

Lecture 2 - HW1, numpy arrays, matplotlib, and git

2020.4.14

Review of Python Basics from Lecture 1

Topics from Lecture 1

Lists

Strings, ints, floats

Dicts

Objects, attributes, methods

(Tuples,)

Functions

{sets}

Importing .py files/libraries

slides, hws, solutions are posted here:
neuropython.com/calendar

Recorded zoom lectures will be accessible via courseworks (more fun in real time though)

HW1 Solutions

Part 1: Working with lists

```
[1]: a = [1,3,5,7,9,11,13,15]  
      b = [2,4,6,8,10,12,14,16]
```

- a) Create a list “c” that contains all of the elements from both a and b (many ways to do this) in any order.

```
[2]: # appending to empty list  
      c = []  
      c.append(a)  
      c.append(b)  
      print(c)
```

```
[[1, 3, 5, 7, 9, 11, 13, 15], [2, 4, 6, 8, 10, 12, 14, 16]]
```

```
[3]: # using extend instead of append
c = []
c.extend(a)
c.extend(b)
print(c)
```

```
[1, 3, 5, 7, 9, 11, 13, 15, 2, 4, 6, 8, 10, 12, 14, 16]
```

```
[4]: # Using two for loops
c = []
for element in a:
    c.append(element)
for element in b:
    c.append(element)

print('c: ',c)
```

```
c: [1, 3, 5, 7, 9, 11, 13, 15, 2, 4, 6, 8, 10, 12, 14, 16]
```

```
[3]: # using extend instead of append
c = []
c.extend(a)
c.extend(b)
print(c)
```

```
[1, 3, 5, 7, 9, 11, 13, 15, 2, 4, 6, 8, 10, 12, 14, 16]
```

```
[4]: # Using two for loops
c = []
for element in a:
    c.append(element)
for element in b:
    c.append(element)

print('c: ',c)
```

```
c: [1, 3, 5, 7, 9, 11, 13, 15, 2, 4, 6, 8, 10, 12, 14, 16]
```

```
[5]: # this is the simplest way
c = a+b
print('c: ',c)
```

```
c: [1, 3, 5, 7, 9, 11, 13, 15, 2, 4, 6, 8, 10, 12, 14, 16]
```

```
[1]: a = [1,3,5,7,9,11,13,15]
      b = [2,4,6,8,10,12,14,16]
```

b) Using slicing techniques, create a list "d" that contains all of the elements from a and b in numerical order

- HINT 1: You'll need to create an empty list "c" before you can assign elements to it.
- HINT 2: If you're going to assign values to this list using slicing, you'll want it to be full of the same number of [None] values as you ultimately expect to fill it with. Try `c = [None] * __ <- length you'll want to the list to eventually be.` Is there a way to find this value using `len()` rather than "hard-coding" it?

```
[6]: d = [0]*len(a+b)
      print('d: ',d)

      d[0::2] = a
      print('d: ',d)

      d[1::2] = b
      print('d: ',d)
```

```
d: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
d: [1, 0, 3, 0, 5, 0, 7, 0, 9, 0, 11, 0, 13, 0, 15, 0]
d: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
```

c) Using a for loop, create a list “e” that contains all of the elements of a and b in numerical order

-HINT 1: You'll need to create an empty list “c” before you can assign elements to it! -HINT 2: Are there any list-associated methods that would make this easier?

```
In [29]: # this uses zip and just a and b
```

```
d = []
for aa,bb in zip(a,b):
    d.append(aa)
    d.append(bb)

print(d)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
```

```
In [82]: # this uses extend using c
```

```
d = []

for i in range(8):
    d.extend(c[i::8])

print('d: ',d)
```

```
d:  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
```

d.) Reassign "e" in a new window by writing a list comprehension that performs the same task as your for loop

```
[7]: e = [c[i::8] for i in range(8)]
print('e: ',e)
e: [[1, 2], [3, 4], [5, 6], [7, 8], [9, 10], [11, 12], [13, 14], [15, 16]]
```

```
[8]: # this is messy and not clear
e = [m for i in range(8) for m in c[i::8] ]
e
```

```
[8]: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
```

d.) Reassign "e" in a new window by writing a list comprehension that performs the same task as your for loop

```
[7]: e = [c[i::8] for i in range(8)]
print('e: ',e)

e: [[1, 2], [3, 4], [5, 6], [7, 8], [9, 10], [11, 12], [13, 14], [15, 16]]
```

```
[8]: # this is messy and not clear
e = [m for i in range(8) for m in c[i::8] ]
e
```

```
[8]: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
```

```
[9]: # this is a nested for loop with list comprehension
e = [j for i in zip(a,b) for j in i]
e
```

```
[9]: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
```

```
[9]: # this is a nested for loop with list comprehension  
e = [j for i in zip(a,b) for j in i]  
e
```

```
[9]: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
```

```
[10]: # let's take a closer look  
for i in zip(a,b):  
    print(i)
```

```
(1, 2)  
(3, 4)  
(5, 6)  
(7, 8)  
(9, 10)  
(11, 12)  
(13, 14)  
(15, 16)
```

```
[11]: e = []  
for i in zip(a,b):  
    for j in i:  
        e.append(j)  
print('e: ',e)
```

```
e: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
```

e.) Using a while loop, create a list "f" that contains all of the elements of a and b in order

```
[12]: i = 0
f = []
while i < len(a):
    f.append(a[i])
    f.append(b[i])
    i += 1
f
```

```
[12]: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
```

Part 2: Create a dict filled with 5 arbitrary username (containing any numbers/lowercase letters you wish)/passwords as key/value pairs.

```
[14]: passwords = {}
passwords['joleen'] = 'covid19'
passwords['shmuel'] = 'COVID'
passwords['rita'] = 'c0r0navirus'
passwords['jaime'] = 'quarantine'
passwords['martha'] = 'sars-cov-2'

print(passwords)

{'joleen': 'covid19', 'shmuel': 'COVID', 'rita': 'c0r0navirus', 'jaime': 'quarantine', 'martha': 'sars-cov-2'}
```

a.) Write a well-documented function called passwordEval that takes in a username and password as input, prints "Username and password accepted!" if the username/password combo are in the dict, and prints "Username/password not found!" if they are not.

- HINT 1: When we say well-documented, we refer to both creating a docstring for the function AND commenting on your code. Comments in code can be created by typing "#" in front of whatever you don't want evaluated – just keep in mind that everything following the "#" on the same line will also be commented out

```
[15]: def passwordEval(username,password):  
    """Checks if username/password are in the dict"""  
  
    # assumes the dict passwords is accessible outside of function  
    if passwords[username] == password:  
        print('Username and password accepted!')  
  
    else:  
        print('Username/password not found!')
```

```
[16]: passwordEval('joleen','covid19')  
passwordEval('joleen','c0r0navirus')  
passwordEval('jaime','quarantine')
```

Username and password accepted!
Username/password not found!
Username and password accepted!

b) We specified that we only want users to have passwords with lowercase letters. How can we always make sure that users are entering only lowercase?

One method is to assert. We didn't learn this in class, but it's pretty straightforward. The format is assert (CONDITION), 'STRING TO PRINT IF FALSE' Using the str.islower() method, add an assertion that usernames must be lowercase into your function. Now test your function to see what happens if you accidentally type a username containing a capital letter!

```
[17]: def passwordEval(username,password):
    """Checks if username/password are in the dict"""

    assert password.islower(),'password must be lower case'

    # assumes the dict passwords is accessible outside of function
    if passwords[username] == password:
        print('Username and password accepted!')

    else:
        print('Username/password not found!')
```

```
[18]: passwordEval('joleen','Covid19')
```

```
-----  
AssertionError                                                 Traceback (most recent call last)  
<ipython-input-18-915521a1117d> in <module>  
----> 1 passwordEval('joleen','Covid19')  
  
<ipython-input-17-fa38a707e56b> in passwordEval(username, password)  
      2     """Checks if username/password are in the dict"""  
      3  
----> 4     assert password.islower(),'password must be lower case'  
      5  
      6     # assumes the dict passwords is accessible outside of function  
  
AssertionError: password must be lower case
```

```
[19]: passwordEval('joleen','covid19')
```

Username and password accepted!

c.) In b, we used an assertion to make sure users enter correct input. Is there a way to do this that takes the burden off the user? Remove your assertion, and instead use the `str.lower()` method to correct for this sort of error. Test it out.

[20]:

```
def passwordEval(username,password):
    """Checks if username/password are in the dict"""

    # make password lowercase
    password = password.lower()
    print(password)

    # assumes the dict passwords is accessible outside of function
    if passwords[username] == password:
        print('Username and password accepted!')

    else:
        print('Username/password not found!')
```

[21]:

```
passwordEval('joleen','Covid19')
```

```
covid19
```

```
Username and password accepted!
```

e.) Create a new dictionary that contains all of the username/password pairs you created in your first dict (is there a Pythonic way to do this instead of retyping it all?), plus 4 new pairs. Write a for loop that runs your passwordEval function on the new dictionary!

- HINT1: How do we iterate through a dictionary? Don't forget about dict.keys()!

```
[1]: import week1 as wk
```

```
[2]: wk.passwordEval('joleen','covid19')
```

```
covid19
```

```
NameError
```

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

```
<ipython-input-2-71c60d9a74d5> in <module>
----> 1 wk.passwordEval('joleen','covid19')
```

```
~/Documents/Code/PythonDataCourse/Homeworks/HW1-jacob/week1.py in passwordEval(username, password)
```

```
 9
```

```
10      # assumes the dict passwords is accessible outside of function
```

```
---> 11      if passwords[username] == password:
```

```
12          print('Username and password accepted!')
```

```
13
```

```
NameError: name 'passwords' is not defined
```

```
[15]: # note that in this function, we are passing in the full dictionary of passwords
def passwordEval(password_dict,username,password):
    """Checks if username/password are in the dict"""

    # make password lowercase
    password = password.lower()

    # assumes the dict passwords is accessible outside of function
    if password_dict[username] == password:
        print('Username and password accepted!')

    else:
        print('Username/password not found!')
```

```
[1]: import week1 as wk
```

```
passwords = {}
passwords['joleen'] = 'covid19'
passwords['shmuel'] = 'COVID'
passwords['rita'] = 'c0r0navirus'
passwords['jaime'] = 'quarantine'
passwords['martha'] = 'sars-cov-2'
```

```
[2]: wk.passwordEval(passwords, 'joleen', 'covid19')
```

Username and password accepted!

```
[3]: password_dict2 = passwords

password_dict2['Moses'] = 'exodus2020'
password_dict2['Jesus'] = 'anastasis20'
password_dict2['moses'] = 'exodus2020'
password_dict2['jesus'] = 'anastasis20'

password_dict2
```

```
[3]: {'joleen': 'covid19',
      'shmuel': 'COVID',
      'rita': 'c0r0navirus',
      'jaime': 'quarantine',
      'martha': 'sars-cov-2',
      'Moses': 'exodus2020',
      'Jesus': 'anastasis20',
      'moses': 'exodus2020',
      'jesus': 'anastasis20'}
```

```
[5]: for key,item in password_dict2.items():
    print('key: '+key+'\nvalue: '+item)
    wk.passwordEval(passwords,key,item)
```

```
key: joleen
value: covid19
Username and password accepted!
key: shmuel
value: COVID
Username/password not found!
key: rita
value: c0r0navirus
Username and password accepted!
key: jaime
value: quarantine
Username and password accepted!
key: martha
value: sars-cov-2
Username and password accepted!
key: Moses
value: exodus2020
Username and password accepted!
key: Jesus
value: anastasis20
Username and password accepted!
key: moses
value: exodus2020
Username and password accepted!
key: jesus
value: anastasis20
Username and password accepted!
```

In [5]: passwords

Out[5]: {'joleen': 'covid19',
 'shmuel': 'COVID',
 'rita': 'c0r0navirus',
 'jaime': 'quarantine',
 'martha': 'sars-cov-2',
 'Moses': 'exodus2020',
 'Jesus': 'anastasis20',
 'moses': 'exodus2020',
 'jesus': 'anastasis20'}

```
In [9]: import week1 as wk  
from copy import copy
```

```
passwords = {}  
passwords['joleen'] = 'covid19'  
passwords['shmuel'] = 'COVID'  
passwords['rita'] = 'c0r0navirus'  
passwords['jaime'] = 'quarantine'  
passwords['martha'] = 'sars-cov-2'  
  
password_dict2 = copy(passwords)  
  
password_dict2['Moses'] = 'exodus2020'  
password_dict2['Jesus'] = 'anastasis20'  
password_dict2['moses'] = 'exodus2020'  
password_dict2['jesus'] = 'anastasis20'  
  
passwords
```

```
Out[9]: {'joleen': 'covid19',  
         'shmuel': 'COVID',  
         'rita': 'c0r0navirus',  
         'jaime': 'quarantine',  
         'martha': 'sars-cov-2'}
```

In [10]: password_dict2

Out[10]: {'joleen': 'covid19',
 'shmuel': 'COVID',
 'rita': 'c0r0navirus',
 'jaime': 'quarantine',
 'martha': 'sars-cov-2',
 'Moses': 'exodus2020',
 'Jesus': 'anastasis20',
 'moses': 'exodus2020',
 'jesus': 'anastasis20'}

```
In [11]: for key,item in password_dict2.items():
    print('key: '+key+'\nvalue: '+item)
    wk.passwordEval(passwords,key,item)
```

```
key: joleen
value: covid19
Username and password accepted!
key: shmuel
value: COVID
Username/password not found!
key: rita
value: c0r0navirus
Username and password accepted!
key: jaime
value: quarantine
Username and password accepted!
key: martha
value: sars-cov-2
Username and password accepted!
key: Moses
value: exodus2020
```

```
KeyError                                                 Traceback (most recent call last)
<ipython-input-11-d8c9991a04c0> in <module>
      1 for key,item in password_dict2.items():
      2     print('key: '+key+'\nvalue: '+item)
----> 3     wk.passwordEval(passwords,key,item)

~/Documents/Code/PythonDataCourse/Homeworks/HW1-jacob/week1.py in passwordEval(password_dict, username, password)
      8
      9     # assumes the dict passwords is accessible outside of function
----> 10    if password_dict[username] == password:
     11         print('Username and password accepted!')
     12

KeyError: 'Moses'
```

```
def passwordEval(password_dict,username,password):
    """Checks if username/password are in the dict"""

    # make password lowercase
    password = password.lower()

    # assumes the dict passwords is accessible outside of function
    if username in password_dict.keys(): # check if key is in dict first
        if password_dict[username] == password:
            print('Username and password accepted!')

    else:
        print('Username/password not found!')
```

```
In [14]: for key,item in password_dict2.items():
    print('key: '+key+'\nvalue: '+item)
    passwordEval(passwords,key,item)
```

```
key: joleen
value: covid19
Username and password accepted!
key: shmuel
value: COVID
key: rita
value: c0r0navirus
Username and password accepted!
key: jaime
value: quarantine
Username and password accepted!
key: martha
value: sars-cov-2
Username and password accepted!
key: Moses
value: exodus2020
Username/password not found!
key: Jesus
value: anastasis20
Username/password not found!
key: moses
value: exodus2020
Username/password not found!
key: jesus
value: anastasis20
Username/password not found!
```

Numpy Arrays (see slides)

Plotting with Matplotlib (see slides)

%matplotlib inline

- sends png to browser
- no panning or zooming
- new figure for each cell
- no changes to previous figures

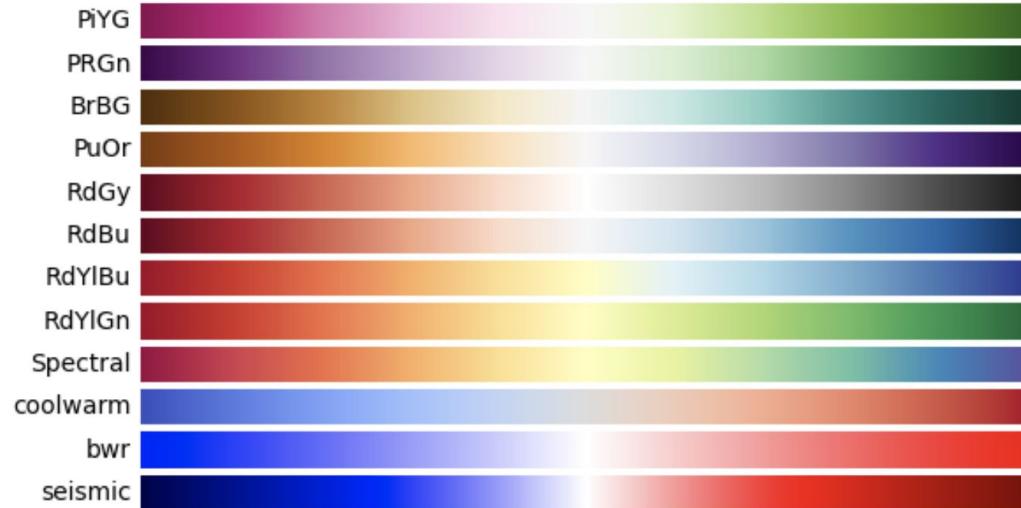
%matplotlib notebook

- interactivity in old notebook
- doesn't work in jupyter lab
- need to create separate figures
- ability to update figures

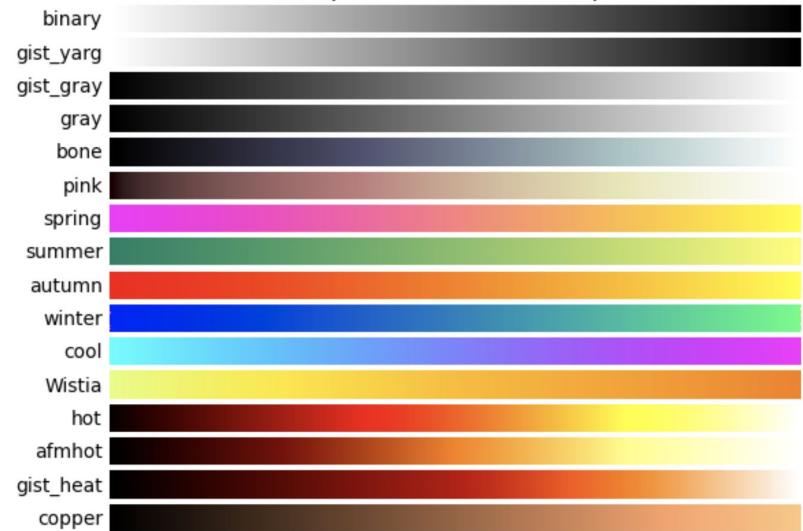
%matplotlib widget

- interactive widget
- all figure features
- need to create separate figures explicitly
- ability to update figures
- installation instructions: <https://github.com/matplotlib/jupyter-matplotlib>

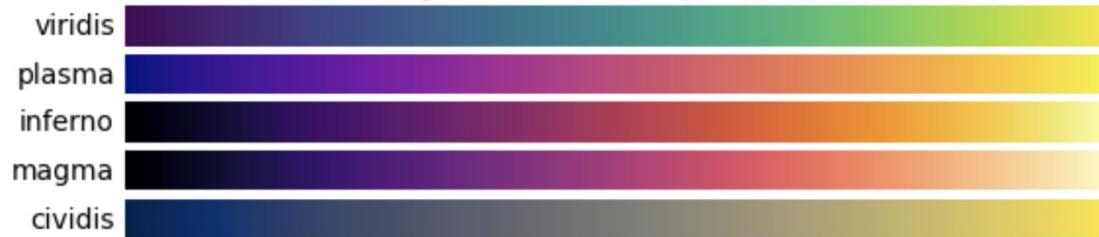
Diverging colormaps



Sequential (2) colormaps



Perceptually Uniform Sequential colormaps

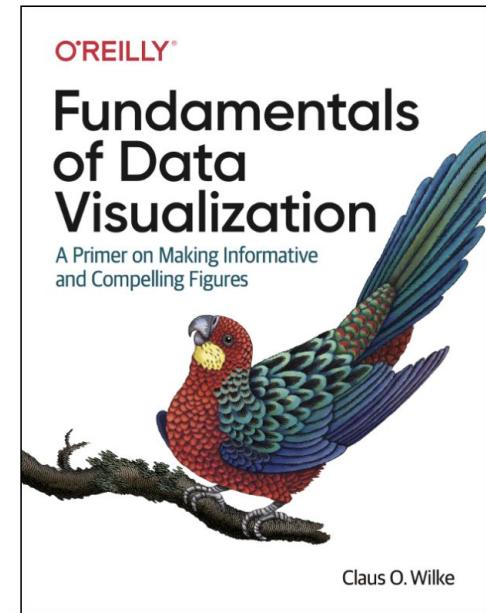
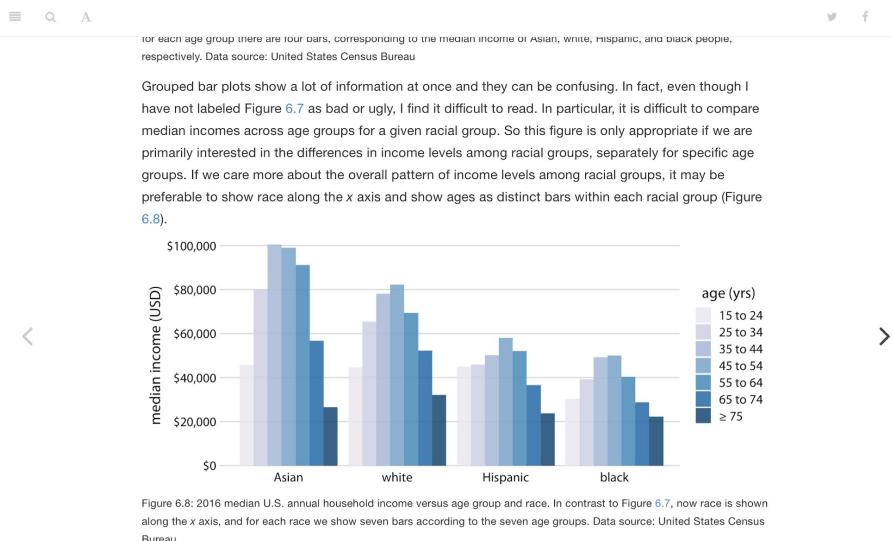


Matplotlib Resources

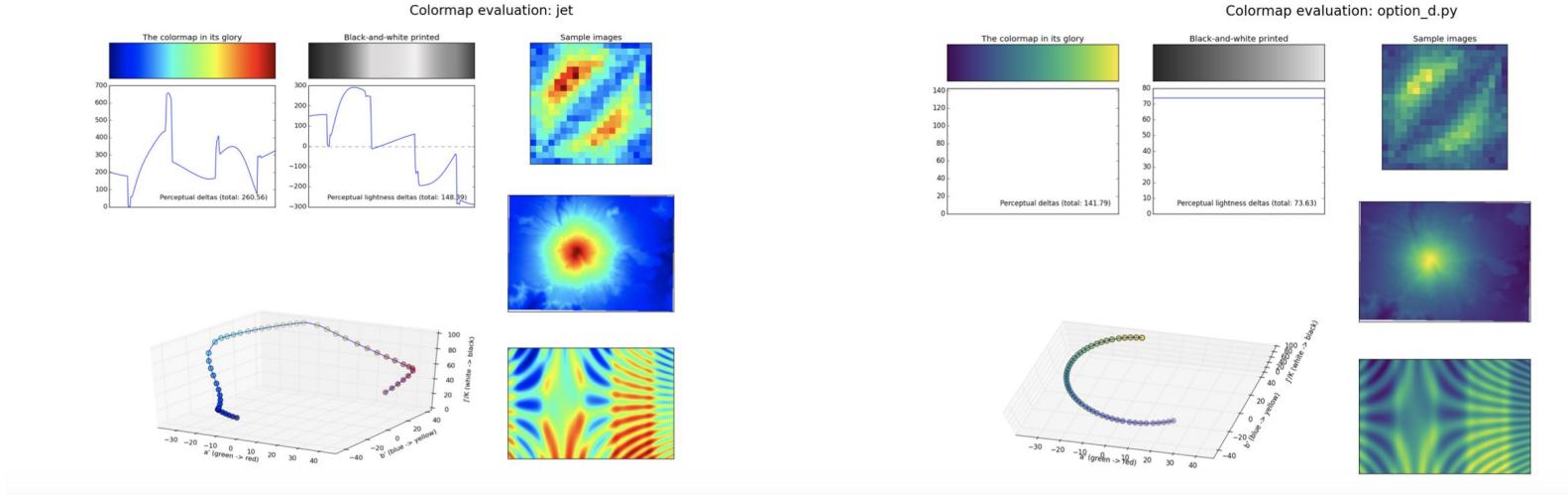
- Matplotlib Tutorials: <https://matplotlib.org/tutorials/>
- Matplotlib Gallery: <http://matplotlib.org/gallery.html>
- Plotting commands summary:
http://matplotlib.org/api/pyplot_summary.html

Fun Resource: <https://serialmentor.com/dataviz/>

Welcome
Preface
1 Introduction
Part I: From data to visualization
2 Visualizing data: Mapping data onto ...
3 Coordinate systems and axes
4 Color scales
5 Directory of visualizations
6 Visualizing amounts
6.1 Bar plots
6.2 Grouped and stacked bars
6.3 Dot plots and heatmaps
7 Visualizing distributions: Histograms ...
8 Visualizing distributions: Empirical cu...
9 Visualizing many distributions at once
10 Visualizing proportions
11 Visualizing nested proportions
12 Visualizing associations among two ...



Use perceptually uniform colormaps



Don't cut the vertical axis

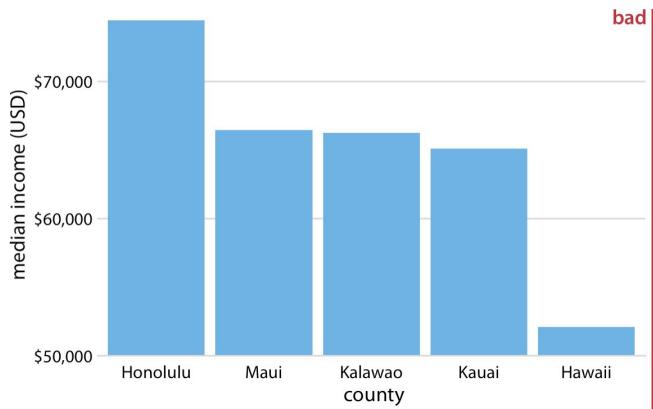


Figure 17.1: Median income in the five counties of the state of Hawaii. This figure is misleading, because the y axis scale starts at \$50,000 instead of \$0. As a result, the bar heights are not proportional to the values shown, and the income differential between the county of Hawaii and the other four counties appears much bigger than it actually is. Data source: 2015 Five-Year American Community Survey.

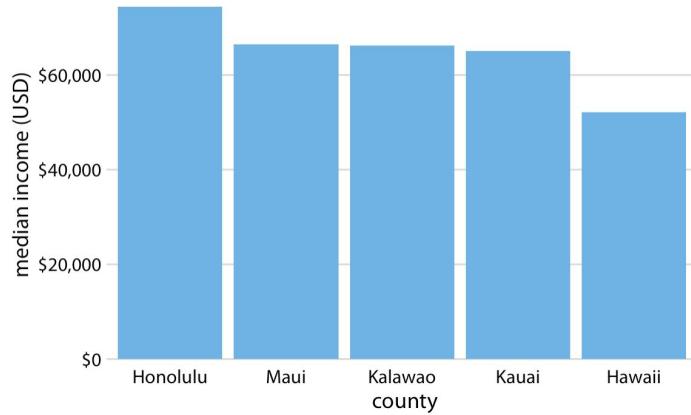
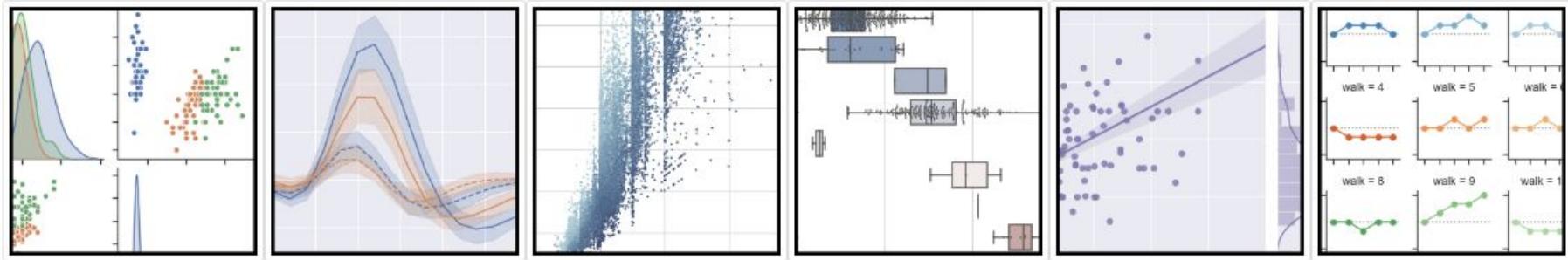


Figure 17.2: Median income in the five counties of the state of Hawaii. Here, the y axis scale starts at \$0 and therefore the relative magnitudes of the median incomes in the five counties are accurately shown. Data source: 2015 Five-Year American Community Survey.

seaborn: statistical data visualization



<https://seaborn.pydata.org/tutorial.html>



Break (5 mins)

Columbia-Neuropythonistas / PythonDataCourse

 Watch ▾ 1

 Star 3

 Fork 3

forked from gucky92/PythonDataCourse

 Code

 Pull requests 0

 Actions

 Projects 0

 Wiki

 Security

 Insights

 Settings

Branch: master ▾

[PythonDataCourse](#) / [Lectures](#) / [Lecture2](#) /

[Create new file](#)

[Upload files](#)

[Find file](#)

[History](#)

This branch is 10 commits ahead of gucky92:master.

 Pull request  Compare

 jacobfulano matplotlib lecture

 Latest commit 79aec7e 13 hours ago

..

 .DS_Store matplotlib lecture 13 hours ago

 lecture2-matplotlib.ipynb matplotlib lecture 13 hours ago

 lecture2-matplotlib.slides.html matplotlib lecture 13 hours ago

 lecture2-numpy-arrays.ipynb lecture2 17 hours ago

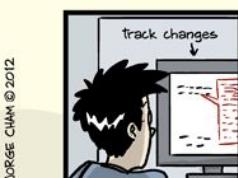
 lecture2-slides-matplotlib.pdf matplotlib lecture 13 hours ago

 lecture2-slides-numpy-arrays.html added html file for numpy slides 17 hours ago

 lecture2-slides-numpy-arrays.pdf added html file for numpy slides 17 hours ago

Version Control with `git`

"FINAL".doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc



FINAL_rev.22.comments49.
corrections.10.#@\$%WHYDID
ICOMETOGRAD SCHOOL?????.doc

Why version control?

Nothing that is committed to version control is ever lost

We have this record of who made what changes when

Version control is the **lab notebook of the digital world**: it's what professionals use to keep track of what they've done and to collaborate with other people.

Every large software development project relies on it, and most programmers use it for their small jobs as well.

And it isn't just for software: books, papers, small data sets, and anything that changes over time or needs to be shared can and should be stored in a version control system

Why git specifically?

It is fast

Work offline - decentralized, everything is on your local machine

Can undo mistakes - almost every action in Git only adds data

The cool kids are doing it



THIS IS GIT. IT TRACKS COLLABORATIVE WORK
ON PROJECTS THROUGH A BEAUTIFUL
DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZIZE THESE SHELL
COMMANDS AND TYPE THEM TO SYNC UP.
IF YOU GET ERRORS, SAVE YOUR WORK
ELSEWHERE, DELETE THE PROJECT,
AND DOWNLOAD A FRESH COPY.

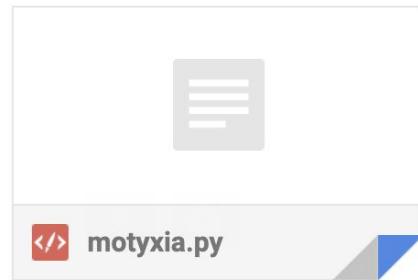


NEVER EMAIL CODE ➤



Jacob Portes <j.portes@columbia.edu>
to Jessica ▾

🕒 Mon, Nov 25, 2019, 4:38 PM



◀ Reply

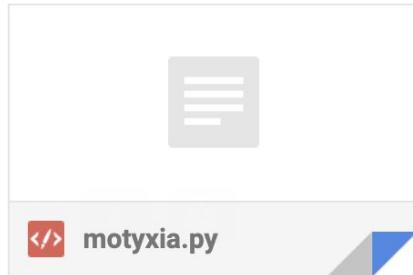
▶ Forward

NEVER EMAIL THE SAME CODE TWICE ➔



Jacob Portes <j.portes@columbia.edu>
to Jessica ▾

🕒 Mon, Nov 25, 2019, 4:54 PM



◀ Reply

▶ Forward

Do not copy same code into every data folder



File/Folder	Last Modified	Type	Size	Content Type
jk169aTm1OctRNAi_060617	Today at 10:59 AM	--		Folder
All Analysis Files	Oct 24, 2019 at 3:22 PM	--		Folder
align_function.m	Mar 26, 2017 at 9:51 PM	642 bytes		Objective-C
back_out_1dfilter_new2.m	Feb 3, 2014 at 6:44 PM	1 KB		Objective-C
back_out_1dfilter.m	Mar 21, 2017 at 12:59 AM	2 KB		Objective-C
FFF_align.m	May 7, 2017 at 7:17 PM	1 KB		Objective-C
IBWread.m	Mar 21, 2017 at 12:59 AM	7 KB		Objective-C
init_read.m	May 18, 2017 at 2:29 PM	8 KB		Objective-C
mean_wave.m	May 21, 2017 at 10:22 PM	584 bytes		Objective-C
read_contrast.m	Mar 21, 2017 at 12:59 AM	4 KB		Objective-C
read_data.m	Jun 1, 2016 at 4:57 PM	9 KB		Objective-C
readIBWbinheader.m	Mar 21, 2017 at 12:59 AM	3 KB		Objective-C
readIBWheaders.m	Mar 21, 2017 at 12:59 AM	14 KB		Objective-C
sto_analysis.m	May 18, 2017 at 2:33 PM	586 bytes		Objective-C
wave_length_align.m	May 7, 2017 at 7:32 PM	262 bytes		Objective-C
ch0_1.ibw	Jun 6, 2017 at 12:27 PM	1.7 MB		Igor Pr...ave File
ch0_2.ibw	Jun 6, 2017 at 12:27 PM	2.8 MB		Igor Pr...ave File
ch0_3.ibw	Jun 6, 2017 at 12:27 PM	4.1 MB		Igor Pr...ave File
ch0_4.ibw	Jun 6, 2017 at 12:27 PM	2.8 MB		Igor Pr...ave File
ch0_5.ibw	Jun 6, 2017 at 12:27 PM	2.8 MB		Igor Pr...ave File
ch0_6.ibw	Jun 6, 2017 at 12:27 PM	2.8 MB		Igor Pr...ave File
ch0_7.ibw	Jun 6, 2017 at 12:27 PM	4.1 MB		Igor Pr...ave File
ch0_8.ibw	Jun 6, 2017 at 12:27 PM	4.1 MB		Igor Pr...ave File
ch0_9.ibw	Jun 6, 2017 at 12:27 PM	2.8 MB		Igor Pr...ave File
ch0_10.ibw	Jun 6, 2017 at 12:27 PM	4.4 MB		Igor Pr...ave File
ch0_11.ibw	Jun 6, 2017 at 12:27 PM	4.4 MB		Igor Pr...ave File
ch0_12.ibw	Jun 6, 2017 at 12:27 PM	4.4 MB		Igor Pr...ave File
ch0_13.ibw	Jun 6, 2017 at 12:27 PM	4.4 MB		Igor Pr...ave File
ch0_14.ibw	Jun 6, 2017 at 12:27 PM	3 MB		Igor Pr...ave File
ch0_15.ibw	Jun 6, 2017 at 12:27 PM	4 MB		Igor Pr...ave File
ch0_16.ibw	Jun 6, 2017 at 12:27 PM	1.2 MB		Igor Pr...ave File



paper_v0

We seem to share a simultaneously naive but nonetheless tenable belief that we can “understand” or at least “interpret” how each cell-type processes information in the fruit fly optic lobe.

paper_v1

...

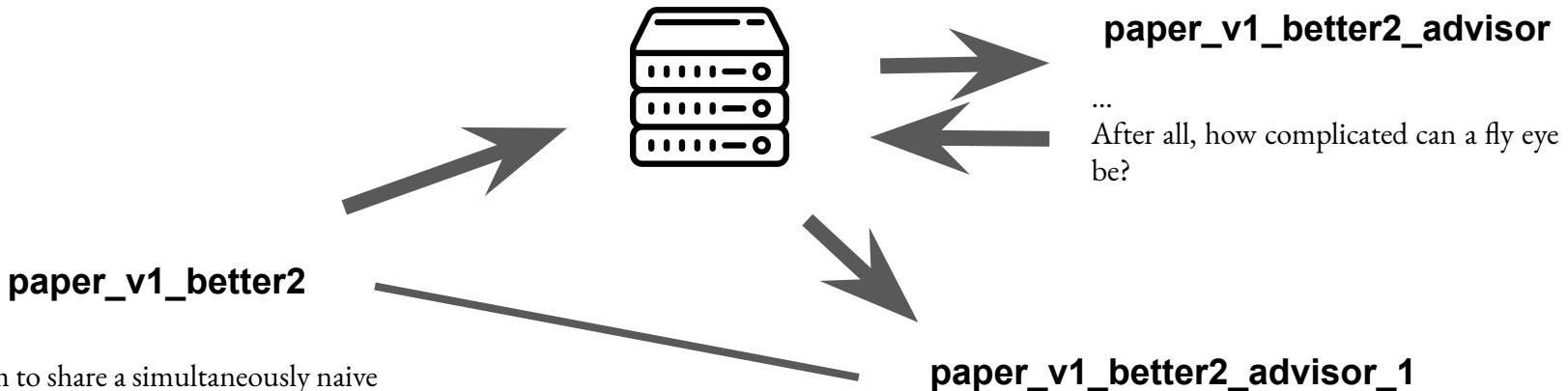
Experimental and theoretical methods and frameworks that have been discarded by vertebrate researchers are still treated with obeisance;

paper_v1_aggressive

...
We are slaves to the myopic methods of the previous generation.

paper_v1_better

...
moving bars do a poor job of revealing the mysteries of V1, but they *might* just work for flies; separation of RGCs into binary ON and OFF cells is equivocal at best, but the the fly eye *might just be that simple*.



We seem to share a simultaneously naive but nonetheless tenable belief that we can “understand” or at least “interpret” how each cell-type processes information in the fruit fly optic lobe. Experimental and theoretical methods and frameworks that have been discarded by vertebrate researchers are still treated with obeisance; moving bars do a poor job of revealing the mysteries of V1, but they *might just* work for flies; separation of RGCs into binary ON and OFF cells is equivocal at best, but the the fly eye *might just be that simple*. The allure of an easily interpretable framework is too great to resist.

paper_v1_better2_advisor
...
After all, how complicated can a fly eye be?

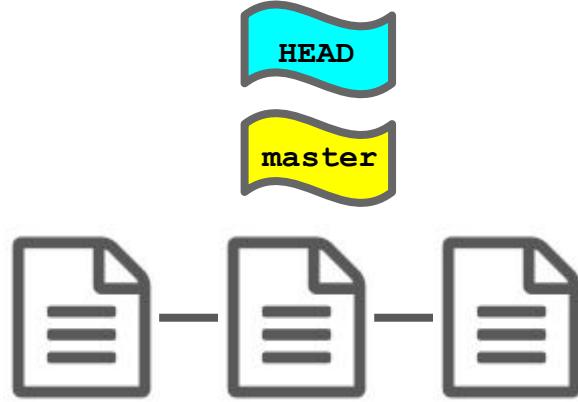
paper_v1_better2_advisor_1

We seem to share a simultaneously naive but nonetheless tenable belief that we can “understand” or at least “interpret” how each cell-type processes information in the fruit fly optic lobe. Experimental and theoretical methods and frameworks that have been discarded by vertebrate researchers are still treated with obeisance; moving bars do a poor job of revealing the mysteries of V1, but they *might just* work for flies; separation of RGCs into binary ON and OFF cells is equivocal at best, but the the fly eye *might just be that simple*. The allure of an easily interpretable framework is too great to resist. After all, how complicated can a fly eye be?



```
git add
```

```
git commit
```



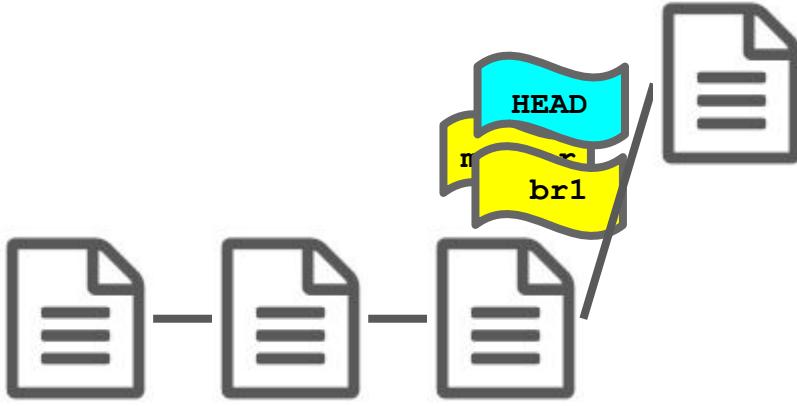
```
git add
```

```
git commit
```



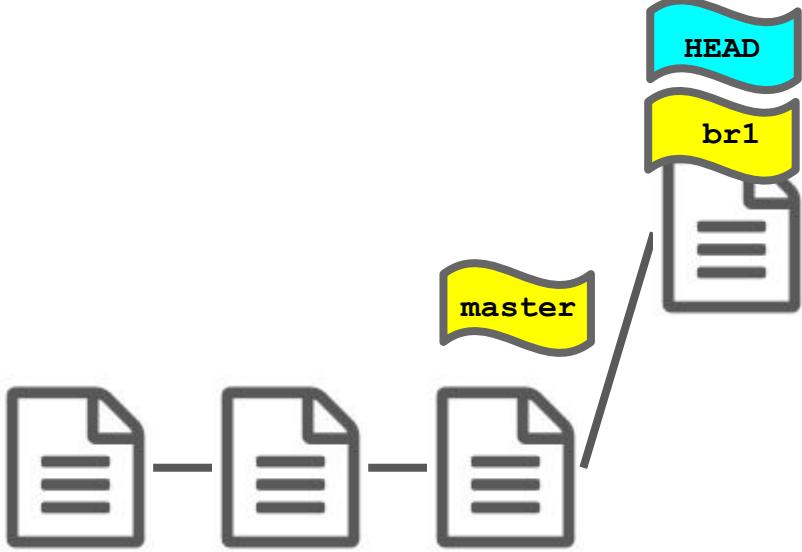
```
git branch br1
```

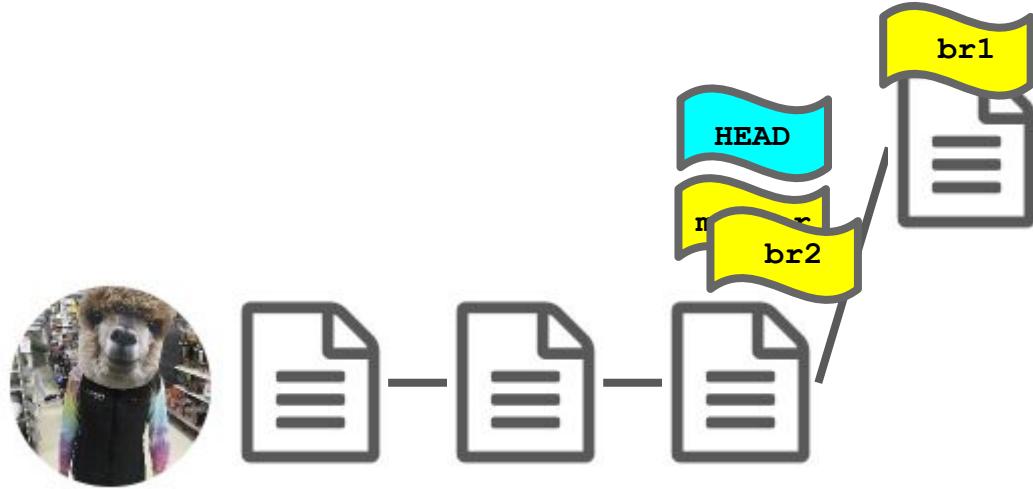
```
git checkout br1
```



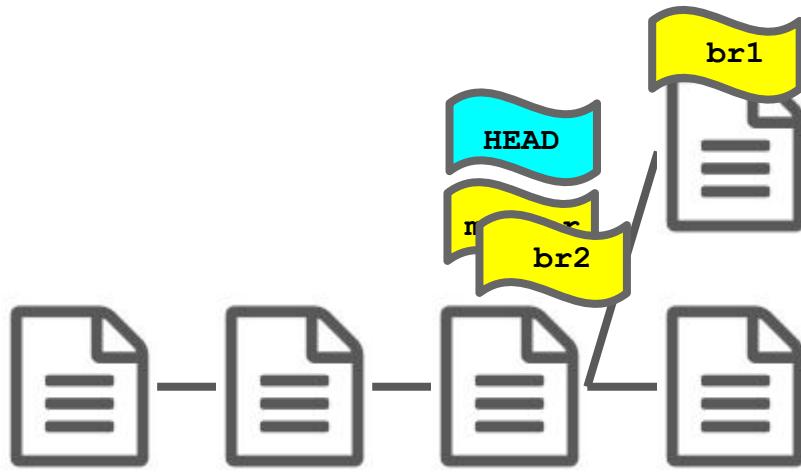
```
git add
```

```
git commit
```



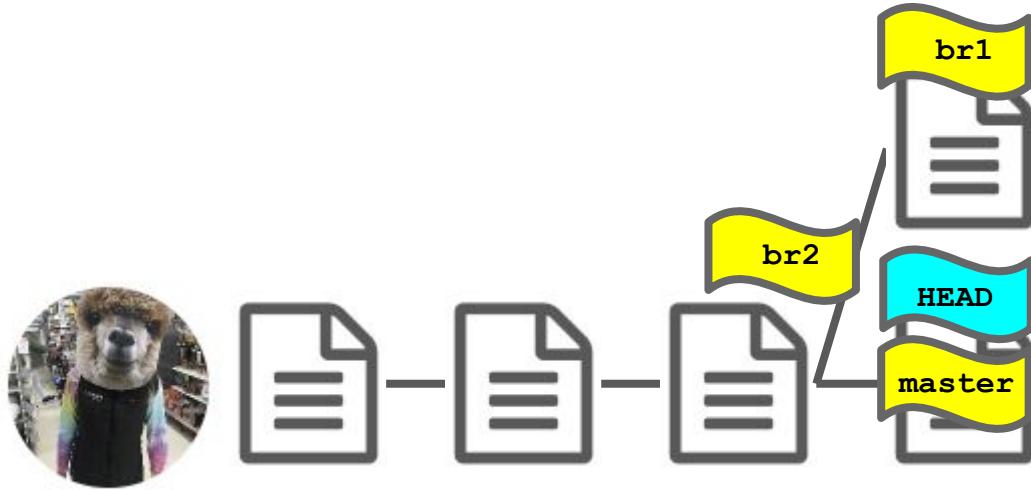


```
git branch br2
```

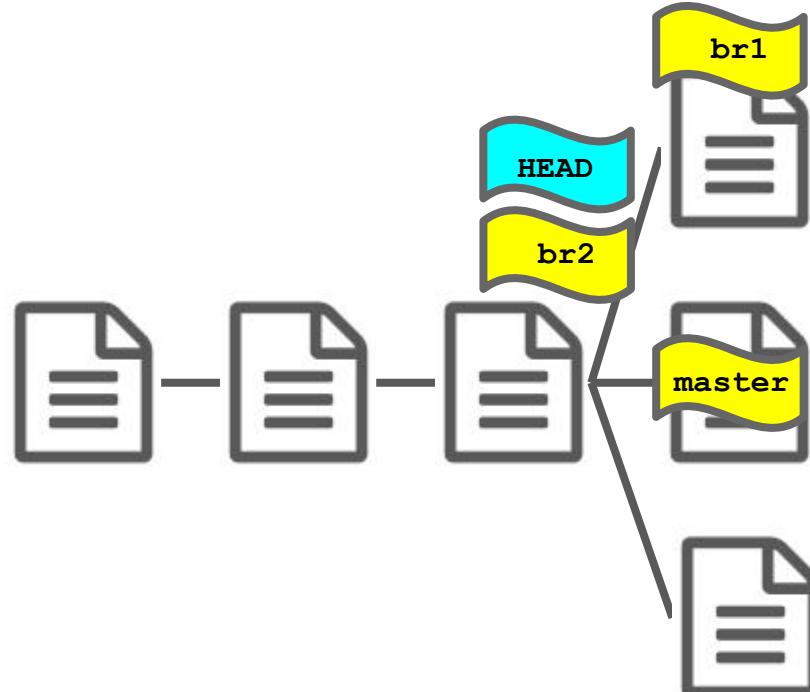


```
git add
```

```
git commit
```

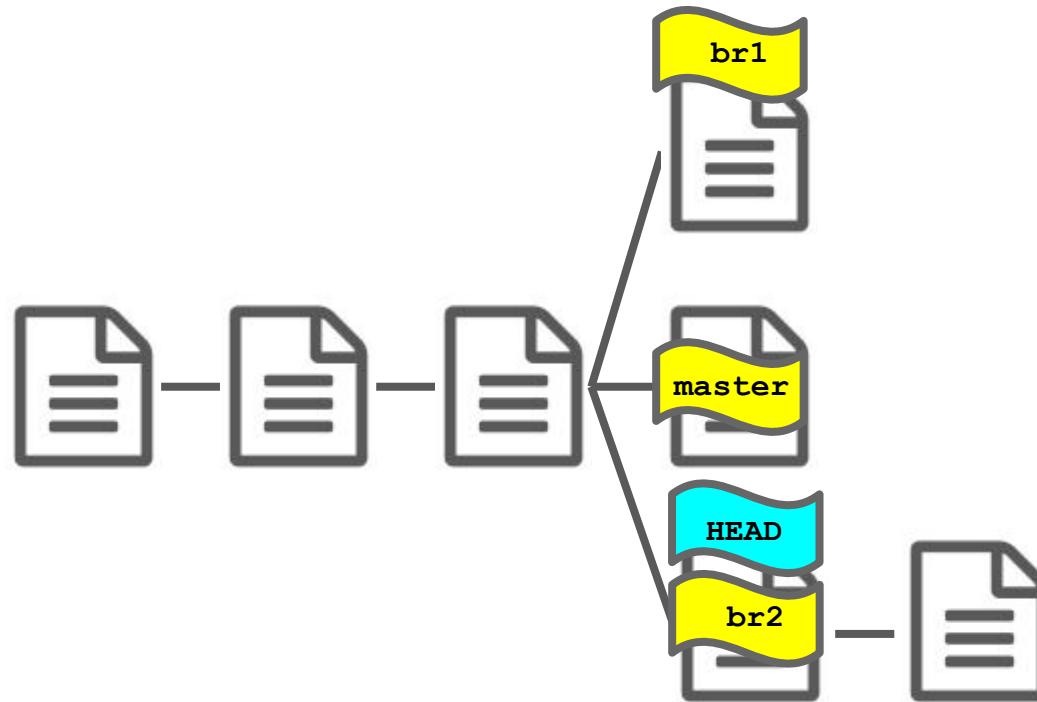


```
git checkout br2
```



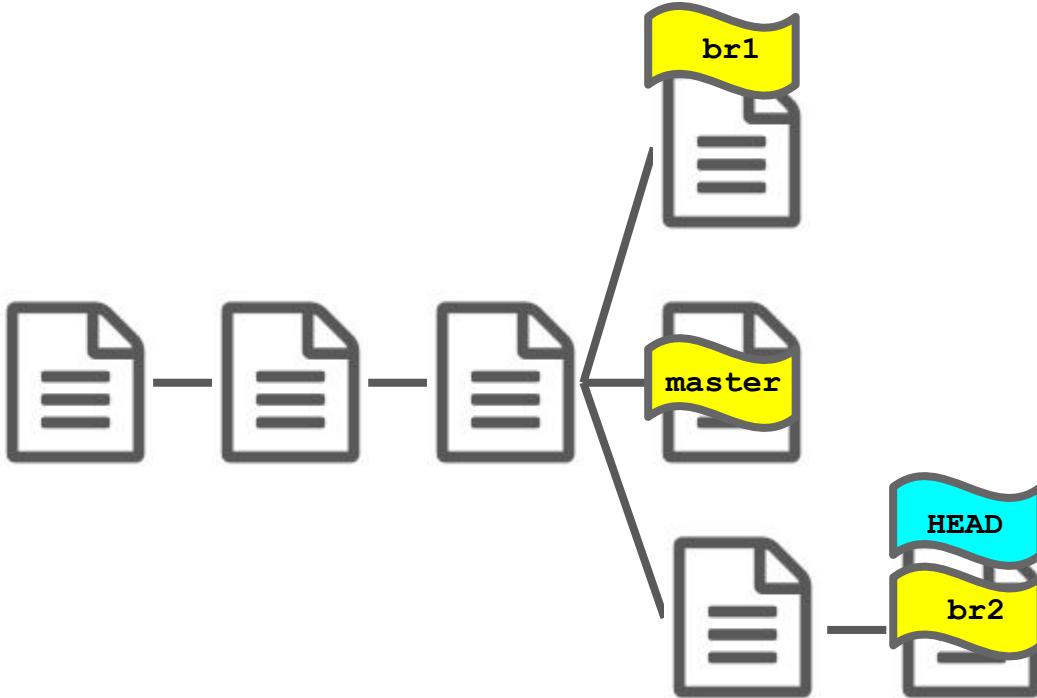
```
git add
```

```
git commit
```



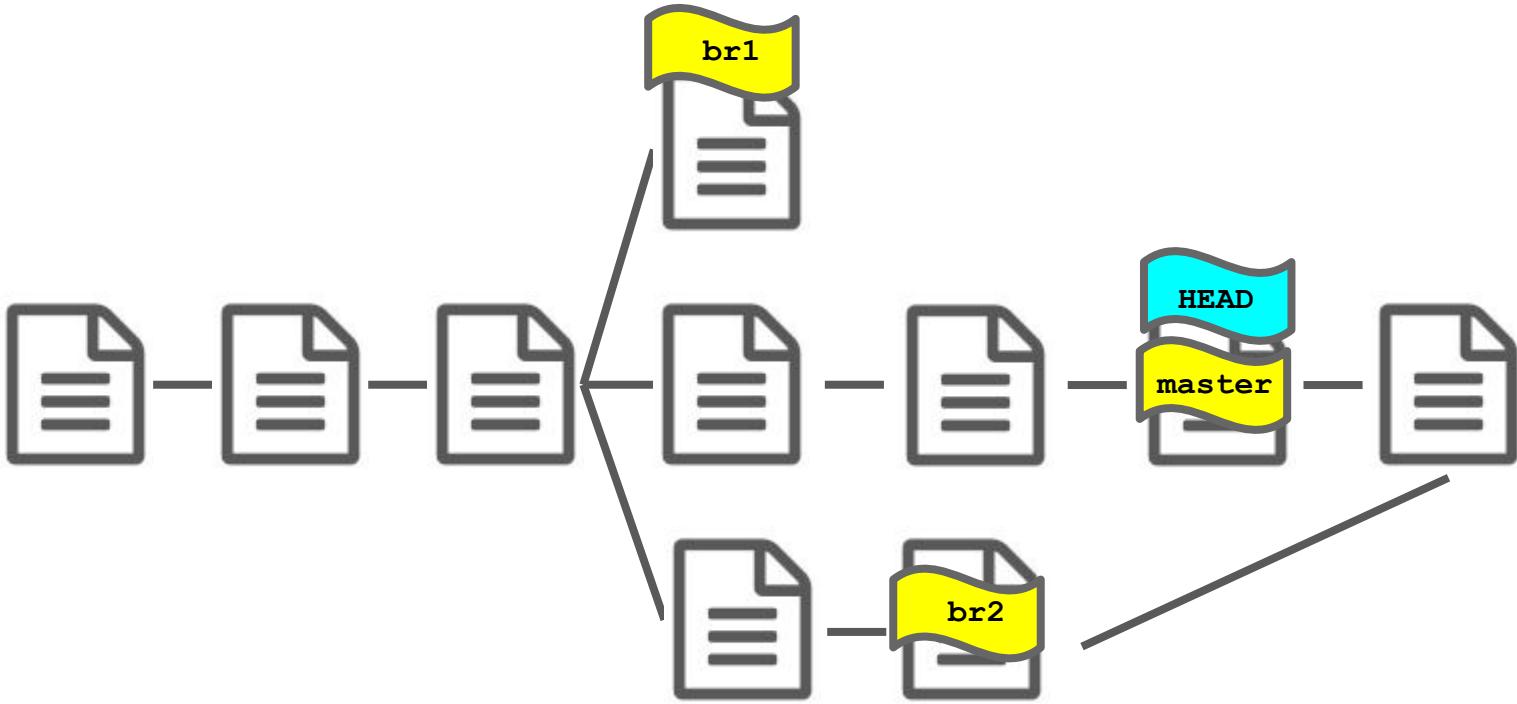
git add

git commit

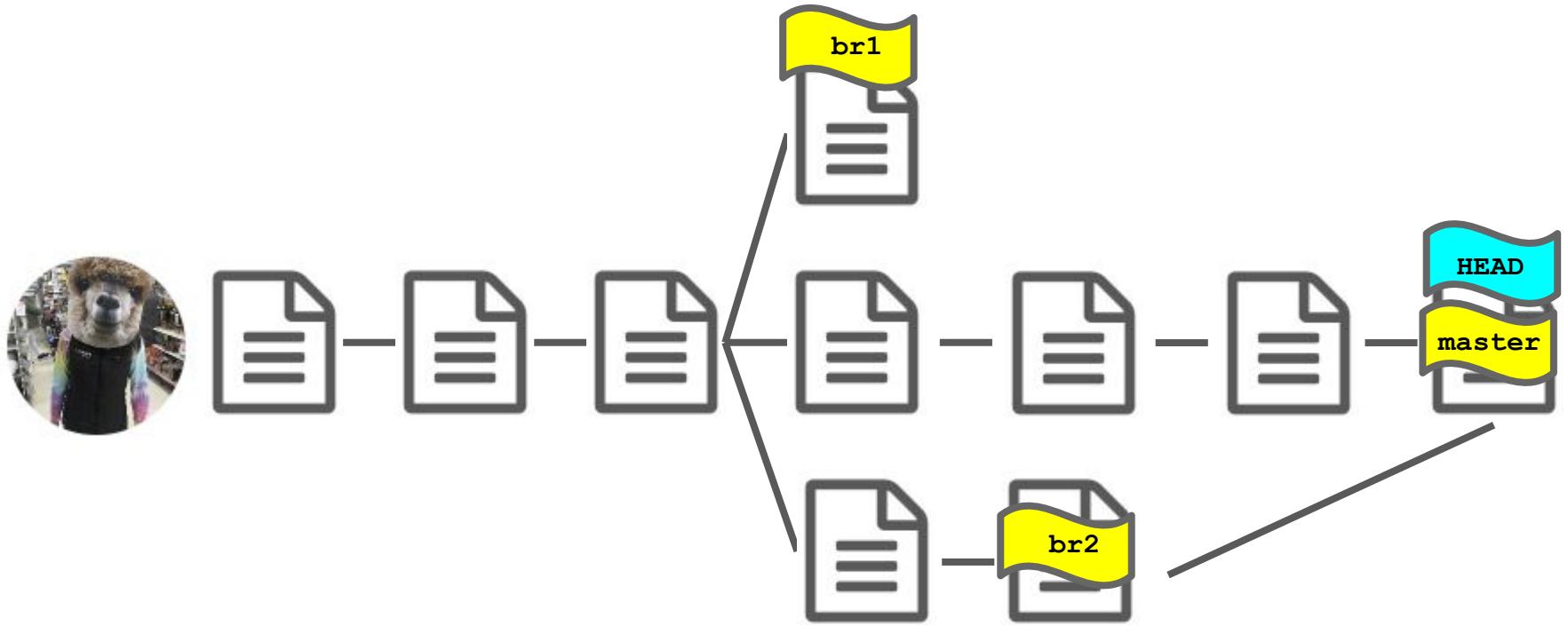


```
git add
```

```
git commit
```



```
git merge br2
```



“Local” version control - keeping it solo

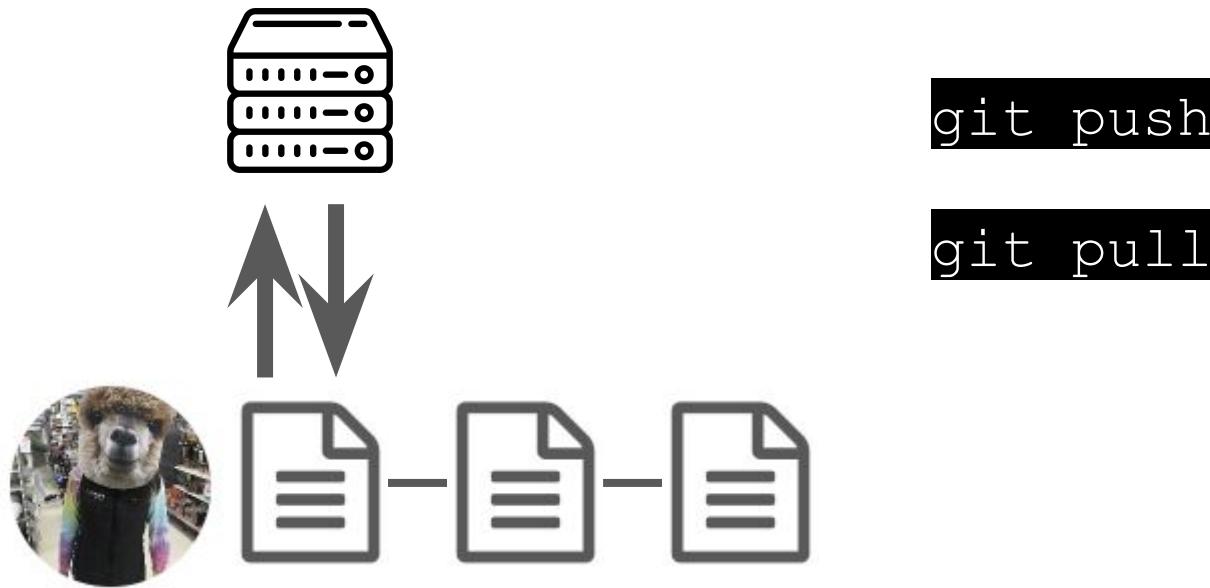
`git add` adds changes in the working directory to the staging area

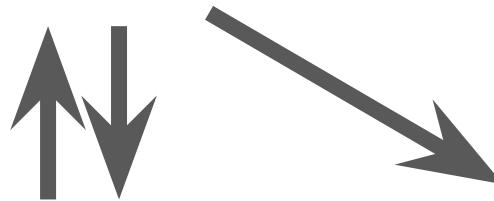
`git commit` records aka “gives id” to changes

`git branch br1` creates new branch

`git checkout br1` switches current workspace to branch “br1”

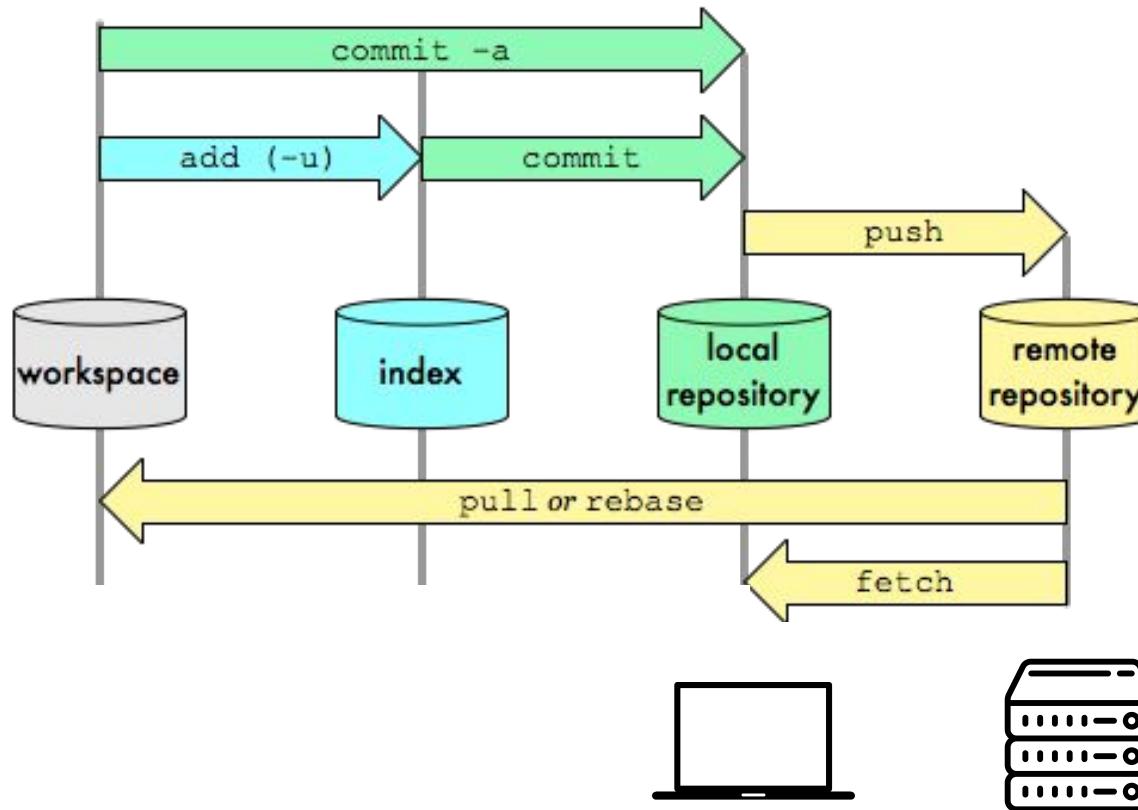
`git merge br1` merges br1 into current branch (e.g. master)





Git Data Transport Commands

<http://osteelle.com>



Cloning a Repository

When you create a repository, it exists as a *remote* repository. You can clone your repository to create a *local* copy on your computer and sync between the two locations.

The screenshot shows a GitHub repository page for 'nytimes / covid-19-data'. The repository has 291 stars, 4k forks, and 1.2k issues. It contains 36 commits, 1 branch, 0 packages, 0 releases, and 3 contributors. A pull request button and a 'Clone or download' button are visible. A tooltip for the 'Clone with HTTPS' button shows the URL <https://github.com/nytimes/covid-19-data>. The repository description is 'An ongoing repository of data on coronavirus cases and deaths in the U.S.' and links to <https://www.nytimes.com/interactive/2020/us/coronavirus-us-cases.html>. The README.md file is titled 'Coronavirus (Covid-19) Data in the United States'.

nytimes / covid-19-data

Watch 291 Star 4k Fork 1.2k

Code Issues 60 Pull requests 2 Projects 0 Wiki Security Insights

covid-19

36 commits 1 branch 0 packages 0 releases 3 contributors View license

Branch: master New pull request Create new file Upload files Find file Clone or download

albertsun New data for 4/12.

LICENSE Initial public release.

README.md Update geographic exception note for New York to add that starts

us-counties.csv New data for 4/12.

us-states.csv New data for 4/12.

README.md

Clone with HTTPS Use SSH

Use Git or checkout with SVN using the web URL.

<https://github.com/nytimes/covid-19-data>

Open in Desktop Download ZIP

17 hours ago

Coronavirus (Covid-19) Data in the United States



GitHub

Hosting

All your code and
documentation in one place

There are hundreds of millions of private, public, and open source repositories hosted on GitHub. Every repository is equipped with tools to help you host, version, and release code and documentation.



Bitbucket

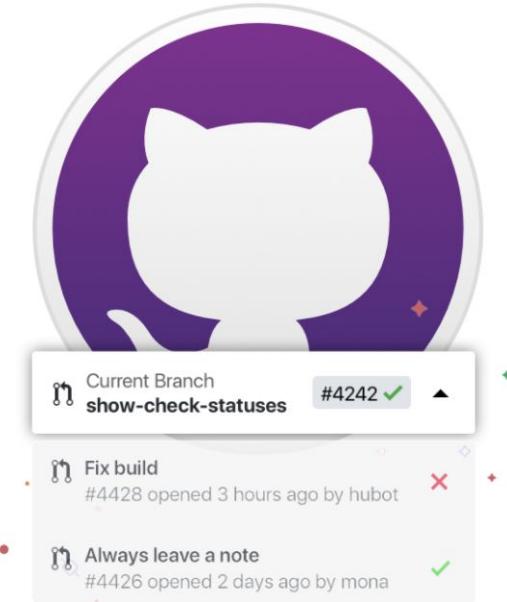


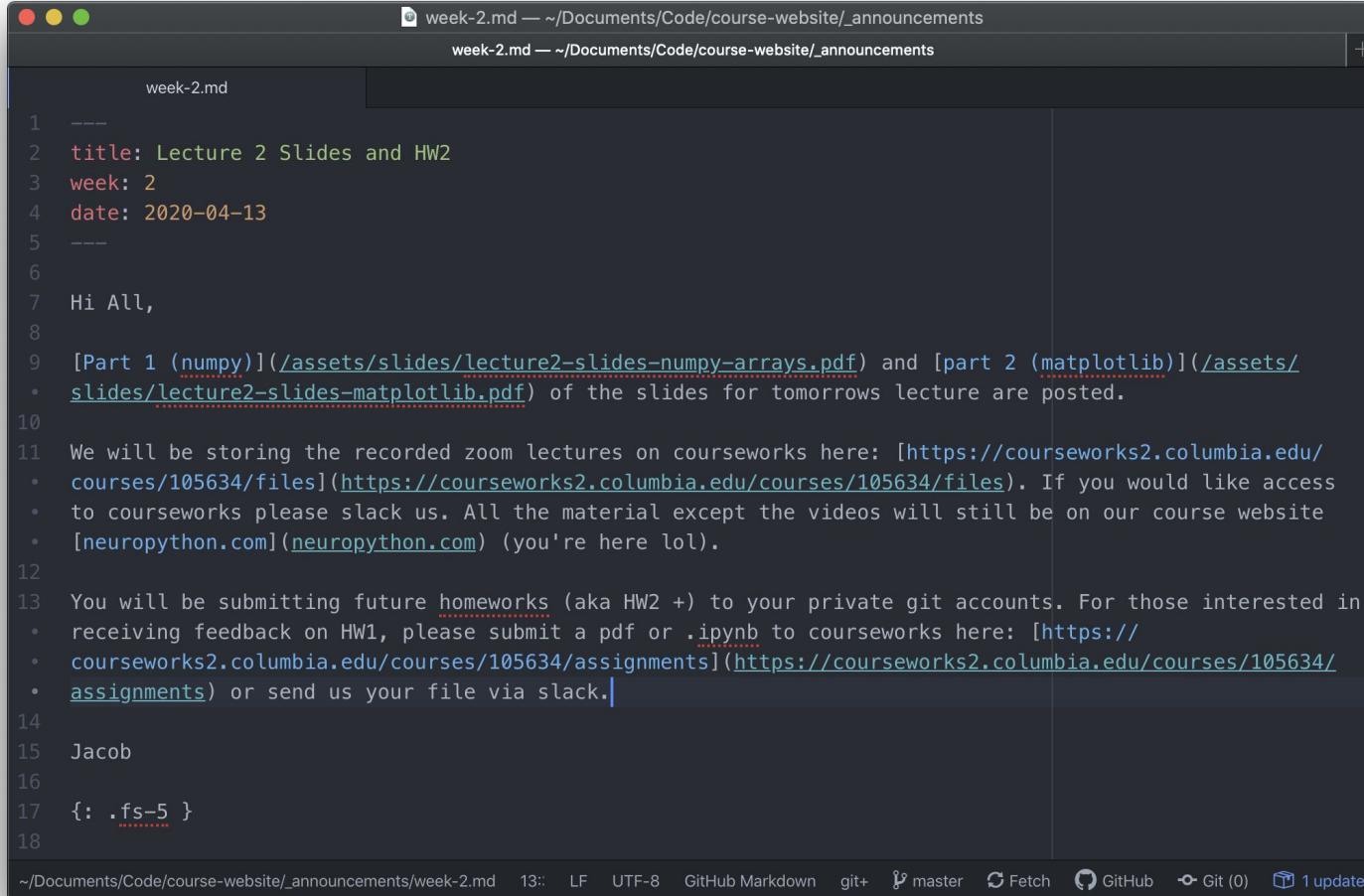
GitLab

GitHub Desktop

Visualize changes across Git and GitHub, and simplify your development workflow.

[Download GitHub Desktop for Windows or macOS →](#)





The screenshot shows a dark-themed code editor window with a title bar containing two tabs: "week-2.md — ~/Documents/Code/course-website/_announcements" and another tab labeled with a plus sign. The main area displays a Markdown file with the following content:

```
1 ---  
2 title: Lecture 2 Slides and HW2  
3 week: 2  
4 date: 2020-04-13  
5 ---  
6  
7 Hi All,  
8  
9 [Part 1 (numpy)](/assets/slides/lecture2-slides-numpy-arrays.pdf) and [part 2 (matplotlib)](/assets/slides/lecture2-slides-matplotlib.pdf) of the slides for tomorrow's lecture are posted.  
10  
11 We will be storing the recorded zoom lectures on courseworks here: [https://courseworks2.columbia.edu/courses/105634/files]. If you would like access  
• to courseworks please slack us. All the material except the videos will still be on our course website  
• [neuropython.com] (neuropython.com) (you're here lol).  
12  
13 You will be submitting future homeworks (aka HW2 +) to your private git accounts. For those interested in  
• receiving feedback on HW1, please submit a pdf or .ipynb to courseworks here: [https://courseworks2.columbia.edu/courses/105634/assignments]  
• [https://courseworks2.columbia.edu/courses/105634/assignments) or send us your file via slack.  
14  
15 Jacob  
16  
17 {:: .fs-5 }  
18
```

Current Repository course-website

Current Branch master

Fetch origin
Last fetched 9 minutes ago

Changes 1 History _announcements/week-2.md

1 changed file

_announcements/week-2.md

@@ -6,11 +6,11 @@ date: 2020-04-13

	6	6
	7	7
	8	8
	9	9
	10	10
	11	11
	12	12
	13	13
	14	14
	15	15
	16	16

Hi All,

-The [part 1 (numpy)](/assets/slides/lecture2-slides-numpy-arrays.pdf) and [part 2 (matplotlib)](/assets/slides/lecture2-slides-matplotlib.pdf) of the slides for tomorrow's lecture are posted.

+[Part 1 (numpy)](/assets/slides/lecture2-slides-numpy-arrays.pdf) and [part 2 (matplotlib)](/assets/slides/lecture2-slides-matplotlib.pdf) of the slides for tomorrow's lecture are posted.

We will be storing the recorded zoom lectures on courseworks here: <https://courseworks2.columbia.edu/courses/105634/files>. If you would like access to courseworks please slack us. All the material except the videos will still be on our course website neuropython.com (you're here lol).

-You will be submitting future homeworks (aka HW2+) to your private git accounts. For those interested in receiving feedback on HW1, please submit a pdf or .ipynb to courseworks here: <https://courseworks2.columbia.edu/courses/105634/assignments>

+You will be submitting future homeworks (aka HW2+) to your private git accounts. For those interested in receiving feedback on HW1, please submit a pdf or .ipynb to courseworks here: <https://courseworks2.columbia.edu/courses/105634/assignments> or send us your file via slack.

Update week-2.md

Description

Jacob

Commit to master

Current Repository course-website

Current Branch master

Fetch origin
Last fetched 10 minutes ago

Changes 1 History _announcements/week-2.md

1 changed file

_announcements/week-2.md

@@ -6,11 +6,11 @@ date: 2020-04-13

	6	6
	7	7
	8	8
	9	9
	10	10
	11	11
	12	12
	13	13
	14	14
	15	15
	16	16

Hi All,

-The [part 1 (numpy)](/assets/slides/lecture2-slides-numpy-arrays.pdf) and [part 2 (matplotlib)](/assets/slides/lecture2-slides-matplotlib.pdf) of the slides for tomorrow's lecture are posted.

+[Part 1 (numpy)](/assets/slides/lecture2-slides-numpy-arrays.pdf) and [part 2 (matplotlib)](/assets/slides/lecture2-slides-matplotlib.pdf) of the slides for tomorrow's lecture are posted.

We will be storing the recorded zoom lectures on courseworks here: [<https://courseworks2.columbia.edu/courses/105634/files>](https://courseworks2.columbia.edu/courses/105634/files). If you would like access to courseworks please slack us. All the material except the videos will still be on our course website [neuropython.com](neuropython.com) (you're here lol).

You will be submitting future homeworks (aka HW2+) to your private git accounts. For those interested in receiving feedback on HW1, please submit a pdf or .ipynb to courseworks here: [<https://courseworks2.columbia.edu/courses/105634/assignments>](https://courseworks2.columbia.edu/courses/105634/assignments)

+You will be submitting future homeworks (aka HW2+) to your private git accounts. For those interested in receiving feedback on HW1, please submit a pdf or .ipynb to courseworks here: [<https://courseworks2.columbia.edu/courses/105634/assignments>](https://courseworks2.columbia.edu/courses/105634/assignments) or send us your file via slack.

fixed some typos\$

Description

Commit to master

Current Repository course-website

Current Branch master

Push origin
Last fetched 11 minutes ago

Changes History

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.



Push 1 commit to the origin remote
You have one local commit waiting to be pushed to GitHub.
Always available in the toolbar when there are local commits waiting to be pushed or `⌘ P`

Push origin

Open the repository in your external editor
Select your editor in [Preferences](#)
Repository menu or `⌘ ⌘ A`

Open in Atom

View the files of your repository in Finder
Repository menu or `⌘ ⌘ F`

Show in Finder

Open the repository page on GitHub in your browser
Repository menu or `⌘ ⌘ G`

View on GitHub

Summary (required)

Description

Commit to master

Committed just now
fixed some typo\$

Undo

Current Repository course-website

Current Branch master

Pushing to origin
remote: Resolving deltas: 100% (...)

Changes History

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.



Push 1 commit to the origin remote
You have one local commit waiting to be pushed to GitHub.
Always available in the toolbar when there are local commits waiting to be pushed or `⌘ P`

Push origin

Open the repository in your external editor
Select your editor in [Preferences](#)
Repository menu or `⌘ ⌘ A`

Open in Atom

View the files of your repository in Finder
Repository menu or `⌘ ⌘ F`

Show in Finder

Open the repository page on GitHub in your browser
Repository menu or `⌘ ⌘ G`

View on GitHub

Summary (required)

Description

Commit to master

Committed just now
fixed some typo\$

Undo

Current Repository
course-website

Current Branch
master

Fetch origin
Last fetched just now

Changes History

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.



Open the repository in your external editor
Select your editor in [Preferences](#)
Repository menu or ⌘ ⌘ A

Show in Finder

View the files of your repository in Finder
Repository menu or ⌘ ⌘ F

View on GitHub

Open the repository page on GitHub in your browser
Repository menu or ⌘ ⌘ G

Summary (required)

Description

Commit to master

Create a free, private github account

<https://github.com/join>

Git Resources

<https://zuckermanbrain.github.io/git-novice/>

- Thorough git tutorial (~30 min)

<http://rogerdudler.github.io/git-guide/>

- Very rough overview (~10 min)

Aside: Extensions to Jupyter Notebook/Lab

- **Hinterland** - autocomplete while typing 😎
 - <https://jupyter-contrib-nbextensions.readthedocs.io/en/latest/nbextensions/hinterland/README.html>
- **Variable inspector** - you can see your vars!! 😂
 - <https://jupyter-contrib-nbextensions.readthedocs.io/en/latest/nbextensions/varInspector/README.html>
- **Autopep8** - formatting for publishing code documentation 😎
 - https://jupyter-contrib-nbextensions.readthedocs.io/en/latest/nbextensions/code_prettify/README_autopep8.html

Conda

There are conda packages for the notebook extensions and the `jupyter_nbextensions_configurator` available from [conda-forge](#). You can install both using

```
conda install -c conda-forge jupyter_contrib_nbextensions
```

This also automatically installs the Javascript and CSS files (using

`jupyter contrib nbextension install --sys-prefix`), so the second installation step below can therefore be skipped.

HW2

- Plotting Coronavirus Data
 - JHU data
<https://data.humdata.org/dataset/novel-coronavirus-2019-ncov-ca-ses>
 - NYT github repo <https://github.com/nytimes/covid-19-data>
- Setting up your own github account
- Bonus: create a personal website with github pages
<https://guides.github.com/features/pages/>
 - <https://jekyllthemes.io/theme/online-cv> website templates

Desktop

CAREER PROFILE

Summarise your career here lorem ipsum dolor sit amet, consectetuer adipiscing elit. You can download this free resume/CV template here. Aenean commodo ligula eget dolor aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu.

EXPERIENCES

Lead Developer Startup Hubs, San Francisco 2015 - Present
Describe your role here lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo.
Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

Senior Software Engineer Google, London 2014 - 2015
Describe your role here lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem.

UI Developer Amazon, London 2012 - 2014
Describe your role here lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu,



Alan Doe
Full Stack Developer

alan.doe@website.com
012 345 6789
blog.webjeda.com
[linkedin.com/in/alandoe](#)
[github.com/alandoe](#)
[@twittername](#)

EDUCATION
MSc in Computer Science
University of London
2011 - 2012
BSc in Applied Mathematics
Bristol University
2007 - 2011

LANGUAGES
English (Native)
French (Professional)

Mobile



Alan Doe
Full Stack Developer

alan.doe@website.com
012 345 6789
blog.webjeda.com
[linkedin.com/in/alandoe](#)
[github.com/alandoe](#)
[@twittername](#)

<https://github.com/sharu725/online-cv>