

Comparatif des outils Python – Formation et pratique

Visual Studio Code

- Avantages : Léger, rapide, personnalisable, nombreuses extensions (Python, Jupyter), terminal intégré.
- Inconvénients : Configuration initiale requise.
- Cas d'usage : Projets perso/pro, data science, scripts Python.

Eclipse + PyDev

- Avantages : IDE robuste, bon pour les profils Java.
- Inconvénients : Interface plus lourde, moins intuitive.
- Cas d'usage : Formation encadrée, gros projets d'entreprise.

Jupyter Notebook

- Avantages : Exécution interactive, visualisation de données.
- Inconvénients : Mauvais pour structurer un projet complet.
- Cas d'usage : Analyse de données, prototypage, formation.

Jupytertext

- Avantages : Conversion de notebook en .py/.md pour versioning.
- Inconvénients : Configuration complémentaire requise.
- Cas d'usage : Documentation, collaboration, pédagogie.

virtualenv

- Avantages : Isole les environnements par projet.
- Inconvénients : Nécessite une bonne habitude de la ligne de commande.
- Cas d'usage : Tous projets Python, propreté de l'environnement.

pyodbc

- Avantages : Connexion simple à SQL Server, utilisé en entreprise.
- Inconvénients : Drivers à installer.
- Cas d'usage : Projets pros avec base de données relationnelles.

Recommandations générales

- Visual Studio Code : pour tous les usages réguliers
- Eclipse : pour les formations encadrées
- Jupyter Notebook : pour l'analyse de données
- virtualenv : pour tous les projets bien organisés
- pyodbc : pour projets pros avec base SQL