

1 Python

A Jupyter Notebook

1 Présentation

Un **Notebook** (calepin en français) intègre le code et sa sortie dans un seul document qui combine visualisations, textes, équations mathématiques et autres médias riches. En d'autres termes, il s'agit d'un document où vous pouvez exécuter du code, afficher la sortie, et également ajouter des explications, des formules, des graphiques et rendre votre travail plus transparent, compréhensible, reproductible et partageable.

Un fichier notebook est reconnaissable par son extension : '.ipynb'.

Jupyter Notebook est un **environnement de calcul interactif (EDI)** qui permet de développer des notebooks pour présenter de manière interactive des projets de science des données.

2 Installation

La façon la plus simple pour un débutant de commencer avec **Jupyter Notebook** est d'installer **Anaconda**.

- 1 **Anaconda** est la distribution Python la plus utilisée pour la science des données et est fournie préchargée avec toutes les bibliothèques et tous les outils les plus populaires.

Avantage : on peut travailler en local.

Inconvénient : tout se fait sur le même ordinateur.


- 2 Se rendre sur le site <https://www.anaconda.com/download/> sélectionner la version (la plus complète) d'**anaconda** selon votre système d'exploitation :



- 3 Télécharger la version sélectionnée puis l'installer.

3 Anaconda

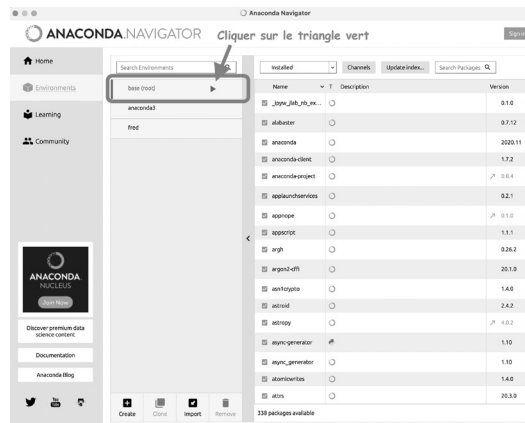
Pour accéder à **Jupyter Notebook** :

- 1 Lancer **anaconda navigator** : .
- 2 Suivre l'une des deux façons suivantes :

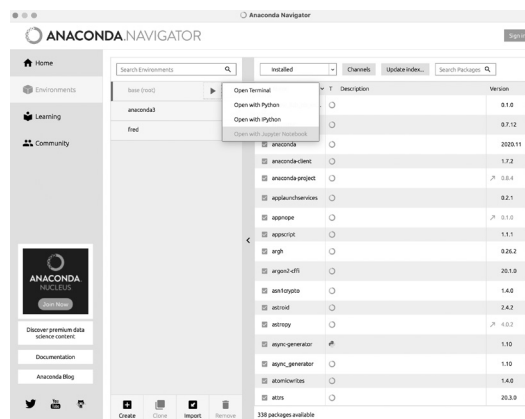
- **Première façon via « Home »**



- **Deuxième façon via « Environments »**



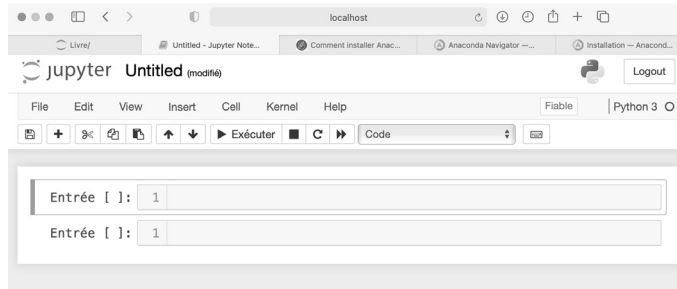
Sélectionner ensuite « Open with Jupyter Notebook » :



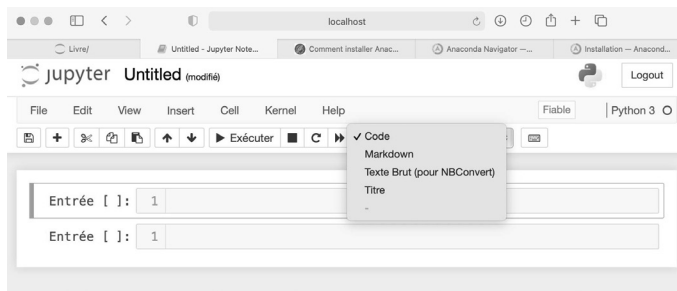
Jupyter Notebook s'ouvre alors dans votre navigateur par défaut et vous permet d'accéder à vos fichiers et données de votre répertoire « utilisateurs ».

4 Nouveau notebook

- Pour créer un nouveau notebook, cliquer sur **▷ New ▷ Python3**.
- Comme on peut le voir ci-dessous, un notebook est un empilement de cellules.

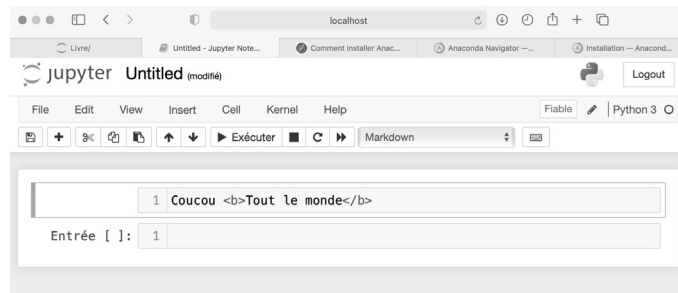


- Pour modifier le type d'une cellule, cliquer sur l'onglet **code**

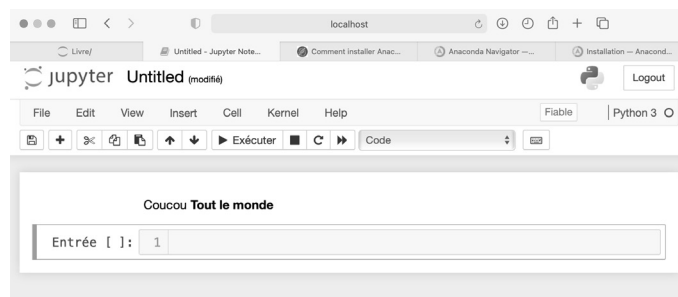


Exemples

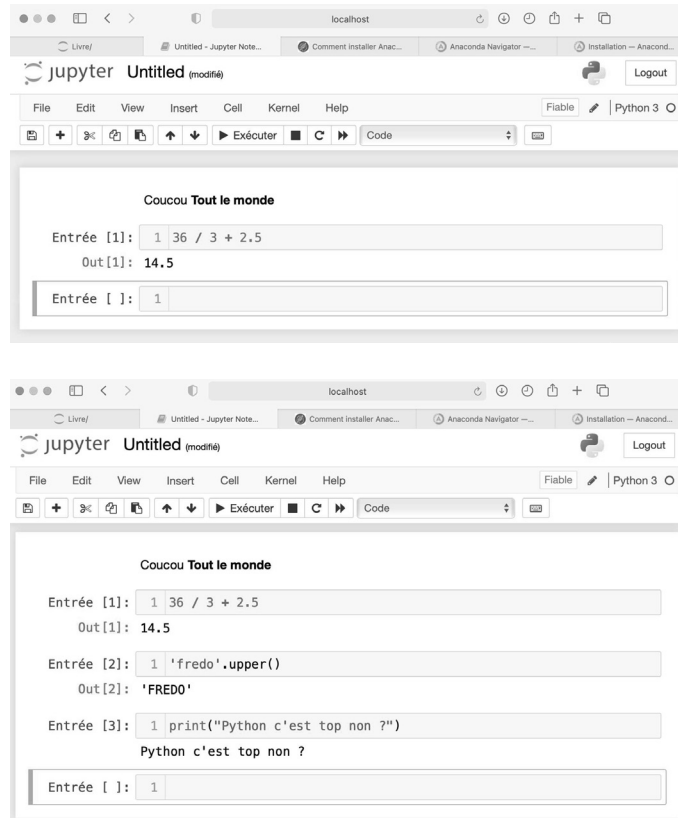
- 1 Ajout d'un texte (qui contient de l'HTML) :



Il suffit ensuite d'exécuter la cellule en appuyant sur **Shift Entrée** ou **▶ Exécuter**.



2 Ajout de codes **Python** :



5 Manipulations complémentaires

Dans un notebook

- Les calculs sont effectués dans un **kernel** (noyau) connecté à l'interpréteur Python 3.
- Toute trace de calcul est conservée en mémoire dans ce noyau. On peut :

Stopper une exécution	Noyau ▷ Interrompre (ou symbole carré noir)
Redémarrer le noyau	Noyau ▷ Redémarrer
Exécuter une ligne	SHIFT + ENTRÉE (exécute une cellule et accède à la cellule suivante) CTRL+ENTRÉE (exécute une cellule et permet de rester dans la cellule)
Enregistrer un notebook	File ▷ Download as ▷ Notebook (.ipynb)
Renommer un notebook	Fichier ▷ Renommer

Dans Jupyter Notebook

Importer un notebook	Upload puis choisir le fichier à importer.
--------------------------------	--

B Rappels de base

Python est un langage de programmation interprété, multiplateforme, inventé par GUIDO VAN ROSSUM dans les années 1990 et soutenu par la PYTHON SOFTWARE FOUNDATION. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

1 Opérations élémentaires

Opérations	+	-	×	/	=	≠	>	<	≤	≥
Python	+	-	*	/	==	!=	>	<	<=	>=

Opérations	a^n	Quotient division euclidienne de a par b	Reste division euclidienne de a par b
Python	a**n	a//b	a%b

2 Variable : type, affichage

Types : string, int, float

x=	Type de x	Commentaires
'baba'	string	x contient une chaîne de caractères <i>C'est une succession de caractères mis entre des guillemets ' ' ou " ".</i>
64	int	x contient l'entier 64
3.5	float	x contient le nombre réel 3,5 <i>On dit que : 3.5 est la représentation flottante du réel 3,5.</i>

Changer de type de variable

Commande	Transformation du type de a en
str (a)	chaîne de caractères
int (a)	nombre entier (<i>lorsque cela est possible</i>)
float (a)	nombre flottant (<i>lorsque cela est possible</i>)

Instruction d'affectation

Commande	Rôle
a = 2	La variable a prend la valeur de 2
a, b = 2, 3	Les variables a et b prennent respectivement les valeurs 2 et 3

Opérations sur les chaînes de caractères

Instruction	Rôle
str (x)	Transforme la variable x en une chaîne de caractères
s.replace('a', 'b')	Remplace l'élément 'a' par l'élément 'b' dans la chaîne de caractères s
s.count('a')	Compte le nombre de 'a' présents dans s
'baba'.upper()	'BABA'
'BABA'.lower()	'baba'
'python'+ '5G'	'python5G' <i>On a concaténé deux chaînes de caractères.</i>

Comment afficher des variables ?

Instruction	Rôle
print (a)	Affiche la variable a
print ('super')	Affiche la chaîne de caractères super

<code>print('l'entier')</code>	Affiche la chaîne de caractères l'entier
<code>print('a = ', a)</code>	Affiche a = suivi de la variable a
<code>print('a = ' + str(a))</code>	

3 Boucles

Ensemble d'entiers consécutifs

Python	Rôle
<code>range(n, m + 1)</code>	Donne l'ensemble des entiers consécutifs de n à m (inclus).
<code>range(10)</code>	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
<code>range(1, 11)</code>	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
<code>range(1, 10, 2)</code>	1, 3, 5, 7, 9

Les différentes boucles

Langage naturel	Code Python
Si condition Alors instructions Sinon autres instructions FinSi	<pre>if condition : instructions else: autres intructions</pre> Code
Pour k allant de n à m Faire instructions FinPour	<pre>for k in range(n, m + 1): instructions</pre> Code
TantQue condition vraie Faire instructions FinTantQue	<pre>while condition : instructions</pre> Code

C Données structurées

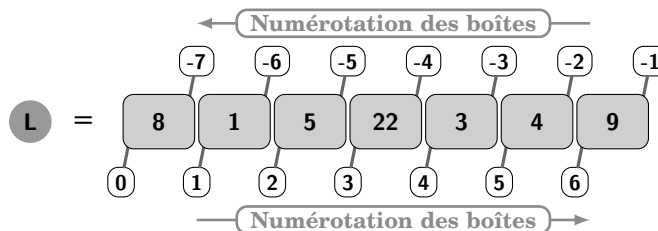
1 Listes

Définition

- Une **liste** est un ensemble d'objets séparés par des virgules et mis entre [].
C'est comme un « tiroir » où chaque élément de la liste est rangé dans une boîte numérotée.
- L'instruction qui permet de créer la liste vide nommée L est `L = []`.

Exemple

La liste `L = [8, 1, 5, 22, 3, 4, 9]` peut-être vue comme ci-dessous :



Actions possibles sur L

Objectif	Instruction	Résultat
Obtenir l'élément n° 0 de L	<code>L[0]</code>	8
Obtenir l'élément n° 1 de L	<code>L[1]</code>	1
Obtenir la liste des éléments de L numérotés de 0 à 2	<code>L[0 :3]</code>	[8,1,5]
Obtenir le dernier élément de L	<code>L[-1]</code>	9
Ajouter le nombre 6 à la liste L	<code>L.append(6)</code>	[8,1,5,22,3,4,9,6]
Concaténer à L la liste [7, 2]	<code>L + [7, 2]</code>	[8,1,5,22,3,4,9,6,7,2]
Compter le nombre de 3 dans L	<code>L.count(3)</code>	1
Donner le nombre d'éléments contenus dans L	<code>len(L)</code>	10
Remplacer la liste L par la liste de ses valeurs triées par ordre croissant	<code>L.sort()</code>	[1, 2, 3, 4, 5, 7, 8, 9, 22]

Parcourir une liste

Pour chaque élément t contenu dans L Faire « Agir » sur t FinPour	<pre>for t in L: ""agir sur "" t</pre> Code
Pour chaque indice k allant de 0 jusque <code>len(L) - 1</code> Faire « Agir » sur $L[k]$ FinPour	<pre>for k in range(len(L)): ""agir sur "" L[k]</pre> Code

Exemples pour générer des listes

Liste	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
Par compréhension	<code>[k for k in range(10)]</code>
Avec append	<pre>L=[] for k in range(10): L.append(k)</pre> Code
Liste	[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
Par compréhension	<code>[2 * k for k in range(1,11)]</code>
Avec append	<pre>L=[] for k in range(1,11): L.append(2 * k)</pre> Code

Exemples d'application

- ❶ Pour rechercher dans une liste L des éléments qui répondent à une condition on peut écrire :

```
[x for x in L if condition]
```

- ❷ La liste des mots de L qui contiennent la chaîne 'republic' s'obtient en utilisant l'instruction :

```
[x for x in L if 'republic' in x]
```

2 Dictionnaires

Le problème d'une liste c'est que l'accès à un de ses éléments nécessite de connaître son indice. Les dictionnaires permettent un accès plus souple aux éléments qu'ils contiennent.

Définition

Un **dictionnaire** D est constitué de valeurs v accessibles avec une clef k en effectuant $D[k] = v$.

Exemple

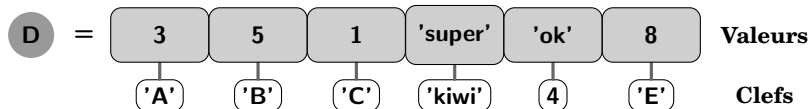
Considérons le dictionnaire D défini par :

Clef	'A'	'B'	'C'	'fred'	4	'E'
Valeur	3	5	1	'super'	'ok'	8

- L'instruction python permettant d'obtenir D est :

```
D = {'A': 3, 'B': 5, 'C': 1, 'fred': 'super', 4 : 'ok', 'E': 8}
```

- Voici une représentation de D voisine de celle d'une liste :



Actions possibles

Instruction	Résultat
Créer un dictionnaire vide D	
$D = \{\}$	
Obtenir la valeur $value$ associée à une clef key d'un dictionnaire D	
$D[key]$	$value$
Soit le dictionnaire $D = \{'A': 3, 'B': 2, 'F': '0'\}$	
$D['A']$	3
Obtenir la liste des clefs de D	
$D.keys()$	<code>dict_keys(['A', 'B', 'F'])</code>
Obtenir la liste des valeurs de D	
$D.values()$	<code>dict_values([3, 2, '0'])</code>
Obtenir le dictionnaire D d'éléments $k : 0$, avec $k \in [0, 3]$	
$\{k : 0 \text{ for } k \text{ in range}(4)\}$	$\{0:0, 1:0, 2:0, 3:0\}$

D Modules et fonctions

Qu'est-ce qu'un module ?

On appelle **module** tout fichier constitué de code **Python** (c'est-à-dire tout fichier avec l'extension **.py**) importé dans un autre fichier ou script.

Pour notre usage, les principaux modules utilisés seront **math**, **pylab**, **random**, **pandas**, **geo-pandas**, **plotly**, **ipywidgets**, **seaborn**, **networkx**, **scipy.stats**, **sympy**, **numpy**.

1 Module math

Le module **math** contient un certain nombre de commandes utiles pour faire des calculs.

En voici quelques unes :