

Algorithmes et Pensée Computationnelle

Exercices – Semaine 2

Pour réaliser cet exercice, vous allez devoir interagir avec système de fichiers (file system) de Jupyter Hub au travers d'un terminal. Nous vous conseillons de garder ce notebook d'instruction ouvert dans un onglet afin de suivre les étapes de l'exercice.

Ouverture d'un terminal dans Jupyter.

Rendez-vous dans l'onglet « Files » de votre Jupyter Hub. En haut à droite cliquez sur New puis Terminal – Notez que le dossier dans lequel vous vous trouvez n'a pas d'importance.

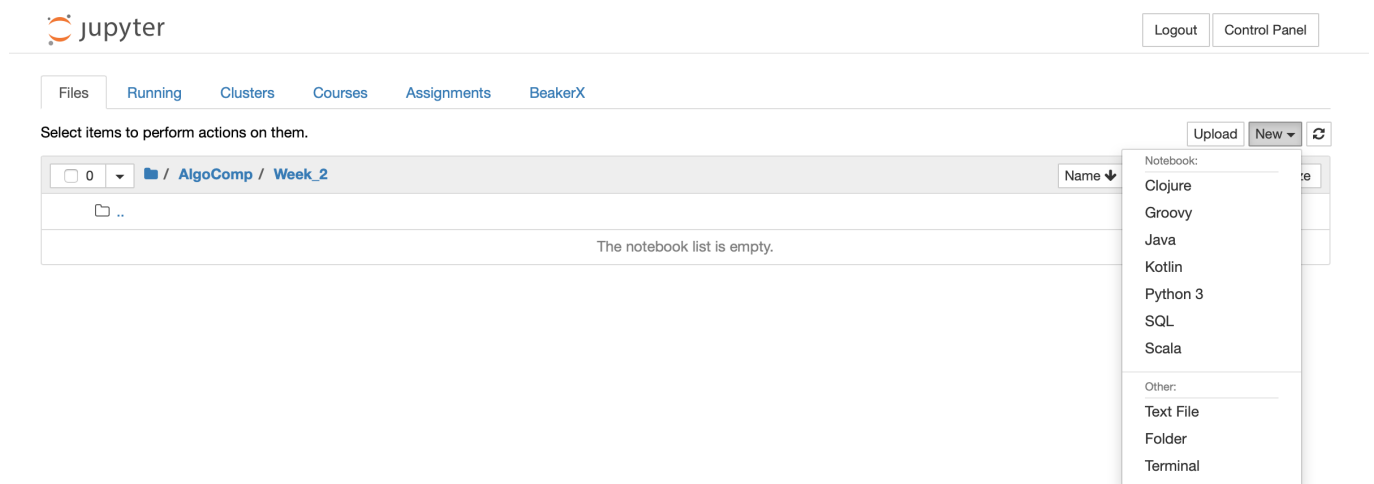


Figure 1 : Lancement d'un nouveau Terminal

Le terminal s'ouvre dans un nouvel onglet de votre navigateur.

Le terminal exécute un « shell » qui sert d'interface entre le système d'exploitation (Operating System – OS) et l'utilisateur. Dans le cas présent, le langage du Shell est le « BASH » (Bourne-Again Shell). Il s'agit d'un langage interprété en mode interactif depuis le terminal ; néanmoins il est également possible d'écrire des scripts en BASH pour les exécuter ultérieurement. En BASH vous pouvez aussi bien communiquer avec votre OS, naviguer dans le système de fichiers (file system), lire des fichiers, lancer des programmes ou encore effectuer des opérations arithmétiques. Nous verrons ultérieurement qu'il est possible d'exécuter d'autre programme que le Shell dans le terminal, comme par exemple un interpréteur python.

Quelques commandes de base

- **echo** : la commande echo retourne dans le terminal une chaîne de caractère qui lui est passé en argument. Essayez en tapant « echo Hello Word ! » par exemple.
- **pwd** : la commande pwd (pour print working directory) retourne le chemin (path) depuis la racine du système de fichiers menant jusqu'au dossier dans lequel vous vous trouvez et à partir duquel la prochaine commande BASH sera exécutée.
- **ls** : la commande ls retourne la liste des fichiers et dossiers présent dans votre répertoire actuel.
- **cd** : la commande cd vous permet de changer de répertoire (directory).

Nous allons maintenant naviguer dans le système de fichiers afin de nous rendre dans le répertoire Week_2.

- Tapez la commande « pwd » afin de vérifier où vous vous trouvez.

- Tapez la commande « cd » afin de revenir dans votre répertoire home si c'est pas déjà le cas.
- Tapez la commande « ls » : en principe, il devrait y avoir un répertoire « AlgoComp ».
- Tapez la commande « cd AlgoComp/ » pour changer de répertoire.
- Tapez la commande « pwd » qui vous confirmera que vous vous trouvez bien dans le répertoire /home/jovyan/AlgoComp.
- Tapez la commande « ls » qui vous montrera les fichiers et dossiers dans le répertoire « /home/jovyan/AlgoComp ».
- Rendez-vous maintenant dans le répertoire Week_2 avec la commande « cd ».

_Notez qu'il aurait aussi été possible de faire directement « cd AlgoComp/Week_2/ » *depuis votre répertoire home*.

Pour remonter d'un niveau dans le files system utilisez « cd .. » « cd ../.. » pour deux niveaux et « cd » pour revenir dans le home directory.

Lancement de python en mode interactif dans le terminal

Pour lancer python dans le terminal exécutez la commande « python ». Votre terminal exécute maintenant un interpréteur python et non plus un Shell. Testez en écrivant « print("Hello Wolrd!") » dans le terminal. Amusez-vous à jouer avec la console en effectuant des opérations arithmétique de base. Pour quitter l'interpréteur python et revenir sur le Shell faites « ctrl + d »

Ecriture d'un programme avec Nano en terminal + Exec dans le shell

Editeur de texte Nano

- Pour voir ou éditer le contenu d'un fichier dans un terminal, il est possible d'utiliser un éditeur de texte nommé Nano.*

Un éditeur de texte permet de modifier des fichiers de texte brut alors qu'un programme de traitement de texte comme Microsoft Word permet de rédiger des documents avec une mise en forme.

Un éditeur de texte peut ainsi vous permettre d'éditer un fichier python.

1. Pour lancer l'éditeur de texte, tapez "nano" et enter dans le terminal. Vous obtiendrez le résultat suivant :

```
tanjamarkotic — nano — 80x24
GNU nano 2.0.6 New Buffer
```



```
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Au bas de la page, vous pouvez voir les commandes principales. ^ signifie ctrl. Attention, vous devez utiliser **ctrl et une lettre minuscule** pour lancer une commande. De plus, il est préférable d'utiliser les flèches de votre clavier pour déplacer le curseur.

1. Afin d'utiliser la commande de recherche ^w, commencez par écrire quelques mots sur nano. Pensez à écrire deux fois la même chose à la suite. Puis, utilisez ^w et enter pour trouver un mot spécifique. Le curseur sera placé à la première apparition du mot recherché. Vous obtiendrez le résultat suivant :

```
tanjamarkotic — nano — 80x24
GNU nano 2.0.6 New Buffer Modified
```

```
Ceci est un test.
Ceci est un test.
```

```
Search: un
^G Get Help ^Y First Line ^R Replace ^W Beg of Par ^M-C Case Sens ^M-R Regexp
^C Cancel ^V Last Line ^T Go To Line ^O End of Par ^M-B Backwards ^P PrevHistory
```

```
tanjamarkotic — nano — 80x24
GNU nano 2.0.6          New Buffer          Modified

Ceci est un test.
Ceci est un test.

[ Search Wrapped ]
^G Get Help    ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text    ^C Cur Pos
^X Exit        ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

Afin de trouver le deuxième mot identique, il suffit de réutiliser la commande ^w et presser enter. Le curseur sera ainsi placé à la deuxième apparition du mot recherché.

1. Pour sortir de la recherche utiliser ^c.
2. Pour sauvegarder, utilisez ^o. Un message apparait si vous utiliser ^c (soit *annuler*) sans sauvegarder. Il suffit d'écrire le nom du nouveau fichier à créer puis presser enter. Vous obtiendrez:

```
tanjamarkotic — nano — 80x24
GNU nano 2.0.6          New Buffer          Modified

Ceci est un test.
Ceci est un test.
```

```
File Name to Write: nanotest
^G Get Help    ^T To Files    M-M Mac Format  M-P Prepend
^C Cancel      M-D DOS Format M-A Append     M-B Backup File
```

1. Pour quitter l'éditeur de texte, utilisez ^x. Vous retrouverez ainsi votre terminal de base. Afin de réouvrir le fichier créer, écrivez "nano nomdufichier" dans la console comme ceci:

```
tanjamarkotic — -bash — 80x24
Last login: Mon Sep 23 18:27:45 on ttys000
(base) wlguest-146-40:~ tanjamarkotic$ nano nanotest
```

Si le fichier est inexistant sur votre ordinateur, un nouveau fichier sera créé.

Hello world avec Nano

- Maintenant, nous allons créer un nouveau programme python dans nano puis l'exécuter dans le terminal.*
1. Afin de créer un nouveau fichier nommé hello.py et ouvrir nano, écrivez **nano hello.py** dans votre terminal comme ceci:

```
tanjamarkotic — -bash — 80x24
Last login: Mon Sep 23 19:12:54 on ttys000
(base) wl3s-217-63:~ tanjamarkotic$ nano hello.py
```

1. Une fois nano ouvert, écrivez votre premier programme **print("Hello World!")**:

```
tanjamarkotic — nano hello.py — 80x24
GNU nano 2.0.6 File: hello.py Modified

print( "hello World!")

[ Read 1 line ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

1. Quittez nano (^x) en sauvegardant les changements effectués sur le fichier hello.py. Une fois de retour sur le terminal de base, exécutez votre nouveau programme python en écrivant **python hello.py** dans votre terminal. Le résultat est le suivant:

```
tanjamarkotic — -bash — 80x24
Last login: Mon Sep 23 19:12:54 on ttys000
[(base) wl3s-217-63:~ tanjamarkotic$ nano hello.py
(base) wl3s-217-63:~ tanjamarkotic$ python hello.py
```

sources:

<https://openclassrooms.com/fr/courses/43538-reprenez-le-contrôle-a-l'aide-de-linux/39267-nano-lediteur-de-texte-du-debutant> (<https://openclassrooms.com/fr/courses/43538-reprenez-le-contrôle-a-l'aide-de-linux/39267-nano-lediteur-de-texte-du-debutant>)

Compilation et lancement d'un programme Java

Contrairement au Python qui est un langage interprété, Java est un langage compilé ce qui veut dire que nous devons compiler le langage avant de l'exécuter. Pour cela nous utilisons un compilateur. dans le cas

In []: