

# Jour-20 Pip

## Python Pip - Gestionnaire de packages Python

### Qu'est-ce que PIP?

PIP signifie programme d'installation préféré. Nous utilisons *pip* pour installer différents packages Python. Le package est un module Python qui peut contenir un ou plusieurs modules ou d'autres packages. Un module ou des modules que nous pouvons installer sur notre application est un package. En programmation, nous n'avons pas à écrire tous les programmes d'utilité, nous les installons à la place et les importons dans nos applications.

### Installation de pip

Si vous n'avez pas installé PIP, permettez-nous de l'installer maintenant. Accédez à votre terminal ou à votre invite de commande et copiez et collez ceci:

```
asabeneh @ asabeneh: ~ $ pip install pip
```

Vérifiez si PIP est installé en écrivant

### Version Pip

```
asabeneh @ asabeneh: ~ $ pip --version
pip 21.1.3 de /usr/local/lib/python3.7/site-packages/pip (python 3.9.6)
```

Comme vous pouvez le voir, j'utilise PIP version 21.1.3, si vous voyez un peu un peu en dessous ou au-dessus, cela signifie que vous avez installé PIP.

Vérifions certains des packages utilisés dans la communauté Python à différentes fins. Juste pour vous faire savoir qu'il existe de nombreux packages disponibles pour une utilisation avec différentes applications.

### Installation de packages à l'aide de PIP

Essayons d'installer *numpy*, appelé Numeric Python. C'est l'un des packages les plus populaires de la communauté d'apprentissage automatique et de science des données.

- Numpy est le package fondamental de l'informatique scientifique avec Python. Il contient entre autres choses:
  - o Un puissant objet de tableau de dimension à n dimensions
  - o Fonctions sophistiquées (radiodiffusion)
  - o Outils pour intégrer C / C ++ et le code FORTRAN
  - o Algèbre linéaire utile, transformée de Fourier et Capacités de nombre aléatoire

```
asabeneh @ asabeneh: ~ $ pip install numpy
```

Laissez-nous commencer à utiliser Numpy. Ouvrez votre shell interactif Python, écrivez Python puis importez Numpy comme suit:

```
asabeneh @ asabeneh: ~ $ python python 3.9.6 (par défaut, 28 juin 2021, 15:26:21) [Clang 11.0.0 (Clang-1100.0.33.8)] sur Darwin Type "Help", "Copyright", "crédits" ou "Licence" pour plus d'informations. >>> Import Numpy >>> numpy.version.version '1.20.1' >>> lst = [1, 2, 3, 4, 5] >>> np_arr = Numpy.array(lst) >>> np_arr ([1, 2, 3, 4, 5]) 1 en (np_arr) 5 >>> np_arr * 2 array ([2, 4, 6, 8, 10]) >>> np_arr + 2 array ([3, 4, 5, 6, 7]) >>>
```

Pandas est une bibliothèque open source et liée à BSD offrant des structures de données et des outils d'analyse de données hautes performances et faciles à utiliser pour le langage de programmation Python. Installons le grand frère de Numpy, **pandas**:

```
asabeneh @ asabeneh: ~ $ pip install pandas
```

```
asabeneh @ asabeneh: ~ $ python python 3.9.6 (par défaut, 28 juin 2021, 15:26:21) [Clang 11.0.0 (Clang-1100.0.33.8)] sur Darwin Type "Help", "Copyright", "crédits" ou "Licence" pour plus d'informations. >>> Importer des pandas
```

Cette section ne concerne pas Numpy ni Pandas, nous essayons ici d'apprendre à installer des packages et à les importer. Si cela est nécessaire, nous parlerons de différents packages dans d'autres sections.

Prévoyons un module de navigateur Web, qui peut nous aider à ouvrir n'importe quel site Web. Nous n'avons pas besoin d'installer ce module, il est déjà installé par défaut avec Python 3. Par exemple, si vous aimez ouvrir n'importe quel nombre de sites Web à tout moment ou si vous aimez planifier quelque chose, ce module **webbrowser** peut être utilisé.

Importez WebBrowser # Module de navigateur Web pour ouvrir des sites Web

```
# liste des URL: Python url_lists = ['http://www.python.org',
```

```
'https://www.linkedin.com/in/asabeneh/',  
'https://github.com/asabeneh', 'https://twitter.com/asabeneh',]
```

# Ouvrez la liste ci-dessus des sites Web dans un onglet différent pour URL dans  
URL\_LISTS: webbrowser.open\_new\_tab (URL)

## Packages désinstallés

Si vous n'aimez pas conserver les packages installés, vous pouvez les supprimer à l'aide de la commande suivante.

```
pip désinstaller packagename
```

## Liste des packages

Pour voir les packages installés sur notre machine. Nous pouvons utiliser PIP suivi de la liste.

```
liste PIP
```

## Package d'exposition

Pour afficher des informations sur un package

```
pip show packagename
```

```
asabeneh @ asabeneh: ~ $ pip show pandas  
Nom: Pandas Version: 1.2.3 Résumé: Structures de données puissantes pour l'analyse des données, les séries chronologiques et les statistiques  
Page d'accueil: http://pandas.pydata.org Auteur: Aucun Author-Email: Aucune Licence: BSD Location: /usr/local/lib/python3.7/site-packages  
a besoin: Python-Dautil, Pytz, Numpy requis-by:
```

If we want even more details, just add --verbose

```
asabeneh @ asabeneh: ~ $ pip show --verbose pandas
Nom: Pandas
Version: 1.2.3
Résumé: Structures de données puissantes pour l'analyse des données, les séries chronologiques et les statistiques
PAGE HAMPARE: http://pandas.pydata.org
Auteur: Aucun
Autor-email: Aucun
Licence: BSD
/usr/local/lib/python3.7
Numpy, pytz, python-dateutil
requis-by: Metadata
Version: 2.1
Installateur: Classificateurs PIP: Statut de développement :: 5 - Langue de programmation stable :: Console :: OS
Langue de programmation :: Python :: 3.7
Langue de programmation :: Python :: 3.8
Langage de programmation :: Cython
Topic :: Scientific / Engineering
```

Points d'entrée:

```
[pandas_plotting_backends]
Matplotlib = Pandas: tracé._matplotlib
```

## Congeler

Générez des packages Python installés avec leur version et la sortie convient à l'utiliser dans un fichier d'exigences. Un fichier `exigence.txt` est un fichier qui doit contenir tous les packages Python installés dans un projet Python.

```
asabeneh @ asabeneh: ~ $ pip Freeze docutils
== 0.11 Jinja2 == 2.7.2 MarkupSafe == 0.19 pygments == 1.6 sphinx == 1.2.2
```

Le Freeze Pip nous a donné les packages utilisés, installés et leur version. Nous l'utilisons avec le fichier `exigence.txt` pour le déploiement.

## Lecture de l'URL

Vous connaissez maintenant comment lire ou écrire sur un fichier situé sur votre machine locale. Parfois, nous aimerions lire un site Web utilisant l'URL ou une API. L'API signifie interface du programme d'application. C'est un moyen d'échanger des données structurées entre les serveurs principalement sous forme de données JSON. Pour ouvrir une connexion réseau, nous avons besoin d'un package appelé *requests* - il permet d'ouvrir une connexion réseau et d'implémenter les opérations CRUD (créer, lire, mettre à jour et supprimer). Dans cette section, nous couvrirons uniquement la lecture du minerai faisant partie d'un crud.

Installons *requests*:

```
asabeneh @ asabeneh: ~ $ requêtes d'installation de pip
```

Nous verrons *get*, *status\_code*, *headers*, *text* et *json* méthodes dans le module *requests*:

- ?

- headers*: pour vérifier les types d'en-tête

- text*: pour extraire le texte de l'objet de réponse récupéré
- json*: Pour extraire les données JSON, lions un fichier TXT à partir de ce site Web, [https://www.w3.org/tr/png/iso\\_8859-1.txt](https://www.w3.org/tr/png/iso_8859-1.txt).

```
importer concuquêtes # Importation du mod de demande       ule

URL = 'https://www.w3.org/tr/png/iso_8859-1.txt' # Texte d'un site Web

Response = requêtes.get (URL) # ouvrant un réseau et récupérant une données

Imprimer (réponse)    print (réponse.status_code) # code d'état, succès: 200 imprime
r (réponse.headers) # Informations en en-têtes INTRÉSIR (Response.Text) # donne t
out le texte de la page
```

```
<Réponse [200] >
200
{'date': 'Sun, 08 Dec 2019 18:00:31 GMT', 'last-modified': 'Fri, 07 Nov 2003 05:51:11 GMT',
'etag': '"17e9- 3cb82080711c0;50c0b26855880-gzip"', 'accept-ranges': 'bytes', 'cache-control':
'Max-Age = 31536000', 'expires': 'lun, 07 décembre 2020 18:00:31 GMT', 'Vary': 'Accept-Enc
oding', 'Content- Encoding': 'GZIP', 'Access-Control-allow-Origin': '', 'Content-Length': '1616
', 'Content-Type ': '', 'text-longueur': '1616 ', 'contenu ':', 'strict-transport-security': '
max-age = 15552000; Comprend des publications; Preload ', 'Content-Security-Policy ': 'Met
re-Insecure-Requests '}
```

- Lisons à partir d'une API. L'API signifie interface du programme d'application. C'est un moyen d'échanger des données de structure entre les serveurs principalement des données JSON. Un exemple d'API: <https://restcountries.eu/rest/v2/all>. Lisons cette API à l'aide du module *requests*.

#### Demandes d'importation

URL = 'https://restcountries.eu/rest/v2/all' # Pays API Response = requests.get (URL) # ouvre un réseau et récupérant une données

Print (Response) # Response Object Print (Response.Status\_Code) # Code de statut, Succès: 200 pays = réponse.json () Imprimer (pays [: 1]) # Nous avons tranché uniquement le premier pays, supprimer le tranchage pour voir tous les pays

<Réponse [200] >

200

```
[{'alpha2code': 'af', 'alpha3code': 'afg', 'altspellings': ['af', 'afġānistān'], 'région': 652230.0, 'Borders': ['irn ', 'pak ', 'tkm ', 'uzb ', 'tjk ', 'chn ']: ': ': ', 'uzb', 'TJK' ['93'], 'Capital': 'Kaboul', 'Cioc': 'afg', 'Currency': [{'code': 'afn', 'name': 'Afghan Afghani', 'Symbol': 'ؑ'}], 'De Money': 'Afghan', 'Flag': 'https://restcouncoun 'gini': 27.8, 'Langues': [{'ISO639_1': 'PS', 'ISO639_2': 'PUS', 'NAM 'native name': 'o 'zbek'}, {'iso639_1': 'tk', 'iso639_2': 'tuk', 'name': 'Turkmen', 'native name': 'türkmen'}],
```

```
'latlng': [33.0, 65.0],  
«Nom»: «Afghanistan»,  
'native name': 'افغانستان',  
«NumericCode»: «004»,  
«Population»: 27657145,  
«région»: «Asie»,  
'régional blocs': [{'acronyme': 'saarc',
```

```
'name': 'South Asian Association for Regional Cooperation', 'autresAcronyms': [], 'autres noms': []], 'Sous-région': 'Asie du Sud', 'Timezones': ['utc + 04:30'], 'TopLeveldomain': ['.af'], 'Traduction 'Afghanistan', 'es': 'afganistán', 'fa': 'افغانستان', 'fr': 'afghanistan', 'hr': 'アフガニスタン', 'it': 'afghanistan', 'ja': 'アフガニスタン'
```

Nous utilisons la méthode *json()* de l'objet de réponse, si nous récupérons les données JSON. Pour TXT, HTML, XML et d'autres formats de fichiers, nous pouvons utiliser *text*.

## Création d'un package

Nous organisons un grand nombre de fichiers dans différents dossiers et sous-repliants en fonction de certains critères, afin que nous puissions les trouver et les gérer facilement. Comme vous le savez, un module peut contenir plusieurs objets, tels que des classes, des fonctions, etc. Un package C et contiennent un ou plusieurs modules pertinents. Un package est en fait un dossier contenant un ou plusieurs fichiers de modules. Créons un package nommé MyPackage, en utilisant les étapes suivantes:

Créez un nouveau dossier nommé MyPacakge à l'intérieur du dossier de 30 jours sur le dossier. Créez un fichier init.py vide dans le dossier mypackage. Créer des modules arithmétique.py et saluer.py avec le code suivant:

```
# mypackage / arithmetics.py # arithmeti
cs.py def add_numbers (* args): total =
0 pour num dans args: total += num retu
rn total def soustraire (a, b): return (a - b
) def multiple (a, b):
```

```
Retour A * b Def Division (A,  
B): Retour A / B Def reste (A,  
B): Retour a% B Def Power (A,  
B): Retour A ** B
```

```
# mypackage / salut.py  
# saluer.py  
Def Greall_Serson (FirstName, LastName): Return f '{FirstName} {LastName}, bienvenue da  
ns 30daysofpython Challenge!'
```

La structure du dossier de votre paquet devrait ressembler à ceci:

```
Mypackage  fique
```

Ordonons maintenant le shell interactif Python et essayons le package que nous avons créé:

```
asabeneh @ asabeneh: ~ / burktop / 30daysofpython $ python python 3.9.6 (par défaut,  
28 juin 2021, 15:26:21) [Clang 11.0.0 (Clang-1100.0.33.8)] sur Darwin Type "Help", "  
Copyright", "crédits" ou "licence" pour plus d'informations. >>> de MyPackage Import  
t Arithmetics >>> arithmetics.add_numbers (1, 2, 3, 5) 11
```

```
>>>Arithmetics.Substract (5, 3) 2
```

```
>>>arithmetics.multiple (5, 3) 1  
5
```

```
>>>arithmetics.division (5, 3)  
1.6666666666666667
```

```
>>>arithmetics.remainder (5, 3) 2  
>>> arithmetics.power (5, 3) 125
```

```
>>>de MyPackage Import Greet  
>>>accorde.greet_person ('asabeneh', 'encoreayeh')  
«Asabeneh YEAYEH, bienvenue au 30Daysofpython Challenge!  
>>>
```



Comme vous pouvez le voir, notre package fonctionne parfaitement. Le dossier du package contient un fichier spécial appelé `init.py` - il stocke le contenu du package. Si nous mettons `init.py` dans le dossier Package, Python Start le reconnaît comme un package. L'`init.py` expose les ressources spécifiées de ses modules à importer dans d'autres fichiers Python. Un fichier `init.py` vide rend toutes les fonctions disponibles lorsqu'un package est importé. L'`init.py` est essentiel pour que le dossier soit reconnu par Python comme un package.

Plus d'informations sur les packages

- Base de données O SQLALCHEMY OU SQLOBJECT - Accès orienté objet à plusieurs systèmes de base de données différents ***pip install SQLAlchemy*** • Développement Web O Django - Framework Web de haut niveau. ***pip install django*** o Flask - micro-framework pour python basé sur werkzeug, Jinja 2. (C'est

BSD sous licence)  
***pip install flask***

- Parser HTML O Belle soupe - Parser HTML / XML conçu pour des projets de revirement rapides comme la tartement d'écran, acceptera une mauvaise majoration. ***pip install beautifulsoup4*** o pyquery - implémente jQuery dans python; Plus vite que BeautifulSoup, apparemment.

- Traitement XML O ElementTree - Le type d'élément est un objet de conteneur simple mais flexible, conçu pour stocker des structures de données hiérarchiques, telles que des infosets XML simplifiés, en mémoire. - NOTE: Python 2.5 et UP ont ElementTree dans la bibliothèque standard

- GUI O PYQT - Bindings pour le framework QT multiplateforme. O TKINTER - La boîte à outils d'interface utilisateur Python traditionnelle.

- Analyse des données, science des données et apprentissage automatique

O Numpy: Numpy (Numeric Python) est connu comme l'une des bibliothèques d'apprentissage automatique les plus populaires de Python. o Pandas: est une analyse des données, la science des données et une bibliothèque d'apprentissage automatique à Python qui fournit des structures de données de haut niveau et une grande variété d'outils pour l'analyse. O SCIPY: SCIPY est une bibliothèque d'apprentissage automatique pour les développeurs d'applications et les ingénieurs. La bibliothèque Scipy contient des modules d'optimisation, d'algèbre linéaire, d'intégration, de traitement d'image et de statistiques. O Scikit-Learn: c'est numpy et scipy. Il est considéré comme l'une des meilleures bibliothèques pour travailler avec des données complexes. O TensorFlow: est une bibliothèque d'apprentissage automatique construite par Google. o Keras: est considéré comme l'une des bibliothèques d'apprentissage automatique les plus cool de Python. Il fournit un mécanisme plus facile pour exprimer des réseaux de neurones. Keras fournit également certains des meilleurs utilitaires pour compiler les modèles, le traitement des ensembles de données, la visualisation des graphiques et bien plus encore.

- Réseau: o Demandes: est un package que nous pouvons utiliser pour envoyer des demandes à un serveur (obtenir, publier, supprimer, mettre)  
*pip install requests*

Vous progressez toujours et vous êtes un chef de 20 étapes jusqu'à votre chemin vers la grandeur. Faites maintenant quelques exercices pour votre cerveau et vos muscles.

## Exercices: Jour 20

1. Lisez cette URL et trouvez les 10 mots les plus fréquents. Romeo\_and\_Juliet = '<http://www.gutenberg.org/files/1112/1112.txt>'

2. Lisez l'API Cats et Cats\_API = '<https://api.thecatapi.com/v1/breeds>' et recherche:

je. Le Min, Max, Mean, médian, écart type du poids des chats en unités métriques.  
ii Le Min, Max, Mean, médian, écart-type de la durée de vie des chats au cours des années.  
iii. Créer une table de fréquence de pays et de race de chats

3. Lisez l'API [des pays](#) et trouvez

je. Les 10 plus grands pays II. Les 10 langues les plus parlées  
s III. Le nombre total de langues dans l'API des pays

4. UCI est l'un des endroits les plus courants pour obtenir des ensembles de données pour la science des données et l'apprentissage automatique. Lisez le contenu de UCL (<https://archive.ics.uci.edu/ml/datasets.php>). Dans les bibliothèques supplémentaires, ce sera difficile, vous pouvez donc l'essayer avec BeautifulSoup4

Félicitations!