

Tutorial d'introduction à l'analyse des données avec Python

Prof Pefura-Yone

Objectifs

- Décrire le type de langage utilisé par « Python »
- Lister les principales fonctionnalités du langage Python
- Décrire les différents types de variables dans Python
- Lister les principales bibliothèques utilisées pour les sciences des données en Python
- Expliquer l'utilisation des principaux environnements de développement intégrés en Python

Plan

1. Langage Python
2. Variables en python
3. Fonctions de base de python
4. Plateforme Anaconda (open-source Individual Edition)
5. Environnement Jupyter
6. Bibliothèques fondamentales

Conclusion

1. Langage Python

- Langage de haut niveau
- Langage orienté objet
- Généraliste avec des nombreux domaine d'utilisation: développement web, jeux, calcul numérique, administration des systèmes, télécommunication...
- Syntaxe orientée sur la lisibilité du code
- Utilisation multiforme: interface interactive (shell), scripts, bibliothèques (modules)

1. Langage Python

Programmation orientée objet (POO)

- Fondement: objets à manipuler plutôt que la logique de manipulation à écrire
- Identification des objets ou données à manipuler: mots, nombre, personnes, étages, barres de défilement...
- Conceptualisation de l'objet en une classe d'objets (table représentant toutes les tables)

1. Langage Python

Programmation orientée objet

- Classe: définit le type de données et les séquences logiques de manipulation de ces données
- Séquence logique distincte: méthode
- Communication entre objets: interfaces nommées messages
- Définition d'une classe réutilisable même par d'autre POO
- Création des données encore indéfini dans le langage lui-même

2. Variables

- Notion fondamentale de ce langage
- Nous allons le plus souvent utiliser Jupiter Notebook pour la manipulation des exemples
- Variable: objet de programmation permettant de stocker une information en mémoire de la machine en attribuant un nom symbolique


2. Variables

Déclaration d'une variable

- Règles bien précises pour définir un nom de variable
- Débute par une lettre ou par le caractère de soulignement (underscore _)
- Suivi par un nombre quelconque de lettres, chiffres ou de caractères de soulignement
- underscore est le seul caractère spécial autorisé
- Par exemple, le tiret du 6 n'est jamais autorisé

2. Variables

Déclaration d'une variable

jupyter Untitled14 Dernière Sauvegarde : il y a 8 minutes (modifié)  Logout

File Edit View Insert Cell Kernel Widgets Help

Fiabilité Python 3

Exécuter

```
Entrée [6]: a=4
            b=7
            c=a+b
            c
```

Out[6]: 11

```
Entrée [11]: Nom='Pefura'
            nom='Yone'
            print(Nom); print(nom)
```

Pefura
Yone

```
Entrée [15]: 1nom='Eric' # ne pas faire ceci
            print(1nom)
```

File "<ipython-input-15-40cda9f1c96a>", line 1

```
    1nom='Eric' # ne pas faire ceci
      ^
```

SyntaxError: invalid syntax

2. Variables

Déclaration d'une variable

Une base de données est un objet!

Adresse de la base dans l'ordinateur

The screenshot shows a Jupyter Notebook window titled 'Untitled14'. The top bar includes the Jupyter logo, the title, and a save status: 'Dernière Sauvegarde : il y a 21 minutes (modifié)'. On the right, there is a Python logo and a 'Logout' button. Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. A toolbar below the menu contains icons for saving, adding cells, undo, redo, and running code. The main area shows two code cells. The first cell, labeled 'Entrée [16]:', contains the code `import pandas as pd`. The second cell, labeled 'Entrée [21]:', contains the code `data=pd.read_csv('C:/Users/DDD/Documents/Cameroon_lung_function/Cameroon lung function.csv')` followed by `data` on a new line. Below the second code cell, the output is displayed as 'Out[21]:' followed by a table of data. The table has 10 columns: 'ID', 'sex', 'age', 'ethnicity', 'height', 'fvc', 'fev', 'fef2575', and 'fevfvc'. It shows 1777 rows of data. The first five rows are highlighted in grey. The last row is also highlighted. Below the table, it says '1777 rows x 9 columns'. A blue arrow points from the text 'Adresse de la base dans l'ordinateur' to the file path in the code cell.

```
Entrée [16]: import pandas as pd
```

```
Entrée [21]: data=pd.read_csv('C:/Users/DDD/Documents/Cameroon_lung_function/Cameroon lung function.csv')
data
```

Out[21]:

	ID	sex	age	ethnicity	height	fvc	fev	fef2575	fevfvc
1	1000948	1	4.950000	1	109	1.230	1.12	1.39	0.91
2	500921	1	5.000000	1	119	1.100	1.10	1.81	1.00
3	1000157	1	5.400000	1	107	0.970	0.91	1.70	0.94
4	1000805	1	5.290000	1	118	0.939	0.94	1.64	1.00
5	1000174	1	5.530000	1	119	1.390	1.34	2.03	0.96
...
1773	1752	2	80.260096	1	163	1.920	1.54	3.46	0.85
1774	1525	2	80.996578	1	160	1.400	1.01	2.41	0.72
1775	1995	2	81.160849	1	160	1.700	1.26	2.96	0.74
1776	1722	2	85.232033	1	170	2.310	2.04	4.35	0.88
1777	1469	2	88.131417	1	160	2.270	1.65	3.92	0.73

1777 rows x 9 columns


2. Variables

Déclaration d'une variable

- Type de variable = sa nature
- variables
 - ✓ nombre entier (int)
 - ✓ nombre réel (float)
 - ✓ chaînes de caractères (str)
- A chacun de ces types de variables est associé un certain nombre d'attributs ou de méthodes


2. Variables

Attributs/Méthodes


jupyter Untitled14 Dernière Sauvegarde : il y a 39 minutes (modifié)  Logout

File Edit View Insert Cell Kernel Widgets Help Fiable Python 3


1777 rows x 9 columns

Entrée [23]:  `# Nombres entiers ou réels`
`x=1`
`y=2`
`print(x+y)`

3

Entrée [24]:  `# Chaînes de caractères`
`m='1'`
`n='2'`
`print(m+n)`

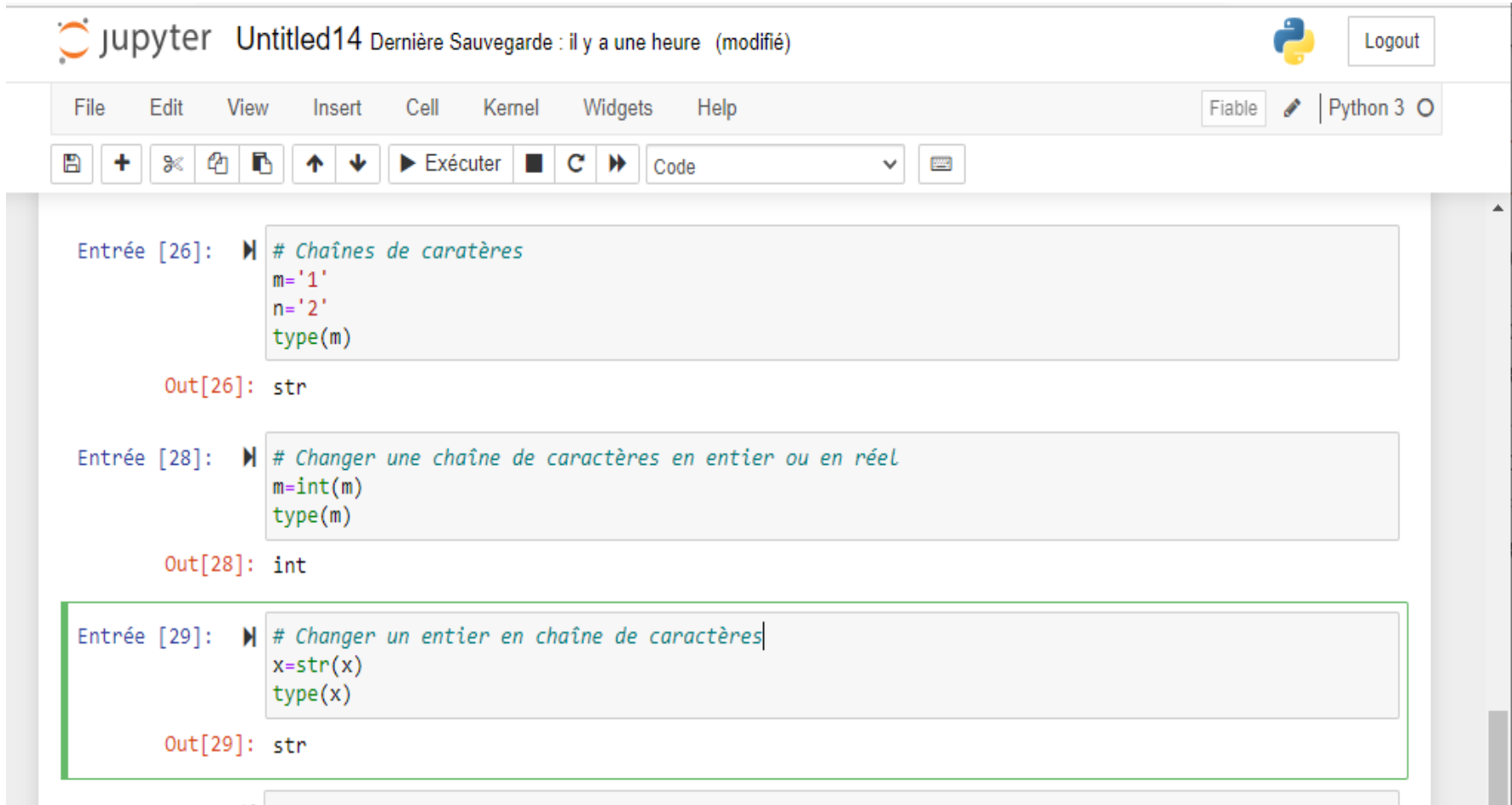
12

Entrée []: 

2. Variables

Convertir les types de variables

- Possibilité de changer/modifier le type de variable



The screenshot shows a Jupyter Notebook interface with the following elements:

- Header:** "jupyter" logo, "Untitled14", "Dernière Sauvegarde : il y a une heure (modifié)", Python logo, and "Logout" button.
- Menu Bar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help.
- Toolbar:** Includes icons for saving, adding cells, undo, redo, and a dropdown menu currently set to "Code".
- Code Cells:**
 - Entrée [26]:**

```
# Chaînes de caractères  
m='1'  
n='2'  
type(m)
```


Out[26]: str
 - Entrée [28]:**

```
# Changer une chaîne de caractères en entier ou en réel  
m=int(m)  
type(m)
```


Out[28]: int
 - Entrée [29]:**

```
# Changer un entier en chaîne de caractères  
x=str(x)  
type(x)
```


Out[29]: str

2. Variables

Attributs/Méthodes

jupyter Untitled14 Dernière Sauvegarde : il y a une heure (modifié)



Logout

File Edit View Insert Cell Kernel Widgets Help

Fiable

Python 3

Exécuter

Out[29]: str

```
Entrée [ ]: # A chaque variable créée dans python, est associé un ensemble d'attributs (ex : type) mais aussi
# un ensemble de méthodes, c'est à dire un ensemble d'opérations de traitement et
# d'exploitation réalisable avec cette variable.

# Pour afficher l'ensemble des méthodes d'une variable, on utilise la commande dir()
```

```
Entrée [31]: print(dir(x))

['_add_', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__',
'__ge__', '__getattr__', '__getitem__', '__getnewargs__', '__gt__', '__hash__', '__init__',
'__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__',
'__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__',
'__sizeof__', '__str__', '__subclasshook__', 'capitalize', 'casefold', 'center', 'count', 'encode',
'endswith', 'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isascii',
'isdecimal', 'isdigit', 'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace', 'istitle',
'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'replace', 'rfind',
'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip',
'swapcase', 'title', 'translate', 'upper', 'zfill']
```

```
Entrée [30]: # Pour afficher l'aide sur toutes les fonctions associées à la variable x, on fait simplement :
print(help(x))
```

No Python documentation found for '1'.
Use help() to get the interactive help utility.
Use help(str) for help on the str class.

Activer Windows
Accédez aux paramètres

3. Fonctions

Opérateurs arithmétiques et logiques

Opérateurs arithmétiques

	Symbole	Exemple
Addition	+	$x=2+3$
Soustraction	-	$z=x-y$
multiplication	*	$y=3*x$
Division (quotient réel)	/	$z=5/2$ # renvoie 2.5
Division (quotient entier)	//	$z=5//2$ # renvoie 2
Puissance	**	$x**2$ # x puissance 2
Reste de la division (modulo)	%	$17\%3$ # renvoie 2
Addition incrémentée	+=	$x+=4$ # ajoute 4 à la valeur initiale de x

Online at <https://mpra.ub.uni-muenchen.de/76653/>
MPRA Paper No. 76653, posted 07 Feb 2017 14:56 UTC

3. Fonctions

Opérateurs arithmétiques et logiques

Soustraction incrémentée

--+

valeur initiale de x
(équivalent à $x=x-4$)

Opérateurs logiques (variables booléennes et valeurs tests)

Egal

==

$x==2$ # Egalité logique ou
mathématique (différent
de l'égalité
d'assignation =)

Strictement inférieur

<

$x<2$

Inférieur ou égal

<=

$x<=2$

Strictement supérieur

>

$x>2$

Supérieur ou égal

>=

$x>=2$

Différent

!=

$x!=2$

Et

and
(équivalent
à &)

$x>2$ and $x<10$ # expression
logique composée

Ou

Or
(équivalent
à |)

$x==2$ or $x==2$ # Expression
logique composée

Opérateur de négation

not
(équivalent
à ~)

x not True

Online at <https://mpra.ub.uni-muenchen.de/76653/>
MPRA Paper No. 76653, posted 07 Feb 2017 14:56 UTC

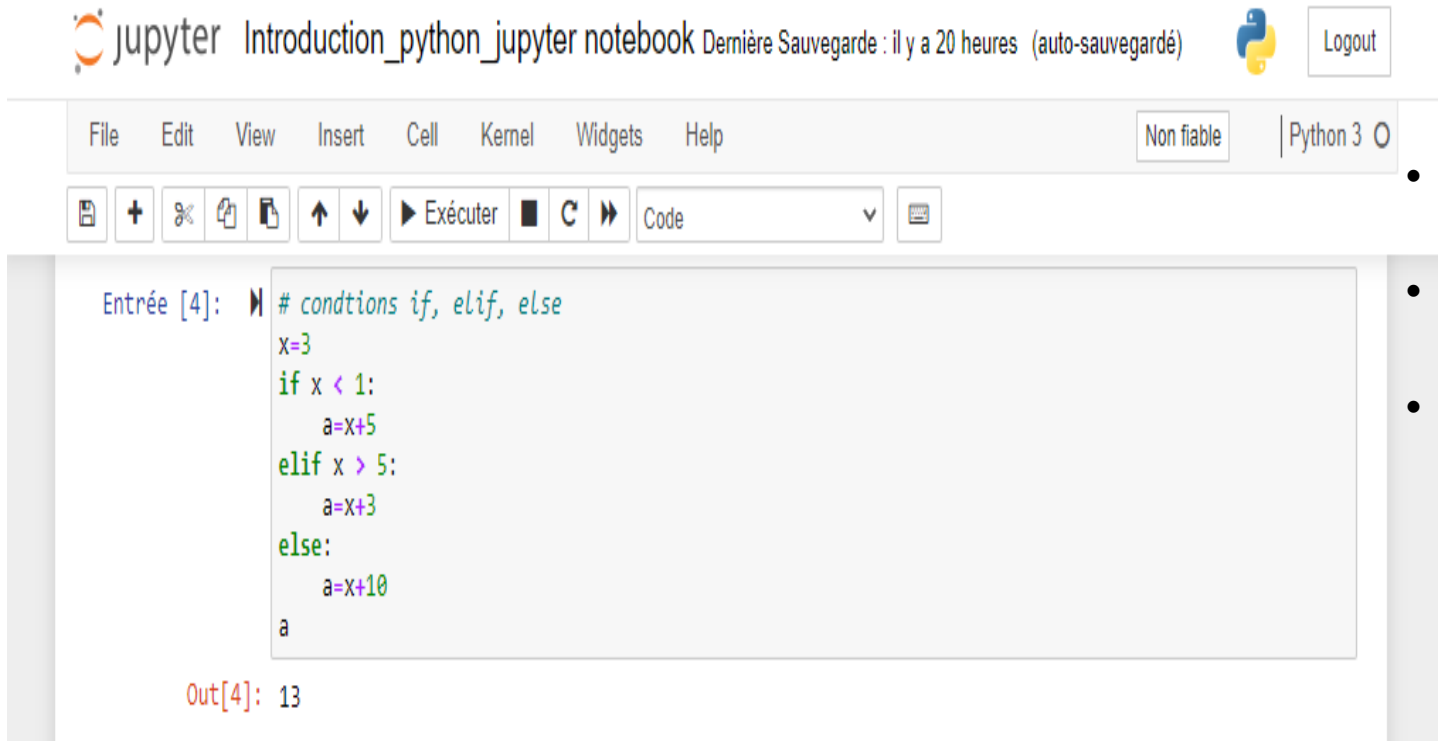
3. Fonctions

Instructions conditionnelles

- Conditions: if, elif, else
- Boucles: for, while

3. Fonctions

Instructions conditionnelles



The image shows a Jupyter Notebook interface. The top bar includes the Jupyter logo, the text "Introduction_python_jupyter notebook", and a save status "Dernière Sauvegarde : il y a 20 heures (auto-sauvegardé)". There is a "Logout" button and a Python 3 logo. The menu bar contains "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". Below the menu bar is a toolbar with icons for saving, adding cells, deleting cells, running cells, and other functions. The main area shows a code cell with the following Python code:

```
Entrée [4]: > # condtions if, elif, else
x=3
if x < 1:
    a=x+5
elif x > 5:
    a=x+3
else:
    a=x+10
a
```

Below the code cell, the output is displayed: "Out[4]: 13".

- Initialisation de la variable
- Ecriture des conditions
- Sortie des résultats

3. Fonctions

Instructions conditionnelles

Boucle: « while » et « for...in »

- Instructions en boucle « while... » exécutées tant que la (les) condition(s) définies par l'expression logiques sont vraies
- Instructions en boucle « for...in » sont exécutées pour chaque élément pris dans un ensemble d'éléments préalablement indiquées (généralement dans un objet range ou list)

3. Fonctions

Instructions conditionnelles

jupyter Introduction_python_jupyter notebook Dernière Sauvegarde : il y a 21 heures (auto-sauvegardé) Python 3

File Edit View Insert Cell Kernel Widgets Help Non fiable

Code

```
a=x+10
a
```

Out[4]: 13

Entrée [16]: `x = 1 # Initialisation de la variable x`
`while (x < 10):`
 `print('La valeur de x est ', x+1)`
 `x = x + 1 # Incrémentement de la variable x`

La valeur de x est 2
La valeur de x est 3
La valeur de x est 4
La valeur de x est 5
La valeur de x est 6
La valeur de x est 7
La valeur de x est 8
La valeur de x est 9
La valeur de x est 10

Entrée []:

3. Fonctions

Instructions conditionnelles

Boucle « for »

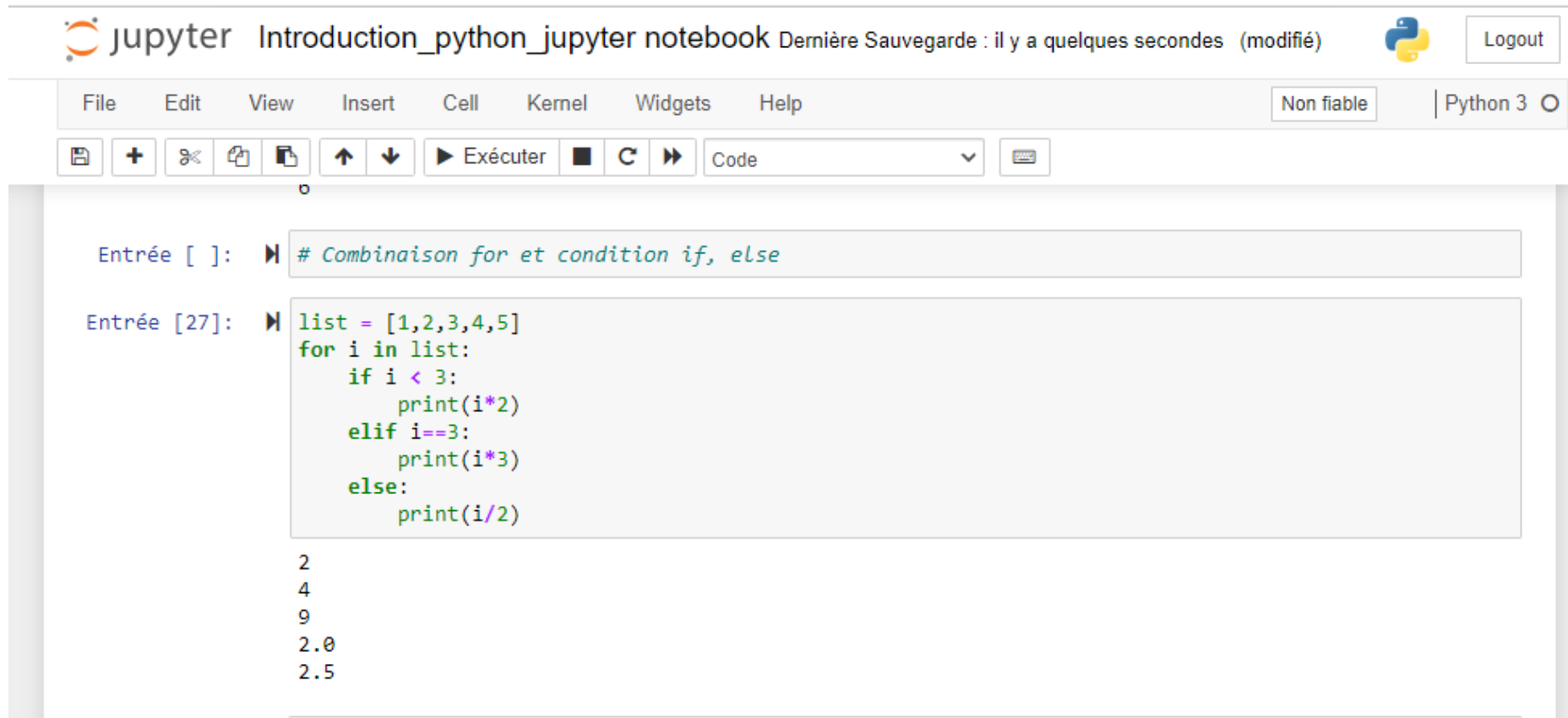
```
Entrée [24]: ► # boucle for  
list=[1, 2, 3, 4, 5]  
  
for i in list:  
    print(i+1)
```

```
2  
3  
4  
5  
6
```

3. Fonctions

Instructions conditionnelles

Combinaison d'instructions



The screenshot shows a Jupyter Notebook interface. The top bar includes the Jupyter logo, the text "Introduction_python_jupyter notebook", a save status "Dernière Sauvegarde : il y a quelques secondes (modifié)", a Python logo, and a "Logout" button. Below this is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". To the right of the menu bar are buttons for "Non fiable" and "Python 3". A toolbar below the menu bar contains icons for saving, adding cells, undo, redo, and running code. The main area shows two code cells. The first cell, labeled "Entrée []:", contains a comment: `# Combinaison for et condition if, else`. The second cell, labeled "Entrée [27]:", contains a loop with conditional logic: `list = [1,2,3,4,5]`, `for i in list:`, `if i < 3:` followed by `print(i*2)`, `elif i==3:` followed by `print(i*3)`, and `else:` followed by `print(i/2)`. The output of the second cell is displayed below the code: `2`, `4`, `9`, `2.0`, and `2.5`.

```
Entrée [ ]:  # Combinaison for et condition if, else

Entrée [27]: list = [1,2,3,4,5]
              for i in list:
                if i < 3:
                  print(i*2)
                elif i==3:
                  print(i*3)
                else:
                  print(i/2)

2
4
9
2.0
2.5
```

3. Fonctions

Fonctions natives

- Plusieurs fonctions de base sont disponibles en Python
- Liste et utilisation disponible à l'adresse:

<https://docs.python.org/fr/3.6/library/functions.html>

3. Fonctions

Vos propres fonctions

Générique des fonctions dans Python

```
def nom_fonction (liste des paramètres):  
    bloc d'instructions
```


3. Fonctions

Vos propres fonctions

jupyter Introduction_python_jupyter notebook Dernière Sauvegarde : il y a 35 minutes (auto-sauvegardé) Python 3

File Edit View Insert Cell Kernel Widgets Help Non fiable

Entrée [33]:

```
# Définir nos propres fonctions
# Par exemple définir une fonction pour calculer l'indice de masse corporelle des individus
def indice_masse_corporelle (poids_kg, taille_m):
    IMC=poids_kg/(taille_m*taille_m)
    print(IMC)
```

Entrée [34]:

```
indice_masse_corporelle (70, 1.72)
```


23.661438615467823

Entrée [39]:

```
def indice_masse_corporelle (poids_kg, taille_m):
    IMC=poids_kg/(taille_m*taille_m)
    return(IMC)
```

Entrée [40]:

```
indice_masse_corporelle (70, 1.72)
```


Out[40]: 23.661438615467823

4. Anaconda



Individual Edition

Your data science toolkit

With over 25 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.



4. Anaconda

Interface Anaconda: launch pour lancer un programme
et install pour installer un programme

Anaconda Navigator

File Help

ANACONDA.NAVIGATOR

Sign in

Home

Environments

Learning

Community

Discover premium data science content

Documentation

Anaconda Blog

Twitter YouTube GitHub

Applications on base (root)

Channels

Refresh

 CMD.exe Prompt 0.1.1 Run a cmd.exe terminal with your current environment from Navigator activated Launch	 Datalore Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team. Launch	 IBM Watson Studio Cloud IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling. Launch	 JupyterLab 3.0.14 An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture. Launch
 Jupyter Notebook 6.3.0 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing their Launch	 Orange 3 3.26.0 Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows Launch	 Powershell Prompt 0.0.1 Run a Powershell terminal with your current environment from Navigator activated Launch	 PyCharm Community 2021.2 An IDE by JetBrains for pure Python development. Supports code completion, listing, and debugging. Launch

Activer Windows

Accédez aux paramètres pour activer Windows.

5. Environnement Jupyter

- Environnement de développement intégré pour Python, R et Julia
- Jupyter fonctionne comme un navigateur web
- Pas besoin de connexion internet pour l'utiliser quand il est installé sur l'ordinateur local
- Mais peut-être utilisé en ligne avec l'interface google colaboratory ou colab

5. Environnement Jupyter

Deux interfaces

- Jupyter notebook
- Jupyter Lab

5. Environnement Jupyter

jupyter notebook au lancement

Cliquez ici pour lancer une page d'édition d'un nouveau projet

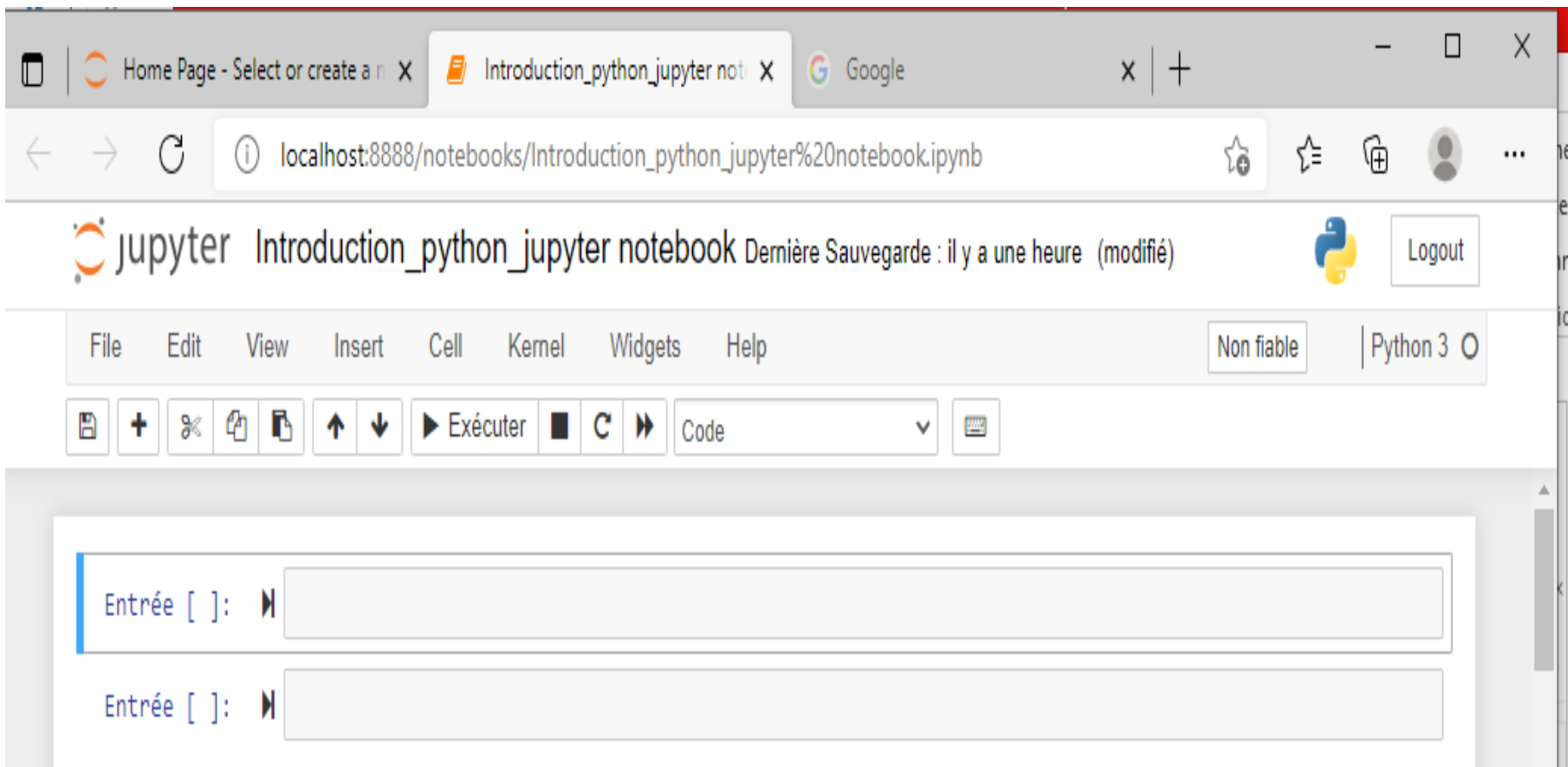
Vos dossiers présents dans votre ordinateur

Name	Last Modified	File size
3D Objects	il y a 6 mois	
anaconda3	il y a 13 jours	
catboost_info	il y a 2 mois	
Contacts	il y a 6 mois	
covid-19	il y a un mois	
dask-worker-space	il y a 13 jours	
Desktop	il y a 8 jours	
Documents	il y a 4 jours	
Downloads	il y a un jour	
Favorites	il y a 6 mois	

IFPERA
Education, Recherche,
Statistiques

5. Environnement Jupyter

Ouverture d'une page pour un nouveau projet dans « notebook »



5. Environnement Jupyter

Premiers pas avec notebook

Home Page - Select or c x Introduction_python_jup x Google x JupyterLab x + - □ ×

← → ↻ ⓘ localhost:8888/notebooks/Introduction_python_jupyter%20notebook.ipynb ☆ ☆ ⌵ 👤 ...

jupyter Introduction_python_jupyter not... Dernière Sauvegarde : il y a quelques secondes (auto-sauvegardé) Python 3 Logout

File Edit View Insert Cell Kernel Widgets Help Non fiable Python 3

📁 + ✂ 📄 📄 ⬆ ⬇ ▶ Exécuter ■ ↺ ▶ Code ▾ 🖨

```
Entrée [ ]: ▶ # Cliquez sur les différents boutons pour savoir et utiliser leur contenu
```

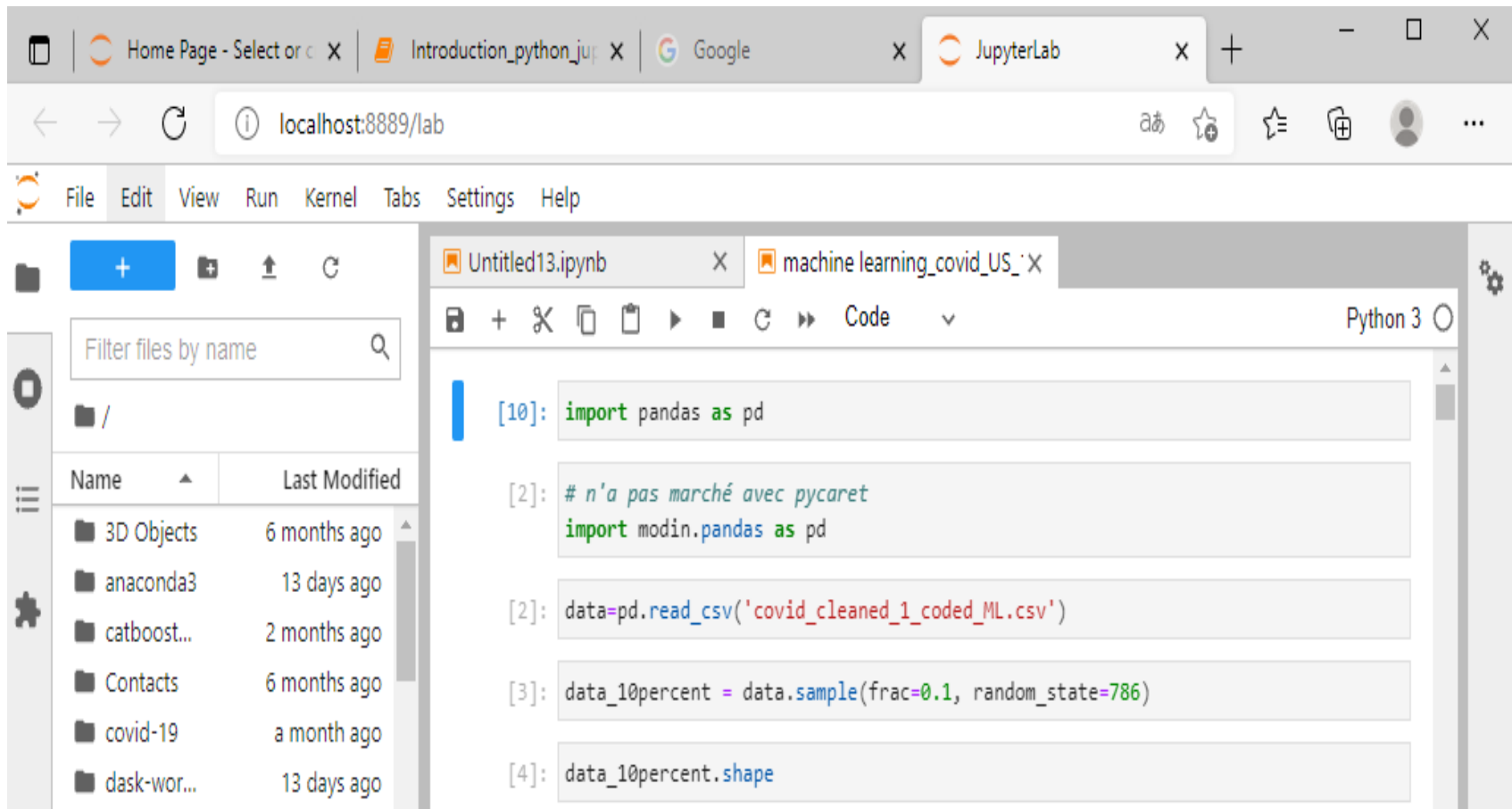
```
Entrée [ ]: ▶ # cellule d'exécution des codes (syntaxes) et de prise de notes.  
# indique les notes et non du code
```

```
Entrée [41]: ▶ 2+4 # voici un exemple d'exécution d'une ligne de code
```

```
Out[41]: 6
```


5. Environnement Jupyter

Premiers pas avec Lab



The screenshot displays the JupyterLab web interface in a browser window. The address bar shows `localhost:8889/lab`. The interface is divided into several panels:

- File Browser (Left):** Contains a search bar "Filter files by name" and a list of files and folders. The list has columns for "Name" and "Last Modified".
- Code Editor (Right):** Shows a notebook with four code cells. The first cell imports `pandas` as `pd`. The second cell contains a comment in French and imports `modin.pandas` as `pd`. The third cell reads a CSV file. The fourth cell samples the data and prints its shape.

Name	Last Modified
3D Objects	6 months ago
anaconda3	13 days ago
catboost...	2 months ago
Contacts	6 months ago
covid-19	a month ago
dask-wor...	13 days ago

```
[10]: import pandas as pd

[2]: # n'a pas marché avec pycaret
import modin.pandas as pd

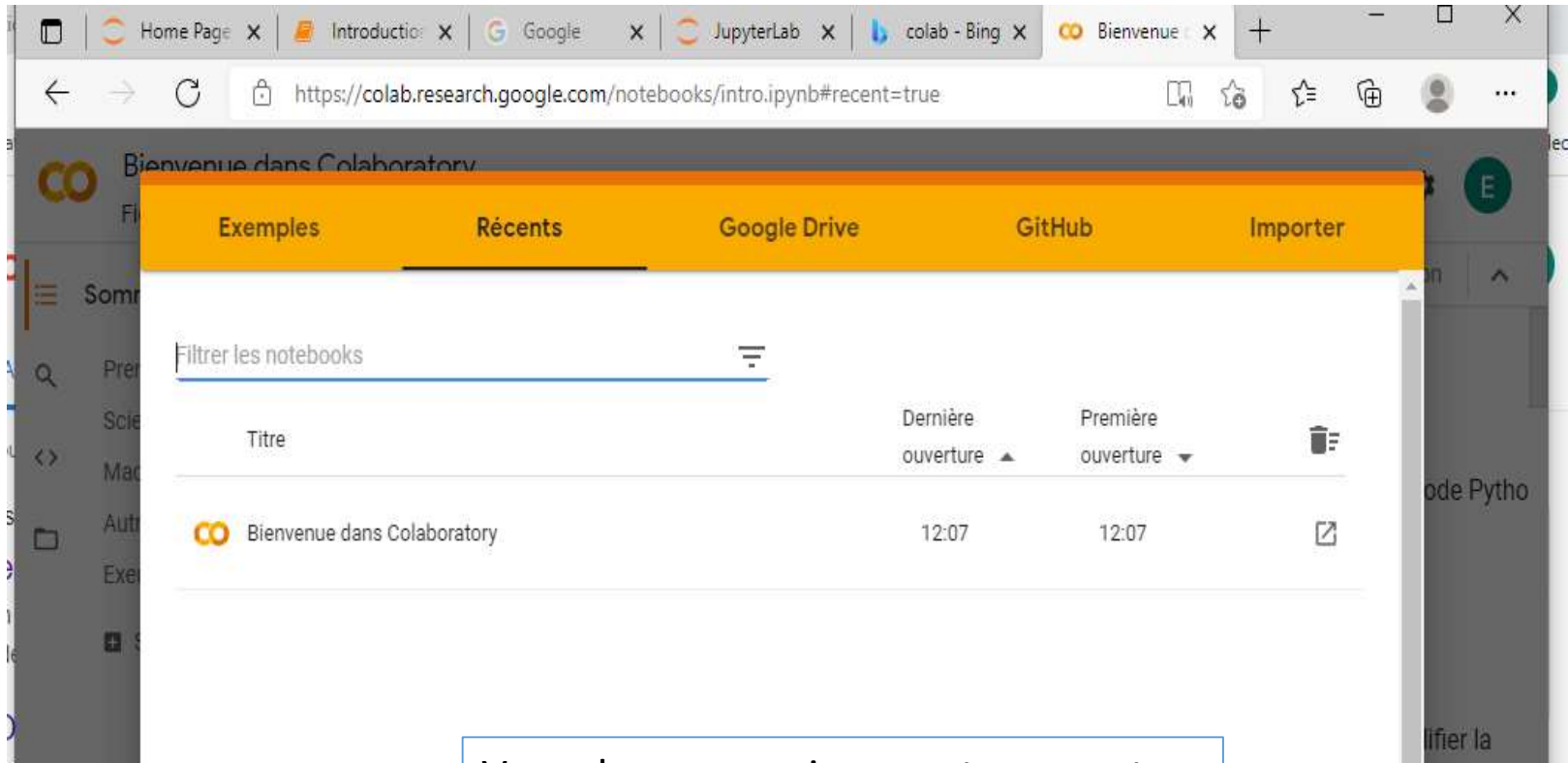
[2]: data=pd.read_csv('covid_cleaned_1_coded_ML.csv')

[3]: data_10percent = data.sample(frac=0.1, random_state=786)



[4]: data_10percent.shape
```

5. Environnement Jupyter

Premiers pas avec colab



The screenshot shows the Google Colaboratory web interface. The browser's address bar displays the URL <https://colab.research.google.com/notebooks/intro.ipynb#recent=true>. The interface features a navigation bar with tabs: 'Exemples', 'Récents' (selected), 'Google Drive', 'GitHub', and 'Importer'. Below the navigation bar, there is a search bar labeled 'Filtrer les notebooks'. A table lists the recent notebooks:

Titre	Dernière ouverture ▲	Première ouverture ▼	
 Bienvenue dans Colaboratory	12:07	12:07	

Vous devez renseigner votre compte mail

5. Environnement Jupyter

Premiers pas avec colab

Cliquez ici pour vous connectez à un environnement de travail

Bienvenue dans Colaboratory

Fichier Modifier Affichage Insérer Exécution Outils Aide

Sommaire

- Premiers pas
- Science des données
- Machine learning
- Autres ressources
- Exemples de machine learning
- + Section

+ Code + Texte Copier sur Drive

Qu'est-ce que Colaboratory ?

Colaboratory, souvent raccourci en "Colab", vous permet d'écrire et d'exécuter du code Python dans votre navigateur. Il offre les avantages suivants :

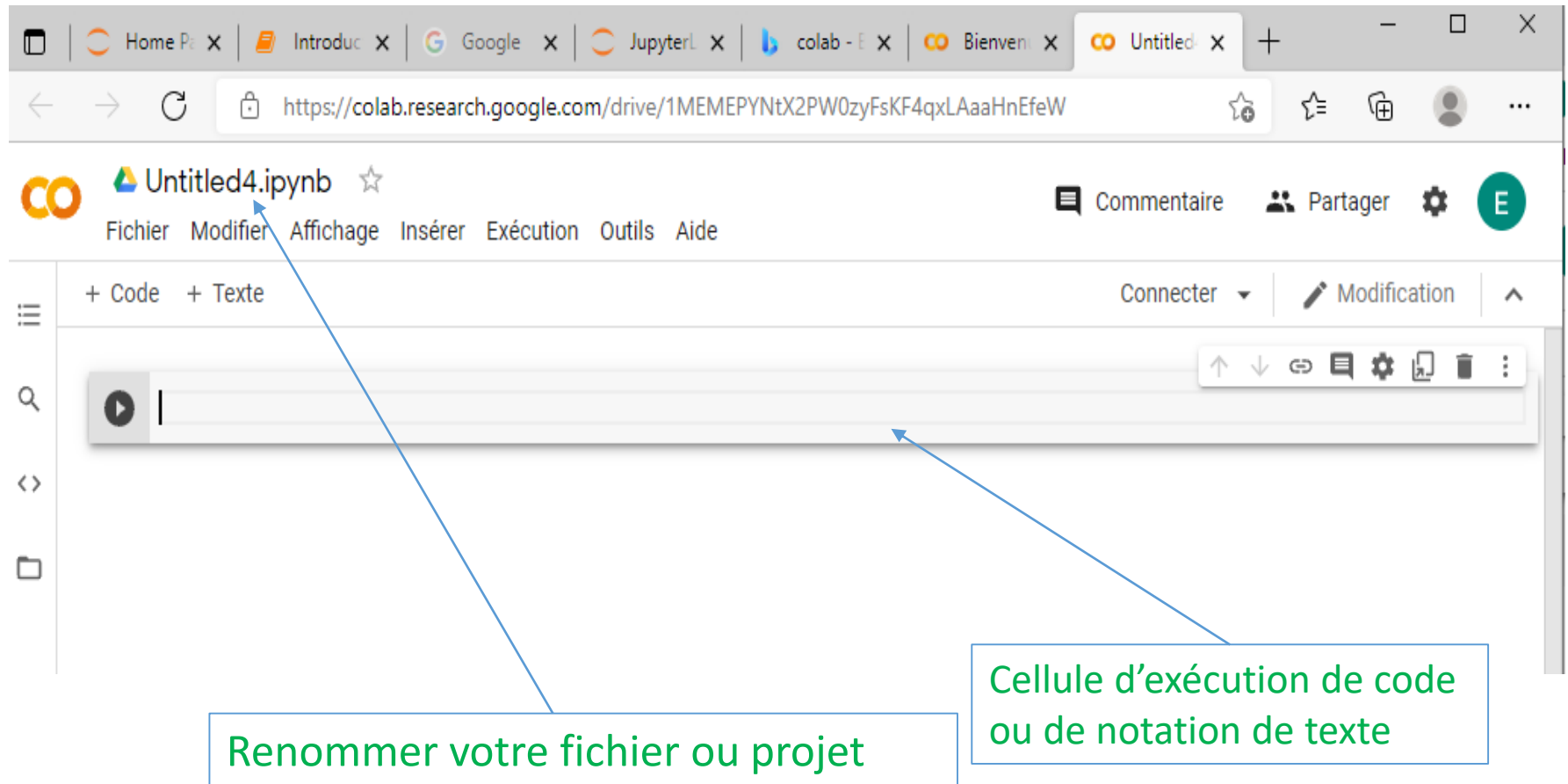
- Aucune configuration requise
- Accès gratuit aux GPU
- Partage facile

Que vous soyez **étudiant**, **data scientist** ou **chercheur en IA**, Colab peut vous simplifier la tâche. Regardez la [présentation de Colab](#) pour en savoir plus ou commencez tout de suite.

Cliquez ici pour ouvrir une page jupyter notebook dans colab

5. Environnement Jupyter

Premiers pas avec colab



The screenshot displays the Google Colab web interface. The browser's address bar shows the URL <https://colab.research.google.com/drive/1MEMEPYNTX2PW0zyFsKF4qxLAaaHnEfeW>. The main header area includes the Colab logo, the file name "Untitled4.ipynb", and a star icon. Below this is a menu bar with options: "Fichier", "Modifier", "Affichage", "Insérer", "Exécution", "Outils", and "Aide". To the right of the menu are buttons for "Commentaire", "Partager", and a settings gear icon, followed by a user profile icon labeled "E". The left sidebar contains icons for file management, search, and code execution. The main workspace area has a toolbar with "+ Code" and "+ Texte" buttons. Below the toolbar is a code cell with a play button icon and a text input field. Two blue arrows point from text boxes to the interface: one points to the "Untitled4.ipynb" text, and the other points to the code cell's toolbar.

Renommer votre fichier ou projet

Cellule d'exécution de code ou de notation de texte

5. Environnement Jupyter

The screenshot shows a Google Colab notebook titled 'Untitled5.ipynb'. The interface includes a top toolbar with icons for file operations, a left sidebar with a file explorer, and a main code area. The code area contains three cells:

- Cell [1]: `import pandas as pd`
- Cell [5]: `from google.colab import files`
`uploaded = files.upload()`
`for fn in uploaded.keys():`
 `print('User uploaded file "{name}" with length {length} bytes'.format(name=fn, length=len(uploaded[fn])))`
- Cell [6]: `data= pd.read_csv('spirometry_data.csv')`
`data.head(3)`

Below the code cells, a file upload dialog is visible, showing a file named 'spirometry_data.csv' (38537 bytes) being uploaded. The file is selected from the local file system.

Annotations in green text boxes with blue arrows pointing to the code:

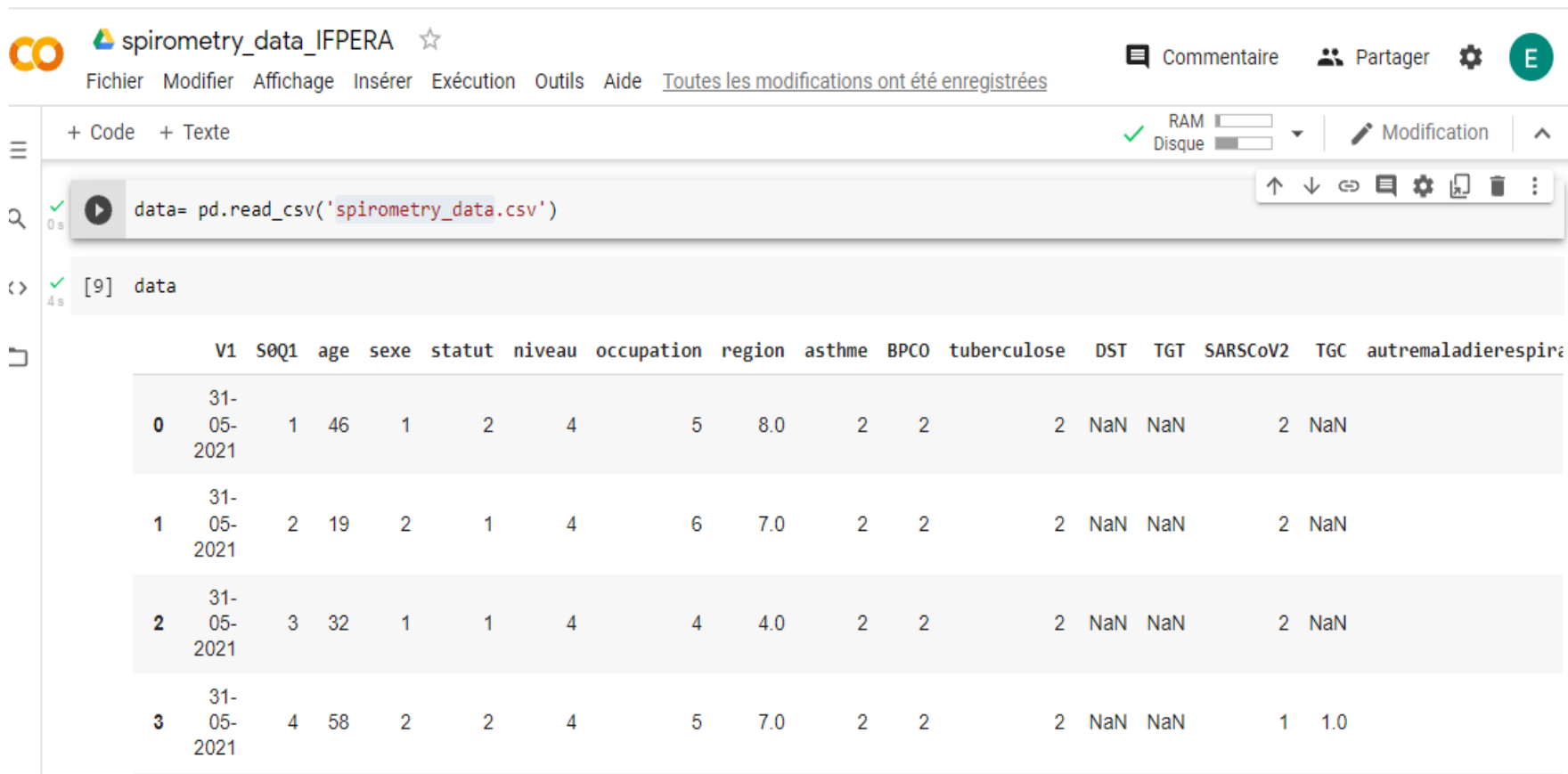
- Importation de la librairie pandas** (points to `import pandas as pd`)
- Copier et coller ce code pour ouvrir une base de données csv avec pandas** (points to the `from google.colab import files` and `files.upload()` code block)
- Sélectionner le fichier csv dans votre ordinateur** (points to the file upload dialog)
- code d'importation de la base des données** (points to `data= pd.read_csv('spirometry_data.csv')`)

The bottom of the notebook shows the first few rows of the data frame:

V1	S0Q1	age	sexe	statut	niveau	occupation	region	asthme	BPCO	tuberculose	DST	TGT	SARSCoV2	TGC	autremaladierespirato
...

5. Environnement Jupyter

Premiers pas avec colab



The screenshot shows a Google Colab notebook interface. At the top, the notebook is titled "spirometry_data_IFPERA". The menu bar includes "Fichier", "Modifier", "Affichage", "Insérer", "Exécution", "Outils", "Aide", and a status message "Toutes les modifications ont été enregistrées". On the right, there are icons for "Commentaire", "Partager", and a user profile "E". Below the menu, there are tabs for "+ Code" and "+ Texte". A RAM and Disque usage indicator is visible. The code cell contains the command `data= pd.read_csv('spirometry_data.csv')`. The output cell shows the variable `[9] data` and a preview of the CSV data.

	V1	S0Q1	age	sexe	statut	niveau	occupation	region	asthme	BPCO	tuberculose	DST	TGT	SARSCoV2	TGC	autresmaladiesrespira
0	31-05-2021	1	46	1	2	4	5	8.0	2	2	2	NaN	NaN	2	NaN	
1	31-05-2021	2	19	2	1	4	6	7.0	2	2	2	NaN	NaN	2	NaN	
2	31-05-2021	3	32	1	1	4	4	4.0	2	2	2	NaN	NaN	2	NaN	
3	31-05-2021	4	58	2	2	4	5	7.0	2	2	2	NaN	NaN	1	1.0	

5. Environnement Jupyter

Premiers pas avec colab

Le colab notebook utilisé dans ce tutorial est disponible à l'adresse:

<https://colab.research.google.com/drive/15DMGRLErG4XpKc1QJF23X-RKnUTkpiAy?usp=sharing>

5. Environnement Jupyter

Premiers pas avec Kaggle kernel

Le Kaggle notebook est disponible à l'adresse:

<https://www.kaggle.com/>

6. Bibliothèques fondamentales

- **Pandas**: manipulation des tableaux des données
- **Mito**: visualisation de la base des données et manipulation directe dans le tableau des données sur le mode excel
- **Numpy**: calcul scientifique et manipulation des tableaux ndarray (tableau à n dimension)
- **Matplotlib**: graphes et visualisation des données
- **Seaborn**: graphes et visualisation des données statistiques
- **Scipy**: mathématiques
- **Statsmodels**: analyse statistique avancée
- **Scikit-learn**: Apprentissage automatique (machine learning)

6. Bibliothèques fondamentales

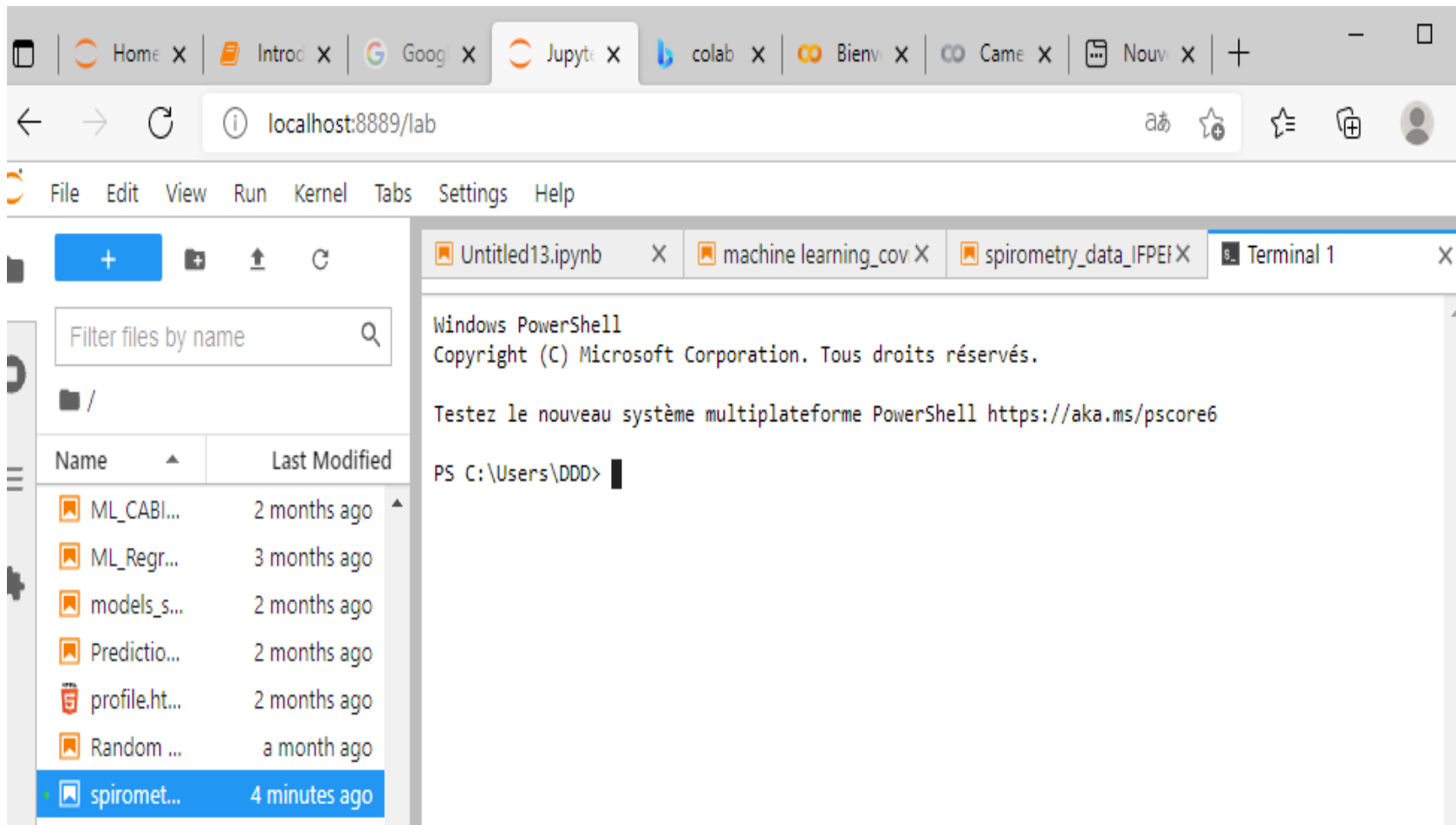
...JE ne peux pas terminer sans vous montrer comment on installe mito...NOUS VERONS pour son utilisation plustard. C'est trop chic quand on est familier avec excel.

Voici le lien pour mito: <https://docs.trymito.io/>

- Pour installer mito: ouvrir un nouveau terminal dans jupyter Lab en cliquant sur « File », puis « New », puis terminal
- Vous obtenez ceci :

6. Bibliothèques fondamentales

mito



The screenshot displays a JupyterLab web interface in a browser window. The browser's address bar shows 'localhost:8889/lab'. The JupyterLab interface includes a top menu bar with 'File', 'Edit', 'View', 'Run', 'Kernel', 'Tabs', 'Settings', and 'Help'. Below the menu is a toolbar with icons for creating new files, opening recent files, and refreshing. The left sidebar contains a file browser with a search bar labeled 'Filter files by name'. The file list shows the following items:

Name	Last Modified
/	
ML_CABl...	2 months ago
ML_Regr...	3 months ago
models_s...	2 months ago
Predictio...	2 months ago
profile.ht...	2 months ago
Random ...	a month ago
spiromet...	4 minutes ago

The right sidebar shows a list of open tabs: 'Untitled13.ipynb', 'machine learning_cov...', 'spirometry_data_IFPER...', and 'Terminal 1'. The 'Terminal 1' tab is active, displaying a Windows PowerShell prompt. The terminal output includes the following text:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\Users\DDD>
```

6. Bibliothèques fondamentales

Mito

- **Exécuter le code suivant dans le terminal** : `python -m pip install mitoinstaller`
- **Puis exécuter le code suivant**: `python -m mitoinstaller install`

Ces deux codes vous installent mito dans votre environnement Jupiter Lab. Exécuter les codes dans l'ordre en « tapant » juste la touche « Entrée »

6. Bibliothèques fondamentales

Mito

Table de données de type excel

The screenshot displays the Mito application interface. On the left is a file explorer with a search bar and a list of files. The main area is divided into a code editor and a data table. The code editor contains two code cells with instructions in French and Python code. The data table, titled 'V1', shows a dataset with columns for date, age, sex, status, level, and occupation. A blue arrow points from the text 'Table de données de type excel' to the data table.

File Explorer (Left):

- Filter files by name
- Name
- Last Modified
- ML_CABl... 2 months ago
- ML_Regr... 3 months ago
- models_s... 2 months ago
- Predictio... 2 months ago
- profile.ht... 2 months ago
- Random ... a month ago
- spiromet... seconds ago
- TB_cohor... 2 months ago
- TB_cohor... 2 months ago
- TB_death... 2 months ago
- TB_death... 2 months ago
- tubercul... 2 months ago
- Untitled.i... 2 months ago
- Untitled1... 2 months ago
- Untitled1... a month ago
- Untitled1... a month ago
- Untitled1... a month ago

Code Editor (Top):

```
[3]: # Vous devez d'abord installer mito en exécutant succesivement dans le terminal
# Les deux codes suivants: python -m pip install
# m mitoinstaller install

[6]: # Exécuter cette cellule puis importer le fichier en cliquant sur Import,
# et trouver le fichier csv dans votre ordinateur et double cliquer dessus
# pour l'ouvrir dans mito.
import mitosheet
mitosheet.sheet()
```

Data Table (Bottom):

	V1	SOQ1	age	sexe	statut	niveau	occupation
1	31-05-2021	1	46	1	2	4	5
2	31-05-2021	2	19	2	1	4	6
3	31-05-2021	3	32	1	1	4	4
4	31-05-2021	4	58	2	2	4	5
5	31-05-2021	5	35	1	1	4	3
6	01-06-2021	7	67	2	2	3	6
7	01-06-2021	7	58	1	2	2	2

Conclusion

- Python est un langage de programmation généraliste utilisant le paradigme P.O.O
- Extensions nombreuses permettant de faire de la science des données
- Utilisation des librairies en général facile quand les syntaxes sont bien maîtrisées
- Nécessité de se former pour chaque librairie fondamentale