

**Pascal ORTIZ**



# **Ocaml dans des feuilles Jupyter**

---

Version du 24 septembre 2018  
Licence CC-BY

# Table des matières

Les feuilles Jupyter pour Ocaml . . . . .	2
Le sous-système Windows pour Linux . . . . .	3
Installer WSL en détail . . . . .	4
L'installation résumée de Jupyter pour Ocaml . . . . .	8
Le script d'installation . . . . .	9
L'installation de ocaml-jupyter en détail . . . . .	10
Difficultés possibles . . . . .	18
Comment importer une feuille Jupyter? . . . . .	18
Comment exporter une feuille Jupyter? . . . . .	19

## Les feuilles Jupyter pour Ocaml

Les feuilles Jupyter Notebook sont des interfaces web pour programmer dans un environnement où on peut intégrer du code de langage de programmation, du texte, des mathématiques en Latex, des images, de la vidéo, du Javascript, etc. Le langage de programmation le plus usuel est Python mais Jupyter permet d'utiliser d'autres « moteurs » tels que Julia, R, Haskell, SageMath ou encore Ocaml.



Au-delà d'être très utilisées dans les data sciences et en calcul scientifique, les feuilles Jupyter Notebook ont un intérêt pédagogique : il est possible de placer du cours et des exercices dans une telle feuille et l'utilisateur peut apprendre de manière autonome en modifiant ou en écrivant du code.

Il est possible d'utiliser en ligne sur Internet des feuilles Jupyter (Cocalc ou Azure par exemple) mais hélas, aucun service ne semble prendre en charge Ocaml.

Donc pour pouvoir utiliser Jupyter pour Ocaml, il faut l'installer sur son ordinateur personnel. L'environnement Windows le plus usuel pour Jupyter Notebook est Anaconda mais le moteur Ocaml n'y est pas disponible.

Pour être plus précis, Jupyter pour Ocaml est un package Opam (un gestionnaire de packages Ocaml) de nom `ocaml-jupyter` créé par Akinori ABE et qui permet d'utiliser une boucle interactive (*repl*) dans des feuilles Jupyter. La page officielle est sur [Opam](#). La page de développement est sur [Github](#).

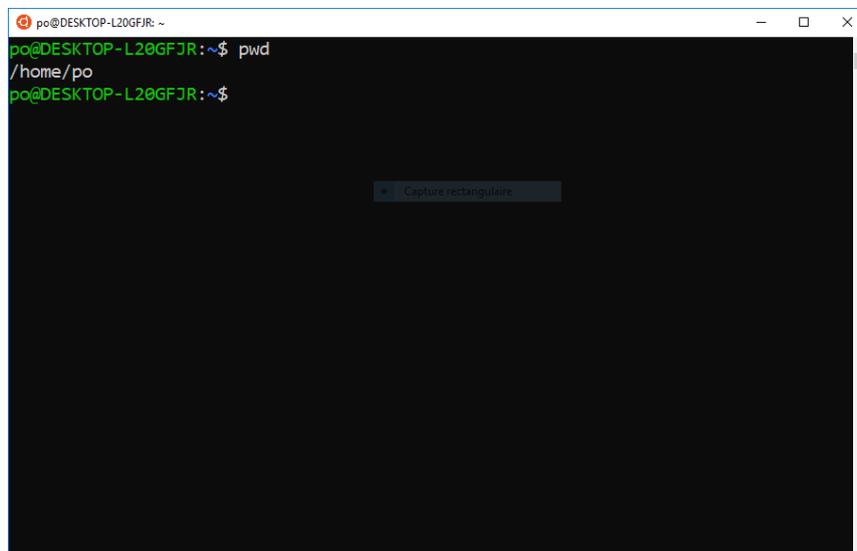
Noter qu'il s'agit bien du langage Ocaml et pas de Caml Light.

### Jupyter pour Ocaml sous Windows

Ocaml étant relativement mal pris en charge nativement par Windows mais très bien sous Linux, on peut parvenir à installer Jupyter pour Ocaml en utilisant le *Windows subsystem for Linux* (WSL). On installe alors Jupyter pour Ocaml comme on l'installerait sous Linux. Une fois installé, c'est le navigateur web sous Windows qui prend en charge l'interface Jupyter.

## Le sous-système Windows pour Linux

Depuis Windows 10, il est possible officiellement d'installer Linux sous Windows. Cette possibilité est offerte à travers le *sous-système Windows pour Linux*, en anglais *Windows subsystem for Linux* (WSL) :

A screenshot of a Windows terminal window titled "po@DESKTOP-L20GFJR: ~". The terminal shows a Linux shell session with the following text:

```
po@DESKTOP-L20GFJR:~$ pwd
/home/po
po@DESKTOP-L20GFJR:~$
```

The terminal background is black, and the text is in green and white. A small grey box with the text "Capture rectangulaire" is visible in the center of the terminal window.

Depuis un Bash (le même que sous Linux), on peut alors installer et exécuter des binaires Linux. Le sous-système peut communiquer avec le système de fichiers de Windows 10. La version de Windows 10 doit être en 64 bits.

Il ne s'agit pas d'une machine virtuelle en sorte que le système est relativement rapide et raisonnable en consommation mémoire. Toutefois, l'environnement graphique de Linux n'est pas pris en charge. Tout se fait donc avec des commandes texte.

Pour simplifier, je parlerai aussi de Bash au lieu de WSL car l'interaction avec le sous-système se fait essentiellement via ce Bash.

### Accès au sous-système

Pour accéder à WSL, il faut :

- placer Windows 10 en mode développeur
- activer WSL (ce qui charge un package)
- redémarrer Windows
- installer une distribution Linux telle que Ubuntu 18.04 depuis le magasin Microsoft (pas d'inquiétude, c'est une installation légère et plutôt rapide)
- définir un nom d'utilisateur et un mot de passe dans le Bash.

Pour l'installation en détail, voir plus bas.

### Deux utilisateurs

Pour la suite, distinguer

- le nom d'utilisateur Linux : c'est le nom sous lequel l'utilisateur se déclare en fin d'installation du WSL et c'est aussi le nom du répertoire /home par défaut quand on ouvre le Bash ; je noterai ce nom `monNomLinux` (pour moi dans les exemples, c'est **po**) ;
- le nom d'utilisateur Windows : c'est le nom sous lequel l'utilisateur s'identifie sous Windows lorsqu'il fournit son mot de passe. Je noterai ce nom `monNomWindows` (pour moi dans les exemples, ce sera encore **po**). Ce nom est aussi le nom d'un dossier Windows : `C:\Users\monNomWindows`.

Ces noms peuvent être distincts.

### Accès au système de fichiers Windows

L'interaction avec WSL se fait essentiellement via une console Bash. Cette console est une console Linux. Elle s'ouvre sur le répertoire utilisateur `/home/monNomLinux`. Depuis cette console, on accède au système de fichiers de Windows via le répertoire `/mnt/c` où `mnt` désigne le répertoire habituel de montage (*mount* dans la terminologie Linux) et `c` fait référence à la partition principale `C :` de Windows. Par exemple, j'accède à mon répertoire personnel `po` de Windows via le chemin `/mnt/c/Users/po`.

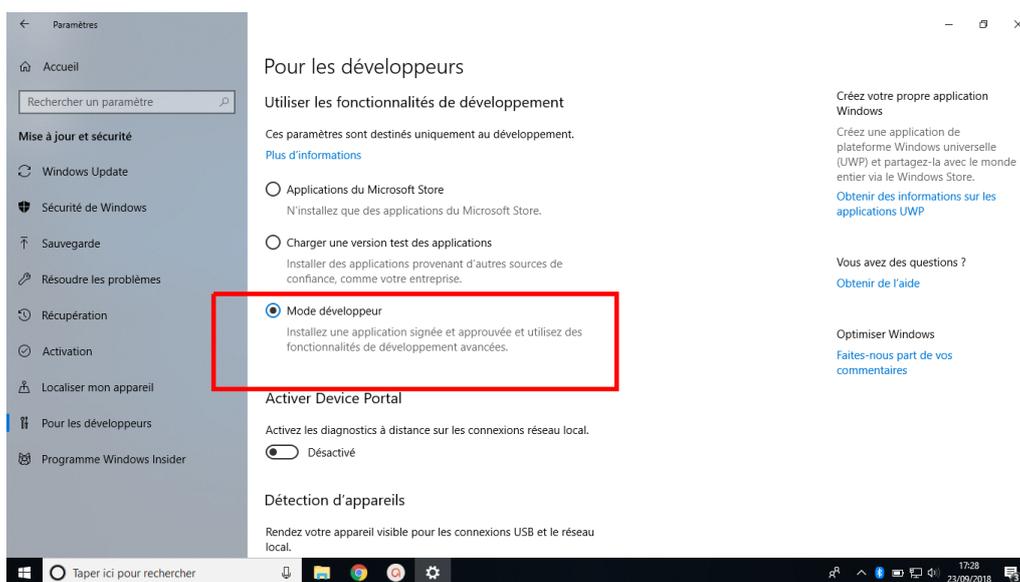
### Cas rare

Si vous n'avez pas de nom d'utilisateur Windows (accès sans mot de passe à Windows, c'est assez rare) ou que votre déclaration de nom d'utilisateur échoue : vous risquez que votre bash WSL soit en root, ce qu'il vaut mieux éviter (et empêche le fonctionnement par défaut de Jupyter). Dans ce cas, autant créer un accès à Windows avec un nom d'utilisateur avant d'installer WSL.

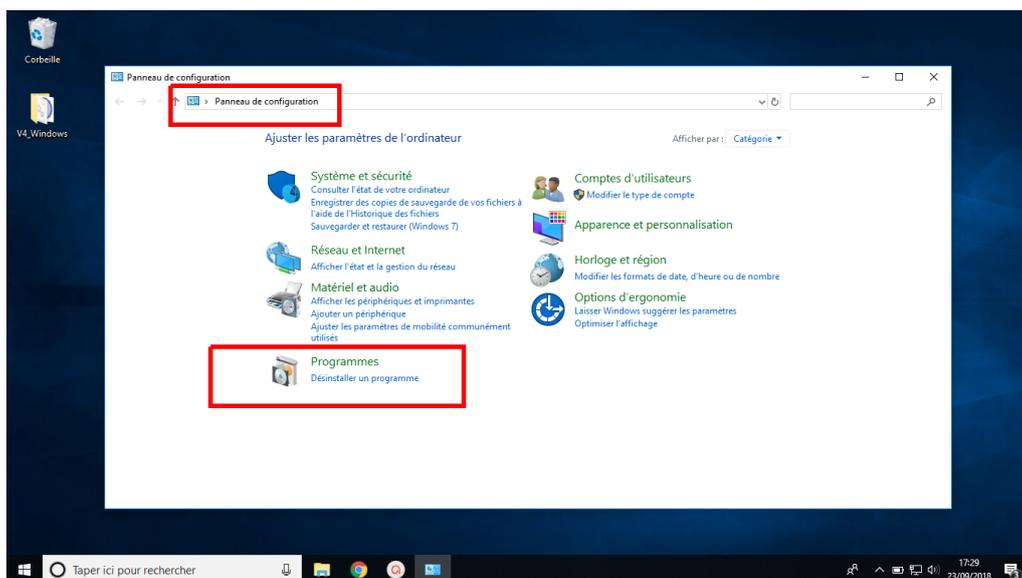
## Installer WSL en détail

Voici comment installer le Bash Ubuntu sous Windows 10. J'utiliserai la dernière LTS d'Ubuntu, la 18.04.

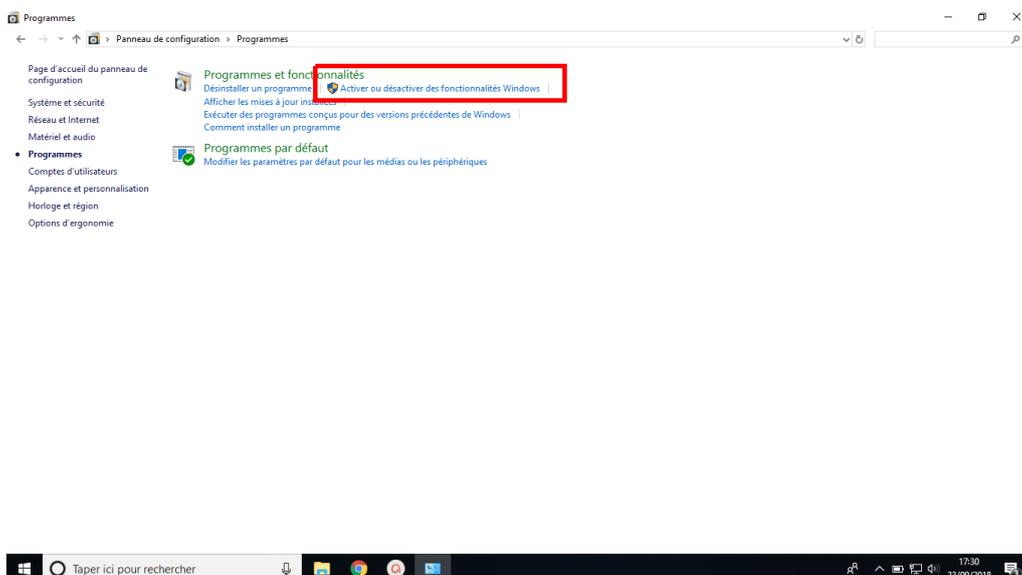
Il faut d'abord placer Windows 10 en mode développeur. Pour cela, dans la barre de recherche des paramètres de Windows, taper **développeur** et choisir "pour les paramètres de développeur". Cocher alors la case Mode développeur :



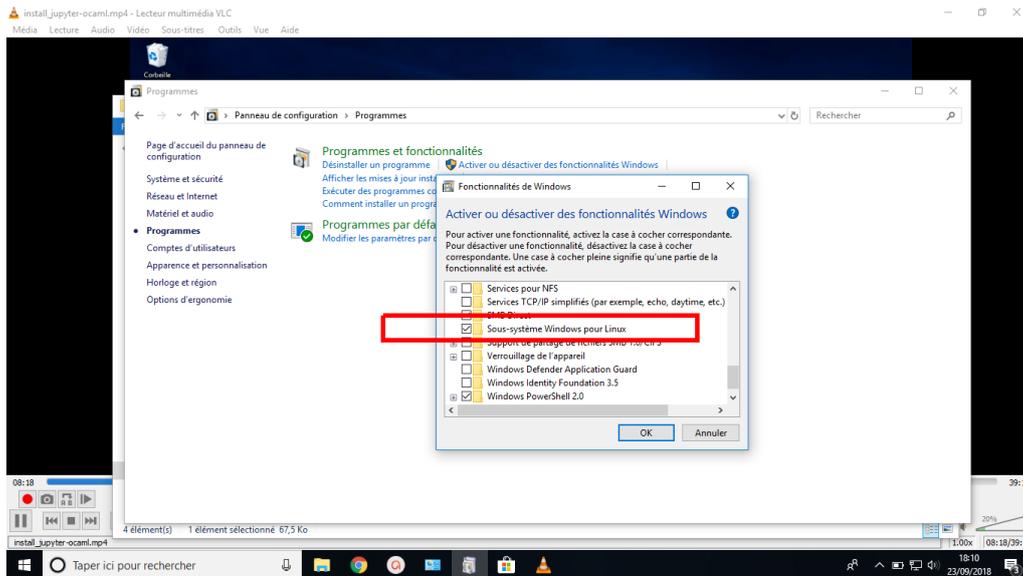
Ensuite, il faut activer WSL. Rechercher dans Cortana la chaîne *panneau de configuration* et cliquer sur *Panneau de configuration* puis encore sur *programmes* :



puis cliquer sur *Activer ou désactiver des fonctionnalités Windows* :

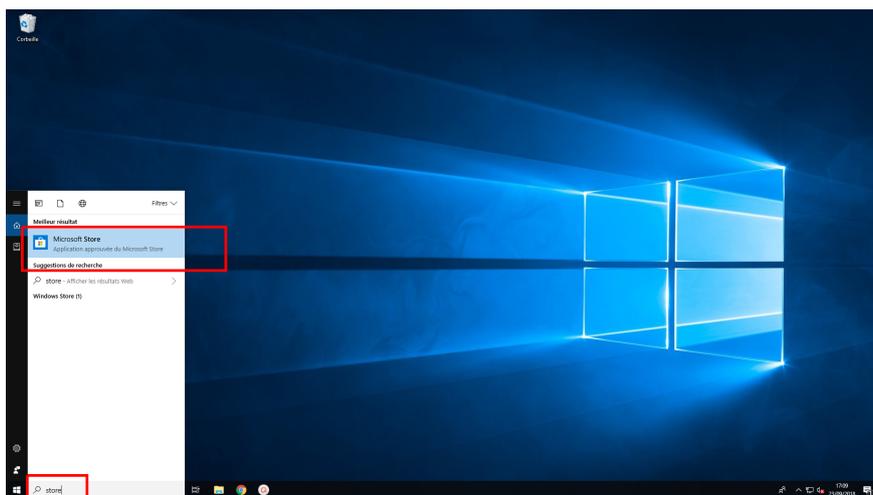


et dans la liste, cocher la case *Sous-système Windows pour Linux* :



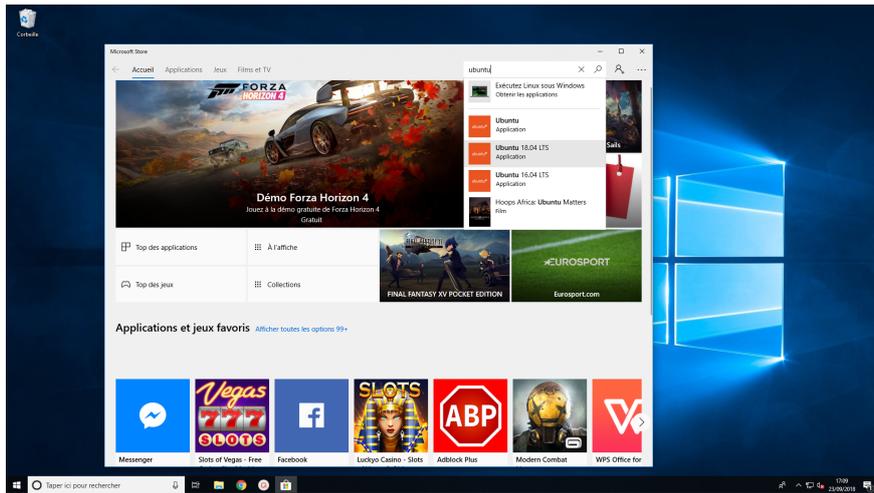
Redémarrer Windows (obligatoire si on veut poursuivre).

Après redémarrage, dans Cortana, taper **Store** et sélectionner le **Microsoft Store** :

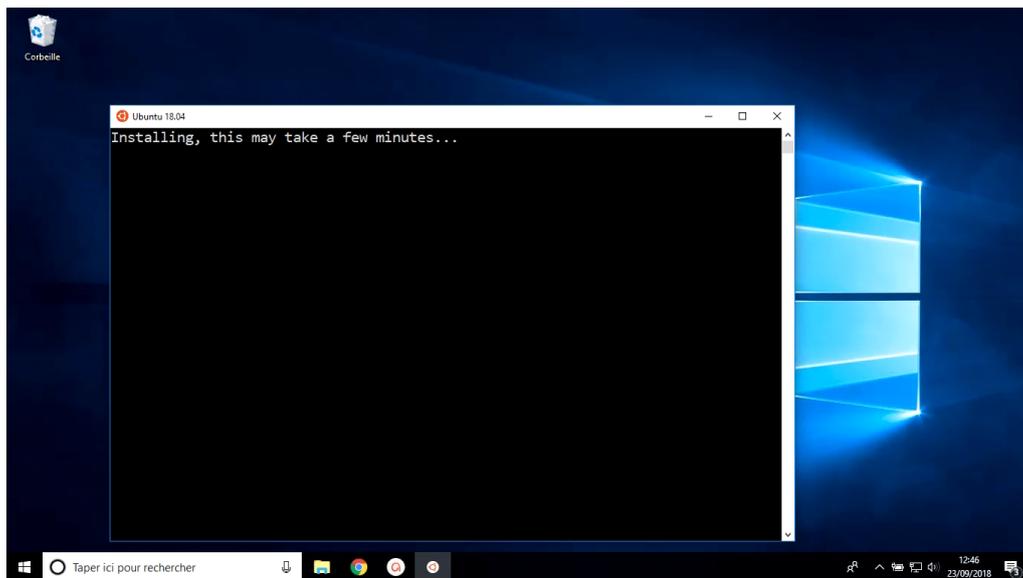


Le Store vous propose de vous connecter, vous pouvez répondre *Non merci*.

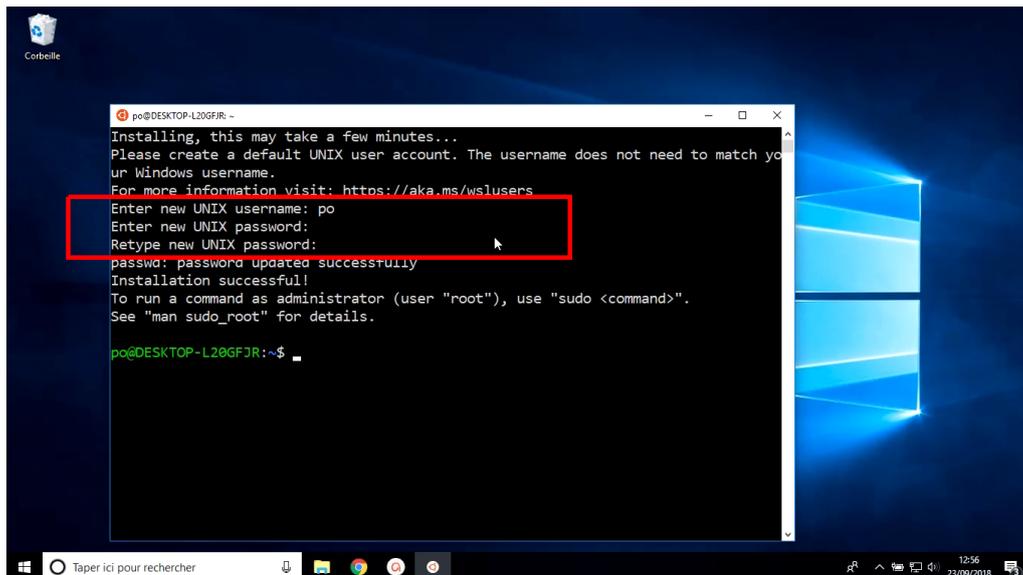
Dans la barre de recherche, chercher *Ubuntu* et sélectionner Ubuntu 18.04 (dernière LTS) :



Choisissez "Obtenir" puis, une fois installé, choisir "Lancer".  
Un terminal va finir par apparaître et dont le titre est Ubuntu 18.04 :



Au bout de quelques minutes, il demande de donner un nom d'utilisateur (voir copie d'écran plus bas) : taper un nom d'utilisateur, pas forcément identique à votre nom d'utilisateur Windows. Donner un mot de passe. Attention, strictement rien n'apparaît quand on tape les caractères du mot de passe, pas d'astérisques ni même aucun curseur. Le système vous demandera de confirmer le mot de passe :



Une fois ces opérations terminées, le Bash est opérationnel.

Le copier-coller dans la console ne semble marcher qu'à la souris et avec un clic droit (ne marche pas au clavier).

Pour désinstaller WSL, il faut

- désinstaller la version de Linux que vous avez installée (aller dans paramètres puis programmes),
- décocher la case d'activation
- quitter le mode développeur.

Si vous avez eu un problème de configuration et que vous voulez réinitialiser WSL, vous pouvez le faire sans désinstaller et réinstaller. Il suffit de réinitialiser :

Menu Démarrer > Paramètres > Applications > Ubuntu 18.04 > Options avancées > Réinitialiser

Toutefois, cette méthode ne marche pas toujours et le plus sûr reste de désinstaller.

## L'installation résumée de Jupyter pour Ocaml

Si vous n'avez pas l'habitude d'installer du logiciel sous Linux, vous pouvez passer ce qui suit. Sinon, sachez qu'il est simple d'installer Jupyter pour Ocaml dans le Bash de Windows (WSL), c'est exactement comme sous Linux.

Voici en résumé la totalité de l'installation. Vous :

- installez WSL qui fournit Bash sous Windows
- téléchargez un script maison d'extension sh nécessaire à l'installation de ocaml-jupyter
- copiez dans votre /home ce script depuis le Bash avec la commande cp (le système de fichiers Windows est accessible depuis /mnt/c où c représente le lecteur C:)
- exécutez la totalité le script avec la commande bash

- relancez le bash (fermer et ré-ouvrir)
- lancez la commande `jupyter notebook` en console depuis le Bash, comme sous Linux,
- copiez à la souris l’url générée que vous collez dans un navigateur Windows
- devriez tomber sur le Dashboard de Jupyter et pouvoir utiliser Jupyter avec un moteur Ocaml.

## Le script d’installation

Pour installer Jupyter pour Ocaml, vous allez exécuter un script dans le Bash (télécharger [ICI](#) et dézipper). Voici le contenu de ce script :

```
# maj
sudo apt update

# jupyter
sudo apt install python-pip
sudo pip install jupyter

# opam
sudo apt install opam

# lib ocaml-jupyter
sudo apt install m4
sudo apt install libzmq3-dev
sudo apt install libgmp-dev
sudo apt install pkg-config
sudo apt install zlib1g-dev
sudo apt install libcairo2-dev

# (indispensable) initialise opam
# A la fin, répondre y pour modification fichiers profile et ocamlinit
opam init

# ocaml-jupyter
opam install jupyter

# hack
mkdir .local
mkdir .local/share
mkdir .local/share/jupyter

# kernel
sudo jupyter kernelspec install --name ocaml-jupyter "$(opam config var share)/jupyter"
```

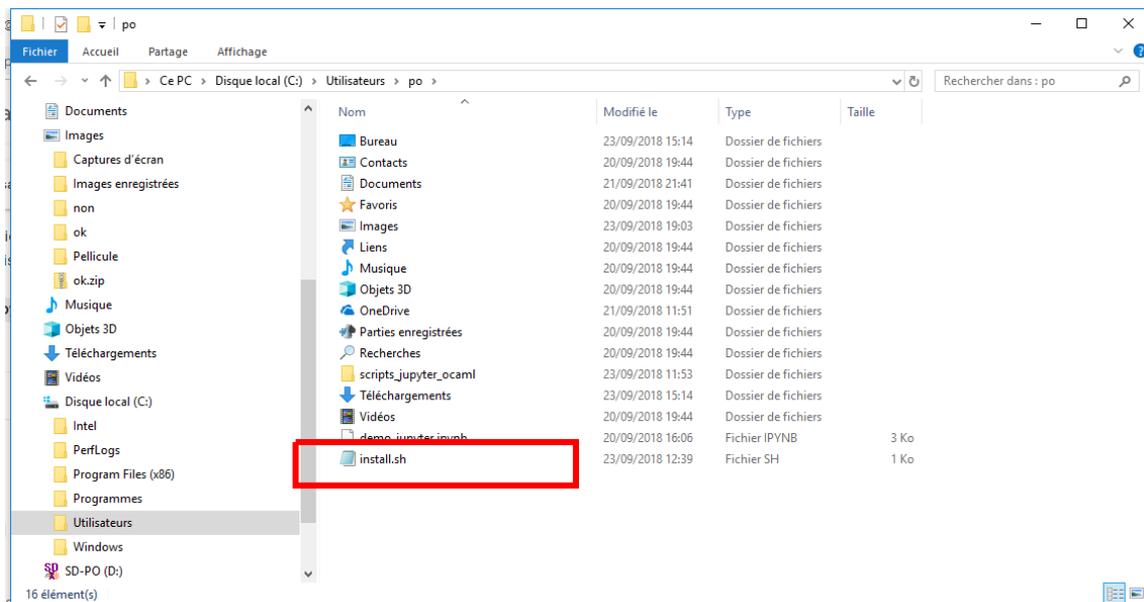
Si vous devez être amené à modifier ce script, faites en sorte de garder des sauts de ligne UNIX (et pas Windows).

Il est commenté (cf. la ligne qui suit chaque hashtag). Quelques explications :

- la première ligne met à jour les paquets de la base de données de apt

- ensuite, Jupyter est installé;
- Opam (gestionnaire de paquets Ocaml) est installé;
- ensuite, des bibliothèques utiles pour ocaml-jupyter sont installées;
- Opam est initialisé; c'est à ce moment-là que l'utilisateur doit choisir une réponse qui n'est pas la réponse par défaut;
- ocaml-jupyter est installé;
- des répertoires sont créés pour contourner une difficulté due à un défaut de droits
- enfin, le kernel Ocaml est déclaré.

Placer le script dans votre répertoire personnel sous Windows : `C:\Users\monNomWindows\install.sh` (par exemple, dans mon cas, monNomWindows est **po**) :



Je n'ai pas installé le package `jupyter-archimedes` qui permet de faire des visualisations 2D, cf. [Github](#) pour pouvoir le faire.

## L'installation de ocaml-jupyter en détail

Pour ce qui suit, vous devez avoir installé le Bash Ubuntu 18.04 sous Windows et devez disposer du mot de passe de super-utilisateur du Bash.

L'installation décrite ci-dessous suit les étapes indiquées dans la page Github du projet [ocaml-jupyter](#); j'ai complété en fonction des succès/insuccès de ma propre installation.

Au moment de l'installation, j'avais un système Windows 10 quasiment inutilisé, sur un portable Dell Latitude. De même, WSL a été installé uniquement pour installer ocaml-jupyter. Si vous utilisez un Bash qui a déjà subi des installations, il se peut que l'installation ci-dessous ne fonctionne pas.

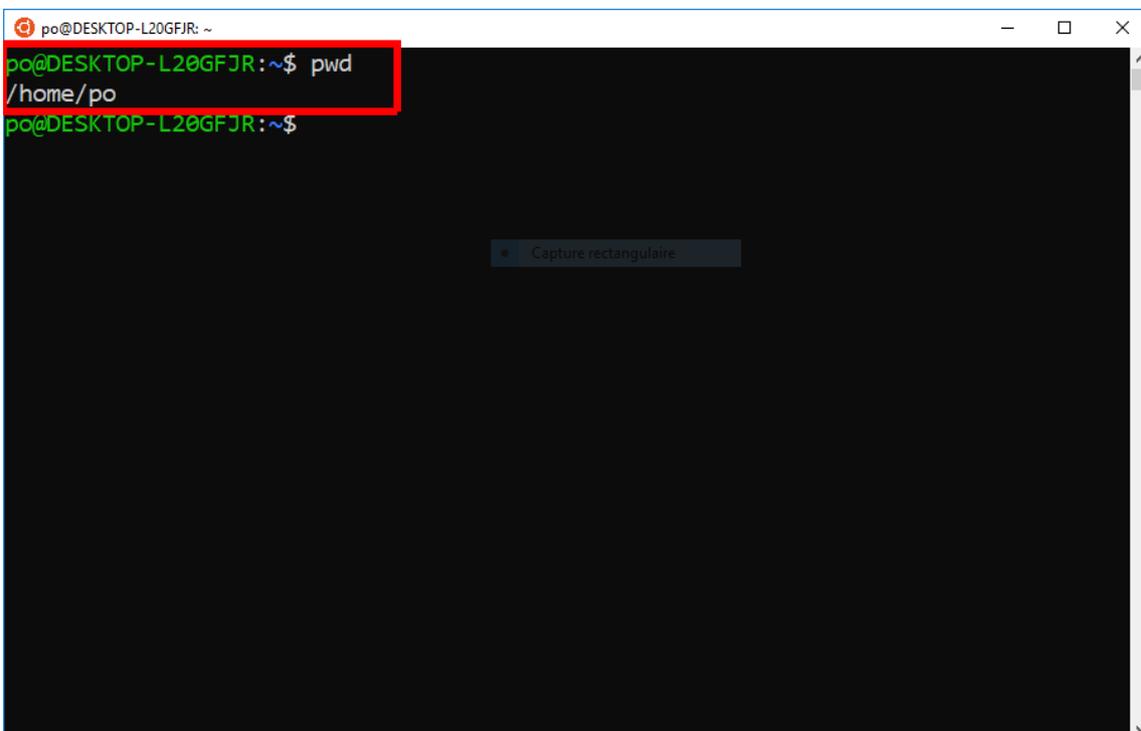
L'installation peut être vue comme complexe si on n'a pas l'habitude d'utiliser une console Bash. Elle est aussi assez longue, elle dure facilement de l'ordre de 30 minutes. Cette durée dépend en fait de la vitesse de votre connexion Internet et de la puissance de votre ordinateur, donc ça peut être 20 minutes ou plus d'une heure.

## Lancement du Bash

Si votre Bash est resté ouvert après que vous l'avez installé, passez à la section suivante. Sinon, il faut le réouvrir. Pour cela : menu Démarrer > Ubuntu.

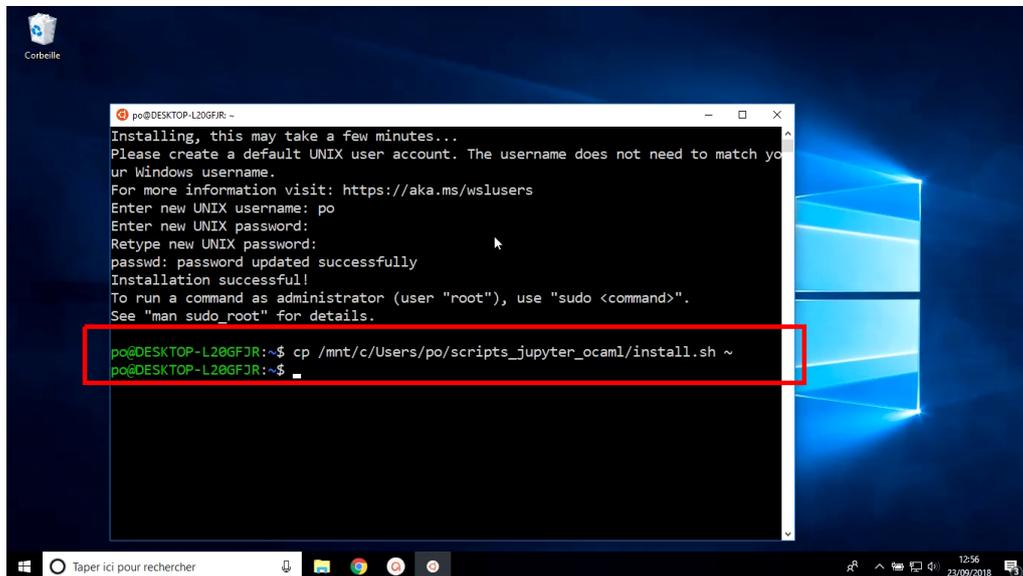


S'assurer que le dossier d'ouverture par défaut du bash est /home/monNomLinux (normalement c'est le cas). En principe, un tilde apparaît dans le prompt ainsi que votre nom d'utilisateur Linux :  
Si vous avez un doute, lancez la commande pwd et vous devriez lire que vous êtes dans /home/monNomLinux :



## Copier le script depuis Bash

Copier "depuis" le Bash le script d'installation depuis leur répertoire `C:\Users\monNomWindows\install.sh` (où il est censé se trouver comme précisé plus haut) vers le répertoire courant :



(sur l'image, l'adresse n'est pas exactement la même).

La commande à saisir dans le Bash est la suivante :

```
cp /mnt/c/Users/monNomWindows/install.sh ~
```

Dans mon cas par exemple, `monNomWindows` est remplacé par `po`.

La fin de la ligne est un caractère tilde qui indique que la destination de la copie est `/home/monNomLinux`, dans mon cas `/home/po`. La commande `cp` permet de copier sous Linux.

Détail : vous pouvez saisir toute la commande mais en fait le système peut le faire largement à votre place : en appuyant sur la touche `TAB`, il faut demander au système de compléter ou de proposer un choix, ce qui vous évite de recopier et évite aussi les erreurs de saisie.

Optionnel : vérifier que les scripts ont bien été copiés :

```
ls
```

ce qui doit afficher

```
install.sh
```

Optionnel : vous pouvez vérifier le contenu du script d'installation en tapant

```
cat install.sh
```

ce qui affichera

```
# maj
sudo apt update

# jupyter
```

```
sudo apt install python-pip
sudo pip install jupyter

# opam
sudo apt install opam

# lib ocaml-jupyter
sudo apt install m4
sudo apt install libzmq3-dev
sudo apt install libgmp-dev
sudo apt install pkg-config
sudo apt install zlib1g-dev
sudo apt install libcairo2-dev

# (indispensable) initialise opam
# A la fin, répondre y pour modification fichier profile et ocamlinit
opam init

# ocaml-jupyter
opam install jupyter

# hack
mkdir .local
mkdir .local/share
mkdir .local/share/jupyter

# kernel
sudo jupyter kernelspec install --name ocaml-jupyter "$(opam config var share)/jupyter"
```

### Exécution du script

Taper dans le Bash la commande

```
bash install.sh
```

et valider :

```
po@DESKTOP-L20GFJR:~$ bash install.sh
[sudo] password for po:
Hit:1 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [83.2 kB]
0% [1 InRelease gpgv 242 kB] [4 InRelease 12.0 kB/83.2 kB 14%]
```

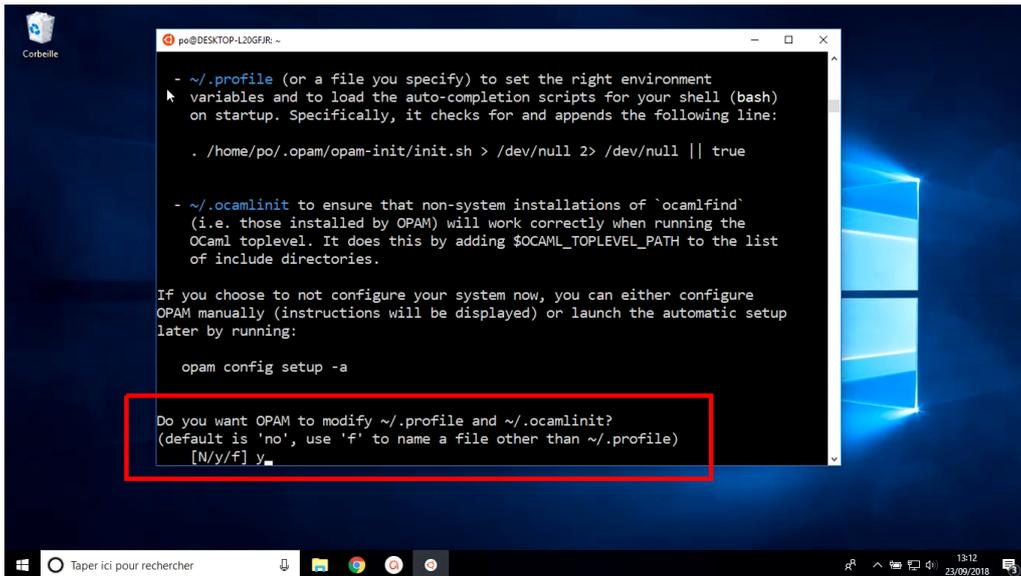
Au total, l'exécution va durer plusieurs dizaines de minutes. Vous serez sollicité à plusieurs reprises.

Une fois (au tout début) pour donner le mot de passe super-utilisateur (cf. image ci-dessus).

A de très nombreuses reprises, une question sera posée et deux ou trois choix de réponses seront possibles dont un unique choix par défaut, écrit en majuscule. A chaque fois **sauf une**, vous pouvez taper sur ENTRÉE ce qui validera le choix par défaut, comme ici :

```
aspud camlp4 clasp darcs gringo ledit libcamlp4-ocaml-dev libfindlib-ocaml
libfindlib-ocaml-dev liblua5.3-0 libncurses5-dev libpthread-stubs0-dev
libtinfo-dev libx11-6 libx11-dev libx11-doc libxau-dev libxcb1-dev
libxdmcp-dev mercurial mercurial-common ocaml ocaml-base ocaml-base-nox
ocaml-compiler-libs ocaml-findlib ocaml-interp ocaml-nox opam-docs unzip
x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
ncurses-doc libxcb-doc kdiff3 | kdiff3-qt | kompare | meld | tkcvs | mgdiff
qct python-mysqldb python-openssl python-pygments wish ocaml-doc tuareg-mode
| ocaml-mode zip
The following NEW packages will be installed:
aspud camlp4 clasp darcs gringo ledit libcamlp4-ocaml-dev libfindlib-ocaml
libfindlib-ocaml-dev liblua5.3-0 libncurses5-dev libpthread-stubs0-dev
libtinfo-dev libx11-dev libx11-doc libxau-dev libxcb1-dev libxdmcp-dev
mercurial mercurial-common ocaml ocaml-base ocaml-base-nox
ocaml-compiler-libs ocaml-findlib ocaml-interp ocaml-nox opam-docs unzip
x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
The following packages will be upgraded:
libx11-6
1 upgraded, 34 newly installed, 0 to remove and 51 not upgraded.
Need to get 88.8 MB of archives.
After this operation, 478 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

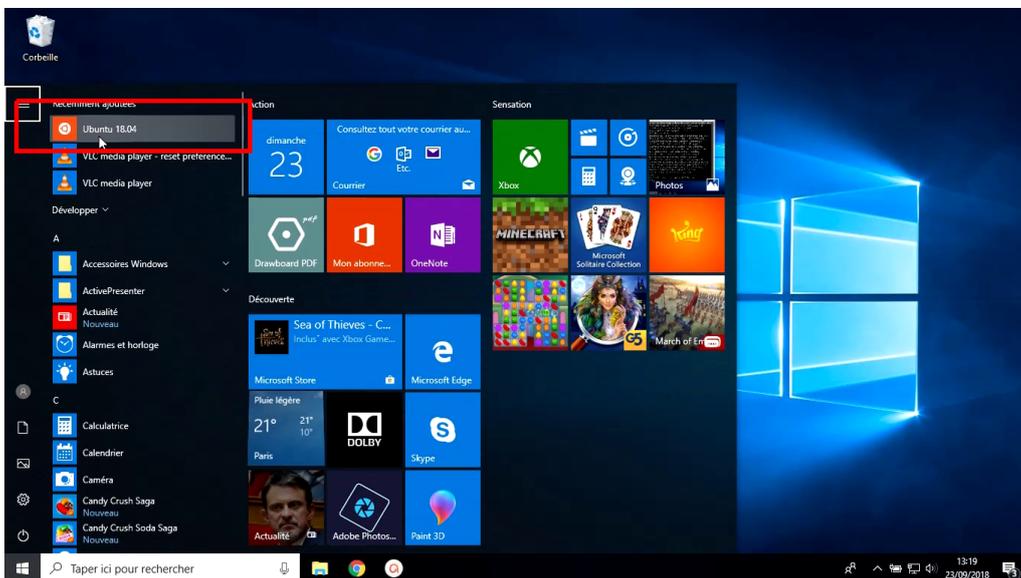
A une seule reprise (lorsqu'il s'agit d'initialiser Opam, plutôt dans le dernier tiers de l'installation, voir l'image ci-dessus) :



il faudra donner un choix autre que celui par défaut. Le choix par défaut est N et le choix à taper est y.

### Lancement de Jupyter pour Ocaml

Une fois le script exécuté et de retour au prompt du Bash, **il faut fermer le Bash** (en cliquant sur la croix de la fenêtre) et le relancer via le Menu démarrer :



Le lancement d'une feuille Jupyter se fait alors en tapant la commande en minuscules :

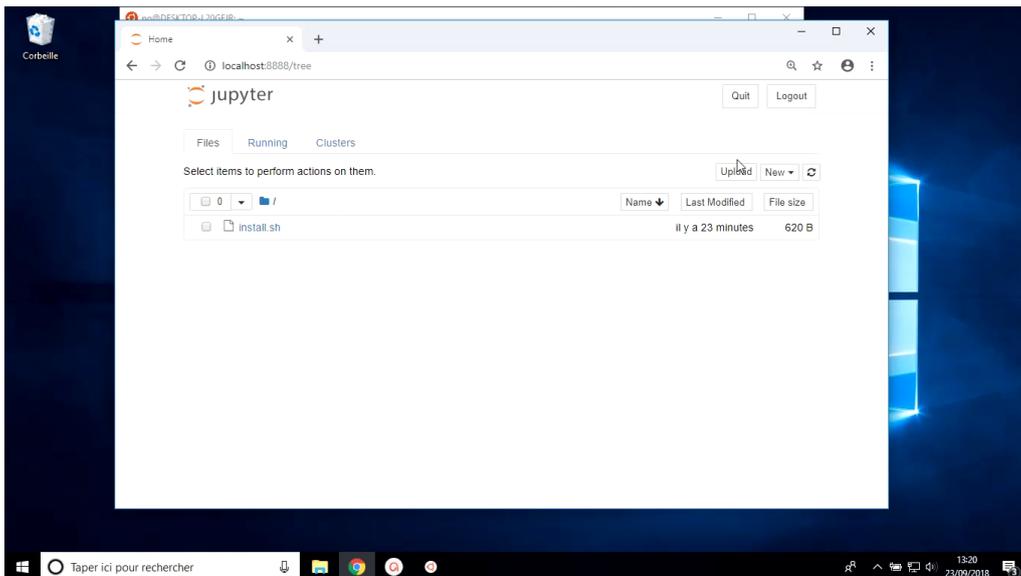
```
jupyter notebook
```

Cette commande crée une URL en http dans la sortie de la console. Sélectionner l'URL à la souris puis copier avec un clic-droit (ne pas s'étonner que cela désélectionne) :

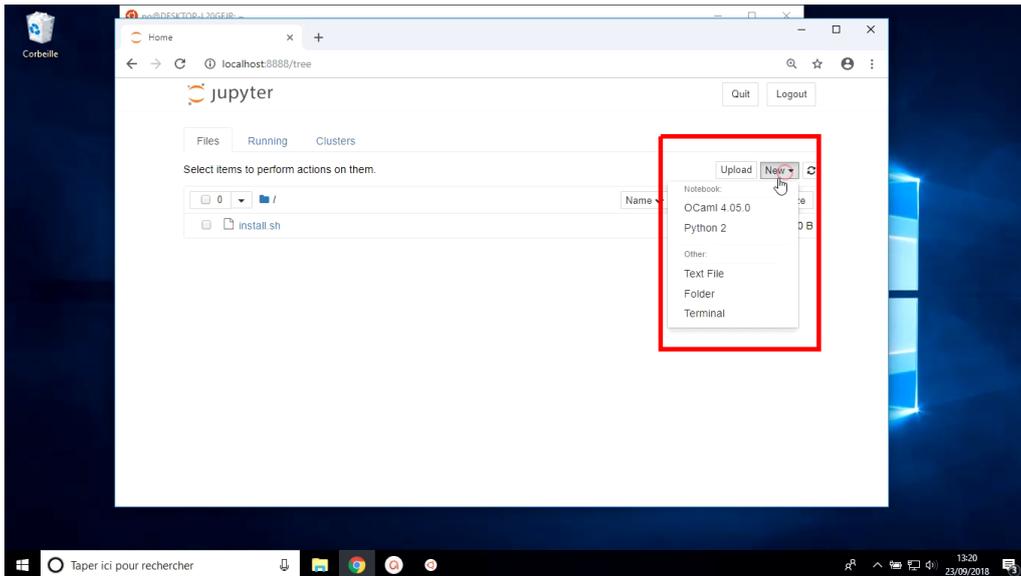
```
po@DESKTOP-L20GFJR:~$ jupyter notebook
[I 13:19:50.888 NotebookApp] Writing notebook server cookie secret to /home/po/.local/share/jupyter/runtime/notebook_cookie_secret
[I 13:19:51.360 NotebookApp] Serving notebooks from local directory: /home/po
[I 13:19:51.360 NotebookApp] The Jupyter Notebook is running at:
[I 13:19:51.361 NotebookApp] http://localhost:8888/?token=c68c692a6a791c4830f635148255fcbbb39dfcd94b8c5633
[I 13:19:51.361 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 13:19:51.367 NotebookApp] No web browser found: could not locate runnable browser.
[C 13:19:51.367 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=c68c692a6a791c4830f635148255fcbbb39dfcd94b8c5633
```

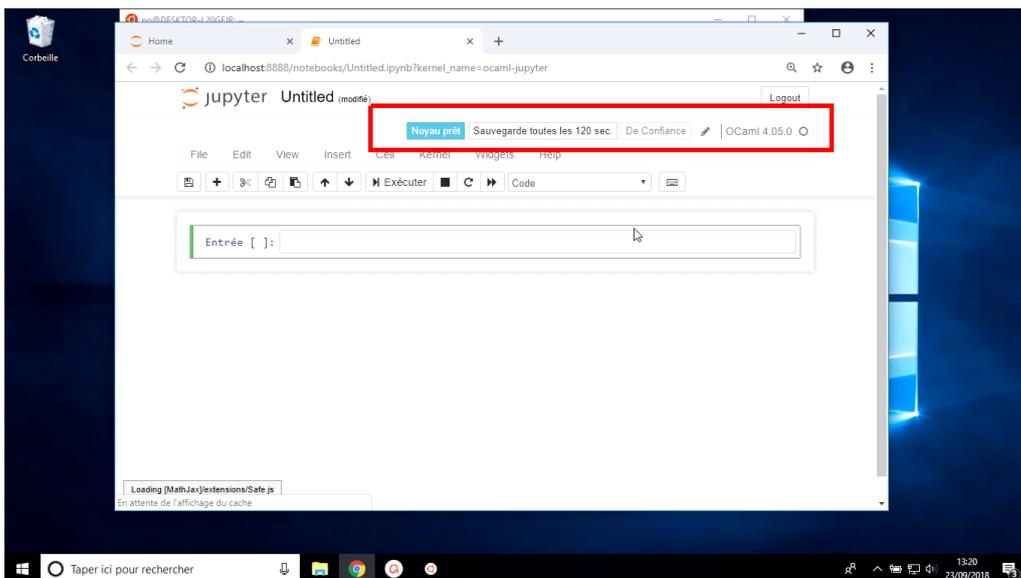
Ouvrir un navigateur Windows (j'ai testé avec Chrome et Microsoft Edge) et coller l'URL dans la barre d'adresse puis valider. Cela ouvre le Dashboard de Jupyter :



Cliquer sur le bouton New et, parmi les deux moteurs disponibles, choisir Ocaml et valider.

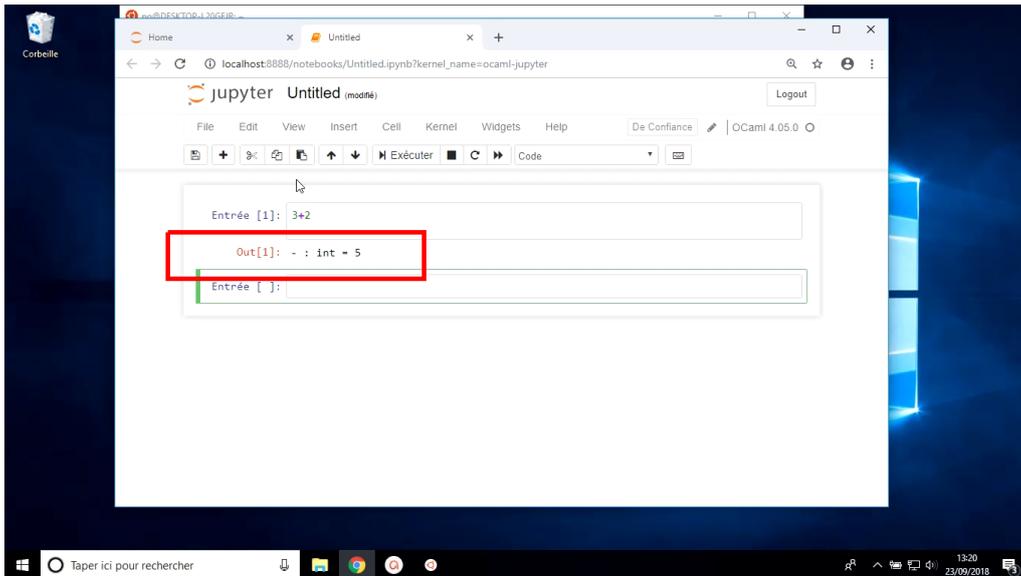


Cela ouvre une feuille Jupyter Notebook avec des cellules pour du code Ocaml. Vous devriez entre-apercevoir une étiquette sur fond bleu clair qui indique **Noyau prêt** :



Le nom du moteur est visible en haut à droite, avec un cercle à fond clair (si c'est foncé, c'est que le moteur ne charge pas).

Essayez du code Ocaml et vérifiez qu'il fonctionne correctement :



Pour quitter, fermer le navigateur (Jupyter sauvegarde automatiquement) et interrompre le serveur qui donne accès à votre feuille Jupyter, en tapant "dans" le Bash, CTRL+C autant de fois que nécessaire ce qui fait réapparaître le prompt du Bash. Par défaut, votre feuille est sauvegardée dans le répertoire de lancement de Jupyter (a priori, votre /home).

## Difficultés possibles

Au cours des multiples tentatives d'installation, j'ai rencontré deux difficultés principales :

- a) D'abord, une impossibilité du chargement du moteur Ocaml (pourtant bien présent dans le menu) dans la feuille Jupyter, le message d'erreur étant :

```
dllwt_unix_stubs.so: cannot open share object file: No such file or directory
```

et un encadré dans le navigateur indiquant

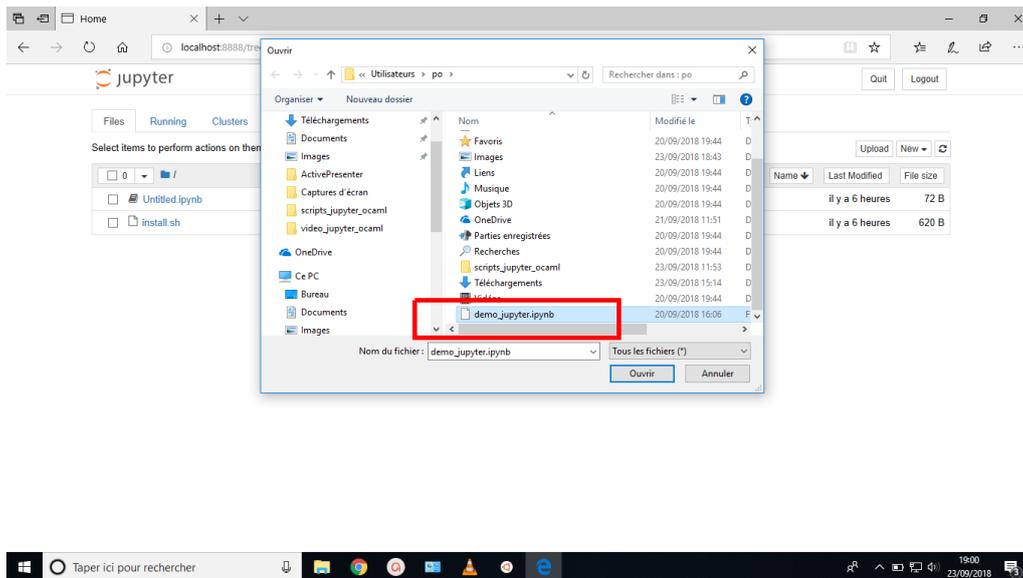
```
Noyau en cours de démarrage, patientez ...
```

- b) Autre difficulté : au moment du lancement de la commande `jupyter notebook`, message d'erreur indiquant que je n'ai pas les droits sur le répertoire `~/local/share`, d'où la création préventive dans le script par l'utilisateur des répertoires utiles (cf. les commandes `mkdir` dans le script). On peut contourner cette difficulté en lançant `jupyter notebook` en root (déconseillé) avec le drapeau `--allow-root` mais cela me ramenait à la difficulté ci-dessus.

## Comment importer une feuille Jupyter ?

Si on dispose d'une feuille jupyter, d'extension `ipynb` et qu'on veut la charger dans Jupyter depuis le Bash, comment fait-on ?

Il suffit d'ouvrir une feuille Jupyter Notebook par la procédure habituelle. Une fois devant le Dashboard, cliquer sur le bouton Upload ce qui ouvre l'arborescence Windows depuis votre répertoire `C:\Users\monNomWindows`. Là chercher et charger votre feuille Jupyter.



De retour au Dashboard, cliquer sur le bouton **Téléverser** puis, cliquer sur le nom de la feuille : cela ouvre la feuille qui, si le moteur est reconnu, sera fonctionnelle.

## Comment exporter une feuille Jupyter ?

On vient de créer ou d'importer une feuille Jupyter Notebook, matérialisée par un fichier d'extension ipynb et présent à la racine de votre home. Vous voulez l'exporter vers votre système de fichiers Windows. Pour cela, dans le menu File de la feuille, choisir **Download as** puis le format ipynb et la feuille est placée par défaut dans le répertoire Téléchargements de Windows.

