

# Workshops w/ Hoffman<sub>2</sub> Python

- **Numerous components of W6 (BS-Seq) use Python programs installable from PyPI**
  - Experience shows this can be a **bog-down point** when trying to have students follow along live (or even offline)
    - **Students have widely-varying/broken configurations** due to prior interactions with one or more of the multitude of Python distributions/managers and former attempts to “get things working” without understanding what doing
      - Shell startup scripts hardcoding weird `${PATH}/env. vars.`, problematic collections of already-installed packages at user level, broken mixtures from multiple Python versions/managers, ...
        - ***“FIXING” can “break” their existing research work!***
- **Most recent teaching (November) tried built-in VENV “virtual environment” support, which worked well**

# W6 Example on Hoffman<sub>2</sub>

```
$ module avail --all --contains --icase adapt  
« ...no matches at all... »
```

**Example tool W6 uses: CutAdapt, which is a Python package and not in module system**

```
$ module avail --all --contains --icase python
```

```
python/2.7.15  
python/2.7.18  
python/3.6.8(default)  
python/3.7.3  
python/3.9.6
```

If you have hardcoded Python things (e.g., in shell startup scripts `~/.bashrc` `~/.bash_profile` `~/.profile` ...), remove them (and re-log in) for the duration of the Workshop. Note `ls` hides dotfiles by default (try `ls -la ~`). (Shell syntax `~USER` expands to home directory; omitted `USER` defaults to yourself.)

```
$ module load python/3.7.3
```

```
$ which python3
```

```
/u/local/apps/python/3.7.3/gcc-4.8.5/bin/python3
```

**Make sure everybody starts by using the same underlying Python install on Hoffman<sub>2</sub>**

```
$ echo "${PATH}" | tr ':' '\n'
```

```
/u/local/apps/python/3.7.3/gcc-4.8.5/bin
```

.....

```
$ python3 -m pip list
```

Package	Version
-----	-----
argon2-cffi	20.1.0
async-generator	1.10

.....

I have previously installed many Python packages (under my home directory in `~/.local`)... this is an example of an existing Python environment that we want to ignore and leave alone for the Workshop

Hoffman<sub>2</sub> module system has adjusted environment variables — including `PATH` — to point to the IDRE-installed Python 3.7.3

# Python VENVs

## CREATE ENVIRONMENT (once)

```
[cokus@n2112 Day1]$ python3 -m venv "${SCRATCH}"/Python-3.7.3-W6-VEENV
```

« ...do **ONCE** to create new environment (clean copy of invoked Python); makes complex filesystem tree at requested place: »

```
[cokus@n2112 Day1]$ ls -l "${SCRATCH}"/Python-3.7.3-W6-VEENV
```

```
drwx-----. 2 cokus matteop 4096 Nov 15 20:29 bin
```

```
drwx-----. 2 cokus matteop 4096 Nov 15 20:27 include
```

.....

```
[cokus@n2112 Day1]$ find "${SCRATCH}"/Python-3.7.3-W6-VEENV | wc -l
```

```
1045
```

« ...it made over 1,000 filesystem objects... »

```
[cokus@n2112 Day1]$ du -c -h "${SCRATCH}"/Python-3.7.3-W6-VEENV
```

```
4.0K /u/scratch/c/cokus/Python-3.7.3-W6-VEENV/include
```

.....

```
18M total
```

« ...it made dozens of directories and used ~18MB of disk... »

## USE ENVIRONMENT (anytime)

```
[cokus@n2112 Day1]$ source "${SCRATCH}"/Python-3.7.3-W6-VEENV/bin/activate
```

« ...do **ANYTIME** to activate the particular environment; adjusts environment variables and `PS1` shell prompt: »

```
(Python-3.7.3-W6-VEENV) [cokus@n2112 Day1]$ echo "${VIRTUAL_ENV}"
```

```
/u/scratch/c/cokus/Python-3.7.3-W6-VEENV
```

```
(Python-3.7.3-W6-VEENV) [cokus@n2112 Day1]$ echo "${PATH}" | tr ':' '\n'
```

```
/u/scratch/c/cokus/Python-3.7.3-W6-VEENV/bin
```

```
/u/local/apps/python/3.7.3/gcc-4.8.5/bin
```

.....

```
(Python-3.7.3-W6-VEENV) [cokus@n2112 Day1]$ which python3
```

```
/u/scratch/c/cokus/Python-3.7.3-W6-VEENV/bin/python3
```

```
(Python-3.7.3-W6-VEENV) [cokus@n2152 Day1]$ python3 -m pip list
```

Package	Version
-----	-----
pip	19.0.3
setuptools	40.8.0

Note: **shell prompt adjusted** to remind you that this virtual environment is activated

Note: **bin directory inside venv is now at the front of your `PATH`**, so unqualified Python-y things will come from there now

Note: basically no Python modules installed — **venv starts clean**

# Now Easy: CutAdapt Install/Run

```
(Python-3.7.3-W6-VEHV) [cokus@n2112 Day1]$ python3 -m pip install --upgrade cutadapt
```

```
Collecting cutadapt
  Downloading https://files.pythonhosted.org/packages/6e/dd/.....whl (162kB)
Collecting xopen~=1.1 (from cutadapt)
  Downloading https://files.pythonhosted.org/packages/24/71/.....whl
Collecting dnaio~=0.5 (from cutadapt)
  Downloading https://files.pythonhosted.org/packages/01/df/.....whl (133kB)
Collecting isal>=0.9.0; ..... (from xopen~=1.1->cutadapt)
  Downloading https://files.pythonhosted.org/packages/18/cf/.....whl (1.0MB)
Installing collected packages: isal, xopen, dnaio, cutadapt
Successfully installed cutadapt-3.5 dnaio-0.6.0 isal-0.11.1 xopen-1.2.1
```

```
(Python-3.7.3-W6-VEHV) [cokus@n2112 Day1]$ which cutadapt
/u/scratch/c/cokus/Python-3.7.3-W6-VEHV/bin/cutadapt
```

```
(Python-3.7.3-W6-VEHV) [cokus@n2112 Day1]$ cutadapt
```

```
This is cutadapt 3.5 with Python 3.7.3 .....
```

```
Run "cutadapt --help" to see command-line options. ....
```

```
cutadapt: error: You did not provide any input file names. Please give me something to do!
```

```
(Python-3.7.3-W6-VEHV) [cokus@n2112 Day1]$ cutadapt --help
```

```
« ..... Long and useful help appears ..... »
```

**PIP/PyPI installs now easy...**

e.g., no need for user-local install options (now default is to venv); PIP in a sane state: should have no package conflicts, etc.; ...

**CutAdapt install location** is also nicely **already at front of `PATH`**

If you're in a shell with an activated Python virtual environment, and you want to stop using that environment, just give the command `deactivate` (no path or parameters; your shell prompt should go back to the way it was). If you want to *permanently delete* the environment, you can delete (`rm -rfv --`) the whole environment's directory tree.

# Using VENV Inside Scripts

```
#!/bin/bash
source /etc/bashrc
module load python/3.7.3
source "${SCRATCH}"/Python-3.7.3-W6-venv/bin/activate
.....
cutadapt ...
.....
```

Shell **scripts** (especially UGE job scripts) **generally need to include same commands used in interactive session**

Hoffman<sub>2</sub> system-level shell startup scripts seem chaotic in terms of what gets setup for “login”/“interactive” shells vs. (e.g.) job script shells... this is the easiest and safest (most general and reliable and enduring) way I’ve been dealing with that for a while.

(Note that generally something has to be done here so that the `module` system is available.)