

1 Python

Python est un langage de programmation moderne, généraliste, orienté objet et de haut niveau.

Caractéristiques techniques.

- **typage dynamique** : Pas besoin de définir le type des variables, des arguments ou des types de retour des fonctions, il est inféré à l'exécution.
- **gestion automatique de la mémoire** : Pas besoin d'allouer ou de dés-allouer explicitement pour les variables et les tableaux de données. Pas de problème de fuite de mémoire.
- **interprété** : Pas besoin de compiler le code. L'interpréteur Python lit et exécute directement le code python.

Avantages.

- L'avantage principal est la facilité de programmer, réduisant le temps nécessaire au développement, débogage et maintien du code.
- Le langage encourage plusieurs bonnes pratiques de programmation :
- L'indentation comme syntaxe, définissant la portée d'un *scope*.
- L'intégration de la documentation au code source.
- Une bibliothèque standard fournie et un large nombre de modules dédiés au calcul scientifique.

Inconvénients.

- Puisque Python est un langage de programmation interprété et typé dynamiquement (*i.e.* lors de l'exécution), l'exécution du code python peut être lente comparée à des langages de programmation compilés et typés statiquement (*i.e.* lors de la compilation), tels que le C ou le Fortran.

2 Notebook (ou Calepin) Jupyter

Le notebook Jupyter est un environnement HTML pour Python, similaire à Mathematic ou à Maple. Il fournit un environnement organisé en cellules interactives qui peuvent être exécutées, permettant l'organisation et la documentation de calculs de façon structurée.

Les commandes sont regroupées dans des cellules suivies de leur résultat après exécution. Ces résultats et commentaires sont stockés dans un fichier spécifique `.ipynb` et sauvegardés. Les commandes \LaTeX sont acceptées pour intégrer des formules, la mise en page est assurée par des balises HTML ou Markdown.

La commande de sauvegarde permet également d'extraire les seules commandes Python dans un fichier d'extension `.py`. C'est une façon simple et efficace de conserver tout l'historique d'une analyse pour en faire une présentation ou créer un tutoriel. Le calepin peut être en effet chargé sous un autre format : page `html`, fichier `.pdf` ou diaporama.

Le projet Jupyter propose cet environnement de calepin pour beaucoup de langages dont Julia, Python et R qui donnent le nom de cet environnement).

L'ouverture d'un navigateur sur un calepin Jupyter est obtenu, selon l'installation, à partir des menus ou en exécutant :

dans une fenêtre de commande.

Une fois le calepin ouvert, on peut :

- Entrer des commandes Python dans une cellule ;
- Cliquer sur le bouton d'exécution de la cellule ;
- Ajouter une ou des cellules de commentaires et balises HTML ou Markdown ;
- Itérer l'ajout de cellules.

Une fois l'exécution terminée :

- Sauver le calepin `.ipynb` ;
- Charger éventuellement une version `.html` pour une page web ou une version `.pdf` ;
- Charger le fichier `.py` regroupant les commandes python pour une version opérationnelle.

3 À faire

1. Télécharger le fichier `R5.A.12-TP1.ipynb`.
2. Pour ouvrir ce fichier, on peut procéder :
 - à partir du menu : `python` » `jupyter` puis ouvrir le fichier.
 - à partir du terminal : à l'aide du navigateur de fichier, placer vous dans répertoire dans lequel vous avez téléchargé le fichier puis `clik droit` » `ouvrir un terminal ici` et ouvrir le fichier.
3. lire attentivement les consignes et exécuter les commandes demandées et observer les résultats obtenus.
Attention à bien analyser les résultats et à bien les comprendre. **Il faut bien ne pas hésiter à poser des questions en cas de doute !**
4. Chercher les deux exercices d'applications suivants.
 - (a) **Deviner une phrase**
 - (b) **TSP : Résolution exacte**