Python TUTORIAL

Python pip

Summary: in this tutorial, you'll learn about Python pip and how to use it to manage third-party packages.

Introduction to Python package index (PyPI)

Python has a rich standard library that you can use immediately. In case you need a package that isn't available in the standard library, you can find it on the Python Package Index (https://pypi.org/) .

The Python Package Index (PyPI) is the largest Python repository. It contains many Python packages developed and maintained by the Python community.

To find a package, you can use the search box. For example, to search packages that deal with HTTP requests, you can simply use the `requests` keyword.

The search result will show many relevant packages. To find detailed information about each package, you can click the corresponding link.

Let's examine the requests package (https://pypi.org/project/requests/) .

Package version

Python packages use semantic versioning which consists of three-part version numbers: major version, minor version, and patch:

```
major.minor.patch
```

The patch number is incremented for minor changes and bug fixes that don't change the way the package works.

The minor version is also incremented for releases that add new features that are backward-compatible.

The major version is incremented for the changes which are not backward compatible.

For example, the `requests` package has version 2.24.0 (at the time of this writing). It has the major version is 2, the minor version is 24, and the patch is zero.

If you use the version `requests` version 2.24.0 in your project, you can upgrade it to any version that has the major version 2, for example, 2.25.1.

If you install a package with a higher major version e.g., 3.0.0, your application may not work correctly.

What is pip

To download the package, you use the command described in the module:

```
pip install requests
```

But what is pip?

pip is the package installer for python. Pip allows you to install packages from PyPI and other repositories.

Python comes with `pip` by default. To check if `pip` is available on your computer, you can open the command prompt (or Powershell) on Windows and type the following command:

```
pip --V
```

It'll show something like this:

```
pip 20.2.4 from C:\Users\<username>\AppData\Roaming\Python\Python38\site-packages
```

...where 20.2.4 is the version and `C:\Users<username>\AppData\Roaming\Python\Python38\site-packages\pip` is the location of pip.

If you use macOS or Linux, you can launch the terminal and use the `pip3` instead of `pip` :

```
pip3 --V
```

Install a package

To install a package from PyPI, you use the following command on Windows:

```
pip install <package_name>
```

And change `pip` to `pip3` on macOS and Linux:

```
pip3 install <package_name>
```

For example, the following command installs the `requests` package:

```
pip install requests
```

From now on, you can use the `requests` package in any project. For example, you can create a new project called `pip-demo` and use the `requests` package.

The following code uses the `requests` package to make an HTTP request to the `https://pypi.org/` and displays the HTTP status code:

```
import requests

response = requests.get('https://pypi.org/')
print(response.status_code)
```

Output:

```
200
```

To install a package with a specific version, you use the following command:

```
pip install <package_name>==<version>
```

The following command installs the `requests` package version 2.20.1:

```
pip install requests==2.20.1
```

List installed packages

To list all installed packages, you use the following `pip` command:

```
pip list
```

It'll return a list of packages installed on your computer like this:

```
Package          Version
---------------- ---------
appdirs          1.4.4
autopep8         1.5.4
certifi          2020.6.20
chardet          3.0.4
colorama         0.4.4
```

```
distlib          0.3.1
filelock         3.0.12
idna             2.10
Pillow           8.0.0
pip              20.2.4
pycodestyle      2.6.0
requests         2.24.0
```

To check what packages are outdated, you use the following command:

```
pip list --outdated
```

Output:

```
Package    Version Latest Type
---------- ------- ------ -----
setuptools 47.1.0  50.3.2 wheel
```

It shows the package name, the installed version and the latest version.

Uninstall a package

To uninstall a package, you use the `pip uninstall` command:

```
pip uninstall <package_name>
```

It'll prompt you for a confirmation like this:

```
Proceed (y/n)?
```

If you type `y` , pip is going to uninstall the package. Otherwise, it won't do so.

List dependencies of a package

When you install a package and if that package uses other packages, pip will install the package and its dependency, and also dependency of dependencies, and so on.

To show the dependencies of a package, you use the following command:

```
pip show <package_name>
```

The following command shows the dependencies of the requests package:

```
pip show requests
```

The `Requires` line lists out the packages used by the requests packages.

```
Requires: urllib3, chardet, idna, certifi
```

Summary

- Python package index provides the third-party Python packages developed and maintained by the Python community.
- Use Python installer for Python (pip) to manage third-party Python packages.