

Python tutorial week 1: Anaconda + Jupyter Notebook

November 26, 2018

Tutorial structure

- **Week 1 (Feb. 28 2018)**
 - Install Python via Anaconda
 - Jupyter Notebook introduction
 - git/github and conda environments
- **Week 2 (Mar. 7 2018)**
 - Python basics
 - More on git/github

Today

- Download week 1 materials
- Download/install Python via Anaconda (or Miniconda)

- Python's history
- Python's structure
- IPython and Jupyter Notebook

Before we start... get this week's material

Grab course materials from GitHub

Recommended: clone the repository using “git clone”

Open a terminal shell, cd into a directory where you'd like to keep contents from this tutorial, and enter the command below. The repository will be downloaded into the folder **ESS-Python-Tutorial**.

```
git clone https://github.com/bairdlangenbrunner/ESS-Python-Tutorial
```

Before we start... get this week's material

Grab course materials from GitHub

If you don't have git on your computer...

Download/unzip the contents the GitHub repository directly at:

<https://github.com/bairdlangenbrunner/ESS-Python-Tutorial>

Before we start... download Anaconda

If you're working on a personal laptop:

Download Anaconda (recommended: Python 3.7)

<https://www.anaconda.com/download/>

There's also a "lightweight" version called Miniconda, if your computer is very short on memory (or internet is slow):

<https://conda.io/miniconda.html>

- If you already have it, run the Anaconda installer on your machine (following default install settings)

Before we start... download Anaconda

If you're working on a remote machine (e.g., greenplanet), download it remotely on the machine:

1. ssh into the remote machine and cd into your home directory
2. wget the install file (Linux and Mac options below):

```
wget https://repo.anaconda.com/archive/Anaconda3-5.3.1-MacOSX-x86_64.sh
```

or

```
https://repo.anaconda.com/archive/Anaconda3-5.3.1-Linux-x86_64.sh
```

Run the sh file, go with the default settings:

```
bash Anaconda3-5.3.1-Linux-x86_64.sh
```

History of Python

PEP 8: Style guide for Python code

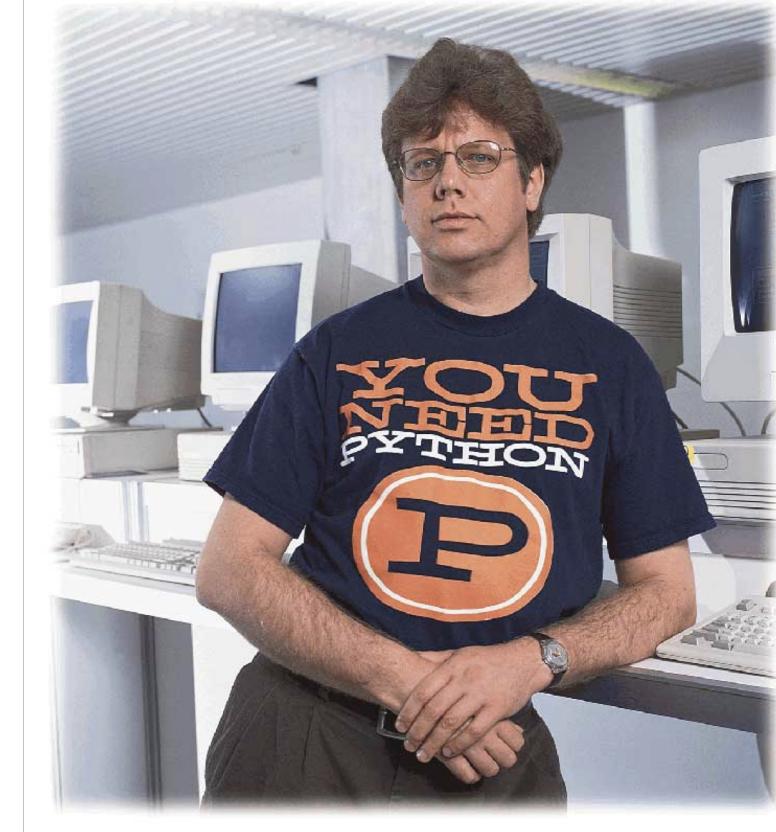
<https://www.python.org/dev/peps/pep-0008/>

Stylized website version:

<https://pep8.org/>

High-level take-home points:

- Indentation matters in Python (4 spaces preferred, not tabs)
- Code should be 79 characters in width, then use continuation character: \
- Use spaces for clarity (a = 5, not a=5).
- But rules should be broken if it's a matter of readability



History of Python

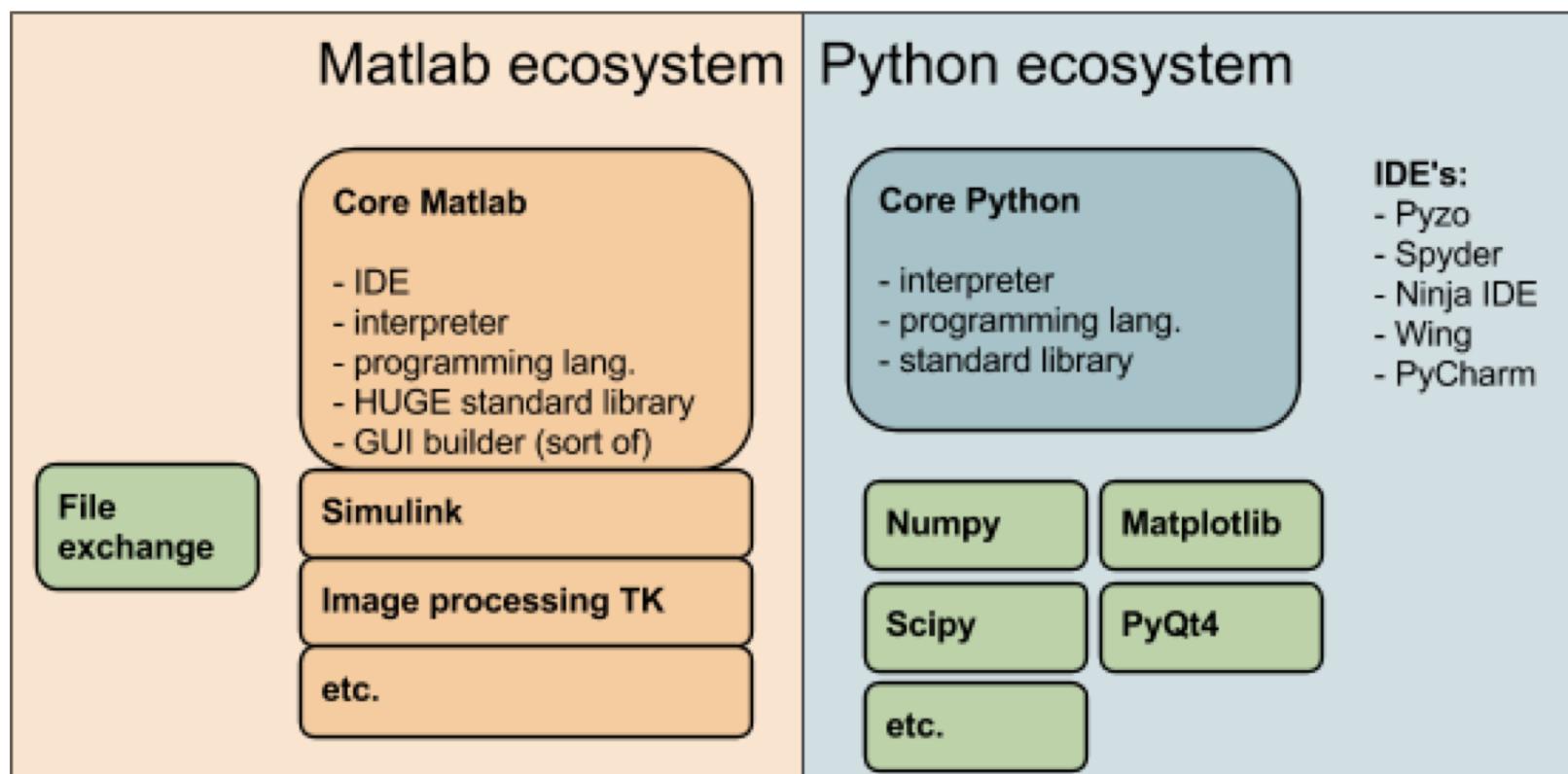
- Python 1
- Python 2 – released 2000
 - Sunset date: was 2015, now 2020
 - Python 2.7 is considered “legacy”
 - Still used, though (e.g., UVCDAT)
 - `print a`
- Python 3 – released 2008
 - Major overhaul to language
 - Most changes are behind the scenes
 - `print(a)`

- Python 1.0 - January 1994
 - Python 1.2 - April 10, 1995
 - Python 1.3 - October 12, 1995
 - Python 1.4 - October 25, 1996
 - Python 1.5 - December 31, 1997
 - Python 1.6 - September 5, 2000
- Python 2.0 - October 16, 2000
 - Python 2.1 - April 15, 2001
 - Python 2.2 - December 21, 2001
 - Python 2.3 - July 29, 2003
 - Python 2.4 - November 30, 2004
 - Python 2.5 - September 19, 2006
 - Python 2.6 - October 1, 2008
 - Python 2.7 - July 3, 2010
- Python 3.0 - December 3, 2008
 - Python 3.1 - June 27, 2009
 - Python 3.2 - February 20, 2011
 - Python 3.3 - September 29, 2012
 - Python 3.4 - March 16, 2014
 - Python 3.5 - September 13, 2015
 - Python 3.6 - December 23, 2016
 - Python 3.7 - June 27, 2018

Python structure (compared to MATLAB)

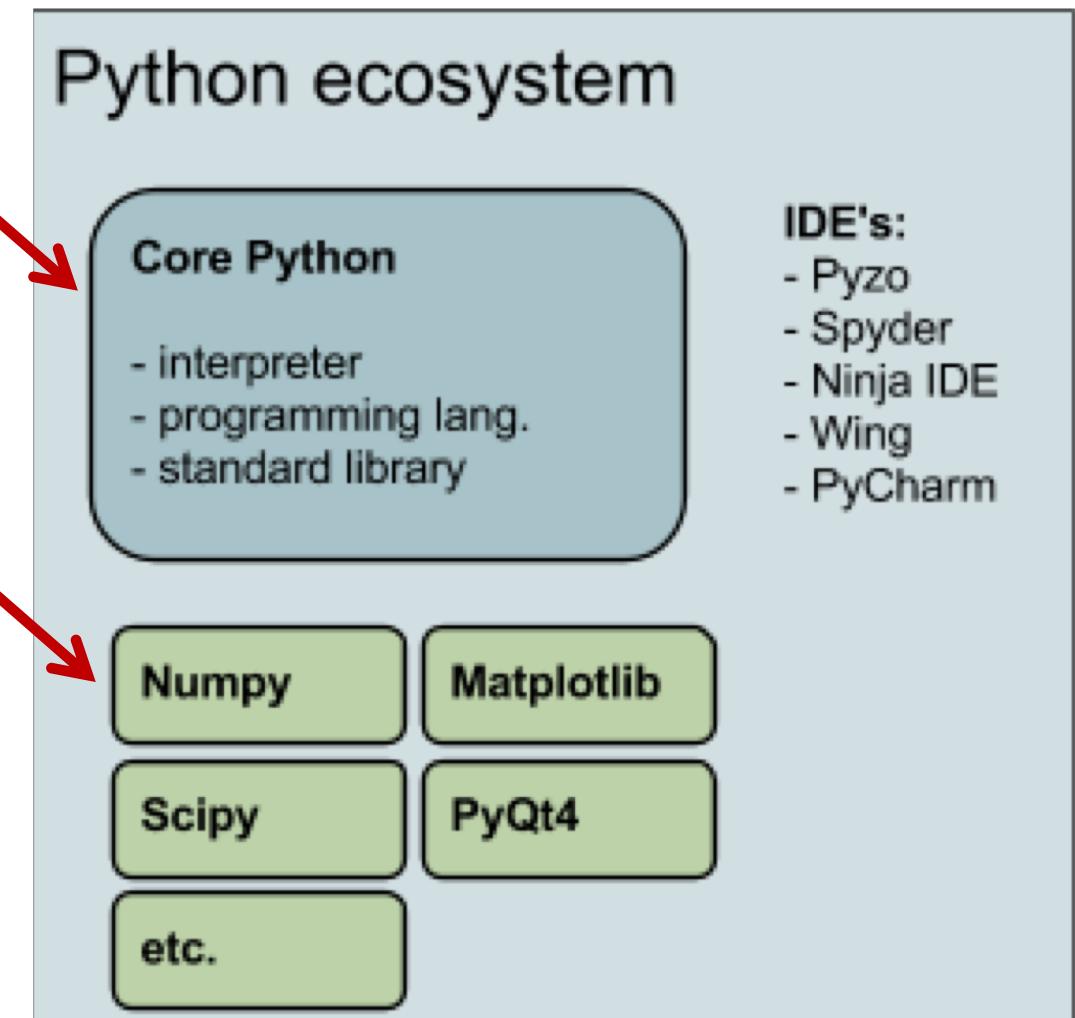
Numpy for MATLAB users (from python.org):

<https://docs.scipy.org/doc/numpy-dev/user/numpy-for-matlab-users.html>



Python structure

- Core Python (lightweight, fast)
- Numerical/scientific capabilities are imported as **modules**
- Most useful: **Numpy, Scipy, Matplotlib**
- IDEs = integrated development environments (like the MATLAB GUI); PyCharm, Spyder, etc.



Anaconda and conda

Anaconda is a free Python distribution that comes with **conda**, a package manager

- Lets you install packages and will check dependencies/versions, keeping [almost] everything up-to-date
- Also allows for different **environments**, so you can have working, parallel builds of Python 2.7 and 3.7

- Comes with a ton of libraries (Numpy, Scipy, Matplotlib, etc.) preinstalled, so most of the work is already done

- **Worth reading:** 30-minute “getting started with conda” tutorial at

<https://conda.io/docs/user-guide/getting-started.html>

- **See also:** `conda-cheatsheet.pdf` in the week1 directory

Anaconda and conda

Once you install an Anaconda Python distribution, check if it's working by typing `python` in a terminal shell:

```
python
```

Something like this should be returned:

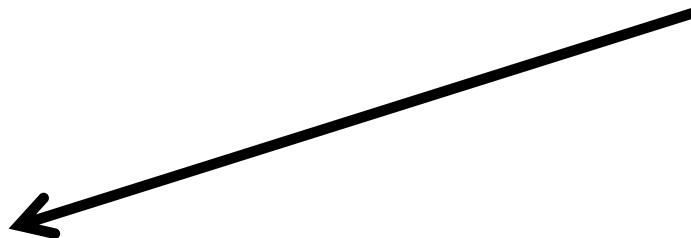
```
Python 3.6.4 |Anaconda custom (64-bit)|  
(default, Jan 16 2018, 12:04:33) [GCC 4.2.1  
Compatible Clang 4.0.1 (tags/RELEASE_401/final)]  
on darwinType "help", "copyright", "credits" or  
"license" for more information.
```

```
>>>
```

conda environments

```
baird$ conda info --envs  
#conda environments:  
#  
base * /Users/baird/anaconda  
basemap_stable /Users/baird/anaconda/envs/basemap_stable  
cdo_stable /Users/baird/anaconda/envs/cdo_stable  
ncl_stable /Users/baird/anaconda/envs/ncl_stable  
nco_stable /Users/baird/anaconda/envs/nco_stable  
py27 /Users/baird/anaconda/envs/py27  
r_stable /Users/baird/anaconda/envs/r_stable  
uvcdat_stable /Users/baird/anaconda/envs/uvcdat_stable
```

base = default
*** = active**



conda environments

```
baird$ conda info --envs
```

```
#conda environments:
```

```
#
```

```
base
```

```
basemap_stable
```

```
cdo_stable
```

```
ncl_stable
```

```
nco_stable
```

```
py27
```

```
r_stable
```

```
uvcdat_stable
```

to use another installed environment, type:

```
source activate nco_stable
```

- * /Users/baird/anaconda
- /Users/baird/anaconda/envs/basemap_stable
- /Users/baird/anaconda/envs/cdo_stable
- /Users/baird/anaconda/envs/ncl_stable
- /Users/baird/anaconda/envs/nco_stable**
- /Users/baird/anaconda/envs/py27
- /Users/baird/anaconda/envs/r_stable
- /Users/baird/anaconda/envs/uvcdat_stable



Anaconda and conda

To list all the Python libraries installed by the active Anaconda/Miniconda environment, type:

```
conda list
```

Anaconda and conda

NOTE: If you're using Mac/Linux, Anaconda supercedes the default Python on your computer by adding a PATH to bash_profile/bashrc

- If you want to revert to the default Python on your machine, comment out the red line below in your bash_profile/bashrc file (Mac):

```
# added by Anaconda 5.1.0 installer
export PATH="/Users/USER_NAME/anaconda/bin:$PATH"
```

IPython and Jupyter Notebook

IPython = Interactive Python

- Command line interface for interactive computing, useful for troubleshooting code
- Used to have a “notebook” option
- **ipython** at the command line

Jupyter Notebook

- What IPython Notebook is *now* called
 - Interactive computing docs (.ipynb files) run via web browser
 - Can be used for other languages, not just Python...
 - **jupyter notebook** at the command line

Jupyter Notebook

Navigate to the folder for this week's class materials,
where jupyter-intro.ipynb is located

Assuming Anaconda was successfully installed, Open Jupyter Notebook
by typing:

jupyter notebook