

Institut Universitaire des Sciences

Devoir 1 - Math Info

Prepared by :

Nom : PIERRE

Prenom : Yann Lelay

Niveau : III - Scs Informatiques

1 - Installer Jupyter Notebook

```
C:\Users\PC>pip install notebook
Collecting notebook
  Using cached notebook-7.4.0-py3-none-any.whl.metadata (10 kB)
Collecting jupyter-server<3,>=2.4.0 (from notebook)
  Using cached jupyter_server-2.15.0-py3-none-any.whl.metadata (8.4 kB)
Collecting jupyterlab-server<3,>=2.27.1 (from notebook)
  Using cached jupyterlab_server-2.27.3-py3-none-any.whl.metadata (5.9 kB)
Collecting jupyterlab<4.5,>=4.4.0rc0 (from notebook)
  Using cached jupyterlab-4.4.0-py3-none-any.whl.metadata (16 kB)
Collecting notebook-shim<0.3,>=0.2 (from notebook)
  Using cached notebook_shim-0.2.4-py3-none-any.whl.metadata (4.0 kB)
Collecting tornado<6.4.2-cp38-abi3-win_amd64.whl.metadata (2.6 kB)
  Using cached tornado-6.4.2-cp38-abi3-win_amd64.whl.metadata (2.6 kB)
Collecting anyio<3.1.0 (from jupyter-server<3,>=2.4.0->notebook)
  Using cached anyio-4.9.0-py3-none-any.whl.metadata (4.7 kB)
Collecting argon2-cffi<21.1 (from jupyter-server<3,>=2.4.0->notebook)
  Using cached argon2_cffi-23.1.0-py3-none-any.whl.metadata (5.2 kB)
Collecting jinja2<3.0.3 (from jupyter-server<3,>=2.4.0->notebook)
  Using cached jinja2-3.1.6-py3-none-any.whl.metadata (2.9 kB)
Collecting jupyter-client<7.4.4 (from jupyter-server<3,>=2.4.0->notebook)
  Using cached jupyter_client-8.6.3-py3-none-any.whl.metadata (8.3 kB)
Collecting jupyter-core!=5.0.*,>=4.12 (from jupyter-server<3,>=2.4.0->notebook)
  Using cached jupyter_core-5.7.2-py3-none-any.whl.metadata (3.4 kB)
Collecting jupyter-events<0.11.0 (from jupyter-server<3,>=2.4.0->notebook)
  Using cached jupyter_events-0.12.0-py3-none-any.whl.metadata (5.8 kB)
Collecting jupyter-server-terminals<0.4.4 (from jupyter-server<3,>=2.4.0->notebook)
  Using cached jupyter_server_terminals-0.5.3-py3-none-any.whl.metadata (5.6 kB)
Collecting nbconvert<6.4.4 (from jupyter-server<3,>=2.4.0->notebook)
  Using cached nbconvert-7.16.6-py3-none-any.whl.metadata (8.5 kB)
Collecting nbformat<5.3.0 (from jupyter-server<3,>=2.4.0->notebook)
  Using cached nbformat-5.10.4-py3-none-any.whl.metadata (3.6 kB)
Collecting overrides<5.0 (from jupyter-server<3,>=2.4.0->notebook)
  Using cached overrides-7.7.0-py3-none-any.whl.metadata (5.8 kB)
```

```

C:\Users\PC>pip show notebook
Name: notebook
Version: 7.4.0
Summary: Jupyter Notebook - A web-based notebook environment for interactive computing
Home-page: https://github.com/jupyter/notebook
Author:
Author-email: Jupyter Development Team <jupyter@googlegroups.com>
License: BSD 3-Clause License

- Copyright (c) 2001-2015, IPython Development Team
- Copyright (c) 2015-, Jupyter Development Team

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
   list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
   this list of conditions and the following disclaimer in the documentation
   and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its
   contributors may be used to endorse or promote products derived from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR

```

2- Écris un programme qui convertit un nombre décimal en hexadécimal en demandant à l'utilisateur de saisir ce nombre.

```

decimal = int(input("Entrer un nombre decimal : "))
hexad = hex(decimal)
print(f"{decimal} en hexadécimal est {hexad[2:].upper()}")

```

Entrer un nombre decimal : 45

45 en hexadécimal est 2D

3- Écris un programme qui convertit un nombre hexadécimal en décimal en demandant à l'utilisateur de saisir ce nombre.

```

hexad = input("Veuillez entrer un nombre hexadecimal : ")
decimal = int(hexad, 16)
print(f"{hexad} en décimal est {decimal}")

```

Veuillez entrer un nombre hexadecimal : 56

56 en décimal est 86

4- Créer un fichier CSV dans Python, charger le et l'afficher

```
import pandas as pd
```

```
data = {  
    "Prenom": ["Peter", "Ismael", "Web", "Badio", "Evens"],  
    "Nom": ["Chery", "St-Amour", "Marcelin", "Robaldo", "Elisee"]  
}
```

```
df = pd.DataFrame(data)
```

```
df.to_csv("Inscrits.csv", index=False)  
print("Fichier CSV créé avec succès !")
```

Fichier CSV créé avec succès !

```
df = pd.read_csv("Inscrits.csv")  
print(df.head())
```

	Prenom	Nom
0	Peter	Chery
1	Ismael	St-Amour
2	Web	Marcelin
3	Badio	Robaldo
4	Evens	Elisee

5- . Créer grande base de données (5000+ lignes) aléatoires

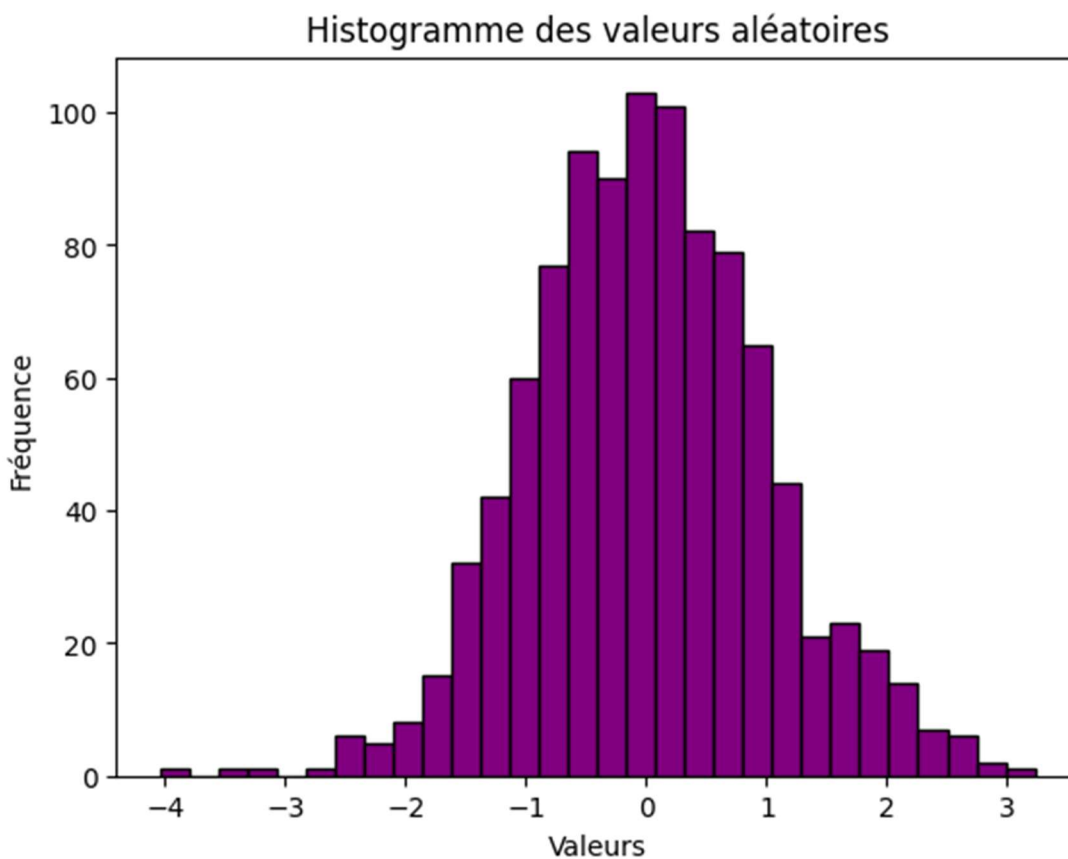
```
import pandas as pd  
import numpy as np  
from faker import Faker  
fake = Faker() # Générateur de données aléatoires  
# Nombre de lignes à générer  
n = 5000  
# Générer des données aléatoires  
data = {  
    'ID': np.arange(1, n+1), # ID de 1 à 1000  
    'Nom': [fake.name() for _ in range(n)], # Noms aléatoires  
    'Âge': np.random.randint(18, 60, size=n), # Âge entre 18 et 60 ans  
    'Ville': [fake.city() for _ in range(n)], # Villes aléatoires  
    'Email': [fake.email() for _ in range(n)], # Emails aléatoires  
    'Date Inscription': [fake.date_this_decade() for _ in range(n)] # Dates  
    d'inscription  
}  
# Convertir en DataFrame Pandas  
df = pd.DataFrame(data)  
# Enregistrer dans un fichier Excel
```

```
df.to_excel('grande_base_de_donnees.xlsx', index=False, engine='openpyxl')
print("Fichier Excel de 5000 entrées généré avec succès !")
```

Fichier Excel de 5000 entrées généré avec succès !

6- Créer 3Graphes

```
import numpy as np
import matplotlib.pyplot as plt
# Générer des données aléatoires
data = np.random.randn(1000)
# Créer l'histogramme
plt.hist(data, bins=30, color='purple', edgecolor='black')
# Ajouter des titres
plt.xlabel("Valeurs")
plt.ylabel("Fréquence")
plt.title("Histogramme des valeurs aléatoires")
# Afficher le graphe
plt.show()
```



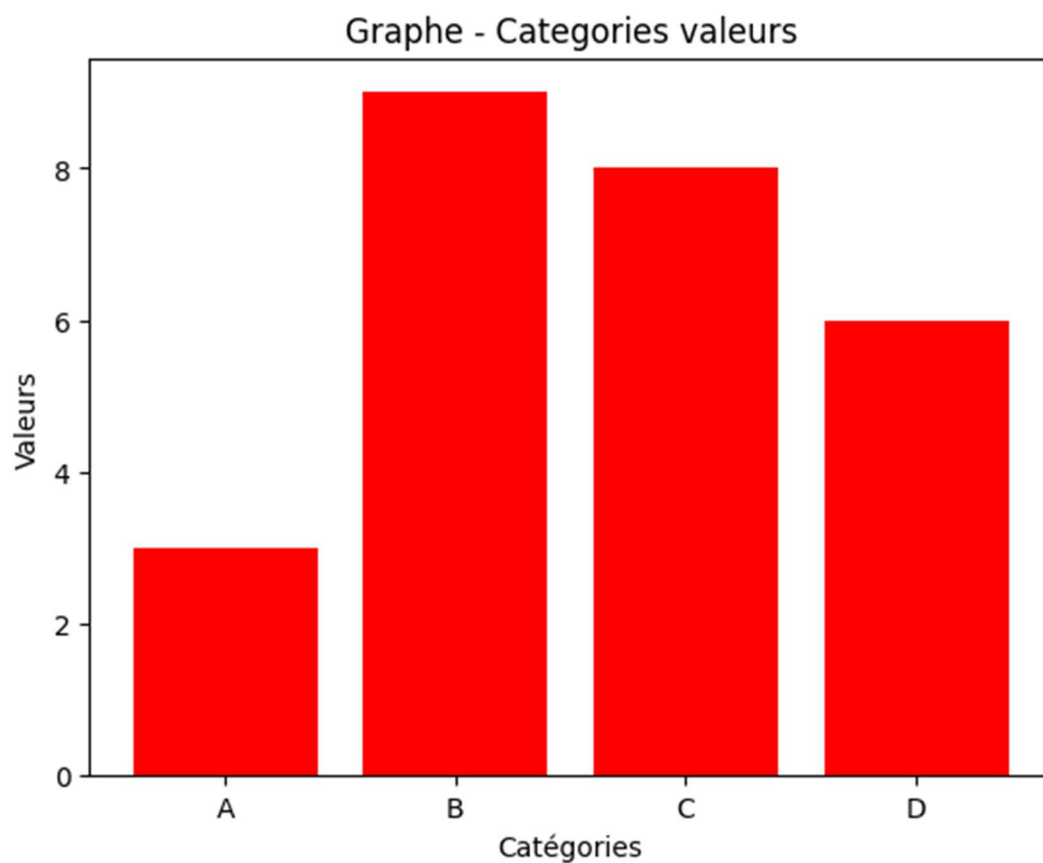
png

```
import matplotlib.pyplot as plt
# Données
categories = ["A", "B", "C", "D"]
```

```
valeurs = [3, 9, 8, 6]
```

```
# Créer Le graphique en barres  
plt.bar(categories, valeurs, color='red')  
# Ajouter des titres  
plt.xlabel("Catégories")  
plt.ylabel("Valeurs")  
plt.title("Graphe - Categories valeurs")  
  
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```

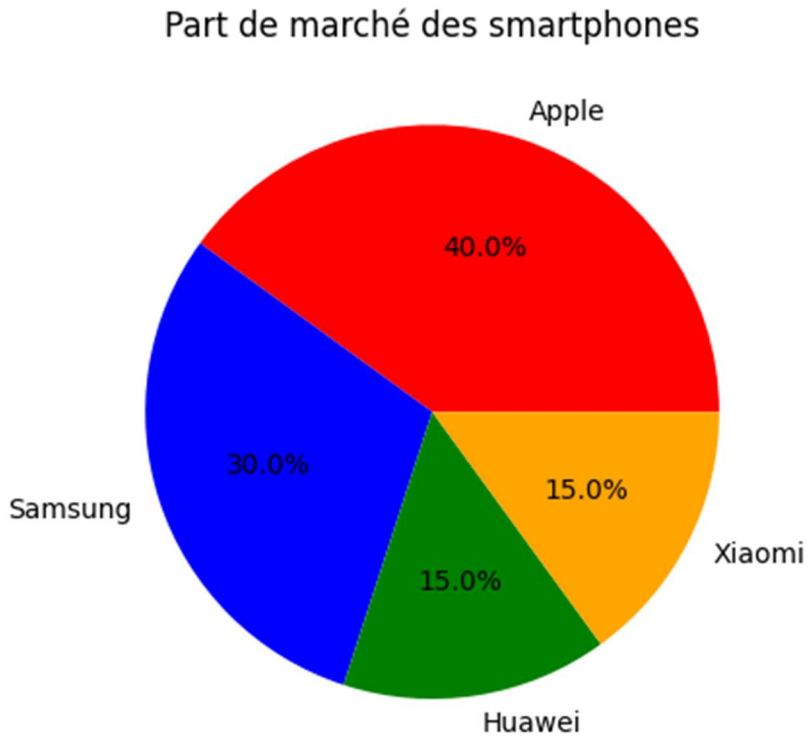


png

```
import matplotlib.pyplot as plt
```

```
# Données  
labels = ["Apple", "Samsung", "Huawei", "Xiaomi"]  
parts = [40, 30, 15, 15]  
# Créer un diagramme circulaire  
plt.pie(parts, labels=labels, autopct='%1.1f%%', colors=['red', 'blue',  
'green', 'orange'])
```

```
# Ajouter un titre  
plt.title("Part de marché des smartphones")  
# Afficher Le graphe  
plt.show()
```



png

Conclusion

Ce Td m'aide dans l'utilisation de nouveau moyen technologique et augmente mes compétences en programmation mathématique avec Python en faisant des conversion decimal en hexadecimal et vice-versa, creation de schema avec la librairie matplotlib.