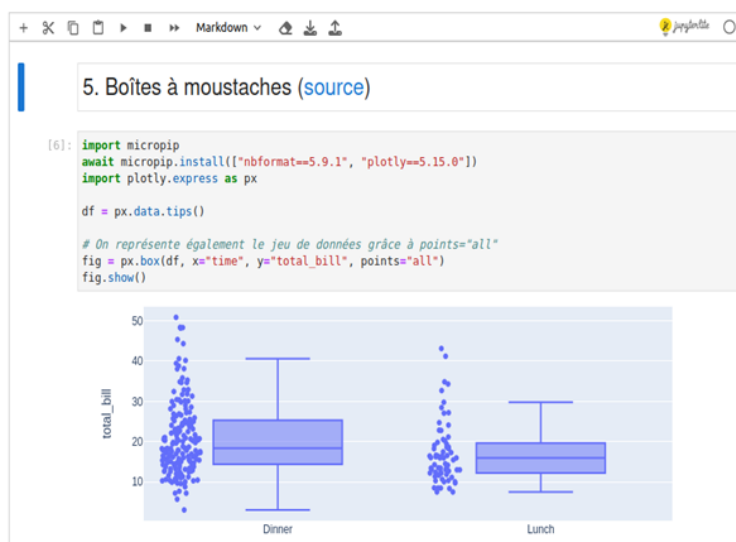


L'outil d'édition de code Python  permet de combiner dans un labdoc du code informatique, des résultats d'exécution et du texte. C'est une version allégée de Jupyter Notebook (JupyterLite) qui s'exécute dans un navigateur.

A – Vue d'ensemble

Le labdoc ci-contre comporte quatre parties :

- La **barre d'outils** permet de
 - insérer et exécuter des cellules
 - importer et exporter des données
 - connaître l'état du noyau de calcul (moteur d'exécution).
- La **cellule de texte** affiche un texte formaté ainsi qu'un lien.
- La **cellule de code** contient du code qui peut être exécuté.
- La visualisation finale est **le résultat d'exécution** de la cellule de code.



B – Fonctionnalités de base de la barre d'outils

Gérez les cellules avec les boutons [Ajouter], [Couper], [Copier], [Coller]

Choisissez leur type avec le menu déroulant :

- Markdown : cellule de texte qui peut être mis en forme.
- Code : cellule de code exécutable.

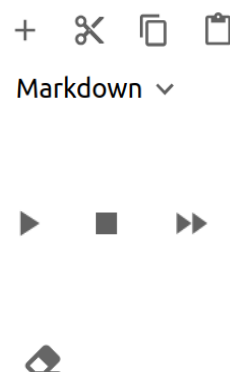
Exécutez le code avec les boutons [Exécuter une cellule], [Arrêter le noyau d'exécution], [Redémarrer le noyau puis exécuter toutes les cellules] ou les raccourcis :

- Ctrl+Entrée : exécute une cellule
- Maj+Entrée : exécute une cellule puis en insère une nouvelle en-dessous

Le bouton gomme permet d'effacer tous les résultats d'exécution.

L'état du noyau est indiqué par l'icône tout à droite :

- ☐ Initialisation (ouverture de l'outil)
-  Disponible (prêt pour une exécution)
-  Occupé (le noyau effectue les calculs)



C – Utiliser des packages préinstallés ou importés

Les packages installés par défaut sont les suivants :

- Les principaux packages de la librairie standard (math, csv, io, ...)
- Matplotlib-inline version 0.1.6
- numpy version 1.23.5
- pandas version 1.5.2
- micropip version 0.2.0

L'utilisation de packages préinstallés dans une cellule se fait simplement avec **import**

```
import numpy as np
import matplotlib.animation
import math
```

micropip permet d'importer des packages publiés sur une plateforme d'hébergement de packages.

Important : indiquez précisément le numéro de version du package importé. Cela assure que votre code reste exécutable si la version du package vient à changer.

Le code ci-contre permet d'utiliser dans une cellule des packages non pré-installés

```
import micropip
# plotly a besoin de nbformat pour fonctionner
await micropip.install(["nbformat==5.9.1", "plotly==5.15.0"])
import plotly.express as px
```

D – Importer et exporter des données tabulaires

Import de données

Pour utiliser un fichier de données tabulaires (par ex. un csv) dans un labdoc code, il faut qu'il soit disponible dans l'outil « Ressources » du rapport.

1.  Cliquez sur le bouton d'import de données pour générer le code ci-contre dans la première cellule
2. Configurez puis exécutez ce code afin d'obtenir les données, au format désiré, dans une variable

```
# séparateur de colonnes
sep = ";"

# format souhaité : "pandas", "numpy", "txt"
data_format = "pandas"
url = "http://.../mes-donnees.csv"

# nom de la variable qui récupère les données
data = import_file(url=url, sep=sep,
data_format=data_format)
```

Export de données

1.  Cliquez sur le bouton d'export de données qui va ajouter une cellule avec le code ci-contre
2. Configurez puis exécutez ce code pour générer un lien de téléchargement d'un fichier .csv

```
# variable contenant vos données
data =

# format de vos données: "pandas", "numpy",
"list"
data_type = "pandas"
export_file(data=data, data_type=data_type,
filename="data.csv")
```

En savoir plus : de nombreux exemples sont disponibles dans la mission publique « LDcode - Exemples de labdocs code sous JupyterLite »