

JUPYTER NOTEBOOK

L'accès à cette application est disponible via le widget « Mes applis »



Table des matières

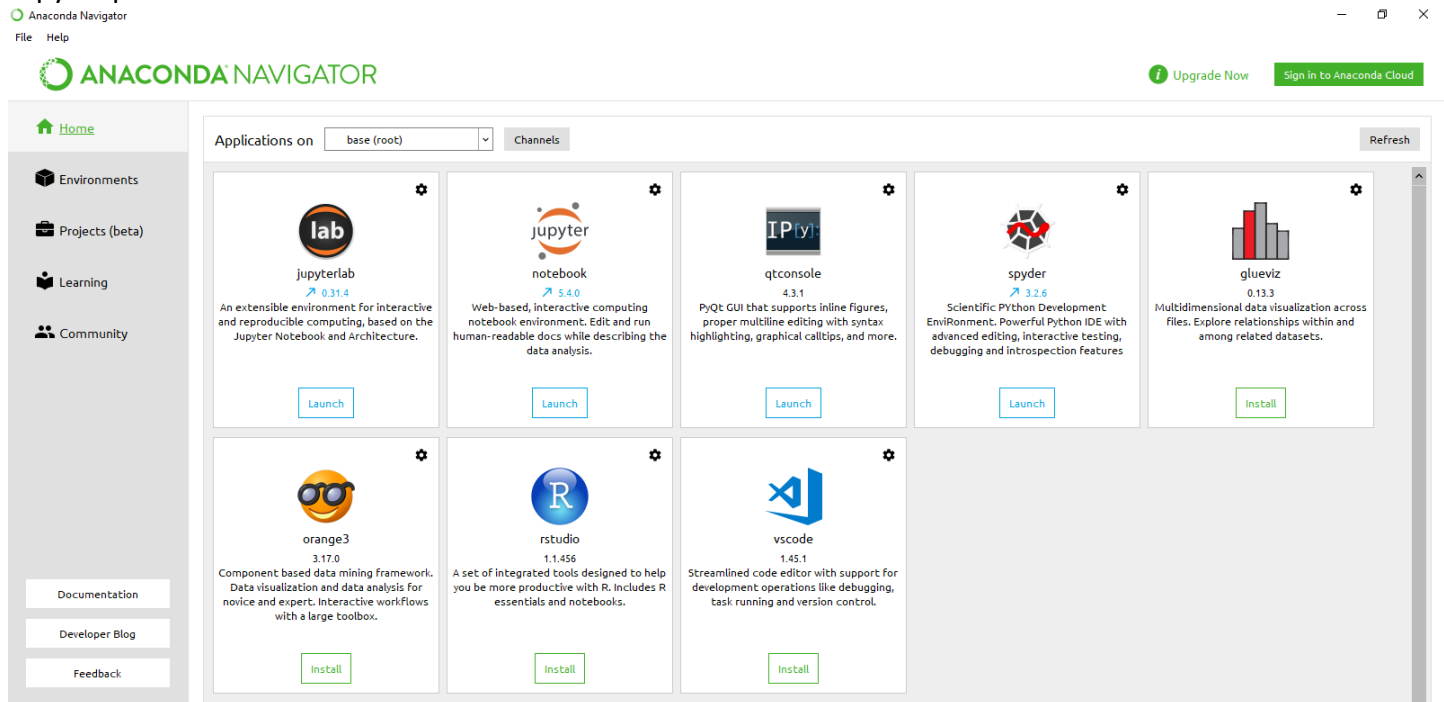
Présentation	2
1. L'environnement	3
2. Un premier bloc-notes	4
2.1. Cellules de code	5
2.2. Console de code.....	6
2.3. Cellule Markdown.....	6
2.4. Cellule brute.....	7
3. Les terminaux	7
4. Gestion des noyaux et des terminaux.....	8
5. Partage de bloc-notes.....	9
6. Exportation de bloc-notes	10
7. Bibliographie.....	10

Présentation

Les notebooks Jupyter sont des documents numériques qui peuvent rassembler du texte, des images, des formules mathématiques et du code informatique exécutable. Ils sont manipulables interactivement via un navigateur web, dans l'application.

Le projet Jupyter pour Julia, Python et R, trois langages utilisés dans la science des données (data science), permet de créer des fichiers bloc-notes d'extension .ipynb (I PYthon NoteBook).

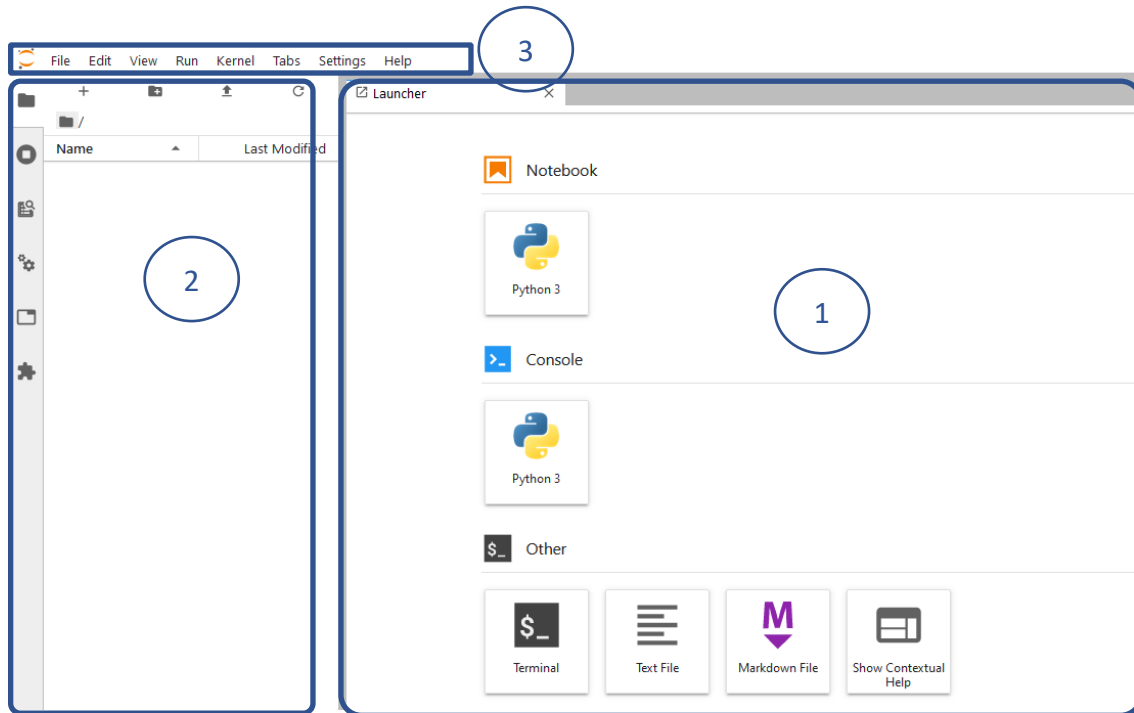
Jupyter peut être installé localement sur votre ordinateur avec la distribution Anaconda.



JupyterLab est une nouvelle solution de développement complètement open source, disponible en ligne via le web sur des plateformes variées :

- Binder (<https://mybinder.org/>)
- Colaboratory (<https://colab.research.google.com/notebooks/welcome.ipynb>)
- Azure Notebooks (<https://notebooks.azure.com/#>)
- Dans l'ENT Lycée Connecté.

1. L'environnement







L'interface JupyterLab se compose d'

- 1 Une zone de travail contenant des onglets de documents et d'activités
- 2 Une barre latérale gauche repliable
- 3 Une barre de menus.

La zone de travail nous propose 3 domaines :

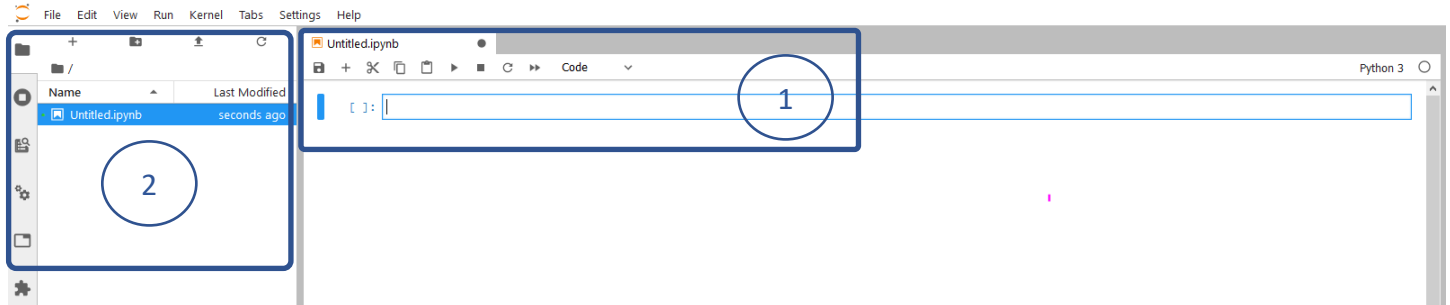
- Un outil d'écriture et d'exécution de Notebook  Notebook
- L'accès à une console Python  Console
- Des outils variés  Other

- Un terminal Linux 
- Un éditeur de texte brut 
- Un éditeur de fichiers Markdown 
- Une zone d'aide contextuelle 

2. Un premier bloc-notes

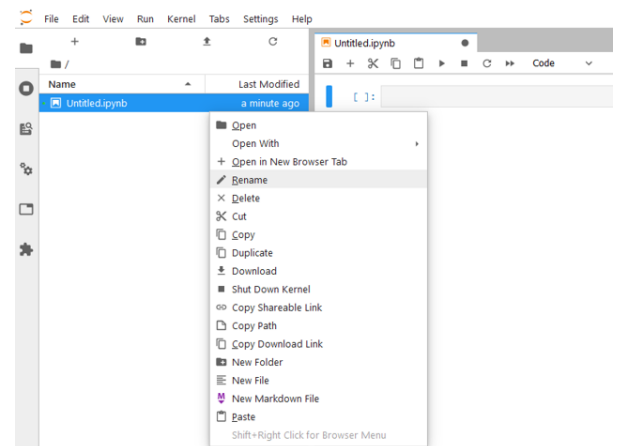


On clique sur la vignette Notebook Python 3

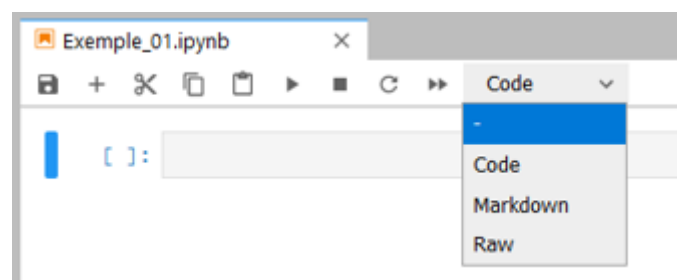


1 Lorsque vous créez un nouveau document de bloc-notes, le nom du bloc - notes, une barre de menus, une barre d'outils et une cellule de code vide s'affichent.

2 Un nouveau fichier est créé avec un nom par défaut. Renommez un fichier en cliquant avec le bouton droit sur son nom dans le navigateur de fichiers et en sélectionnant « Renommer » dans le menu contextuel :



Le bloc-notes se compose d'une séquence de cellules. Une cellule est un champ de saisie de texte multiligne, et son contenu peut être exécuté en utilisant Shift-Enter, ou en cliquant soit sur le bouton "Lecture" de la barre d'outils, soit sur Cellule, Exécuter dans la barre de menus. Le comportement d'exécution d'une cellule est déterminé par le type de cellule. Il existe trois types de cellules :



2.1. Cellules de code

Une cellule de code vous permet d'éditer et d'écrire un nouveau code, avec une coloration syntaxique et une tabulation.

Le langage de programmation que vous utilisez dépend du noyau (kernel) et le noyau par défaut (IPython) exécute le code Python 3.

Lorsqu'une cellule de code est exécutée, le code qu'elle contient est envoyé au noyau associé au bloc-notes. Les résultats renvoyés par ce calcul sont ensuite affichés dans le bloc-notes en tant que sortie de la cellule.

Il est important que les cellules de code soient évaluées dans le bon ordre. En fait, évaluer un programme sous forme de notebook revient à le découper en petits fragments, et si on exécute ces fragments dans le désordre, on obtient naturellement un programme différent.

```
[1]: a = 10

[2]: print(a)
10

[3]: for i in range(5):
      print(a + i)
10
11
12
13
14

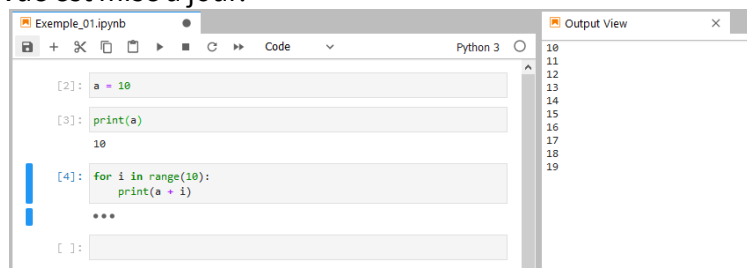
[ ]:
```

Il est possible de :

- Glisser et déposer les cellules pour réorganiser votre bloc-notes.
- Glisser les cellules entre les blocs-notes pour copier rapidement leur contenu
- Réduire et développer le code et la sortie à l'aide du menu Affichage ou du bouton de réduction bleu à gauche de chaque cellule.

Par ailleurs, une fonctionnalité intéressante est la création de vues. Si vous avez affiché des données dans votre bloc-notes, elles apparaissent sous la forme d'une sortie (output).

Dans JupyterLab mettez-vous au niveau de la sortie et utilisez le bouton de droite de la souris afin de choisir l'option « Create new view for Output ». Cette sortie apparaît alors dans une nouvelle fenêtre et reste disponible lorsque vous vous trouvez ailleurs dans votre code. De plus, si vous relancer la cellule d'affichage de cette sortie avec des modifications, la vue est mise à jour.



L'indentation (Tab) est automatique et les info-bulles (Shift Tab) donnent accès à l'aide contextuelle.

```
chaîne = 'un caractère ou une chaîne de caractère'
type(chaîne)

Init signature: type(self, /, *args, **kwargs)
Docstring:
type(object_or_name, bases, dict)
type(object) -> the object's type
type(name, bases, dict) -> a new type
Type:
type
Subclasses:
ABCMeta, EnumMeta, _TemplateMetaclass, MetaHasDescriptors, _ProtocolMeta,
NamedTupleMeta, LexerMeta, StyleMeta, _NormalizerMeta, CachedMetaClass, ...
```

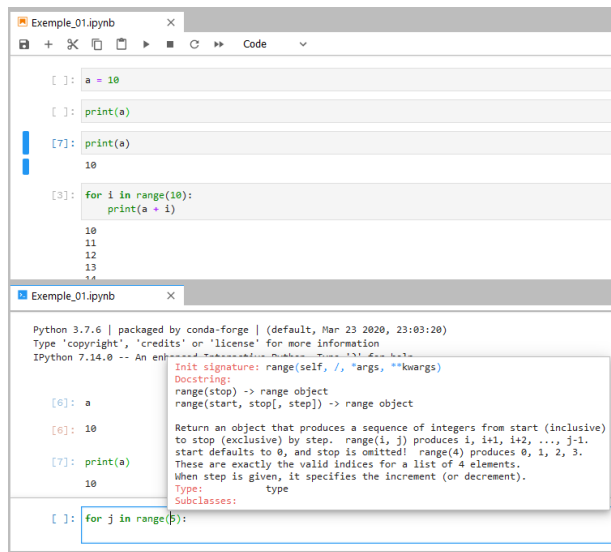
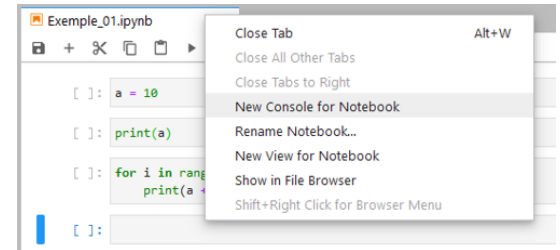


Cette zone peut être maintenue visible par l'outil zone d'aide contextuelle

2.2. Console de code

Vous pouvez connecter une console de code à un noyau de bloc-notes pour avoir un journal des calculs effectués dans le noyau, et l'ordre avec lequel ils ont été effectués. La console de code jointe fournit également un endroit pour inspecter interactivement l'état du noyau sans changer le bloc-notes.

Faites un clic droit sur un bloc-notes et sélectionnez « Nouvelle console pour bloc-notes » :



Dans la console, exécutez le code à l'aide de **Shift Enter**

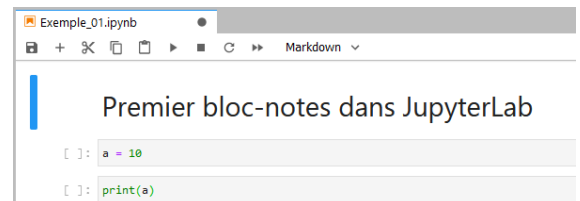
Utilisez les flèches haut et bas pour parcourir l'historique du code précédemment exécuté.

Dans l'état actuel la console n'est pas associée au bloc-notes.

L'indentation (**Tab**) et les info-bulles (**Shift Tab**) fonctionnent comme dans le bloc-notes.

2.3. Cellule Markdown

Le langage Markdown fournit un moyen simple de spécifier les parties du texte à mettre en évidence (italique), en gras, les listes de formulaires, etc.



Le code Markdown est converti en texte enrichi formaté correspondant après avoir exécutée la cellule Markdown.

Dans les cellules Markdown, vous pouvez également inclure les formules mathématiques de manière simple, en utilisant la notation LaTeX standard : $\frac{1}{T} \int_a^{a+T} f(t) dt$ pour les formules en ligne et $\frac{1}{T} \int_a^{a+T} f(t) dt$ pour les formules affichées.

La valeur moyenne en ligne $\frac{1}{T} \int_a^{a+T} f(t) dt$

La valeur moyenne en ligne $\frac{1}{T} \int_a^{a+T} f(t) dt$

La valeur moyenne affichée dans la page $\frac{1}{T} \int_a^{a+T} f(t) dt$

La valeur moyenne affichée dans la page

$$\frac{1}{T} \int_a^{a+T} f(t) dt$$



2.4. Cellule brute

Les cellules brutes fournissent un endroit où vous pouvez écrire directement la sortie. Les cellules brutes ne sont pas évaluées dans le bloc-notes.

Elles ne seront converties qu'après l'export du bloc-notes.

Dans le bloc-notes la cellule n'est pas évaluée.

Un exemple de cellule brute ci-dessous

```
<!DOCTYPE html>
<html lang = "fr">
<head>
<meta charset = "utf-8">
<title>page web exemple</title>
</head>
<body>
<p>Cellule RAW de code HTML</p>
</body>
</html>
```

Lors d'un export en HTML on peut observer

Un exemple de cellule brute ci-dessous

Cellule RAW de code HTML

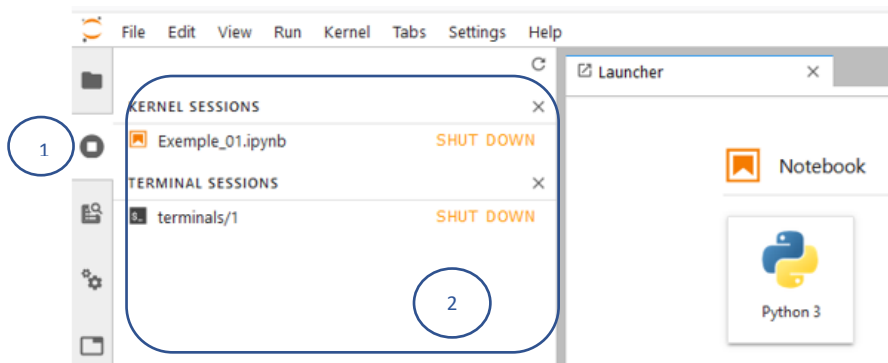
3. Les terminaux

Les terminaux JupyterLab fournissent une prise en charge du shell système (bash) :



```
jovyan@a71a4240c8a5: /
jovyan@a71a4240c8a5:~$ cd /
jovyan@a71a4240c8a5:/$ ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys usr var
jovyan@a71a4240c8a5:/$
```

4. Gestion des noyaux et des terminaux

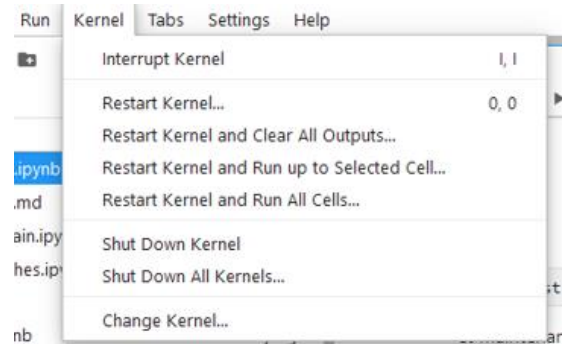


1 Le panneau « En cours d'exécution » dans la barre latérale gauche affiche une liste de tous les noyaux et terminaux en cours d'exécution sur tous les blocs-notes, consoles de code et répertoires.

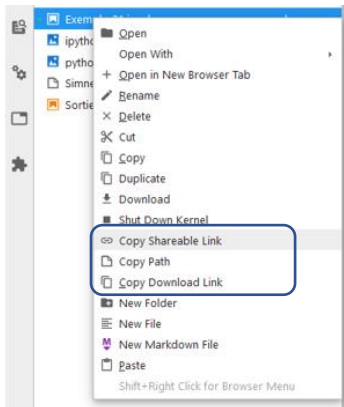
2 Lorsque vous fermez un document de bloc-notes, une console de code ou un terminal, le noyau ou le terminal sous-jacent exécuté sur le serveur continue de fonctionner. Cela vous permet d'effectuer des actions de longue durée et de revenir plus tard.

Important : Les noyaux ne sont pas fermés immédiatement lorsqu'on quitte JupyterLab et même Lycée Connecté. Une bonne habitude est donc de le faire manuellement.

Le menu Kernel (Noyau) offre la possibilité de réinitialiser le noyau, effacer les sorties dans le bloc-notes et exécuter un ensemble de cellules.



5. Partage de bloc-notes



Trois formes de lien vers un bloc-notes sont disponibles dans JupyterLab. Voici un exemple de leur forme :

Copy Path

Exemple_01.ipynb

Ce lien est utilisable pour lier un bloc-notes à un autre bloc-notes

Copy Shareable Link

https://jupyter.lyceeconnecte.fr/hub/user-redirect/lab/tree/Exemple_01.ipynb

Utilisé directement (ce lien renvoie sur l'authentification ENT) ou avec une authentification préalable, ce lien ouvre JupyterLab mais ne permet pas d'accéder au fichier.

Copy Download Link

https://jupyter.lyceeconnecte.fr/user/prof.graves/files/Exemple_01.ipynb?_xsrf=2%7C6d90059a%7Ca74bdc1267af4c1258b2aaed781f991b%7C1590089676

Utilisé directement (ce lien renvoie sur l'authentification ENT) ou avec une authentification préalable, ce lien ouvre JupyterLab mais ne permet pas d'accéder au fichier.

Dans l'état actuel ces deux derniers liens ne fonctionnent pas.

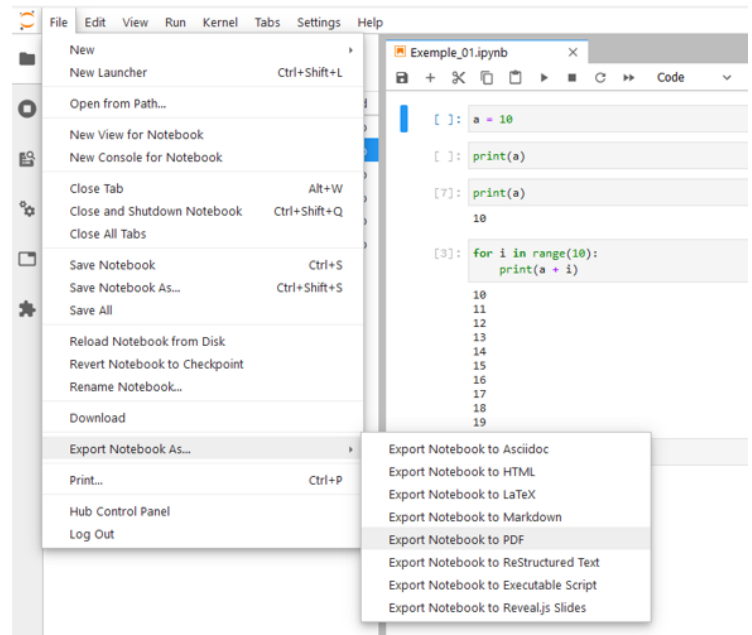
Il est possible de télécharger le bloc-notes afin de le partager avec d'autres utilisateurs.





6. Exportation de bloc-notes

JupyterLab vous permet de télécharger ou d'exporter vos fichiers de bloc-notes jupyter (.ipynb) dans d'autres formats de fichiers



Dans l'état actuel l'export PDF ne fonctionne pas. On pourra utiliser :

- L'impression dans un fichier .pdf
- L'export en HTML, puis imprimer en .pdf depuis Chrome en enlevant les entêtes.

7. Bibliographie

<https://jupyter-notebook.readthedocs.io/en/stable/index.html>

<https://jupyterlab.readthedocs.io/en/stable/index.html>

https://python.sdv.univ-paris-diderot.fr/18_jupyter/#185-les-magic-commands