

Python - PIP

A biblioteca padrão do Python é uma grande coleção de módulos e pacotes prontos para uso. Além desses pacotes, um programador Python geralmente precisa usar determinadas bibliotecas de terceiros. Pacotes Python de terceiros são hospedados em um repositório chamado Python Package Index (<https://pypi.org/>) .

Para instalar um pacote deste repositório, você precisa de uma ferramenta gerenciadora de pacotes. PIP é um dos gerenciadores de pacotes mais populares.

O utilitário PIP é instalado automaticamente com a distribuição padrão do Python, especialmente a partir da versão 3.4. Está presente na pasta scripts dentro do diretório de instalação do Python. Por exemplo, quando o Python 3.11 está instalado em um computador Windows, você pode encontrar pip3.exe na pasta C:\Python311\Scripts.

Se o **pip** não estiver instalado por padrão, ele poderá ser instalado seguindo o procedimento a seguir.

Baixe o script get-pip.py do seguinte URL -

```
https://bootstrap.pypa.io/get-pip.py
```

Para instalar, execute o script acima no prompt de comando -

```
c:\Python311>python get-pip.py
```

Na pasta de scripts, pip e pip3 estão presentes. Se pip for usado para instalar um determinado pacote, sua versão compatível com Python 2.x será instalada. Portanto, para instalar a versão compatível com Python 3, use pip3.

Instale um pacote

Para instalar um determinado pacote do PyPi, use o comando de instalação com PIP. O comando a seguir instala a biblioteca Flask na instalação atual do Python.

```
pip3 install flask
```

O pacote, junto com suas dependências, se houver, será instalado a partir do repositório PyPI. O comando acima produz o seguinte log no terminal -

```
Collecting flask
  Downloading Flask-2.2.3-py3-none-any.whl (101 kB)
  101.8/101.8 kB 3.0 MB/s eta 0:00:00
Collecting Werkzeug>=2.2.2
```



```

Downloading Werkzeug-2.2.3-py3-none-any.whl (233 kB)
233.6/233.6 kB 7.2 MB/s eta 0:00:00
Collecting Jinja2>=3.0
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
133.1/133.1 kB 8.2 MB/s eta 0:00:00
Collecting itsdangerous>=2.0
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting click>=8.0
  Downloading click-8.1.3-py3-none-any.whl (96 kB)
96.6/96.6 kB 5.4 MB/s eta 0:00:00
Requirement already satisfied: colorama in
c:\users\mlath\appdata\roaming\python\python311\site-packages (from
click>=8.0->flask) (0.4.6)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.2-cp311-cp311-win_amd64.whl (16 kB)
Installing collected packages: MarkupSafe, itsdangerous, click,
Werkzeug, Jinja2, flask
Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.2 Werkzeug-2.2.3
click-8.1.3 flask-2.2.3 itsdangerous-2.1.2

```

Por padrão, a versão mais recente disponível do pacote desejado é instalada. Para especificar a versão necessária,

```
pip3 install flask==2.0.0
```

Para testar se a instalação do pacote foi concluída, abra o interpretador Python e tente importá-lo e verifique a versão. Se o pacote não tiver sido instalado com êxito, você receberá um `ModuleNotFoundError`.

```

>>> import flask
>>> print (flask.__version__)
2.2.3
>>> import dummypackage
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'dummypackage'

```

O utilitário PIP funciona com -

- PyPI (e outros índices) usando especificadores de requisitos.
- URLs do projeto VCS.
- Diretórios de projetos locais.



- Arquivos de origem locais ou remotos.

Usar requisitos.txt

Você pode realizar a instalação do pacote imediatamente, mencionando a lista de pacotes necessários em um arquivo de texto denominado requisitos.txt.

Por exemplo, o arquivo requisitos.txt a seguir contém uma lista de dependências a serem instaladas para a biblioteca FastAPI.

```
anyio==3.6.2
click==8.1.3
colorama==0.4.6
fastapi==0.88.0
gunicorn==20.1.0
h11==0.14.0
idna==3.4
pydantic==1.10.4
sniffio==1.3.0
starlette==0.22.0
typing_extensions==4.4.0
uvicorn==0.20.0
```

Agora use a opção `-r` no comando de instalação PIP.

```
pip3 install -r requirements.txt
```

O utilitário PIP é usado junto com os seguintes comandos -

desinstalação do pip

Este comando é usado para desinstalar um ou mais pacotes já instalados.

Sintaxe

```
pip3 uninstall package, [package2, package3, . . .]
```

Isso desinstalará os pacotes junto com as dependências.

Exemplo

```
pip3 uninstall flask
```



Será solicitada a confirmação da remoção antes de prosseguir.

```
pip3 uninstall flask
Found existing installation: Flask 2.2.3
Uninstalling Flask-2.2.3:
  Would remove:
    c:\python311\lib\site-packages\flask-2.2.3.dist-info\*
    c:\python311\lib\site-packages\flask\*
    c:\python311\scripts\flask.exe
Proceed (Y/n)?
```

lista pip

Este comando fornece uma lista de pacotes instalados, incluindo editáveis. Os pacotes são listados em uma ordem de classificação que não diferencia maiúsculas de minúsculas.

Sintaxe

```
pip3 list
```

As seguintes opções estão disponíveis com o comando pip list -

-o, --outdated: Lista pacotes desatualizados

```
pip3 list --outdated
```

Package	Version	Latest	Type
-----	-----	-----	-----
debugpy	1.6.6	1.6.7	wheel
ipython	8.11.0	8.12.0	wheel
pip	22.3.1	23.0.1	wheel
Pygments	2.14.0	2.15.0	wheel
setuptools	65.5.0	67.6.1	wheel

-u, --uptodate: Lista pacotes de atualização

```
pip3 list --uptodate
```

Package	Version
-----	-----
click	8.1.3
colorama	0.4.6
executing	1.2.0
Flask	2.2.3
jedi	0.18.2
Jinja2	3.1.2
python-dateutil	2.8.2



pyzmq	25.0.2
six	1.16.0
Werkzeug	2.2.3

DE ANÚNCIOS



Cruit

a new way to hire talent

show de pip

Este comando mostra informações sobre um ou mais pacotes instalados. A saída está no formato de cabeçalho de correio compatível com RFC.

Sintaxe

```
pip3 show package
```

Exemplo

```
pip3 show flask
Name: Flask
Version: 2.2.3
Summary: A simple framework for building complex web applications.
Home-page: https://palletsprojects.com/p/flask
Author: Armin Ronacher
Author-email: armin.ronacher@active-4.com
License: BSD-3-Clause
Location: C:\Python311\Lib\site-packages
Requires: click, itsdangerous, Jinja2, Werkzeug
Required-by:
```

DE ANÚNCIOS

pip congelar

Este comando gera pacotes instalados no formato de requisitos. Todos os pacotes são listados em uma ordem de classificação que não diferencia maiúsculas de minúsculas.

Sintaxe

```
pip3 freeze
```

A saída deste comando pode ser redirecionada para um arquivo de texto com o seguinte comando -

```
pip3 freeze > requirements.txt
```

DE ANÚNCIOS

baixar pip

Este comando baixa pacotes de -

- PyPI (e outros índices) usando especificadores de requisitos.
- URLs do projeto VCS.
- Diretórios de projetos locais.
- Arquivos de origem locais ou remotos.

Na verdade, pip download faz a mesma resolução e download que pip install, mas em vez de instalar as dependências, ele coleta as distribuições baixadas no diretório fornecido (padrão para o diretório atual). Este diretório pode posteriormente ser passado como o valor para pip install --find-links para facilitar a instalação offline ou bloqueada de pacotes.

Sintaxe

```
pip3 download somepackage
```

pesquisa pip

Este comando procura pacotes PyPI cujo nome ou resumo contém a consulta fornecida.

Sintaxe

```
pip3 search query
```

configuração do pip

Este comando é usado para gerenciar a configuração local e global.

Subcomandos

- **list** - Liste a configuração ativa (ou do arquivo especificado).
- **edit** - Edite o arquivo de configuração em um editor.
- **get** - Obtenha o valor associado a `command.option`.
- **set** - Defina o `command.option = valor`.
- **unset** - Desative o valor associado a `command.option`.
- **debug** - Liste os arquivos de configuração e os valores definidos neles.

As chaves de configuração devem ser nomes de comandos e opções separados por pontos, com o prefixo especial "global" afetando qualquer comando.

Exemplo

```
pip config set global.index-url https://example.org/
```

Isso configuraria o URL do índice para todos os comandos.

```
pip config set download.timeout 10
```

Isso configuraria um tempo limite de 10 segundos apenas para comandos "pip download".