IFT 3225
Rapport de Projet 1

Mohamed Terbaoui
Mehdi Qostali

8 mars 2025

1 Objectif du Projet

Ce travail propose deux commandes essentielles : `extract` et `genere`. L'idée est de collecter des ressources (images et vidéos) d'une page HTML et, par la suite, de générer une page HTML5 dynamique qui les affiche sous différentes formes (tableau, galerie, carrousel). Les scripts sont fonctionnels dans l'environnement Linux du DIRO et respectent les normes décrites dans le cahier des charges.

2 Rôle des membres

Mohamed a travaillé sur la commande **extract**, la page HTML et le script, tout en supervisant et en corrigeant l'intégralité du travail.

Mehdi a travaillé sur la commande **genere**, a apporté une contribution minimale à `extract`, et a pris en charge l'élaboration du rapport PDF.

3 Liens des Pages Traitées et Pages Générées

Voici trois pages HTML sources et les pages générées correspondantes qui sont héberges au DIRO :

1. **URL 1 :**
   `https://zone.votresite.ca/site-web-blogue/photo-video-audio/photos-et-videos:-quand-en-integrer-sur-votre-site-et-ou-vous-les-procurerr/gCsl7aSdzo/`
   Page web générée :
   `https://www-ens.iro.umontreal.ca/hiver/~terbaoum/ImageAndVideoScrapper/page1.html`
2. **URL 2 :**
   `http://www-labs.iro.umontreal.ca/~felipe/brand_new_home/creative-design/public_html/index.php?lg=fr`
   Page web générée :
   `https://www-ens.iro.umontreal.ca/hiver/~terbaoum/ImageAndVideoScrapper/page2.html`

3. **URL 3 :**
   `https://www.w3schools.com/html/html5_video.asp`
   Page web générée :
   `https://www-ens.iro.umontreal.ca/hiver/~terbaoum/ImageAndVideoScrapper/page3.html`

Dans chaque page créée, on retrouve une liste complète des ressources et un mécanisme JavaScript pour passer d'une présentation en tableau à une vision galerie ou carrousel.

4 Langages et Librairies Utilisés

— **Python 3** pour les scripts `extract.py` et `genere.py`.
— **argparse** pour définir et lire les arguments de la commande extract.py.
— **BeautifulSoup4 (bs4)** pour parser les balises HTML (`<img>` et `<video>`).
— **Requests** ou `urllib` pour récupérer les pages web.
— **HTML5, CSS et JavaScript** pour la page générée par `genere.py`.
— Aucune commande CSS dans le HTML : la feuille de style est externe.
— Le JavaScript (DOM) est séparé du code HTML, gérant l'apparition du carrousel, de la galerie, etc.

5 Traitements Particuliers

— **Gestion des fichiers .svg** : S'ils apparaissent parmi les balises <**img**>, on va les traiter comme des images classiques (à afficher).
— **Filtrage par regex (`-r <regex>`)** : Permet de n'extraire que les ressources dont le nom de fichier matche le String fournie (comme mentionné dans le projet on a juste implémenter pour des textes et pas des expressions régulieres) .
— **Alt et Src** : Les attributs `alt` et `src` des images sont listés dans l'ordre prévu, et pour les vidéos, on récupère `src` ou le premier `<source>`.

6 Compatibilité et Modes d'Exécution

Les deux commandes sont :

Commande `extract`

— **-r** <**regex**> : Filtre sur le nom des ressources (images/vidéos).
— **-p** <**path**> : Copie localement les ressources dans `<path>`.
— **-i** : Ignore les `<img>`.
— **-v** : Ignore les `<video>`.
— **-h** : Affiche l'aide (synopsis + auteurs).

Exemple d'utilisation :

```
python3 extract.py -r "jpeg" -p "mypath" "http://www-labs.iro.umontreal.ca/~
    felipe/..."
```

Commande genere

— Lit en entrée standard la liste produite par **extract.py**.
— **-h** : Affiche l'aide (synopsis + auteurs).

Exemple d'utilisation :

```
python3 extract.py "https://zone.votresite.ca/..." | python3 genere.py "mapage.
    html"


Depuis le rpertoire au DIRO (par ex. /www/ens/hiver/terbaoum/public_html/
    ImageAndVideoScrapper), on peut excuter :

python3 extract.py -r "regex" -p "path" -i -v <url> | python3 genere.py [
    optionnel: "fichier.html"]
```

Exécution depuis root :

```
~$ python3 /www/ens/hiver/terbaoum/public_html/ImageAndVideoScrapper/extract.py
     "https://www.w3schools.com/html/html5_video.asp" | python3 /www/ens/hiver/
        terbaoum/public_html/ImageAndVideoScrapper/genere.py "page1.html"
```

7 Conclusion

Ce projet permet d'exécuter les scripts sans avoir à installer quoi que ce soit, car toutes les bibliothèques sont déjà incluses dans un environnement dédié. L'outil fonctionne bien pour extraire des images et des vidéos, mais plusieurs améliorations sont possibles.

L'interface web pourrait être améliorée pour offrir un meilleur rendu visuel. De plus, il faudrait ajouter plus de logique pour bien gérer tous les formats d'images et de vidéos, car certains types peuvent ne pas être reconnus correctement.

Il existe le problème de permissions, certains site web ne permettent pas de faire du scrapping sur leurs videos, par exemple youtube, il faudrait utiliser une librairie propre a cet effet.

Enfin, une amélioration importante serait d'ajouter une fonction de recherche récursive pour explorer automatiquement toutes les pages d'un site web, ce qui rendrait le scraping plus complet et efficace.