

02_prise-en-main-notebooks-jupyter

November 29, 2023

1 Activité : prise en main d'un notebook Jupyter

Adapté de C. Poulmaire / AEIF

Cette activité a pour objectif de vous faire rapidement explorer les **fonctionnalités de base** de l'interpréteur interactif **jupyter** en mode **Notebook**.

1.1 Exercice 1 : édition et exécution d'une cellule de code

Un notebook jupyter consiste en **cellules** individuelles pouvant contenir du **code** Python, du **texte**, ou des **graphiques**. Commençons par les cellules de type Code.

1. Placez le curseur dans la 1ère cellule Entrée[]: ci-dessous puis frappez <enter>. Que se passe-t-il ? Rien d'autre que l'insertion d'un saut de ligne... C'est normal, car dans une cellule de type "Code" (c'est-à-dire préfixée par Entrée[]:) on est par défaut en mode **édition de code** Python !

Pour **exécuter le code** Python d'une cellule, il faut en effet frapper <maj-entrée> (ou cliquer sur le bouton , ou encore utiliser le menu **Cellule > Exécuter les cellules et sélectionner la suivante**). Essayez !

```
[ ]: 15 + 24
```

```
[ ]: 'Hello tout le monde !'
```

Une ligne Sortie[]: devrait être apparue en dessous de la 1ère expression, affichant le résultat de celle-ci ! Le curseur est alors automatiquement passé dans la cellule d'en-dessous que vous pouvez exécuter à son tour.

*Notez qu'il est aussi possible de frapper <ctrl-entrée> (identique à **Cellule > Exécuter les cellules**). Dans ce cas, le curseur **reste dans la cellule** après l'exécution du code.*

2. Exécutez maintenant le code de la cellule ci-dessous.

```
[ ]: a = 6      # affectation de la valeur 6 à la variable a
     b = 10     # affectation de la valeur 10 à la variable b

     print('La somme de ces 2 nombres vaut :', a + b)
```

Remarquez que l'exécution du code ci-dessus n'a retourné aucune valeur. Il n'y a pas de prompt Sortie[], uniquement l'affichage d'un texte.

3. Peut-on **modifier le code** d'une cellule puis le **ré-exécuter** ? Bien sûr !
Expérimentez cela dans la cellule ci-dessus en modifiant la valeur des variables a et b, puis frappez à nouveau <maj-entrée>

1.2 Exercice 2 : création de cellules de code

Comment puis-je créer **mes propres cellules** de code ? Tout simplement de la façon suivante :

sélectionnez d'abord la cellule juste au-dessus (resp. juste en dessous) de l'emplacement où vous désirez insérer votre propre cellule

cliquez sur le bouton ou faites Insérer > Insérer une cellule après (resp. Insérer > Insérer une cellule avant)

1. Exercez-vous en créant, juste en-dessous de la présente cellule, une nouvelle cellule.
2. Insérez-y le code `print('Waouh... ma première cellule !')`, puis exécutez le.

1.3 Exercice 3 : création de cellules de texte (markdown)

Il existe 2 types de cellules de texte, selon le choix effectué à l'aide du menu déroulant de droite ou avec **Cellule > Type de cellule >...** : - **Texte Brut** : cellules de texte sans aucun formatage, directement éditables sans devoir *double-cliquer* dedans ; - **Markdown** : cellules dans lesquelles le texte est soumis à un formatage selon la syntaxe Markdown.

1. Créez juste ci-dessous une cellule de type **Texte Brut** . Pour cela, créez une cellule par le procédé standard (elle sera donc de type Code), puis changez son type à l'aide du menu déroulant. Constatez que la cellule reste toujours en mode édition.
2. Pour le format **Markdown** on peut également n'y mettre que du texte sans aucun formatage ou, si l'on est curieux, aller consulter le lien [Rédigez en Markdown !](#) - ou même regarder (en double cliquant dans les cellules) le code des cellules de texte du présent notebook! Créer de nouveau juste en dessous une cellule de type **Markdown** et saisir un texte dedans - le format est à votre choix!

Pour plus d'informations : [un Notebook assez complet sur les cellules au format Markdown](#)

1.4 Exercice 4 : destruction de cellules

Pour détruire une cellule, que ce soit du code ou du texte, sélectionnez-la puis cliquez sur le bouton (ou faites **Edition > Supprimer les cellules**). Exercez-vous en détruisant les deux cellules ci-dessous. Puis rétablissez-les avec **Edition > Annuler la suppression des cellules**

```
[ ]: # Une cellule de code à jeter !!!
```

Une cellule de texte à jeter

Notez qu'il existe également : le bouton (correspondant au menu **Edition > Copier les cellules**) ainsi que le bouton (correspondant au menu **Edition > Coller les cellules**)

1.5 Exercice 5 : sauvegarder et charger un notebook

1. N'oubliez pas de le sauvegarder régulièrement votre travail dans un notebook avec le bouton ci-dessus (et bien entendu lorsque vous avez terminé de travailler!).
2. Téléchargez (au format ipynb) votre notebook si vous le souhaitez au moyen du menu Fichier > Enregistrer sous....
3. Vous pouvez également charger un notebook le menu Fichier > Charger un notebook...).

1.6 Exercice 6 : une première utilisation d'un notebook avec du code Python

1.6.1 Exécuter les cellules suivantes - comme vous l'aviez fait vendredi dans la console

```
[ ]: (2 + 6) * 3
```

```
[ ]: 2 + 6 * 3
```

```
[ ]: 3 - 4 ** 4
```

```
[ ]: (3 - 4) ** 4
```

```
[ ]: 4 // 2
```

```
[ ]: 5 // 2
```

```
[ ]: 4 % 2
```

```
[ ]: 5 % 2
```

1. A quoi correspond l'opérateur "//"? (répondez ci-dessous)
2. A quoi correspond l'opérateur "%"?

1.6.2 Python possède également de nombreuses fonctions mathématiques; exécuter les cellules suivantes et déterminer ce que font les différentes fonctions mathématiques proposées.

```
[ ]: from math import *           # importation d'une bibliothèque : on comprendra plus  
    ↪ tard ...
```

```
[ ]: abs(8)
```

```
[ ]: abs(-8)
```

```
[ ]: round(1/8,0)
```

```
[ ]: round(1/8,1)
```

```
[ ]: round(1/8,2)
```

```
[ ]: floor(1.25)
```

```
[ ]: floor(5)
```

```
[ ]: floor(-3.2)
```

```
[ ]: sqrt(1)
```

```
[ ]: sqrt(2)
```

```
[ ]: sqrt(3)
```

1.7 Quelques raccourcis clavier bien utiles à noter dans un coin...

On retrouve tous les raccourcis dans le menu « Aide », mais il est bon de retenir ce qui suit.

Il faut d'abord cliquer sur la gauche de la cellule (partie blanche) puis, pour

1. Copier : taper c

Couper : x

Coller : v

Supprimer : d,d (c'est-à-dire taper deux fois sur d)

2. Insérer une cellule avant : a (comme above)

Insérer une cellule après : b(comme below)

3. Mettre en Markdown, éditer,

Pour mettre la cellule en Markdown : m

Pour mettre la cellule en code : y

Pour éditer une cellule en Markdown : enter (entrée)

4. Pour exécuter une cellule : shift + enter

5. Annuler la dernière action :

z pour annuler une suppression de cellule

Ctrl + z pour annuler ce que l'on vient de taper (seulement en mode édition)

1.8 ***