Where Python lives, and how to talk to it

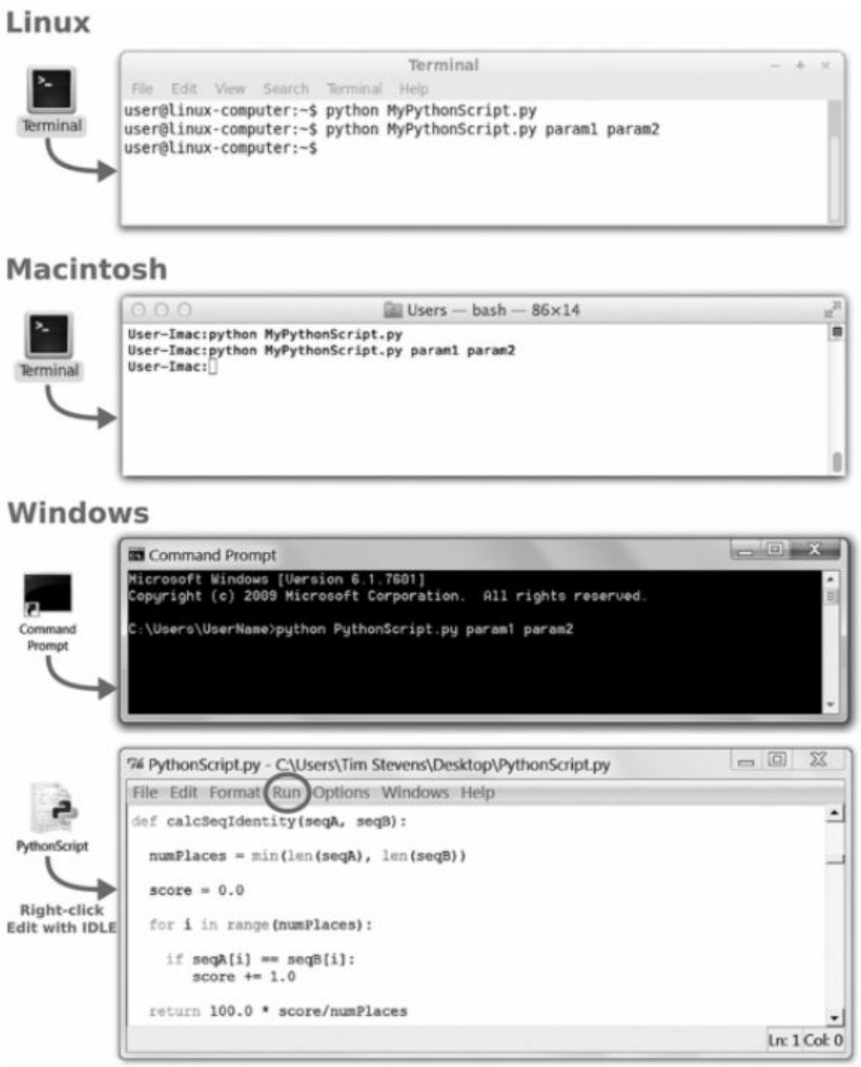
BILD 62

Objectives for today

- Identify various ways of writing and running Python code
- Introduce Jupyter Notebooks
- Learn basics of Python syntax

There are multiple ways to interact with the Python interpreter

- Command line
  - Line-by-line coding
  - Running “Scripts”



```
user@linux-computer:~$ python MyPythonScript.py
user@linux-computer:~$ python MyPythonScript.py param1 param2
user@linux-computer:~$
```

```
User-Imac:python MyPythonScript.py
User-Imac:python MyPythonScript.py param1 param2
User-Imac:
```

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\UserName>python PythonScript.py param1 param2
```

PythonScript.py - C:\Users\Tim Stevens\Desktop\PythonScript.py
File Edit Format Run Options Windows Help
PythonScript
Right-click
Edit with IDLE
Ln: 1 Col: 0

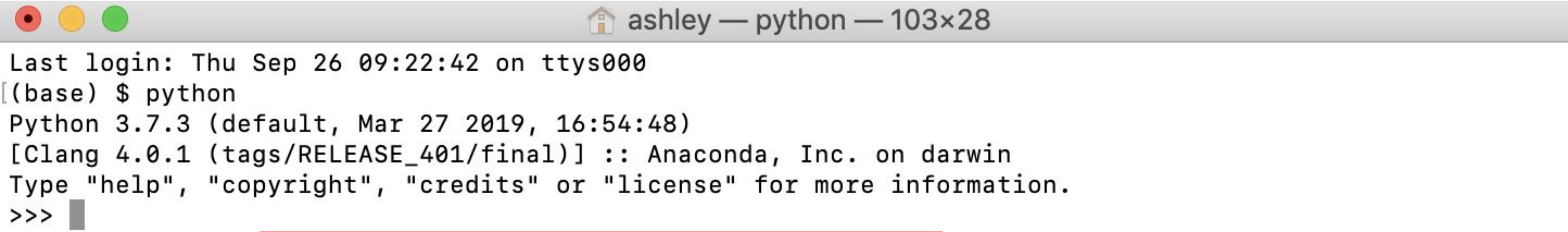
```
def calcSeqIdentity(seqA, seqB):

  numPlaces = min(len(seqA), len(seqB))

  score = 0.0

  for i in range(numPlaces):

    if seqA[i] == seqB[i]:
      score += 1.0

  return 100.0 * score/numPlaces
```

Running a Python script from different operating systems

(from http://www.cambridge.org/pythonforbiology)

If you have a Mac

- Macs ship with Python already installed.
- You can check which version by opening **Terminal** & typing
  `python --version`
  - **For this course, we'll be using Python 3.7** (or above).



ashley — python — 103×28

```
Last login: Thu Sep 26 09:22:42 on ttys000
(base) $ python
Python 3.7.3 (default, Mar 27 2019, 16:54:48)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

The ">>>" tells you you're inside the Python prompt, and the computer is ready for some code!

Useful Linux Commands

In Jupyter Notebook, add a `!` in front to use these. E.g., **`!pwd`**

Command	**Description**
`pwd`	Print working directory
`ls`	List contents
`cd`	Change directory
`cp`	Copy files from the current directory to a different directory
`mv`	Move or rename files
`mkdir`	Make a directory
`touch`	Create a blank file

More details: https://www.hostinger.com/tutorials/linux-commands
https://jakevdp.github.io/PythonDataScienceHandbook/01.05-ipython-and-shell-commands.html

There are multiple ways to interact with the Python interpreter

- Command line
  - Line-by-line coding
  - Running “Scripts”
- Integrated Development Environments
  - Folks have strong opinions about these, and each have pros/cons.
  - A few good options are:
    - Spyder (Included with Anaconda, the recommended install)
    - Visual Code (https://code.visualstudio.com/download)
- Jupyter Notebook *— most of what we’ll do in this course*

Integrated Development Environments (IDEs)

- Help you write, debug, and compile code
    - **Compiling** is the process of translating your **source code** into **machine code**
- Useful because they have features like **line numbers** and **syntax highlighting**, which colors your code based on the syntax.
- Often have auto-completion, memory for commands, and provide information about functions

Anaconda is an open-source distribution of Python, focused on scientific computing in Python.

Includes:

- "Conda," a package management tool
- Useful code packages
- A couple applications for editing & running code:
    - Spyder (Python IDE)
    - Jupyter Notebooks

A few notes

Macs have a native installation of Python.

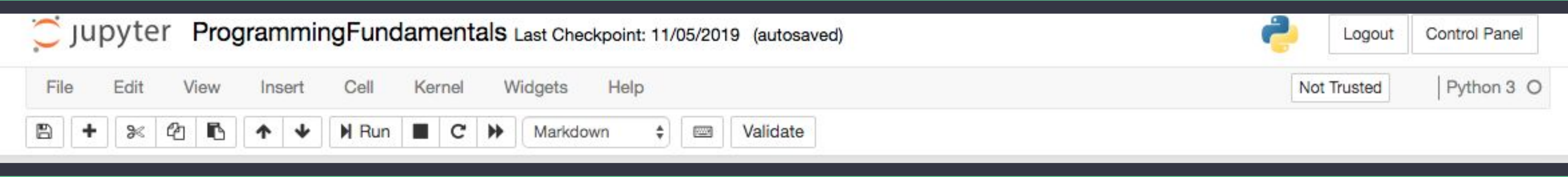
- It may be older & will not include the extra packages that you will need for this class, and is best left untouched.
- Downloading Anaconda will install a separate, independent install of Python, leaving your native install untouched.

Windows does not require Python natively and so it is not typically pre-installed.

If you're not sure which Python your computer is using, ask it (in Python):

```
>>> which python
```

Introduction to the UCSD DataHub & Jupyter Notebooks



About Jupyter Notebooks

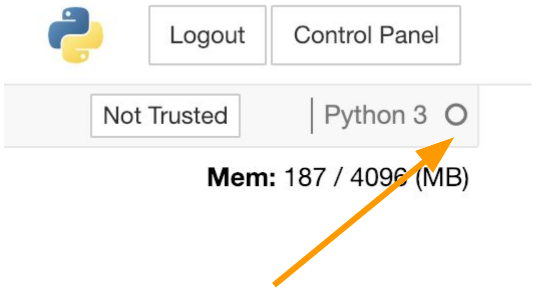
- Jupyter is a loose acronym for Julia, Python, and R
- Run in a web browser!
- Usefully, it will show plots directly in the notebook as you work your way through, performing analyses in real-time
- Two main components:
    - **Kernel**: the engine that runs the code
    - **Dashboard**: landing page where you can see the notebooks you’ve created

Using Jupyter Notebooks

- **Cell**: the main organizational structure of the notebook
    - Use **Shift+Enter** to run a cell (or press Run)
    - You can run cells out of order, and move cells around!
    - Cells can be **code** (the default) or **markdown** (descriptive text or images)
        - Code cells have `In[ ]:`
            - If there is a star (`In[]*:`), that means your cell is running
        - Change between code & markdown using dropdown menu (or keyboard shortcuts)
        - Turns **green** in edit mode

Using Jupyter Notebooks (continued)

- Processing-intensive cells will take > 10 seconds to run, but your code may also get stuck in a cell.
  - Interrupt a stuck cell using Kernel > Interrupt
- **If you change anything in the cell, you need to re-run it.**
- For help:
  - Help > User Interface tour
  - Help > Keyboard Shortcuts



You can tell if the kernel is busy by whether or not the circle next to Python 3 (upper right corner) is filled or not. (filled = busy)

In today’s Jupyter Notebook, we’ll do the following:

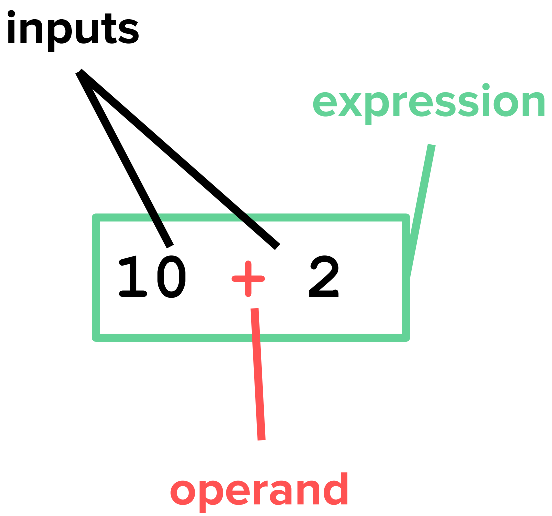
- Edit and run code and markdown cells in Jupyter Notebooks
- Use **basic arithmetic operations** in Python
- Assign **variables** and manipulate them
- Interpret basic errors while running Jupyter Notebooks
- Identify fundamental rules of Python syntax

Expressions describe how to combine pieces of data (e.g., add them!)

Symbol	Name	Sample Usage
=	Equal sign	Assign variable
#	Pound sign; hashtag	Line comments
[ ]	Brackets	Indexing & Slicing
( )	Parentheses	Using functions
{ }	Curly Brackets	Defining dictionary
' '	Single quotes	Creating string
" "	Double quotes	Creating string
_	Underscore	In variable names
!	Explanation point	To test not equal (!=)
\	Back slash	Delineate line break
:	Colon	Indexing

Basic arithmetic operators in Python

Symbol	Operation	Usage
+	Addition	`10+2`
–	Subtraction	`10-2`
*	Multiplication	`10*2`
/	Division	`10/2`
**	Exponent	`10**2`
%	Modulo	`10%2`



If you want a whole number (floor division), use // instead.

Basic arithmetic operators in Python

- The default order of operations is the same as in mathematics! (PEMDAS)
- Use parentheses to specify that you want an operation to happen first.

Storing values

We can store values in variables, e.g.:

```
variable_1 = 48
```

value

name

Variables can be text, integers, or floats (with decimals), e.g.:

```
text_string = “hello”
```

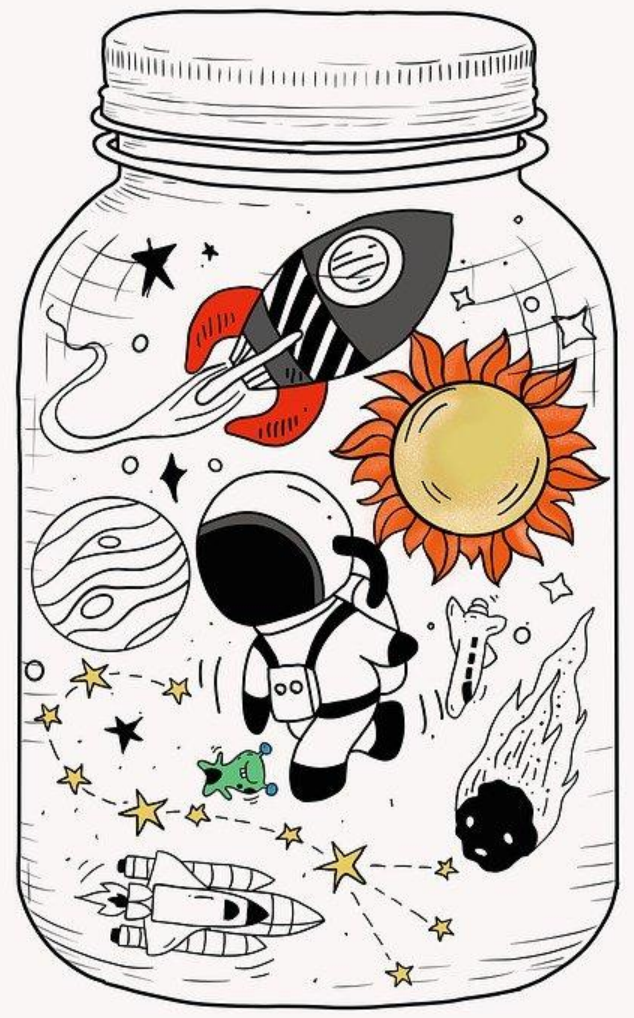
Storing values

We can store values in variables, e.g.:

```
variable_1 = 48
```

We use an equal sign to *assign* the value to a name, but it's not the same thing as saying they are equal.

In other words, we're storing that value in the variable. (Think of them like cookie jars)

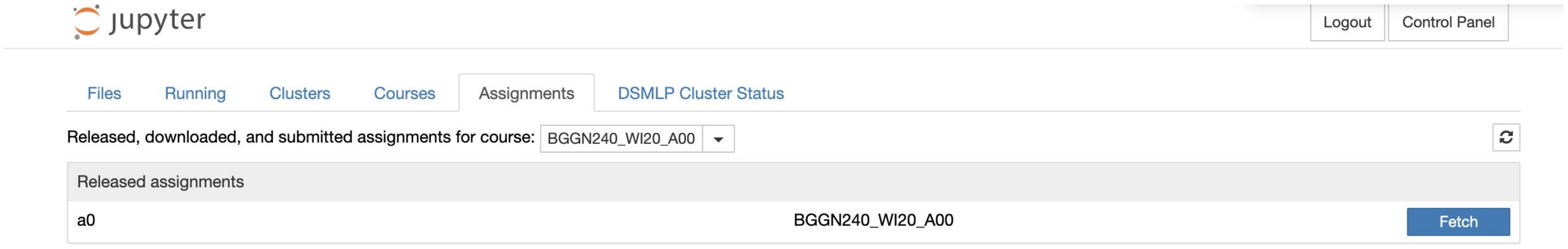


Submitting assignments

Done on the DataHub, due Fridays at 8am

Instructions to submit assignments

1. Log into the UCSD datahub by going to http://datahub.ucsd.edu and using your UCSD email & password to login.
2. Open the container for our course by choosing it and clicking **Launch Environment**. Note: You can use first container, without allen-brain-observatory.
3. Go to the Assignments tab, and look for our first assignment (a0) under "Released assignments":



Released, downloaded, and submitted assignments for course: BGGN240_WI20_A00

Released assignments

a0 BGGN240_WI20_A00

Instructions to submit assignments

4. Click the blue **Fetch** button.

5. Click on the assignment to open the Jupyter Notebook.

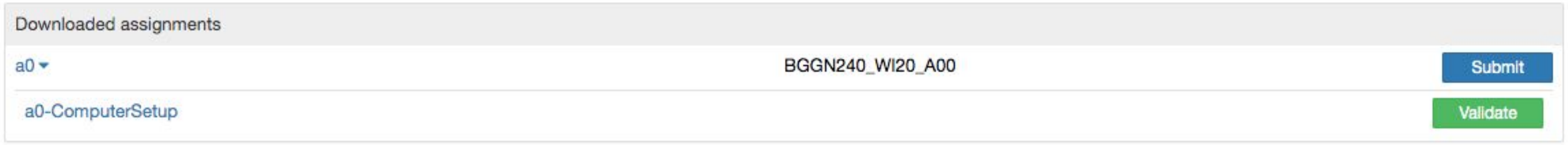
6. Follow the instructions within the notebook. For longer notebooks, you may want to save periodically (in addition to the autosaving Jupyter will do for you).

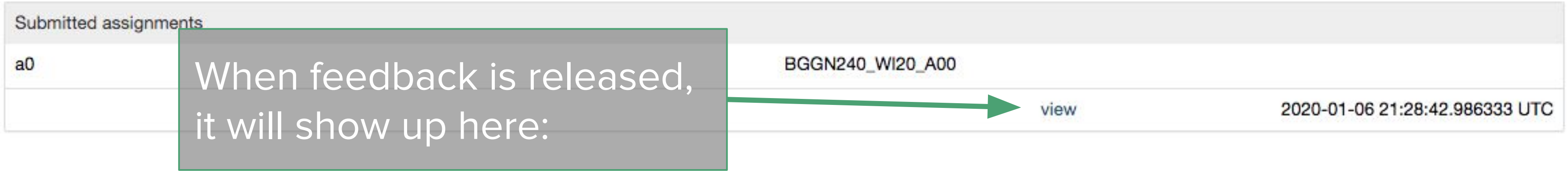
7. When you're done, save the notebook and close it.

Instructions to submit assignments

8. Click **Validate** to ensure you've passed all of the visible tests. (It will turn green once you've validated it).

9. Click **Submit** to submit your assignment. If you submit multiple times, your most recent submission will be graded.





When feedback is released, it will show up here:

Note: Assignment deadlines on Datahub are in UTC (and cannot be changed, annoyingly).

All assignments are due Friday at 8am.

Let’s get into a Jupyter notebook! Use the magic link to sync up your DataHub with our Materials folder, and open notebook 02.

Topics from today

- There are multiple ways to write and run Python code
- Writing and running markdown and code cells in Jupyter Notebook
- Basic rules of writing expressions and assigning variables in Python
- Python syntax rules:
    - Spaces and white space do not matter
    - Capitalization matters
    - Some words are protected
- Functions we learned: `slice` , `print`

Resources

Jupyter Notebooks:

- DataQuest "Learn and Install Jupyter Notebooks" (Note, parts of this require coding syntax you may not know yet)
- Official Jupyter documentation
- Example notebooks
- A Gallery of Interesting Jupyter Notebooks
- Software Carpentry: Running & Quitting Jupyter Notebooks