

# Mise à jour de Python vers la version 3.6 et mise à jour de jupyter

## 1. Constatation d'un bug

- Exécutez dans ce jupyter notebook sur votre Raspberry, le code dans la cellule ci-dessous :

```
In [1]: d3 = {4*i: chr(64+i) for i in range(1, 5)}  
print(d3)
```

```
{8: 'B', 12: 'C', 4: 'A', 16: 'D'}
```

1) Quel est le résultat affiché par ce code ?

Réponse : {8: 'B', 12: 'C', 4: 'A', 16: 'D'}

2) Quel est le résultat attendu ?

Réponse : {4: 'A', 8: 'B', 12: 'C', 16: 'D'}

- Pour corriger ce bug, nous allons installer une version plus récente de Python.

## 2. Installation d'une version ultérieure de Python : la version 3.6

La mise à jour de la version de Python dans jupyter notebook nécessite deux phases :

- Une première phase de changement de version de Python (amélioration ou "upgrade").
- Une deuxième phase de réinstallation de jupyter afin que jupyter notebook prenne en compte la version upgradée de Python.

## **PHASE 1 : Installation de berryconda3 qui contient Python 3.6**

a) Vérification de la version de Python :

Ouvrez un nouveau terminal et dans ce terminal, tapez :

```
python
```

puis appuyez sur la touche Entrée.

On voit que la version de Python est la 2.7

Les 3 chevrons du prompt montrent que nous ne sommes pas dans le bash, mais dans Python. On pourrait ici exécuter des commandes Python. Par exemple `print(12**2)`. Mais ce n'est pas le but. Pour en sortir et retourner au bash, tapez `exit()` puis appuyez sur la touche Entrée.

b) Avant toute nouvelle installation, assurez-vous de mettre à jour le système d'exploitation Linux. Dans un terminal, saisissez la ligne de commande

```
sudo apt-get update
```

Attendez que les éventuelles mises à jour soient terminées. Puis dans le terminal saisissez :

```
sudo apt-get upgrade -y
```

A nouveau, attendez que les mises à jour soient terminées.

c) Téléchargement du fichier d'installation : Une fois revenu au bash, tapez la ligne de commande (pour la copier ici utilisez le clic droit de la souris + Copier et dans le terminal, utilisez le menu Edition / Coller)

```
wget https://github.com/jjhelmus/berryconda/releases/download/v2.0.0/Berryconda3-2.0.0-Linux-armv7l.sh
```

d) Vérifier la présence du fichier qu'on vient de télécharger :

```
ls -al
```

(ls permet de lister et l'option -al permet de voir tout et de façon détaillée).

On voit que pour Berryconda3-2.0.0-Linux-armv7l.sh

les permissions sont -rw-r--r-- c'est à dire que :

- Le premier - indique que c'est un fichier (et non un dossier)
- Le premier groupe rw- indique que les droits du propriétaire (c'est à dire vous qui venez de télécharger) sont read: oui, write: oui, execute: non.
- Le deuxième groupe r-- indique que les droits du groupe des utilisateurs du fichier sont read: oui, write: non, execute: non.
- Le troisième groupe r-- indique que les droits des utilisateurs hors du groupe des utilisateurs de ce fichier sont read: oui, write: non, execute: non.

e) Modifier les droits pour rendre exécutable le fichier d'installation :

```
chmod +x Berryconda3-2.0.0-Linux-armv7l.sh
```

f) Vérifier à nouveau les droits :

```
ls -al
```

Cette fois, les permissions sont -rwxr-xr-x c'est à dire que :

- Le premier - indique que c'est un fichier (et non un dossier)
- Le premier groupe rwx indique que les droits du propriétaire (c'est à dire vous qui venez de télécharger) sont read: oui, write: oui, execute: oui.
- Le deuxième groupe r-x indique que les droits du groupe des utilisateurs du fichier sont read: oui, write: non, execute: oui.
- Le troisième groupe r-x indique que les droits des utilisateurs hors du groupe des utilisateurs de ce fichier sont read: oui, write: non, execute: oui.

Donc comme on peut exécuter le fichier, on peut continuer.

g) L'extension .sh indique que c'est un fichier script qui peut être exécuté avec un interpréteur de type shell (comme le bash de notre terminal). Dans le terminal, il suffit de commencer par écrire ./ puis d'écrire le nom du fichier et enfin d'appuyer sur la touche Entrée.

```
./Berryconda3-2.0.0-Linux-armv7l.sh
```

Répondez par Entrée ou yes quand c'est demandé. Allez jusqu'au bout, c'est à dire jusqu'à ce que vous

ayez à nouveau le prompt (l'invite de commandes) du bash. Rappel : les trois chevrons >>> *ne sont pas le prompt du bash* mais le prompt de Python.

h) Rebootez votre Raspberry. Vous ouvrirez à nouveau ce jupyter notebook après le redémarrage.

i) Vérification de la version de Python : Ouvrez un nouveau terminal, et tapez :

```
python
```

puis appuyez sur la touche Entrée On voit que maintenant la version de Python a été upgradée de 2.7 à 3.6

## PHASE 2 : Réinstallation de jupyter

a) La présence des trois chevrons montre qu'on est dans Python. Quittez Python en tapant l'instruction `exit()`

Dans le terminal, les trois chevrons sont alors remplacés par l'invite de commande du bash.

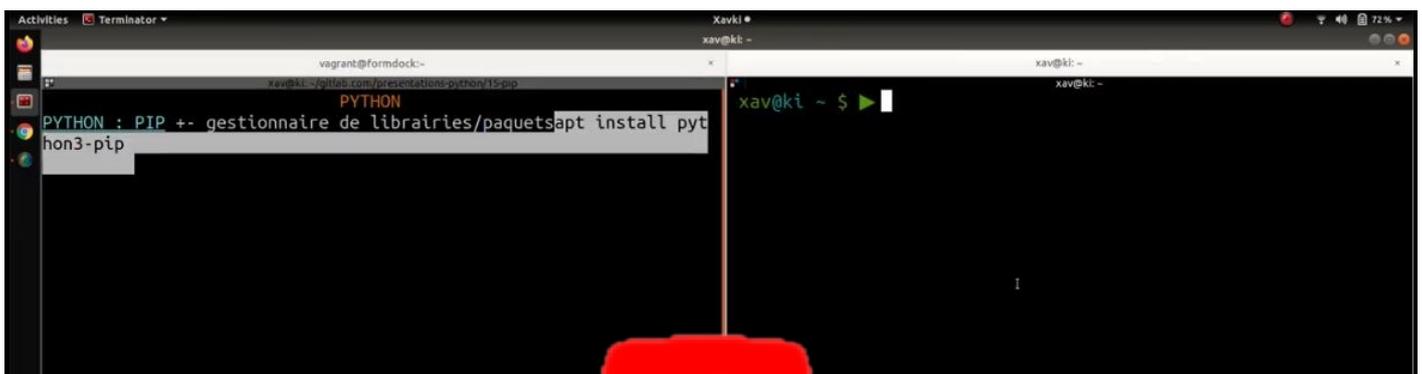
b) Dans le terminal, tapez la ligne de commande

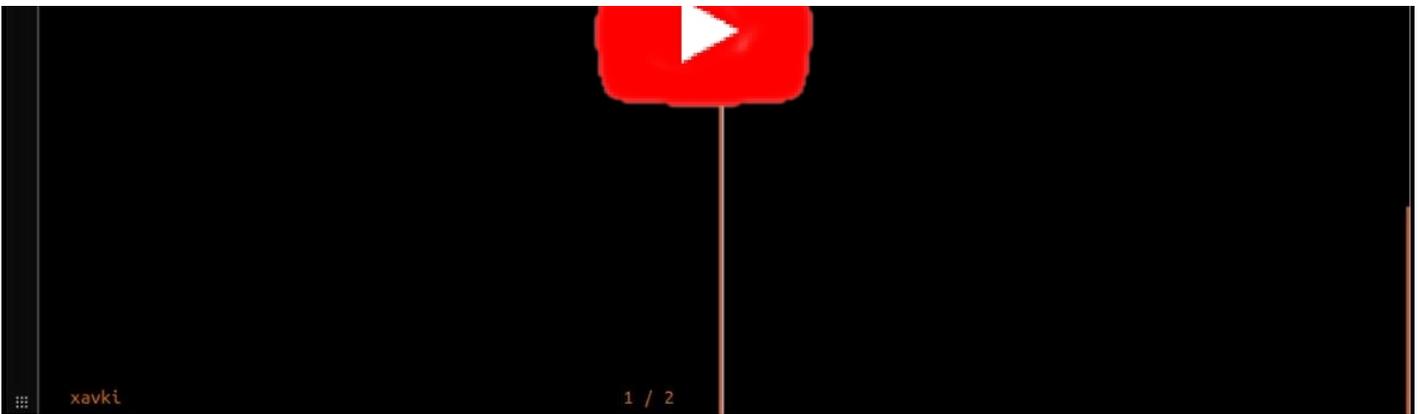
```
python3 -m pip install jupyter
```

Patiencez un certain temps que l'installation soit terminée. Même si par moments il semble que l'installation soit figée, soyez patient. Il faut attendre le retour de l'invite de commandes du bash.

c) Une fois l'installation terminée, pour finir, mettez à jour le programme *pip* en saisissant la commande proposée qui apparaît tout à la fin en caractères orange dans le terminal.

*pip* est un programme gestionnaire (Pip Installs Python) qui est utilisé pour installer et gérer des paquets (c'est à dire des modules ou "bibliothèques" écrits en Python).





[http://www.astrovirtuel.fr/jupyter/19\\_pnsi\\_cours/gestionnaire\\_de\\_libraries\\_python.mp4](http://www.astrovirtuel.fr/jupyter/19_pnsi_cours/gestionnaire_de_libraries_python.mp4)

*Si vous êtes intéressé d'en savoir un peu plus sur le programme pip, voyez la vidéo (10mn) tirée de la chaîne YouTube de xavki ([https://www.youtube.com/channel/UCs\\_AZuYXi6NA9tkdbhjtHQ](https://www.youtube.com/channel/UCs_AZuYXi6NA9tkdbhjtHQ)) "Python débutant 15. Pip gestionnaire de librairies".*

d) Dernière opération : rebootez votre Raspberry. Vous ouvrirez à nouveau ce jupyter notebook après le redémarrage.

### 3. Vérification de la nouvelle version de Python

- Exécutez le code ci-dessous pour connaître la version de Python qui est maintenant associée à ce jupyter notebook :

```
In [1]: import sys  
print(sys.version)
```

```
3.6.1 | packaged by rpi | (default, Apr 20 2017, 19:35:19)  
[GCC 4.9.2]
```

3) Quel est le résultat affiché par ce code ?

Réponse : 3.6.1 | packaged by rpi | (default, Apr 20 2017, 19:35:19) [GCC 4.9.2]

## 4. Verification de la disparition du bug

- Exécutez dans ce jupyter notebook sur votre Raspberry, le code dans la cellule ci-dessous :</font>

```
In [3]: d4 = {4*i: chr(64+i) for i in range(1, 5)}  
print(d4)  
  
{4: 'A', 8: 'B', 12: 'C', 16: 'D'}
```

4) Quel est le résultat affiché par ce code ?

Réponse : {4: 'A', 8: 'B', 12: 'C', 16: 'D'}

## 5. Installation du module Python pillow spécialisé dans la manipulation des images

5) Ouvrez un terminal sur votre Raspberry et exécutez les lignes suivantes :

```
sudo apt-get install libjpeg-dev  
(répondre 0 pour oui à la question)
```

```
python -m pip install pillow
```