

Package Management

Essentially, you have two options to manage Python packages: conda and pip. Here's a good discussion about the differences between the two:

<http://jakevdp.github.io/blog/2016/08/25/conda-myths-and-misconceptions/#:~:text=In%20short%2C%20pip%20is%20a.any%20package%20within%20conda%20environments.>

For our purposes, they are basically interchangeable but you should be aware of the differences

Conda

Conda will search Anaconda Repositories and try to find the package you want to install:

<https://anaconda.org/>

For example, the requests library:



















The screenshot shows the Anaconda repository search results for the 'requests' package. At the top, there is a search bar with 'requests' entered and a green search button. Below the search bar, there are filters for 'Type: All', 'Access: All', and 'Platform: All'. The search results are displayed in a table with columns: 'Favorites', 'Downloads', 'Package (owner / package)', and 'Platforms'. The first result is for the 'requests' package by 'conda-forge', version 2.25.1. The package description is 'Requests is an elegant and simple HTTP library for Python, built with ♥.' The platforms listed are linux-64, linux-aarch64, linux-ppc64le, noarch, osx-64, win-32, and win-64.

Favorites	Downloads	Package (owner / package)	Platforms
7	9324814	conda-forge / requests 2.25.1 Requests is an elegant and simple HTTP library for Python, built with ♥.	linux-64 linux-aarch64 linux-ppc64le noarch osx-64 win-32 win-64



















Then, it just drops the packages into this folder under your installation of Anaconda:

PC > Local Disk (C:) > Users > Will Tirone > anaconda3 > pkgs				
Name	^	Date modified	Type	Size
_ipyw_jlab_nb_ext_conf-0.1.0-py38_0		1/21/2021 2:58 PM	File folder	
alabaster-0.7.12-py_0		1/21/2021 2:57 PM	File folder	
anaconda-2020.11-py38_0		1/21/2021 2:57 PM	File folder	
anaconda-client-1.7.2-py38_0		1/21/2021 2:57 PM	File folder	
anaconda-navigator-1.10.0-py38_0		1/21/2021 2:57 PM	File folder	
anaconda-project-0.8.4-py_0		1/21/2021 2:57 PM	File folder	
argh-0.26.2-py38_0		1/21/2021 2:57 PM	File folder	
argon2-cffi-20.1.0-py38he774522_1		1/21/2021 2:57 PM	File folder	
asn1crypto-1.4.0-py_0		1/21/2021 2:57 PM	File folder	
astroid-2.4.2-py38_0		1/21/2021 2:57 PM	File folder	
astropy-4.0.2-py38he774522_0		1/21/2021 2:57 PM	File folder	
async_generator-1.10-py_0		1/21/2021 2:57 PM	File folder	
atomicwrites-1.4.0-py_0		1/21/2021 2:57 PM	File folder	
attrs-20.3.0-pyhd3eb1b0_0		1/21/2021 2:57 PM	File folder	
autopep8-1.5.4-py_0		1/21/2021 2:57 PM	File folder	
babel-2.8.1-pyhd3eb1b0_0		1/21/2021 2:57 PM	File folder	
backcall-0.2.0-py_0		1/21/2021 2:57 PM	File folder	
backports.functools_lru_cache-1.6.1-py_0		1/21/2021 2:57 PM	File folder	
backports.shutil_get_terminal_size-1.0.0-...		1/21/2021 2:57 PM	File folder	
backports.tempfile-1.0-py_1		1/21/2021 2:57 PM	File folder	
backports.weakref-1.0.post1-py_1		1/21/2021 2:57 PM	File folder	
backports-1.0-py_2		1/21/2021 2:57 PM	File folder	
bcrypt-3.2.0-py38he774522_0		1/21/2021 2:57 PM	File folder	

If we look at the requests library, we can find the same files on github:

» This PC » Local Disk (C:) » Users » Will Tirone » anaconda3 » pkgs » requests-2.24.0-py_0 » site-packages » requests				
Name	Date modified	Type	Size	
 <code>_init__.py</code>	2/16/2021 12:46 PM	PY File	5 KB	
 <code>_version__.py</code>	2/16/2021 12:46 PM	PY File	1 KB	
 <code>_internal_utils.py</code>	2/16/2021 12:46 PM	PY File	2 KB	
 <code>adapters.py</code>	2/16/2021 12:46 PM	PY File	21 KB	
 <code>api.py</code>	2/16/2021 12:46 PM	PY File	7 KB	
 <code>auth.py</code>	2/16/2021 12:46 PM	PY File	10 KB	
 <code>certs.py</code>	2/16/2021 12:46 PM	PY File	1 KB	
 <code>compat.py</code>	2/16/2021 12:46 PM	PY File	2 KB	
 <code>cookies.py</code>	2/16/2021 12:46 PM	PY File	18 KB	
 <code>exceptions.py</code>	2/16/2021 12:46 PM	PY File	4 KB	
 <code>help.py</code>	2/16/2021 12:46 PM	PY File	4 KB	
 <code>hooks.py</code>	2/16/2021 12:46 PM	PY File	1 KB	
 <code>models.py</code>	2/16/2021 12:46 PM	PY File	34 KB	
 <code>packages.py</code>	2/16/2021 12:46 PM	PY File	1 KB	
 <code>sessions.py</code>	2/16/2021 12:46 PM	PY File	30 KB	
 <code>status_codes.py</code>	2/16/2021 12:46 PM	PY File	5 KB	
 <code>structures.py</code>	2/16/2021 12:46 PM	PY File	3 KB	
 <code>utils.py</code>	2/16/2021 12:46 PM	PY File	30 KB	

Here's github:

master requests / requests /		Go to file	Add file	...
sigmavirus24 Merge pull request #5681 from mateusduboli/5677		8c211a9	on Jan 24	History
..				
 <code>_init__.py</code>	Upgrade to chardet 4.x	3 months ago		
 <code>_version__.py</code>	v2.25.1	2 months ago		
 <code>_internal_utils.py</code>	adding unicode_is_ascii utility function	4 years ago		
 <code>adapters.py</code>	Revert "Use urllib for chunked requests"	12 months ago		
 <code>api.py</code>	Merge branch 'master' into master	2 years ago		
 <code>auth.py</code>	Updated references to previous requests/requests GitHub path	2 years ago		
 <code>certs.py</code>	cleanup certs.py	4 years ago		
 <code>compat.py</code>	Remove internal use of unnecessary compat shim for OrderedDict	2 years ago		
 <code>cookies.py</code>	Use comprehensions whenever possible	2 years ago		
 <code>exceptions.py</code>	Normalize exception docstrings (#5338)	12 months ago		
 <code>help.py</code>	remove final remnants from 2.6	2 years ago		
 <code>hooks.py</code>	Use comprehensions whenever possible	2 years ago		
 <code>models.py</code>	Clarify that PreparedRequests should not be manually created	8 months ago		
 <code>packages.py</code>	Fix requests.packages not having package attributes	4 years ago		
 <code>sessions.py</code>	Merge pull request #5681 from mateusduboli/5677	last month		
 <code>status_codes.py</code>	Fix formatting of status code example	2 years ago		
 <code>structures.py</code>	Remove internal use of unnecessary compat shim for OrderedDict	2 years ago		
 <code>utils.py</code>	updated 'get_encoding_from_headers' to return utf-8 if the content ty...	3 months ago		

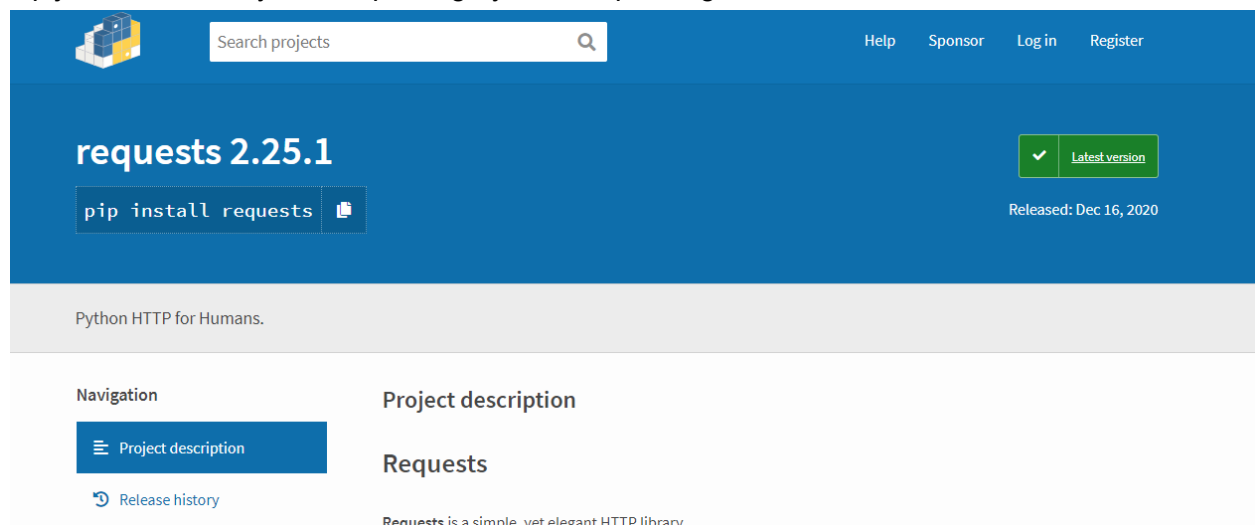
Using Pip instead:

You'll notice that *pip install requests* also works, and will drop the package in the same Anaconda folder.

```
(base) C:\Users\Will Tirone>pip install requests
Requirement already satisfied: requests in c:\users\will tirone\anaconda3\lib\site-packages (2.24.0)
Requirement already satisfied: urllib3<1.25.0,!>1.25.1,<1.26,>=1.21.1 in c:\users\will tirone\anaconda3\lib\site-packages (from requests) (1.25.11)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\will tirone\anaconda3\lib\site-packages (from requests) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in c:\users\will tirone\anaconda3\lib\site-packages (from requests) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\will tirone\anaconda3\lib\site-packages (from requests) (2020.6.20)
```

(notice the first line - the site packages folder. That's where we were just looking earlier)

Pip just searches PyPi for a package you're requesting:

The image shows the PyPI project page for the 'requests' package, version 2.25.1. The page has a blue header with the PyPI logo, a search bar, and links for Help, Sponsor, Log in, and Register. Below the header, the package name 'requests 2.25.1' is prominently displayed. To the right of the package name is a green button with a checkmark and the text 'Latest version'. Below the package name is a button that says 'pip install requests' with a copy icon. To the right of this button, it says 'Released: Dec 16, 2020'. Below the main content area, there is a section for 'Navigation' with links to 'Project description' and 'Release history'. To the right of this is the 'Project description' section, which includes the title 'Requests' and a brief description: 'Requests is a simple, yet elegant HTTP library.'

Virtual Environments:

Creating Virtual Environments

Python "Virtual Environments" allow Python [packages](#) to be installed in an isolated location for a particular application, rather than being installed globally. If you are looking to safely install global command line tools, see [Installing stand alone command line tools](#).

Problems will arise if your code depends on a certain version of a package that, say, gets updated and breaks your production code. We want to freeze the packages in their current versions. Virtual environments are just a way to contain packages and freeze the version of them you're using.

You can do this with Python's venv module or conda.

The cheat sheet located here is very helpful and I reference it often:

<https://docs.conda.io/projects/conda/en/latest/user-guide/cheatsheet.html>

You'll notice the **(base)** text here: that's just the current environment I'm in. When you install a new package, you're actually installing it into the base environment.

```
(base) C:\Users\Will Tirone>
```

We can check which packages we have installed with “conda list”. We'll see there are hundreds in the base environment:

```
(base) C:\Users\Will Tirone>conda list
# packages in environment at C:\Users\Will Tirone\anaconda3:
#
# Name                                Version                                Build      Channel
_ipyw_jlab_nb_ext_conf               0.1.0                                py38_0
alabaster                             0.7.12                               py_0
anaconda                             2020.11                              py38_0
anaconda-client                       1.7.2                                py38_0
anaconda-navigator                    1.10.0                               py38_0
anaconda-project                      0.8.4                                 py_0
argh                                  0.26.2                               py38_0
argon2-cffi                           20.1.0                              py38he774522_1
asn1crypto                            1.4.0                                 py_0
astroid                               2.4.2                                py38_0
astropy                               4.0.2                                py38he774522_0
async_generator                       1.10                                 py_0
atomicwrites                          1.4.0                                 py_0
attrs                                 20.3.0                              pyhd3eb1b0_0
autopep8                              1.5.4                                 py_0
babel                                 2.8.1                                pyhd3eb1b0_0
backcall                              0.2.0                                 py_0
backports                             1.0                                   py_2
backports.functools_lru_cache         1.6.1                                py_0
backports.shutil_get_terminal_size    1.0.0                                py38_2
backports.tempfile                    1.0                                   py_1
backports.weakref                     1.0.post1                            py_1
bcrypt                                3.2.0                                py38he774522_0
beautifulsoup4                        4.9.3                                pyhb0f4dca_0
bitarray                              1.6.1                                py38h2bbff1b_0
bkcharts                              0.2                                   py38_0
blas                                  1.0                                   mkl
bleach                                3.2.1                                py_0
blosc                                 1.20.1                               h7bd577a_0
bokeh                                  2.2.3                                py38_0
boto                                   2.49.0                               py38_0
bottleneck                            1.3.2                                py38h2a96729_1
brotlipy                              0.7.0                                py38he774522_1000
```

Let's make a new virtual environment:

```
(base) C:\Users\Will Tirone>conda create --name code_lou python=3.7
WARNING: A space was detected in your requested environment path
'C:\Users\Will Tirone\anaconda3\envs\code_lou'
Spaces in paths can sometimes be problematic.
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\Will Tirone\anaconda3\envs\code_lou

  added / updated specs:
    - python=3.7

The following packages will be downloaded:
```

package	build	
ca-certificates-2021.1.19	haa95532_0	122 KB
certifi-2020.12.5	py37haa95532_0	141 KB
openssl-1.1.1j	h2bbff1b_0	4.8 MB
pip-21.0.1	py37haa95532_0	1.8 MB
python-3.7.9	h60c2a47_0	14.4 MB
setuptools-52.0.0	py37haa95532_0	711 KB
vc-14.2	h21ff451_1	8 KB
vs2015_runtime-14.27.29016	h5e58377_2	1007 KB
wheel-0.36.2	pyhd3eb1b0_0	33 KB
wincertstore-0.2	py37_0	14 KB
Total:		23.0 MB

```
The following NEW packages will be INSTALLED:

ca-certificates      pkgs/main/win-64::ca-certificates-2021.1.19-haa95532_0
certifi              pkgs/main/win-64::certifi-2020.12.5-py37haa95532_0
openssl              pkgs/main/win-64::openssl-1.1.1j-h2bbff1b_0
pip                  pkgs/main/win-64::pip-21.0.1-py37haa95532_0
python               pkgs/main/win-64::python-3.7.9-h60c2a47_0
setuptools           pkgs/main/win-64::setuptools-52.0.0-py37haa95532_0
sqlite               pkgs/main/win-64::sqlite-3.33.0-h2a8f88b_0
vc                   pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime       pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel                pkgs/main/noarch::wheel-0.36.2-pyhd3eb1b0_0
wincertstore         pkgs/main/win-64::wincertstore-0.2-py37_0
zlib                 pkgs/main/win-64::zlib-1.2.11-h62dcd97_4

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

We have to deactivate the current env:

```
(base) C:\Users\Will Tirone>conda deactivate
```

Now we can see we're in the new environment I created



```
C:\Users\Will Tirone>conda activate code_lou  
(code_lou) C:\Users\Will Tirone>
```

We can check what kind of environments we have total:

```
(code_lou) C:\Users\Will Tirone>conda env list  
# conda environments:  
#  
base                C:\Users\Will Tirone\anaconda3  
code_lou            * C:\Users\Will Tirone\anaconda3\envs\code_lou
```

The only thing this is doing is dropping new folders under the “envs” folder in my anaconda install:

is PC > Local Disk (C:) > Users > Will Tirone > anaconda3 > envs

Name	Date modified	Type	Size
 code_lou	2/24/2021 10:40 AM	File folder	
 .conda_envs_dir_test	2/24/2021 10:40 AM	CONDA_ENVS_DI...	0 KB

If we run `conda list` again we see we only have a few packages, the ones that are default installed when conda makes a new virtual env:

```
(code_lou) C:\Users\Will Tirone>conda list
# packages in environment at C:\Users\Will Tirone\anaconda3\envs\code_lou:
#
# Name                          Version          Build      Channel
ca-certificates                2021.1.19        haa95532_0
certifi                        2020.12.5        py37haa95532_0
openssl                        1.1.1j           h2bbff1b_0
pip                            21.0.1           py37haa95532_0
python                         3.7.9            h60c2a47_0
setuptools                     52.0.0           py37haa95532_0
sqlite                         3.33.0           h2a8f88b_0
vc                             14.2             h21ff451_1
vs2015_runtime                 14.27.29016      h5e58377_2
wheel                          0.36.2           pyhd3eb1b0_0
wincertstore                   0.2              py37_0
zlib                           1.2.11           h62dcd97_4
```

Let's install pandas in our new environment:

```
(code_lou) C:\Users\Will Tirone>conda install pandas
```

And we can run `conda list` again to see that pandas (and its dependencies, like numpy, were installed):

```
(code_lou) C:\Users\Will Tirone>conda list
# packages in environment at C:\Users\Will Tirone\anaconda3\envs\code_lou:
#
# Name                          Version          Build      Channel
blas                           1.0              mkl
ca-certificates                2021.1.19        haa95532_0
certifi                        2020.12.5        py37haa95532_0
intel-openmp                   2020.2           254
mkl                             2020.2           256
mkl-service                    2.3.0            py37h196d8e1_0
mkl_fft                        1.2.1            py37h46781fe_0
mkl_random                     1.1.1            py37h47e9c7a_0
numpy                           1.19.2           py37hadc3359_0
numpy-base                     1.19.2           py37ha3acd2a_0
openssl                        1.1.1j           h2bbff1b_0
pandas                         1.2.2            py37hf11a4ad_0
pip                            21.0.1           py37haa95532_0
python                         3.7.9            h60c2a47_0
python-dateutil                2.8.1            pyhd3eb1b0_0
pytz                           2021.1           pyhd3eb1b0_0
setuptools                     52.0.0           py37haa95532_0
six                            1.15.0           py37haa95532_0
sqlite                         3.33.0           h2a8f88b_0
vc                             14.2             h21ff451_1
vs2015_runtime                 14.27.29016      h5e58377_2
wheel                          0.36.2           pyhd3eb1b0_0
wincertstore                   0.2              py37_0
zlib                           1.2.11           h62dcd97_4
```


Frequently, you'll see python packages on GitHub with a requirements.txt file. How do you make your own?

```
(code_lou) C:\Users\Will Tirone>pip freeze > requirements.txt
```

This will make a text document of the files you have installed and put it in your current working directory.

 requirements.txt	2/24/2021 11:00 AM	Text Document	10 KB
--	--------------------	---------------	-------

Now, I'm in a new environment (notice the text in parentheses) and we can install the same packages we used in the other environment using this:

```
(another_test_env) C:\Users\Will Tirone>pip install requirements.txt
```