

Librairie Mendeleev

La librairie [Mendeleev](#) est complète et évoluée

- Package repository sur PyPI : <https://pypi.org/project/mendeleev/>
- Page officielle, description et code source : <https://github.com/Immentel/mendeleev>
- Documentation complète : <https://mendeleev.readthedocs.io/en/stable/>
 - Tutoriels : <https://mendeleev.readthedocs.io/en/stable/tutorials.html>
- Notebook Jupyter (exemples) :
 - https://nbviewer.jupyter.org/github/Immentel/mendeleev/blob/master/docs/source/notebooks/01_intro_to_mendeleev.ipynb (tutoriel introductif)
 - https://nbviewer.jupyter.org/github/Immentel/mendeleev/blob/master/docs/source/notebooks/02_tables.ipynb (accessing the data in bulk)
 - https://nbviewer.jupyter.org/github/Immentel/mendeleev/blob/master/docs/source/notebooks/03_plotting.ipynb (plotting tutorial)
- Installation via pip, ou la commande conda, ou l'interface de Anaconda, suivant l'environnement utilisé :
 - `pip install -user mendeleev`
 - `conda install -c conda-forge mendeleev=0.5.2`
 - `conda install -c Immentel mendeleev=0.6.1` (version plus récente)

Contrairement à ce qu'on trouve dans la documentation, il semble que le canal (channel) à référencer est celui de **conda-forge**, plutôt que Immentel.

En ligne de commande (console), cela donnerait ceci : `conda install -c conda-forge mendeleev=0.6.1`

références :

- <https://anaconda.org/Immentel/mendeleev/files> (limité à la version 0.4.5)
- <https://anaconda.org/conda-forge/mendeleev/files> (actualisé pour la dernière version 0.6.1)

- Données utilisables, en ligne : <http://mendeleev.herokuapp.com/>

Utilisation dans Colaboratory

- Créer une première cellule de code permettant l'installation de la librairie mendeleev :

```
!pip install mendeleev
```

- Fichier exemple :

mendeleev_primer_01.ipynb

Exemples de programmes simples

IonizationEnergy-01.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

"""
Library references :
* https://pypi.org/project/mendeleev/
* https://mendeleev.readthedocs.io/en/stable/
* https://github.com/lmmentel/mendeleev
"""

from mendeleev import element
import matplotlib.pyplot as plt

x, y = range(1,108), [element(i).ionenergies[1] for i in range(1,108)]
for i in range(1,108):
    print(x[i-1], y[i-1])

plt.figure()
plt.plot(x, y)
plt.savefig("ionenergies.png")
plt.show()
```

boiling-melting-temperatures-01.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

"""
Library references :
* https://pypi.org/project/mendeleev/
* https://mendeleev.readthedocs.io/en/stable/
* https://github.com/lmmentel/mendeleev
"""

from mendeleev import element

# on peut accéder aux valeurs en utilisant le symbole de l'élément
print(element('Na').atomic_number)
print(element('Na').melting_point)
print(element('Na').boiling_point)

# on peut aussi accéder aux mêmes valeurs par nombre atomique
print(element(11).melting_point, element(11).boiling_point)

# On peut parcourir une liste d'éléments, par exemple les 18 premiers
for atnum in range(1, 19):
```

```
print(element(atnum).atomic_number,
      element(atnum).symbol,
      element(atnum).name,
      element(atnum).melting_point,
      element(atnum).boiling_point,
      )
```

elements-names-01.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Jan  8 19:11:22 2021

@author: villersd
"""
import string
from mendeleev import element
print(list(range(6)))
for ele in element([1, 2, 3, 4, 5, 6]):
    print(ele.name)

for ele in element(list(range(1,119))):
    print(ele.symbol,)

symbols = [element(i).symbol for i in range(1,119)]
print(symbols)

# recherche de lettres non utilisées pour des symboles chimiques à une
# seule
# lettre
# https://docs.python.org/release/3.8.5/library/string.html
print(string.ascii_uppercase)
nonsymbols = [U for U in string.ascii_uppercase if U not in symbols]
print(nonsymbols)
```

Jupyter notebooks

Application : intégrer les parties de code suivante dans un notebook, après avoir installé la librairie mendeleev :

```
from mendeleev import element
for Z in range(1,19):
    print(Z,element(Z))
```

```
from mendeleev import get_table
ptable = get_table('elements')
```

```
ptable.info()
```

```
cols = ['atomic_number', 'symbol', 'name', 'atomic_radius',  
'covalent_radius_pyykko', 'en_pauling']  
ptable[cols].head(19)
```

```
type(ptable)  
ptable.plot(x='atomic_number', y='covalent_radius_pyykko')
```

Tableau périodique

- [Un tableau périodique avec le module Python "mendelev"](#) David Alberto
 - Voir aussi le package LaTeX [pgf-PeriodicTable](#)

From:
<https://dvillers.umons.ac.be/wiki/> - **Didier Villers, UMONS - wiki**

Permanent link:
<https://dvillers.umons.ac.be/wiki/teaching:progappchim:mendelev?rev=1668782761>

Last update: **2022/11/18 15:46**

