

Installer un projet en lisant son README

Objectif : installer et lancer un projet qu'on n'a pas écrit,
sans autre consigne que sa documentation, et voir ce que la
commande enchaîne

`cours1/1j_trajet/` `≈ 15'`

Installer un projet sur sa seule documentation

Le projet `trajet` fabrique une vidéo commentée du chemin de la gare à l'école. Aucune consigne ici : tout est dans son `README`.

	Ce qu'il faut faire
1	copier <code>depart/trajet/</code> dans <code>travail/</code> , puis lire son <code>README</code>
2	les taper, dans l'ordre
3	<code>magick --version</code> et <code>ffmpeg --version</code>
4	<code>trajet</code> , puis ouvrir <code>trajet.mp4</code>

`pyproject.toml` déclare zéro dépendance, et le projet ne tourne pourtant pas sans `environment.yml` : ce dont il dépend n'est pas du Python.

Ce que la commande enchaîne

Quatre étapes, chacune produisant le fichier que la suivante consomme. Ce sont les trois modules de `src/trajet/`, appelés dans cet ordre.

Étape	Qui la fait	Entrée
lecture	<code>etapes.py</code>	<code>data/etapes.csv</code>
tracé	<code>images.py</code> , par <code>magick</code>	<code>data/carte.png</code> + les étapes
commentaires	<code>montage.py</code>	les étapes et leurs durées
montage	<code>montage.py</code> , par <code>ffmpeg</code>	les images + le <code>.srt</code>

Les deux fichiers du milieu sont du texte : les ouvrir. `etapes.csv`, le `.csv` de la grille des extensions et le `content.xml` du `.odt` — trois fois le même constat, le contenu utile est du texte.

En option : les commandes, une à une

`outils_video.ipynb` reprend les commandes `magick` et `ffmpeg` du projet, isolées, avec leur résultat affiché sous chacune.

Ce que le notebook montre	Ce qu'on y voit
<code>subprocess.run</code>	la commande donnée en liste, un élément par argument, donc aucun espace à protéger
<code>magick identify, -resize, -draw</code>	l'image produite, affichée dans la cellule qui l'a faite
<code>ffprobe, puis ffmpeg -f concat</code>	la liste de montage écrite à la main, et la vidéo qu'elle donne

Ce notebook ne tourne pas dans le navigateur : `subprocess` demande au système de lancer un programme, ce qu'un navigateur ne fait pas. Il s'ouvre dans l'éditeur ou dans JupyterLab, `trajet_ensg` actif.