

Python

Cours 8 - Prog. Scientifique et Modules Externes

Steve Lévesque, Tous droits réservés © où applicables

Table des matières

- 1 Utilisation de modules externes populaires "open-source"
- 2 Exemple 1 - Module Numpy
- 3 Exemple 2 - Module Sklearn

Utilisation de modules externes populaires "open-source" pour aider à la programmation en AI (1/6)

Rechercher "npm install (nom du module, i.e. numpy)" sur votre navigateur favori.

Utilisation de modules externes populaires "open-source" pour aider à la programmation en AI (2/6)

Cliquer sur le premier résultat qui apparaît en lien avec le module (en norme) : le site <https://pypi.org/project/numpy/>

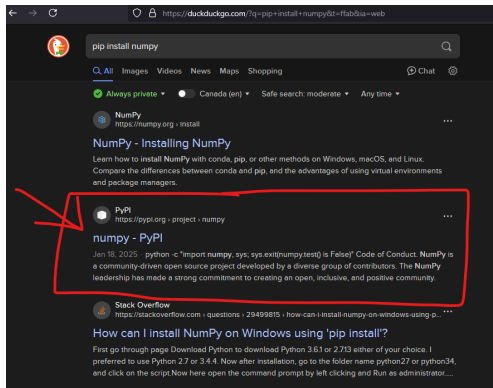


Figure: <https://duckduckgo.com/?q=pip+install+numpy&t=ffab&ia=web>

Utilisation de modules externes populaires "open-source" pour aider à la programmation en AI (3/6)

Copier la commande d'installation sur le site : i.e. "pip install numpy" pour cet exemple.

Attention : Si vous devinez le nom du module, il est possible que le module ne soit pas le bon, ou même un virus !

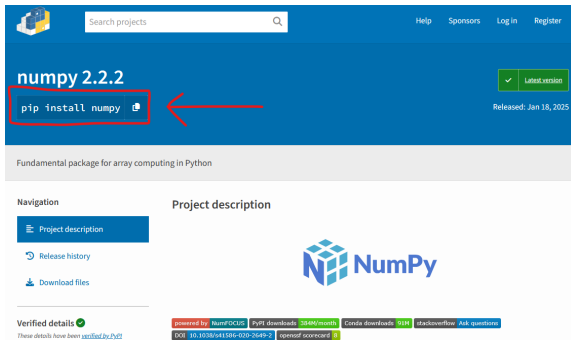
A screenshot of the NumPy project page on the Python Package Index (PyPI). The page has a blue header with the NumPy logo, a search bar, and links for Help, Sponsors, Log in, and Register. The main content area shows "numpy 2.2.2" with a green "Latest version" button. Below this, the command "pip install numpy" is displayed in a box, with a red arrow pointing to it from the right. The page also includes a "Project description" section with the NumPy logo, a "Verified details" section, and a "Navigation" sidebar with links to Project description, Release history, and Download files. At the bottom, there are statistics for downloads and a list of sponsors.

Figure: <https://pypi.org/project/numpy/>

Utilisation de modules externes populaires "open-source" pour aider à la programmation en AI (4/6)

Aller dans la console de votre IDE avec le `.venv` d'activé.

Utilisation de modules externes populaires "open-source" pour aider à la programmation en AI (5/6)

Écrire la commande copiée précédemment, attendre, et le module s'installera tout seul.

Utilisation de modules externes populaires "open-source" pour aider à la programmation en AI (6/6)

Il est maintenant possible de profiter du module !

NB : Il est pertinent de suivre les étapes et valider le nom du module sur PyPI puisque les noms des modules ne sont pas toujours simples et intuitifs dans le registre, et, dans les pires cas, le module peut être un virus.

Exemple 1 - Module Numpy

```
1  # Steve Levesque, All rights reserved
2
3  """
4  (.venv) PS D:Steve> pip install numpy
5  Collecting numpy
6     Obtaining dependency information for numpy from https://files.pythonhosted.org/
7     Downloading numpy-1.26.4-cp312-cp312-win_amd64.whl.metadata (61 kB)
8         ----- 61.0/61.0 kB 819.4 kB/s eta 0:00:00
9  Downloading numpy-1.26.4-cp312-cp312-win_amd64.whl (15.5 MB)
10     ----- 15.5/15.5 MB 3.6 MB/s eta 0:00:00
11  Installing collected packages: numpy
12  Successfully installed numpy-1.26.4
13
14  [notice] A new release of pip is available: 23.2.1 -> 24.0
15  [notice] To update, run: python.exe -m pip install --upgrade pip
16  """
```

Exemple 1 - Module Numpy

```
1  # Steve Levesque, All rights reserved
2
3  # The module installed earlier, made by experts and open-source
4  import numpy as np
5
6  arr = np.array([8, 7, 6, 1, 2, 3, 4, 5])
7  arr = np.sort(arr)
8
9  # [1 2 3 4 5 6 7 8]
10 print(arr)
```

Exemple 2 - Module Sklearn

Listing:

https://scikit-learn.org/stable/auto_examples/classification/plot_digits_classification.html

```
1  # Author: Gael Varoquaux <gael dot varoquaux at normalesup dot org>
2  # License: BSD 3 clause
3  from sklearn import datasets, metrics, svm
4  from sklearn.model_selection import train_test_split
5
6  # MNIST Project : load database and flatten images
7  digits = datasets.load_digits()
8  data = digits.images.reshape((len(digits.images), -1))
9
10 # Create a classifier: a support vector classifier
11 clf = svm.SVC(gamma=0.001)
12 # Split data into 50% train and 50% test subsets
13 X_train, X_test, y_train, y_test = train_test_split(
14     data, digits.target, test_size=0.5, shuffle=False
15 )
16
17 # Learn the digits on the train subset
18 clf.fit(X_train, y_train)
19 # Predict the value of the digit on the test subset
20 predicted = clf.predict(X_test)
```

Bibliographie

- <https://pypi.org/project/numpy/>
- https://scikit-learn.org/stable/auto_examples/classification/plot_digits_classification.html