



Develop
Intelligence



Learning Solutions to Attract, Retain,
and Grow your top technical talent.

Setup

- Python 2.7.x
- Python 3.6.x is fine too, but as most libraries are still in 2.7.x, this class will focus on that
- Virtual environments
- pip (package management)
- A terminal (Mac terminal or iTerm2 are fine)
- A text editor of your choice
- IDEs like Eclipse, IDLE, PyCharm are fine too but I'll be teaching basics with plain text

Python 2.7.x

- 🕒 On Mac, you should have this already pre-installed
- 🕒 Test it out with

```
$ python --version  
Python 2.7.10
```

```
$ python  
>>> print "Hello, World!"
```

- On Mac, you would need to install this from <http://python.org>

```
$ python3 --version  
Python 3.5.1
```

```
$ python3  
>>> print "Hello, World!"  
File "<stdin>", line 1  
  print "Hello, World!"  
      ^
```

```
SyntaxError: Missing parentheses in call to 'print'  
>>> print("Hello, World!")  
Hello, World!
```

- Python package management
- Registry of packages on [PyPi](https://pypi.org/)
- Verify you have pip installed:

```
$ pip --version  
pip 9.0.1 from /usr/local/lib/...
```

```
$ pip --help  
$ pip freeze # or pip list  
$ pip show <pkg-name>
```

Virtual Environments

- ⦿ An **absolute, non-negotiable, must have** for any type of serious development work
- ⦿ Avoids versioning nightmare
- ⦿ No more having eight versions of a package on your laptop

```
$ pip install virtualenv
```

```
$ pip install virtualenvwrapper
```

virtualenvwrapper (installing)

- 🕒 This is a really, really nice way to manage
- 🕒 Edit shell startup file (I use ~/.bash_profile):

```
# for virtualenvwrapper
export WORKON_HOME=$HOME/.virtualenvs
export PROJECT_HOME=$HOME/Code
source /usr/local/bin/virtualenvwrapper.sh
```

- 🕒 Then run:

```
$ source ~/.bash_profile
$ workon
```

virtualenvwrapper (creating)



Develop
Intelligence

```
$ python
```

```
>>> import requests
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
ImportError: No module named requests
```

```
$ mkvirtualenv test
```

```
(test)$ pip install requests
```

```
(test)$ python
```

```
>>> import requests
```

```
>>> requests.get("http://google.com")
```

```
<Response [200]>
```

```
>>> exit()
```


virtualenvwrapper (deleting)

```
(test)$ pip freeze # show installed packages
(test)$ which python # what do we see here?
(test)$ deactivate
$ which python
$ workon test
(test)$ deactivate
$ rmvirtualenv test
$ workon test
ERROR: Environment 'test' does not exist. Create
it with 'mkvirtualenv test'.
```

Lab 0: Using a virtualenv

- Create a new virtualenv
- Install the Fabric SSH automation library for Python
- Install the boto v3 library for AWS automation
- Verify you can import them in a terminal
- **Q:** Which versions of each did you install?
- **Q:** Where is this virtualenv located on disk?
- Delete the virtual environment