



Lecture 1-2: Programming Environment

Dr. James Chen, Animal Data Scientist, School of Animal Sciences

2023 APSC-5984 SS: Agriculture Data Science

What Is Programming Environment?

- **I. Why Python?**
- **II. Jupyter Notebook**
- **III. Git and GitHub**
- **IV. What is Unix shell?**
- **V. Integrated Development Environment (IDE)**

I. Why Python?

1. Interpreted vs. Compiled language



Interpreted language:

- **Examples: Python, Perl, R, Matlab**
- The interpreter executes the codes **line by line**.
- Faster **development process** as you can validate the results without compiling the entire program.
- Suitable for exploring new ideas or developing lightweight applications

Compiled language:

- **Examples: C/C++, Java, Fortran, Visual Basic**
- The **entire program** will be compiled and built before the execution.
- Faster **execution time**.
- Suitable for developing large and complicated applications

I. Why Python?

2. Dynamic vs. Static typing

Python is a **dynamically-typed language**: Variables DO NOT have static types, such as integer, string, or floating number.



Pros:

Higher flexibility for development

Cons:

Maybe slower and less efficient (memory-wise)
as the interpreter has to determine the type of a
variable at runtime



C is a statically-typed language

```
#include <stdio.h>
int main()
{
    int x = 5; → int: specify the variable type as an integer
    printf("x is %d\n", x);
    return 0;   ↓
}
```

the main() function MUST return an integer value

→ int: specify the variable type as an integer

%d to specify the variable type
(i.e., integer) in a formatted string

Lecture 1-2: Programming environment

I. Why Python?

2. Dynamic vs. Static typing

Python is a **dynamically-typed language**: Variables DO NOT have static types, such as integer, string, or floating number.



Python is a dynamically-typed language

```
x = 5
print(f"x is {x}")

x = "hello"
print(f"x is {x}")
```

You can assign whatever data type
you want to the variable 'x'



C is a statically-typed language

```
#include <stdio.h>

int main()
{
    int x = 5; → int: specify the variable type as an integer
    printf("x is %d\n", x);
    return 0;      ↓
}
```

the main() function MUST return an integer value

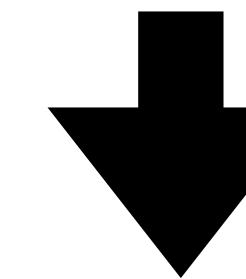
→ int: specify the variable type as an integer

%d to specify the variable type
(i.e., integer) in a formatted string

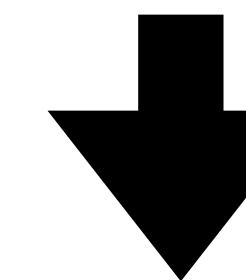
I. Why Python?

3. Large and active community

Many resources are available to help you with your problems. Such as forums, blogs, and tutorials



Wide range of libraries and frameworks



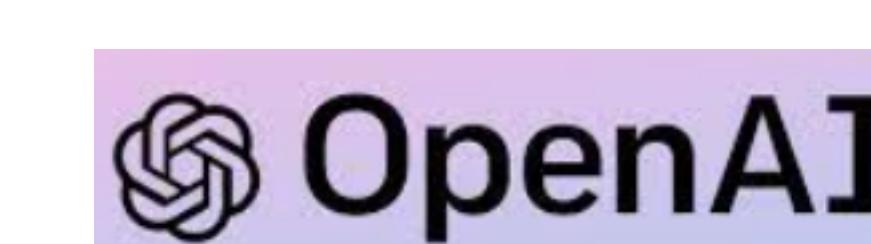
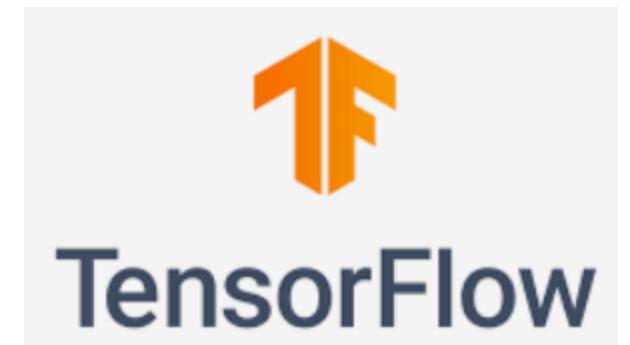
Scientific computation



Visualization



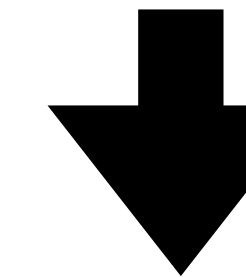
Machine learning



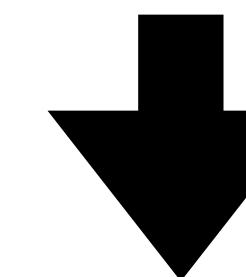
I. Why Python?

3. Large and active community

Many resources are available to help you with your problems. Such as forums, blogs, and tutorials



Wide range of libraries and frameworks



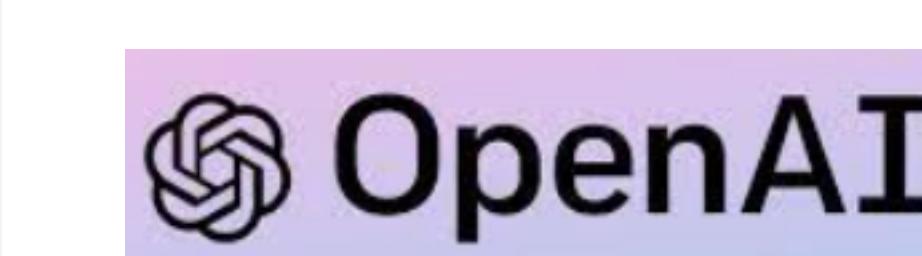
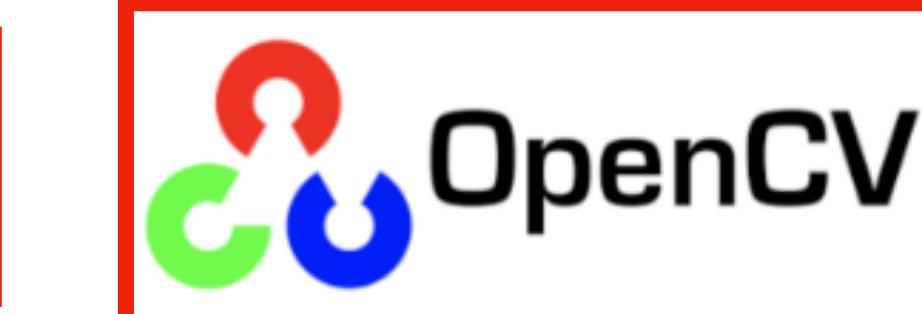
Scientific computation



Visualization



Machine learning



II. Jupyter Notebook

It is a web-based interactive computing platform

- Good for prototyping and testing codes
- Easy to document and share your work

You can share the notebook through

a PDF file or an interactive webpage

with others

Data

Multivariate normal distribution

First, we can simulate a structured data that is sampled from two different multivariate normal distributions.

```
In [194]: # number of clusters
k = 2
# number of data points in each cluster
n = 50
# total number of data points
N = n * k
# cluster properties: means and standard deviation
means = [[40, 40], [60, 70]]
stds = [[[80, 0],
          [0, 70]],
         [[60, 10],
          [10, 90]]]

# use np.random.multivariate_normal() to sample the datasets
data = []
for mean, std in zip(means, stds):
    data += [np.random.multivariate_normal(mean=mean, cov=std, size=n)]
data = np.concatenate(data)

# validation
print("Shape: ", data.shape)
print(data[:20])
```

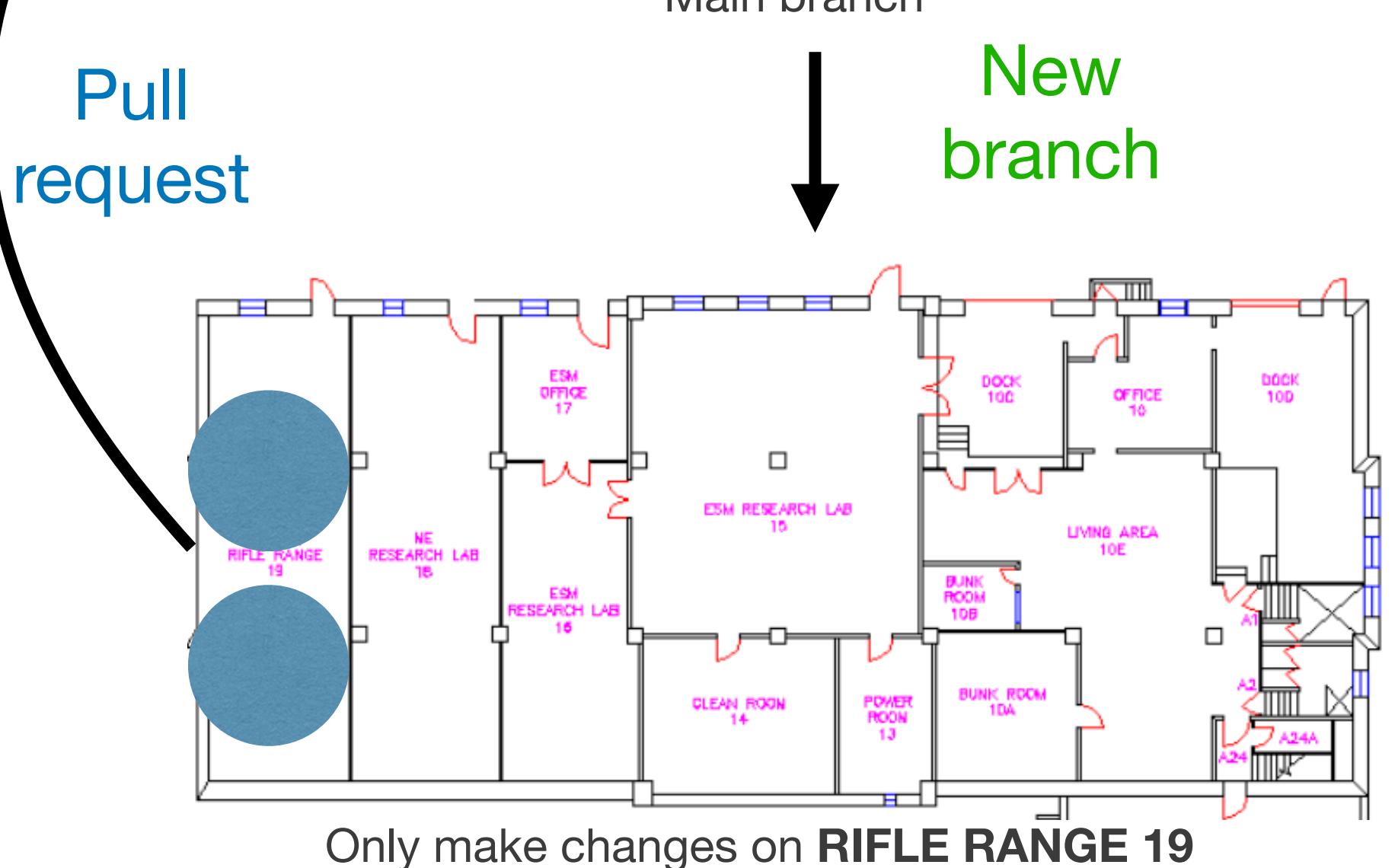
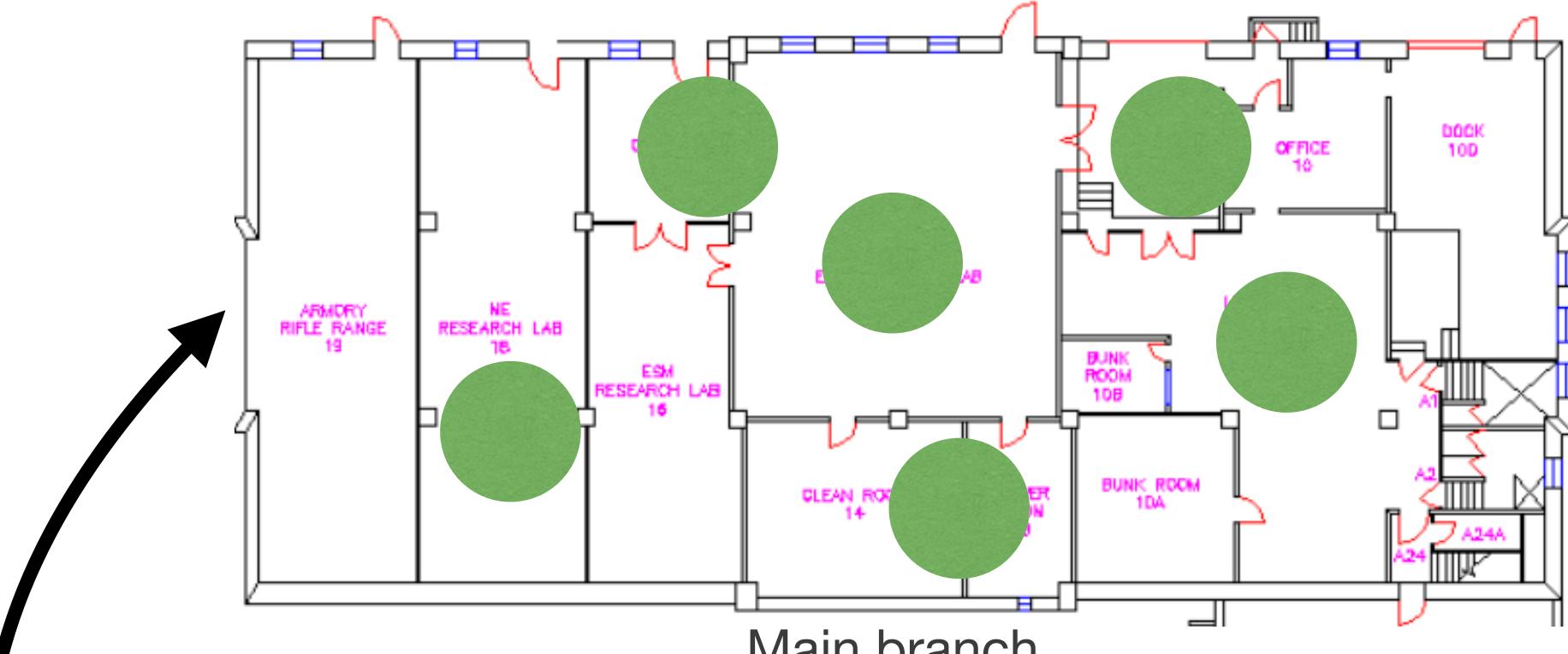
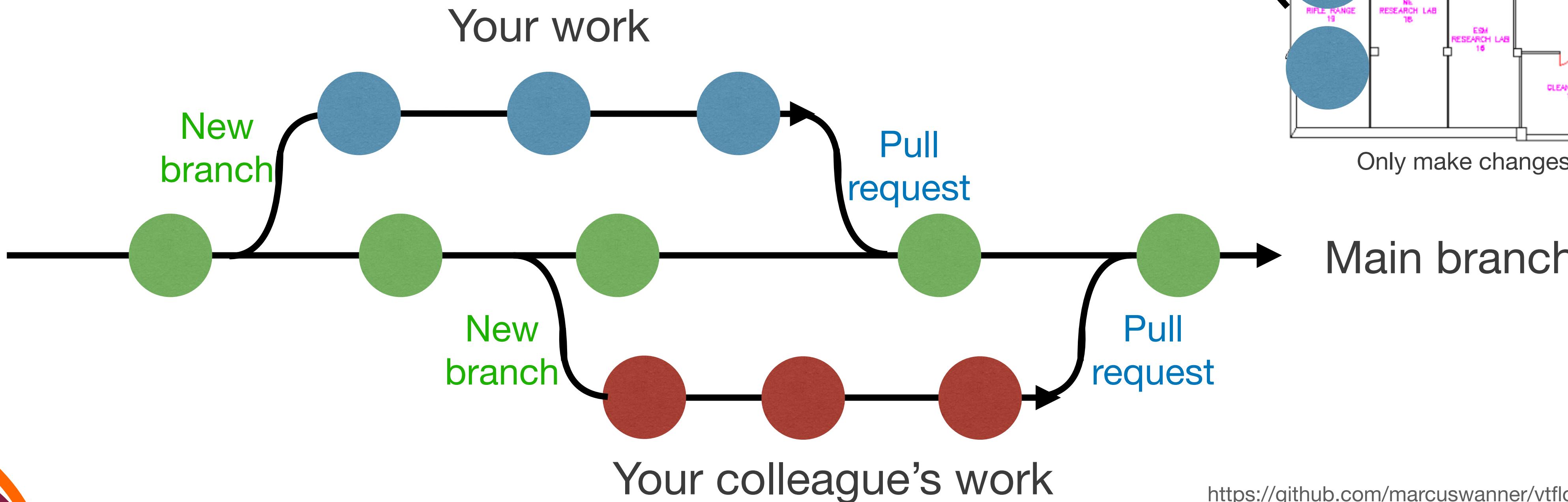
```
Shape: (100, 2)
[[51.66251802 47.84412914]
 [39.71618565 50.12966424]
 [35.30204586 30.11518838]
 [41.21290983 37.45870684]
 [33.50222942 27.44831926]
 [45.59088463 45.32844912]
 [42.53108047 46.10790442]
 [28.29073007 52.30456678]
 [38.31237176 34.84785694]
 [29.25183562 43.67950456]
 [40.58293457 25.99292839]
 [45.3154831 45.79598333]
 [22.58292304 39.54979532]
 [33.92459233 43.18324273]
 [38.06019806 40.15521412]
 [38.81693873 47.34977002]
 [43.93189465 42.68070032]
 [46.3391774 51.42474463]
 [42.13973106 38.61929858]
 [45.20335028 38.9488359 ]]
```

Lecture 1-2: Programming environment

III. Git and GitHub

A time machine for your coding project!

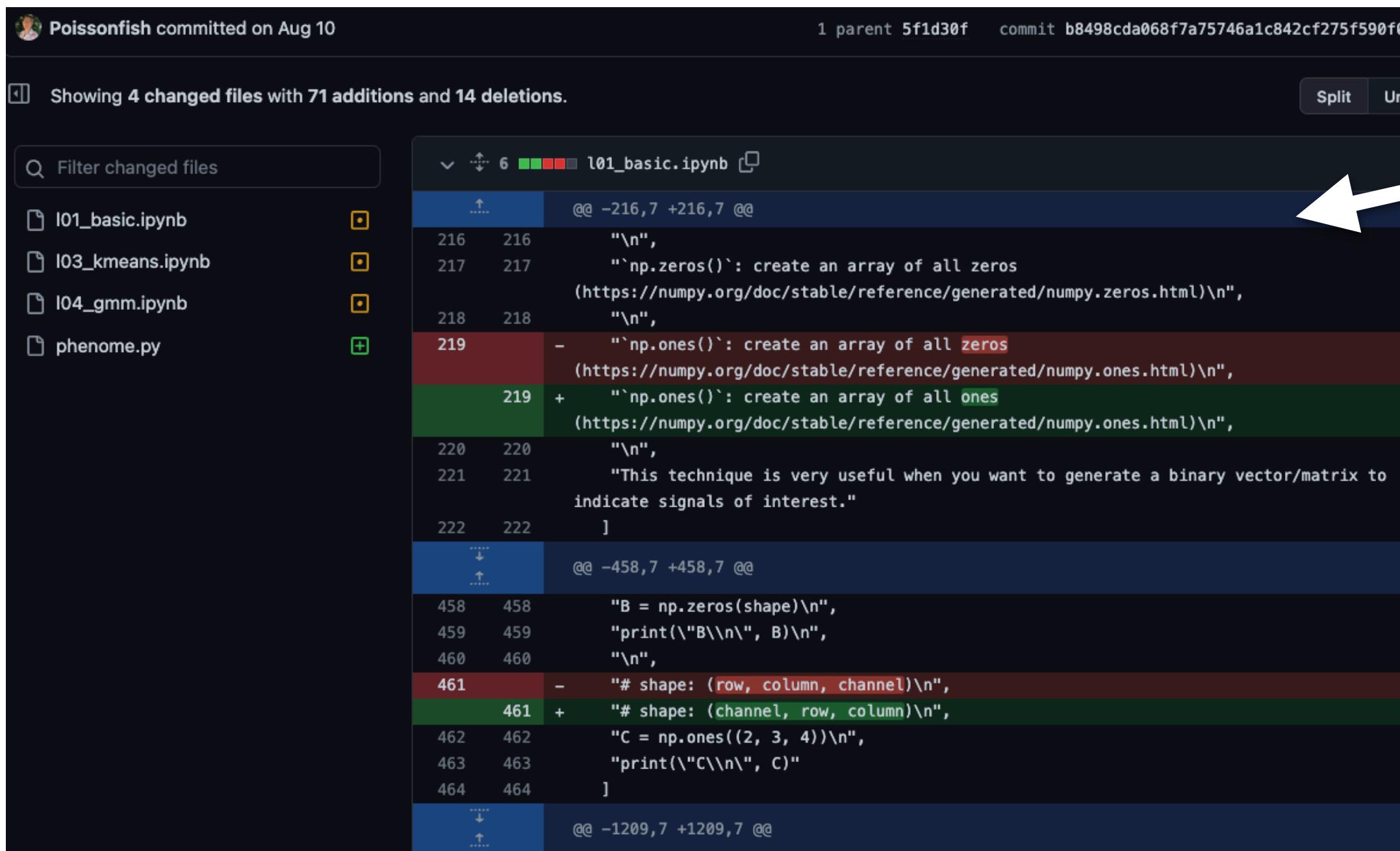
- **Git** is a system that allows you to track and manage changes to your code over time
- It also allows you to merge changes from multiple sources (collaborators) into a single codebase.



III. Git and GitHub

A time machine for your coding project!

- **Git** is a system that allows you to track and manage changes to your code over time
- It also allows you to merge changes from multiple sources (collaborators) into a single codebase.



Poissonfish committed on Aug 10

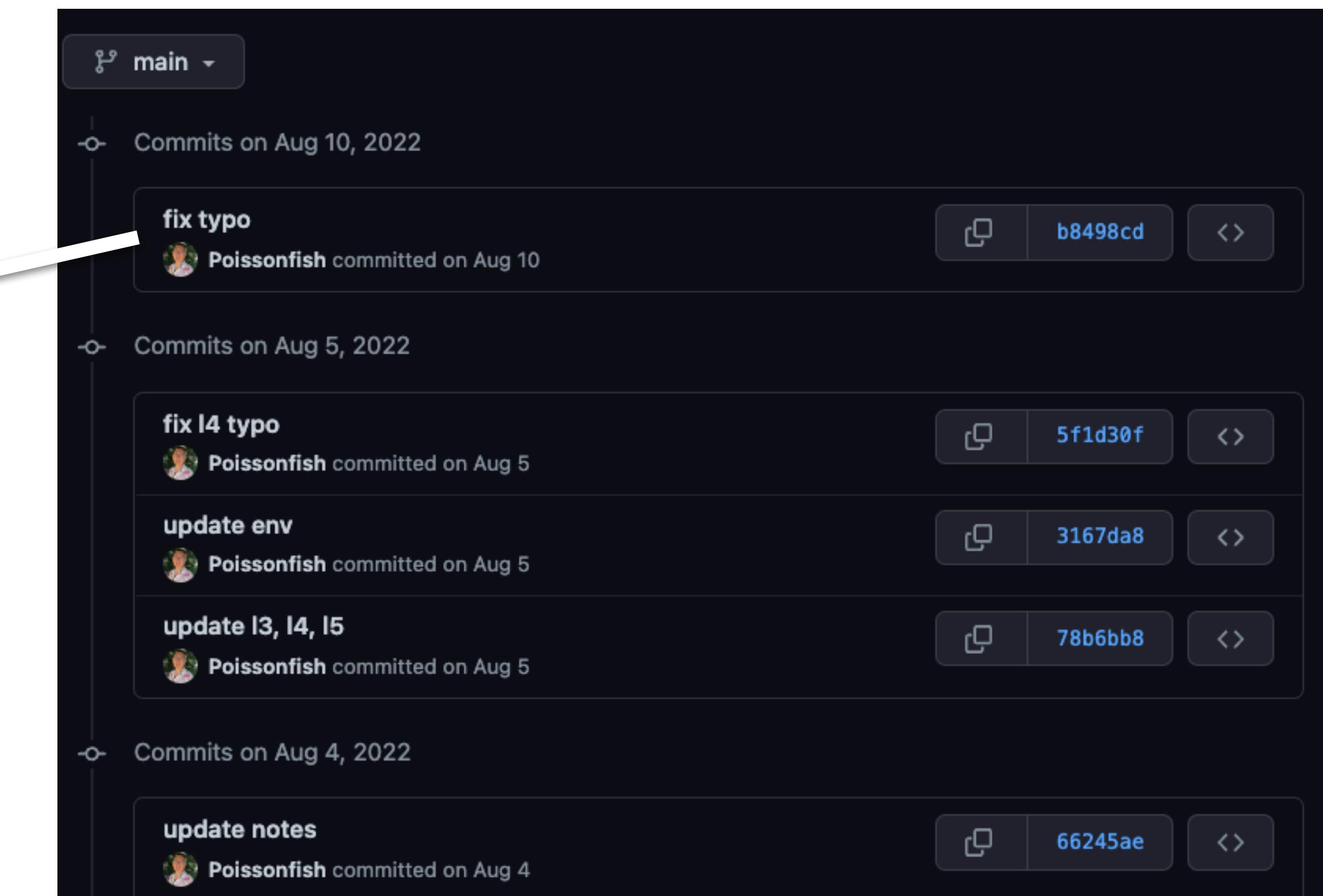
Showing 4 changed files with 71 additions and 14 deletions.

l01_basic.ipynb

```
diff --git a/l01_basic.ipynb b/l01_basic.ipynb
--- a/l01_basic.ipynb
+++ b/l01_basic.ipynb
@@ -216,7 +216,7 @@ 
 216 216      "\n",
 217 217      ``np.zeros()``: create an array of all zeros
 218 218      (https://numpy.org/doc/stable/reference/generated/numpy.zeros.html)\n",
 219 219 -    ``np.ones()``: create an array of all zeros
 219 219 +    ``np.ones()``: create an array of all ones
 220 220      (https://numpy.org/doc/stable/reference/generated/numpy.ones.html)\n",
 221 221      "\n",
 221 221      "This technique is very useful when you want to generate a binary vector/matrix to
 222 222      indicate signals of interest."
 222 222 ]
```

1 parent 5f1d30f commit b8498cda068f7a75746a1c842cf275f590f0adaa

GitHub is a web-based, public platform that implements Git system.



Track every change you made

Historical records of your repository

Lecture 1-2: Programming environment

IV. Unix Shell

- **Unix shell** is a command-line interface that allows you to interact with the operating system.
- The major way to interact with a remote high-performance computer (e.g., VT ARC)
- You can use shell commands to:

Execute scripts

Copy files

Move / rename files

Delete files

Create folders

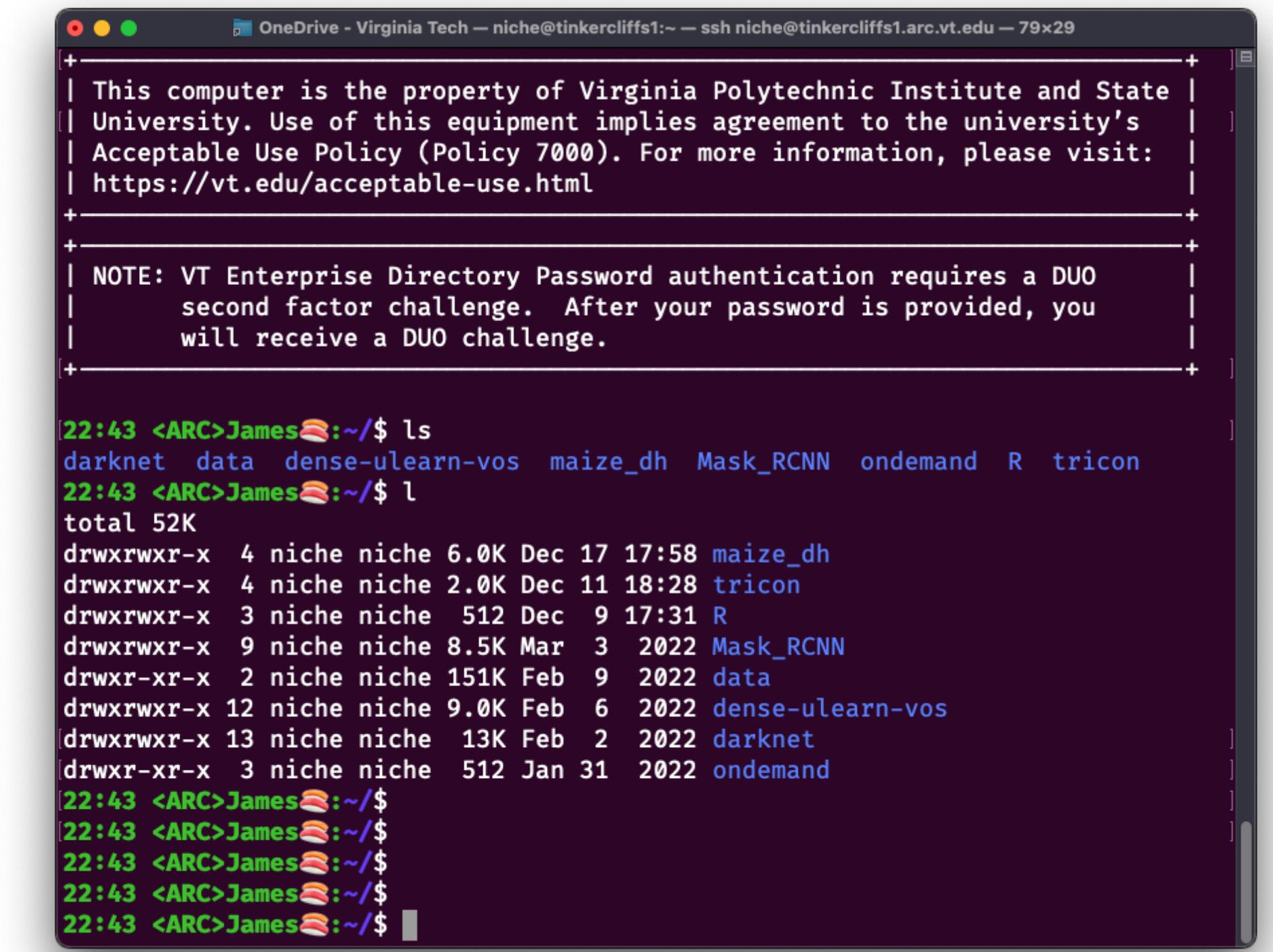
List files

Write texts to files

Display file contents

Change the working directory

Show the current working directory



This screenshot shows a terminal window titled "OneDrive - Virginia Tech - niche@tinkercliffs1:~ ssh niche@tinkercliffs1.arc.vt.edu - 79x29". It displays a message about university equipment usage and a note about VT Enterprise Directory Password authentication requiring a DUO second factor challenge. Below this, the user runs the 'ls' command, listing several files and directories including 'darknet', 'data', 'dense-ulearn-vos', 'maize_dh', 'Mask_RCNN', 'ondemand', 'R', and 'tricon'. The terminal then prompts for a password, indicating a DUO challenge has been received.

```
| This computer is the property of Virginia Polytechnic Institute and State |
| University. Use of this equipment implies agreement to the university's |
| Acceptable Use Policy (Policy 7000). For more information, please visit: |
| https://vt.edu/acceptable-use.html |
|
| NOTE: VT Enterprise Directory Password authentication requires a DUO |
| second factor challenge. After your password is provided, you |
| will receive a DUO challenge. |
|
[22:43 <ARC>James:~/]$ ls
darknet data dense-ulearn-vos maize_dh Mask_RCNN ondemand R tricon
[22:43 <ARC>James:~/]$ l
total 52K
drwxrwxr-x 4 niche niche 6.0K Dec 17 17:58 maize_dh
drwxrwxr-x 4 niche niche 2.0K Dec 11 18:28 tricon
drwxrwxr-x 3 niche niche 512 Dec 9 17:31 R
drwxrwxr-x 9 niche niche 8.5K Mar 3 2022 Mask_RCNN
drwxr-xr-x 2 niche niche 151K Feb 9 2022 data
drwxrwxr-x 12 niche niche 9.0K Feb 6 2022 dense-ulearn-vos
drwxrwxr-x 13 niche niche 13K Feb 2 2022 darknet
drwxr-xr-x 3 niche niche 512 Jan 31 2022 ondemand
[22:43 <ARC>James:~/]$ 
[22:43 <ARC>James:~/]$ 
[22:43 <ARC>James:~/]$ 
[22:43 <ARC>James:~/]$ 
[22:43 <ARC>James:~/]$ 
```

A screenshot of the VT ARC remote server

V. Integrated Development Environment (IDE)

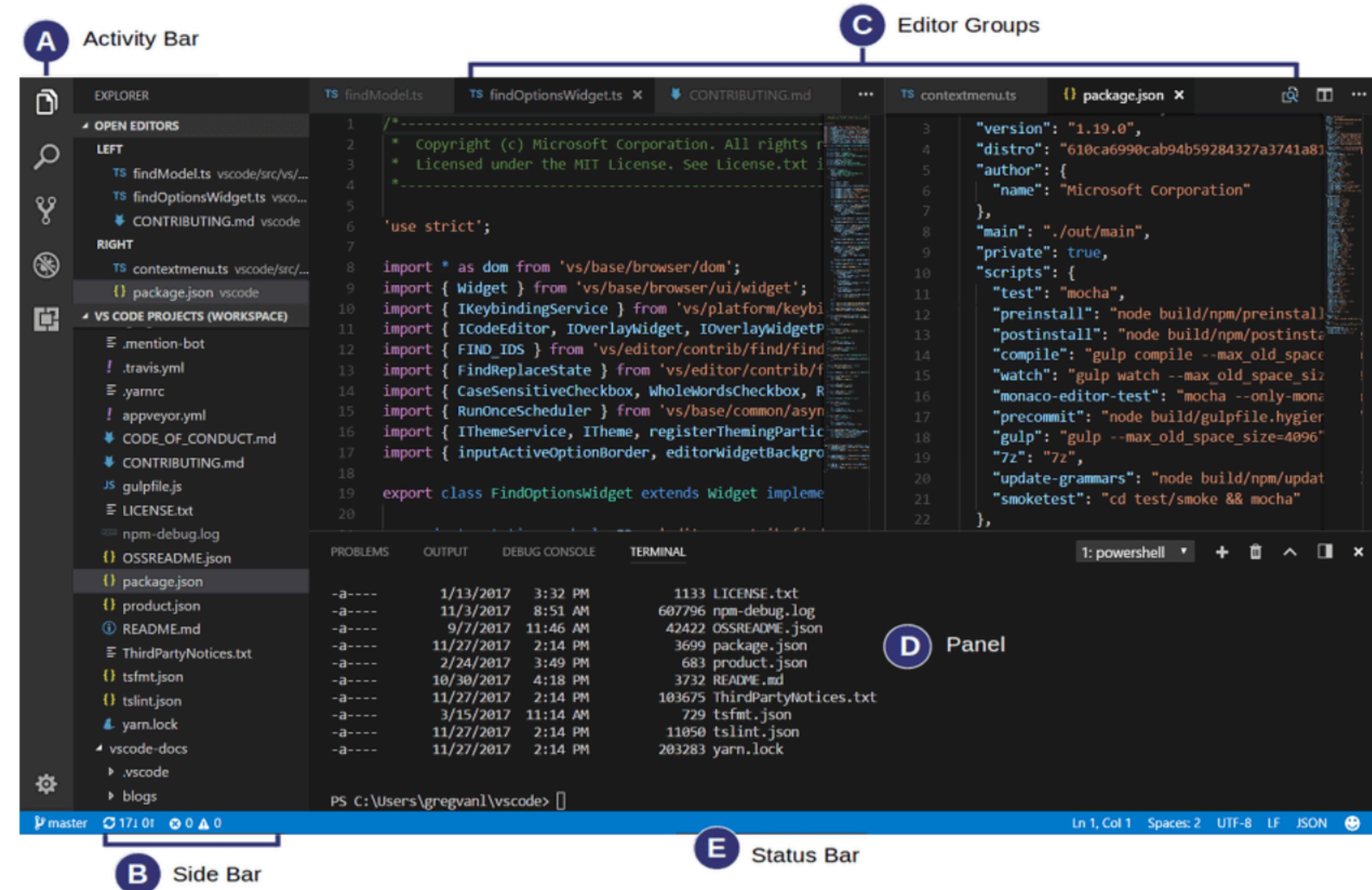
Integrated development environment (**IDE**) is a software application that provides a comprehensive set of tools for software development

Code editor

Automation tools

Warnings and suggestions

Code builder
(e.g., LaTeX)



V. Integrated Development Environment (IDE)

IDE put everything together!

The screenshot shows a Jupyter Notebook interface integrated into a larger development environment. The top navigation bar indicates the notebook is titled "Pi-Computer-Vision".

Left Sidebar (File Explorer):

- OPEN EDITORS: Shows multiple notebooks, including "221120_segformer_train.ipynb" which is currently active.
- PI-COMPUTER-VISION directory structure:
 - __pycache__
 - .ipynb_checkpoints
 - .vscode
 - arc
 - data
 - James
 - models
 - node_modules
 - notebooks (containing __pycache__ and .ipynb_checkpoints)
 - out
 - packages
 - test (containing __pycache__, .ipynb_checkpoints, .pytest_cache, data, test_file_io.py, test_models.py)
 - .env
- SOURCE CONTROL REPOSITORIES: Shows a commit message "Message (%Enter to commit on "master")" and a list of changes in the "notebooks" folder.
- SOURCE CONTROL: Shows a "Commit" button and a list of changes.

Middle Area (Code Editor):

Cell [4] displays code to show an image and its shape:

```
plt.imshow(ds[5]["image"])
print(np.array(ds[5]["image"])[::2, ::2, :]) # first 2x2 pixels
print("Shape: ", np.array(ds[5]["image"]).shape)
```

Output:

```
[[[222 164 127]
 [222 164 127]]

 [[220 162 125]
 [220 162 125]]]
Shape: (512, 768, 3)
```

A preview image of a bedroom scene is shown below the code.

Cell [5] displays code to show an annotation and its shape:

```
plt.imshow(ds[5]["annotation"])
print(np.array(ds[5]["annotation"]))
print("Shape: ", np.array(ds[5]["annotation"]).shape)
```

Output:

```
[[6 6 6 ... 6 0 0]
 [6 6 6 ... 6 0 0]
 [6 6 6 ... 6 0 0]
 ...
 [4 4 4 ... 4 4 0]
 [4 4 4 ... 4 4 0]
 [0 0 0 ... 0 0 0]]
Shape: (512, 768)
```

Right Area (Terminal and File List):

The terminal window shows a file listing for "av" (likely a alias for "av" command) and a "bash - _03_Papers" session showing a directory listing of "_03_Papers" contents.

```
av
drwx----- 4 niche staff 128B Nov 13 20:54 talks
drwx----- 6 niche staff 192B Nov 11 11:31 _02_Grants_Docs
drwx----- 3 niche staff 96B Oct 6 00:40 _12_Reviews
drwx----- 5 niche staff 160B Oct 6 00:39 car
drwxr-xr-x+ 5 niche staff 160B Jan 28 2022 ..
(transformer) 22:59 James:OneDrive - Virginia Tech/
(transformer) 22:59 James:OneDrive - Virginia Tech/ ls
DC_MidJourney4_please_generate_a_landing_web_page_layout_for_a_
_57661d8e-c5fd-46e1-966a-25d235246c3c.png
DC_MidJourney4_please_generate_a_landing_web_page_layout_for_a_
_5992112d-e8d2-4791-a873-58fe6d1fb6ee.png
Icon?
Microsoft Teams Chat Files
_01_Lab
_02_Grants
_02_Grants_Docs
_03_Papers
_03_Papers_Docs
_04_Software
_05_Talks_Travels
_05_Teaching
_06_Mautushi
_06_Shihong
_06_Xiaohui
_07_Writing
_11_Admin
_12_Reviews
_99_Arc
_99_Random
car
eructation_01.wav
notebooks
talks
(transformer) 22:59 James:OneDrive - Virginia Tech/ cd _03_P
apers
(transformer) 22:59 James:OneDrive - Virginia Tech/ ls
DH_maize Tricon eigen_aug preg_detection
(transformer) 22:59 James:OneDrive - Virginia Tech/ ls -alht
ls: t: No such file or directory
(transformer) 22:59 James:OneDrive - Virginia Tech/ ls -alht
total 24
drwx-----@ 31 niche staff 992B Dec 25 23:51 ..
drwx----- 29 niche staff 928B Dec 21 18:47 DH_maize
-rw-r--r--@ 1 niche staff 8.0K Dec 16 17:08 .DS_Store
drwx----- 5 niche staff 160B Dec 16 14:00 eigen_aug
drwxr-xr-x 7 niche staff 224B Dec 16 13:58 .
drwxr-xr-x@ 23 niche staff 736B Dec 15 18:08 Tricon
drwx----- 14 niche staff 448B Dec 5 16:24 preg_detection
(transformer) 22:59 James:OneDrive - Virginia Tech/ ls
```

V. Integrated Development Environment (IDE)

IDE put everything together!

