UNIVERSITÉ JEAN MONNET SAINT-ÉTIENNE

Info. TP3: tableaux numpy

- https://learn.heeere.com/python
- https://learn.heeere.com/2018-infospichi-8a82
- https://learn.heeere.com/2018-info-8c99 (cours de L1)
- https://practice.heeere.com/

(au moment du rendu : sous forme d'archive)

À la fin, vos réponses seront rendues dans une archive zip : cette archive contiendra un fichier `compte-rendu.txt` (avec votre **nom**, **groupe**, et les réponses aux questions précédées d'une apostrophe) et vos **fichiers pythons**.

Pour faire une **archive zip** de votre TP pour le rendu, exécutez, dans le terminal :

```
cd ~/info-spichi/
zip -r tp3.zip tp3/
```

Important : mise en place

Suivre les consignes des TP précédents, et travailler dans un dossier `tp3`.

Exercice 1 – Plateforme d'exercices

Cet exercice est à réaliser par chaque étudiant. Si vous n'avez qu'une machine pour deux, faites le avec un utilisateur, puis l'autre. Si vous ne le faites pas vous serez **pénalisés sur votre note pratique**.

'Q1) Votre identifiant sur la plateforme d'exercice est de la forme : `etu-???????` en remplaçant les points interrogation par votre utilisateur. Quel est votre identifiant ?

Q2) Allez sur la « claroline connect » sur le site du cours « Info (L2 SPI/CHI) ». Sur la page d'accueil, un encadré « Résultat » vous montre `?????? / 999999`. Le nombre à 6 chiffre à la place des points d'interrogation est votre mot de passe.

Q3) Maintenant que vous avez vos identifiant, vous pouvez aller directement sur `https://practice.heeere.com` (lien présent sur la page d'accueil du cours).

'Q4) Clickez sur « pratiquer les exercices du moment » puis quand la question s'affiche, clickez sur le gros point d'interrogation et lire l'aide de réponse. Que faut-il répondre si le programme donné est incorrect, n'a pas la bonne structure ou crée un problème quand on l'exécute avec `python3` ?

Q5) Maintenant que vous savez répondre, faites les 5 questions proposées (en essayant de comprendre les erreurs que vous feriez éventuellement).

Q6) Refaites encore 5 questions. Si vous n'avez qu'une machine pour deux, utilisez la session et les identifiants de l'autre membre du binôme (après avoir cliqué sur « logout » en haut à gauche.

Exercice 2 – Téléchargement, Assertion et Échauffement

Dans le dossier `tp3`, téléchargez et décompressez l'archive du TP3, téléchargeable sur le site du cours. Cela peut être réalisé directement depuis le terminal avec les commandes :

```
wget https://learn.heeere.com/2018-infospichi-8a82/raw/fichiers-tp3.zip
unzip fichiers-tp3.zip
```

La plupart des fichiers nécessaires vous sont fournis et vous devez les compléter :

- ne jamais modifier ou supprimer les sections de code commençant par le commentaire `# vvv NE PAS MODIFIER vvv` et terminées par `# ^^^ NE PAS MODIFIER ^^^`
- parfois le programme donné s'exécute mais ne fait pas ce qu'il faut, parfois il ne s'exécute même pas

- le but est de faire que le programme s'exécute jusque la fin, sans erreur (parfois sans rien afficher)
- il faut passer le temps nécessaire pour comprendre le code donné, en particulier les parties non-modifiables

IMPORTANT: comme toujours, pensez à bien sauver et exécuter vos programmes et à bien lire les messages d'erreur (numéro de ligne, etc) donnés à l'exécution.

Pour cet exercice, travaillez dans un dossier `ex2` lui même dans le dossier `tp3`.

'Q7) Ouvrez (avec emacs) le fichier `somme.py` sans le modifier pour l'instant et exécutez le avec `python3`. Jusqu'où semble s'exécuter le programme ? Que fait selon vous l'instruction `assert` ?

Note : `assert` n'est pas une fonction mais une *instruction* du langage Python. Tout comme l'instruction `return`, `assert` ne nécessite pas de parenthèses. Cette instruction permet de vérifier qu'une proposition (expression booléenne) est vraie et interrompt le programme si ce n'est pas le cas.

Q8) Ouvrez maintenant `assertflottant.py`. Comprenez bien le « problème » et faîtes que le programme s'exécute (aide : l'assertion peut ne pas être vraie à cause d'imprécisions de calcul).

Q9) Faîtes de même avec `multiplieajoute.py` (ouvrez le et faîtes qu'il fonctionne).

Exercice 3 – Numpy : création de tableaux (*arrays*) (de type `np.ndarray`)

Pour cet exercice, travaillez dans un dossier `ex3` lui même dans le dossier `tp3`.

Q10) Ouvrez et faîtes fonctionner `vecteurs.py`.

Q11) Ouvrez et faîtes fonctionner `peignes.py`.

Q12) Ouvrez et faîtes fonctionner `tableaux2d.py`.

Exercice 4 – Numpy : accès simple (arrays)

Pour cet exercice, travaillez dans un dossier `ex4` lui même dans le dossier `tp3`.

Q13) Ouvrez et faîtes fonctionner `lecture.py` (en accédant aux valeurs du tableau `a`).

Q14) Ouvrez et faîtes fonctionner `ecriture.py`.

Exercice 5 – Numpy : opérations élément-par-élément

Pour cet exercice, travaillez dans un dossier `ex5` lui même dans le dossier `tp3`.

Q15) Ouvrez et faîtes fonctionner `operations.py`.

Exercice 6 – Numpy : sous-tableaux

Pour cet exercice, travaillez dans un dossier `ex6` lui même dans le dossier `tp3`.

Q16) Ouvrez et faîtes fonctionner `trancheslistes.py`.

Q17) Ouvrez et faîtes fonctionner `tranchesnumpy.py`.