibliothèque1.rapport_auteurs("PDF", 'liste_auteurs')
```

## bibli

---

- est une sous-classe de simple\_bibli, elle hérite donc de ses fonctions
- prend en argument un dossier
- a une fonction 'alimenter' supplémentaire à simple\_bibli

### **Comment l'utiliser ?**

Bibliothèque2 = bibli('dossier2')

alimenter : La fonction prend en argument une url, crée une instance de PDF ou de EPUB à partir des pdf ou epub référencés dans la page web puis l'ajoute au dossier et à sa liste.

Bibliothèque2.alimenter(url)

## bibli\_scrap

---

- est une sous-classe de bibli, elle hérite donc de ses fonctions
- prend en argument un dossier
- une liste des url visitées lui est attribuée
- a une fonction 'scrap' supplémentaire à simple\_bibli

### **Comment l'utiliser ?**

Bibliothèque3 = bibli\_scrap('dossier3')

scrap : La fonction réalise un web scraping destiné à alimenter la bibliothèque. Cette méthode prend en argument une url, une profondeur et un nombre maximum de fichier à télécharger (ainsi qu'un nombre initial de lien ajouté qui est initié à zéro). Elle récupère la page web référencée par url, puis télécharge tous les ressources pdf et epub qui y sont référencées. Ensuite, elle extrait de cette page tous les liens vers d'autres pages web, et réitère le processus précédent sur chacune d'elles. Le processus se réitère jusqu'à ce que l'un des critères d'arrêt soit vérifié.

Bibliothèque3.scrap(url, profondeur, nbmax)