Computational Methods for Linguists

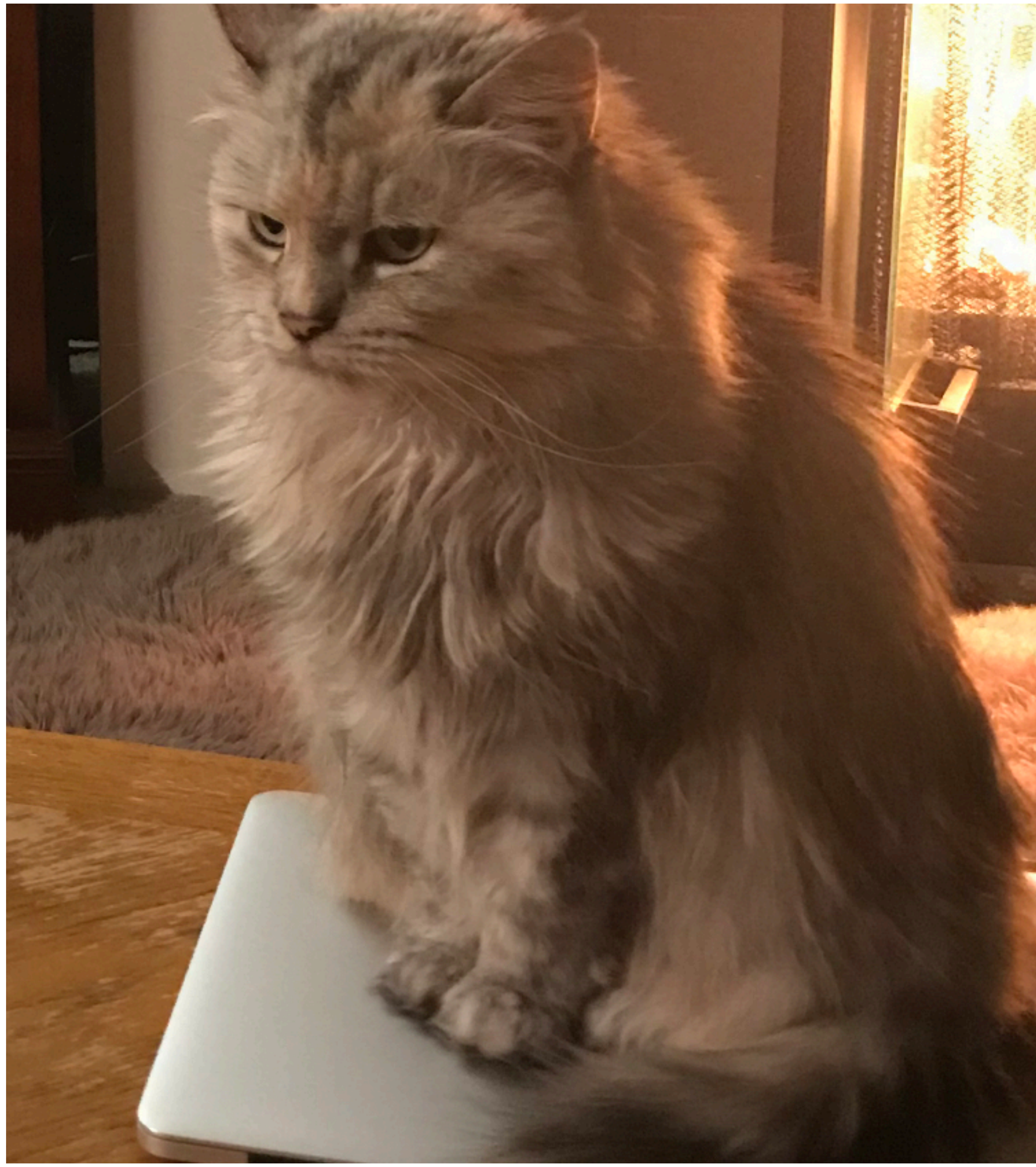
Ling 471

Olga Zamaraeva (Instructor)
Yuanhe Tian (TA)

04/08/21

Reminders

- First **blog post** due today (5 posts)
  - Responses due by next lecture (April 13)
  - Each student not blogging responds to **one** post
- **Assignment 1** due April 13
  - If you still don't have **patas** access:
    - Email Olga now with info: when you requested it
    - If you don't have access by EOD Friday, you can get an **extension for Part 4**
- **Assignment 2** will be published by Apr. 13-15



Plan for today

- IDE: First steps programming
- Command line
- Virtual environments
- Version control (git)
- See Olga getting confused during demos!
    - Please interrupt the demos and ask questions!

Integrated Development Environments
aka IDEs

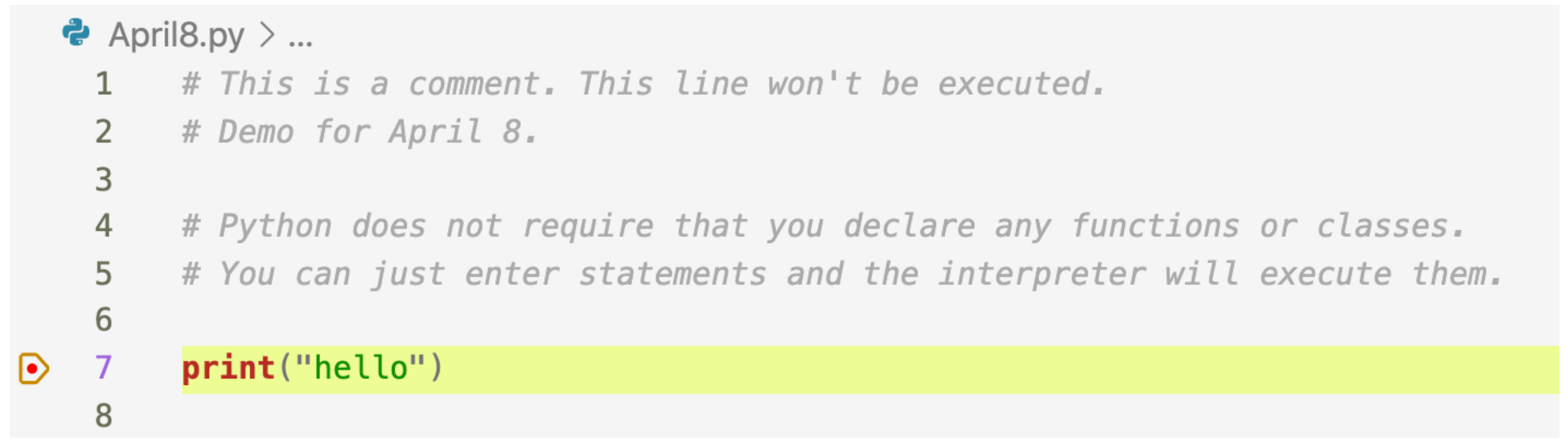
- Source code editor + build automation + debugger
  - (and more)
  - Editor: to enter text for the program
    - Recall a program is **just text**
  - Build automation: (path to) compiler/interpreter etc.
  - **Debugger: inspect program state step by step**
- We are using **Visual Studio Code**
  - Popular for python: **Pycharm**
  - VS Code supports **a variety** of languages
  - **Demo:** setting up a python project and debugging

https://medium.com/analytics-vidhya/difference-between-text-editor-and-ide-integrated-development-environment-73f8b2368de6

Programming first steps
(preview of next week!)

- To program, you need to understand:
  - Input/output
  - Assignment
    - Variables, types
  - Math and logical operators
  - Control flow:
    - Conditionals
    - Repetition (loops)
  - Function, classes, inheritance, object properties, keywords

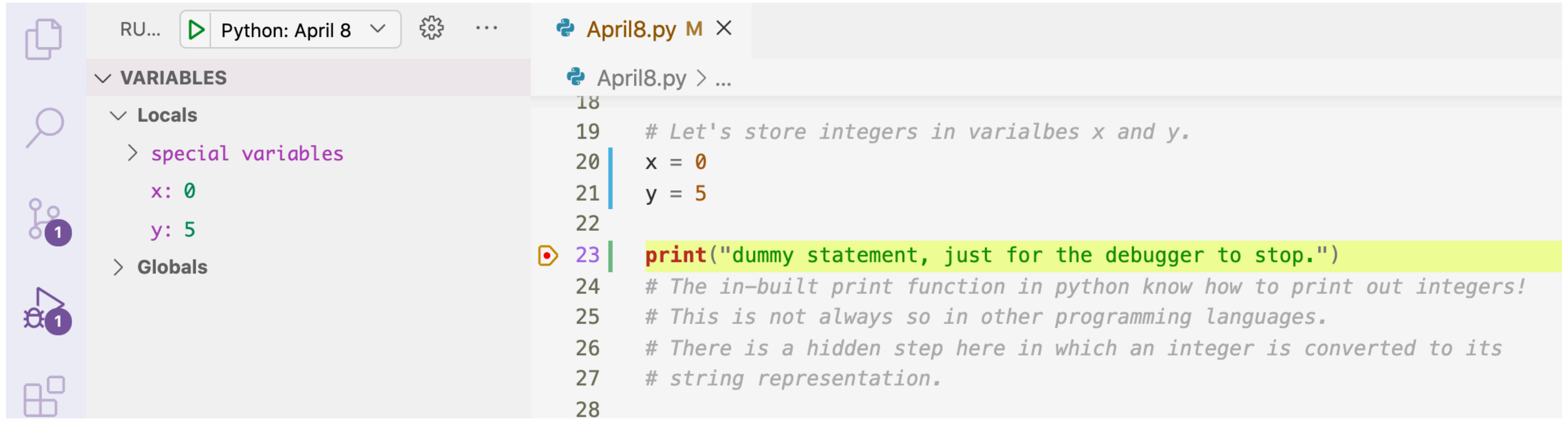
Debugging
In VS Code



April8.py > ...

```python
# This is a comment. This line won't be executed.
# Demo for April 8.

# Python does not require that you declare any functions or classes.
# You can just enter statements and the interpreter will execute them.

print("hello")

```

- Checklist:
  - Set up the debugger with a running config
  - Entry point of a program
  - Stepping into a function
  - Stepping over statements
  - Inspecting program state
  - Stopping and resuming/restarting



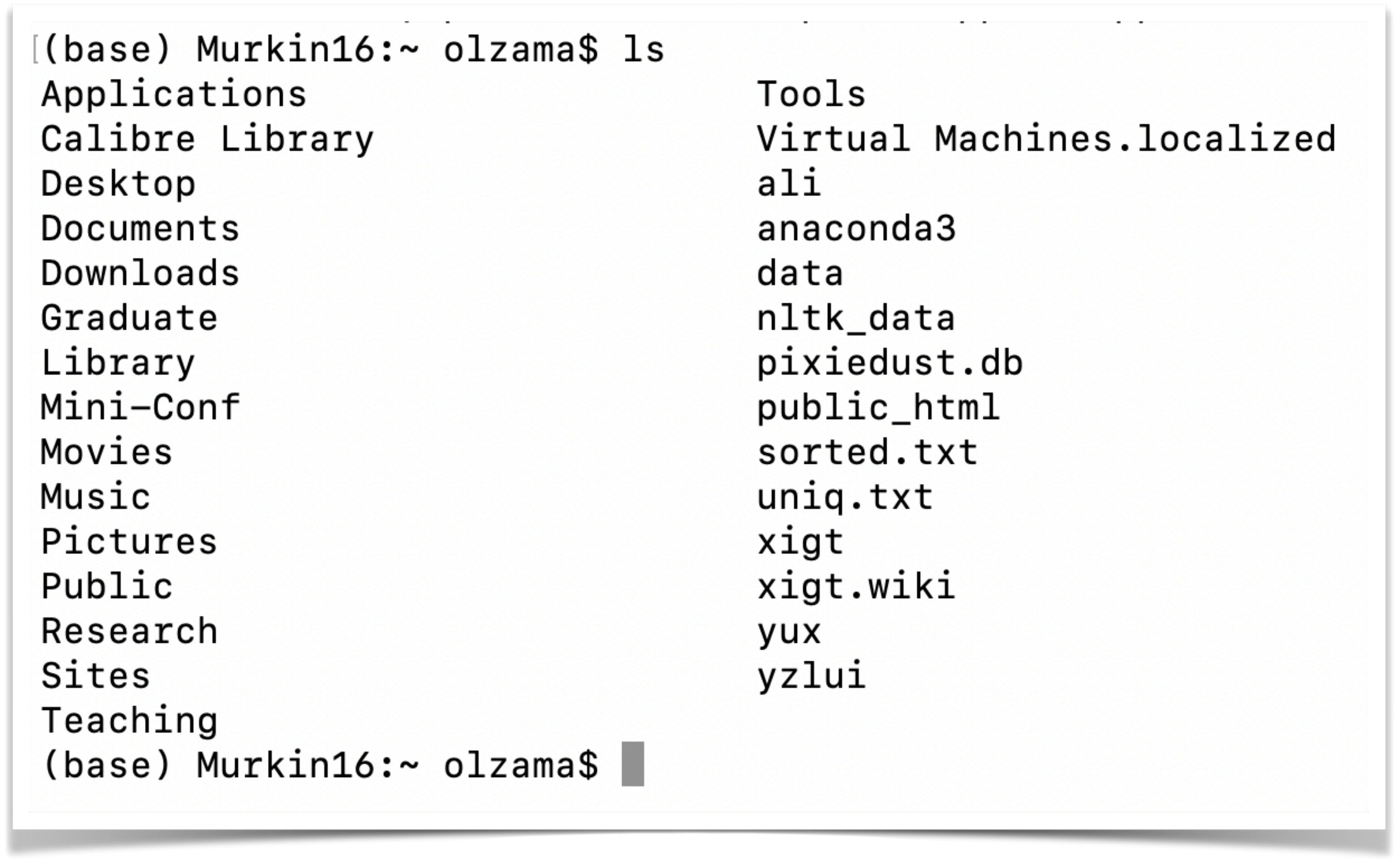
```
x: 0
y: 5
```

April8.py > ...

```python
# Let's store integers in varialbes x and y.
x = 0
y = 5

print("dummy statement, just for the debugger to stop.")
# The in-built print function in python know how to print out integers!
# This is not always so in other programming languages.
# There is a hidden step here in which an integer is converted to its
# string representation.

```

Command line

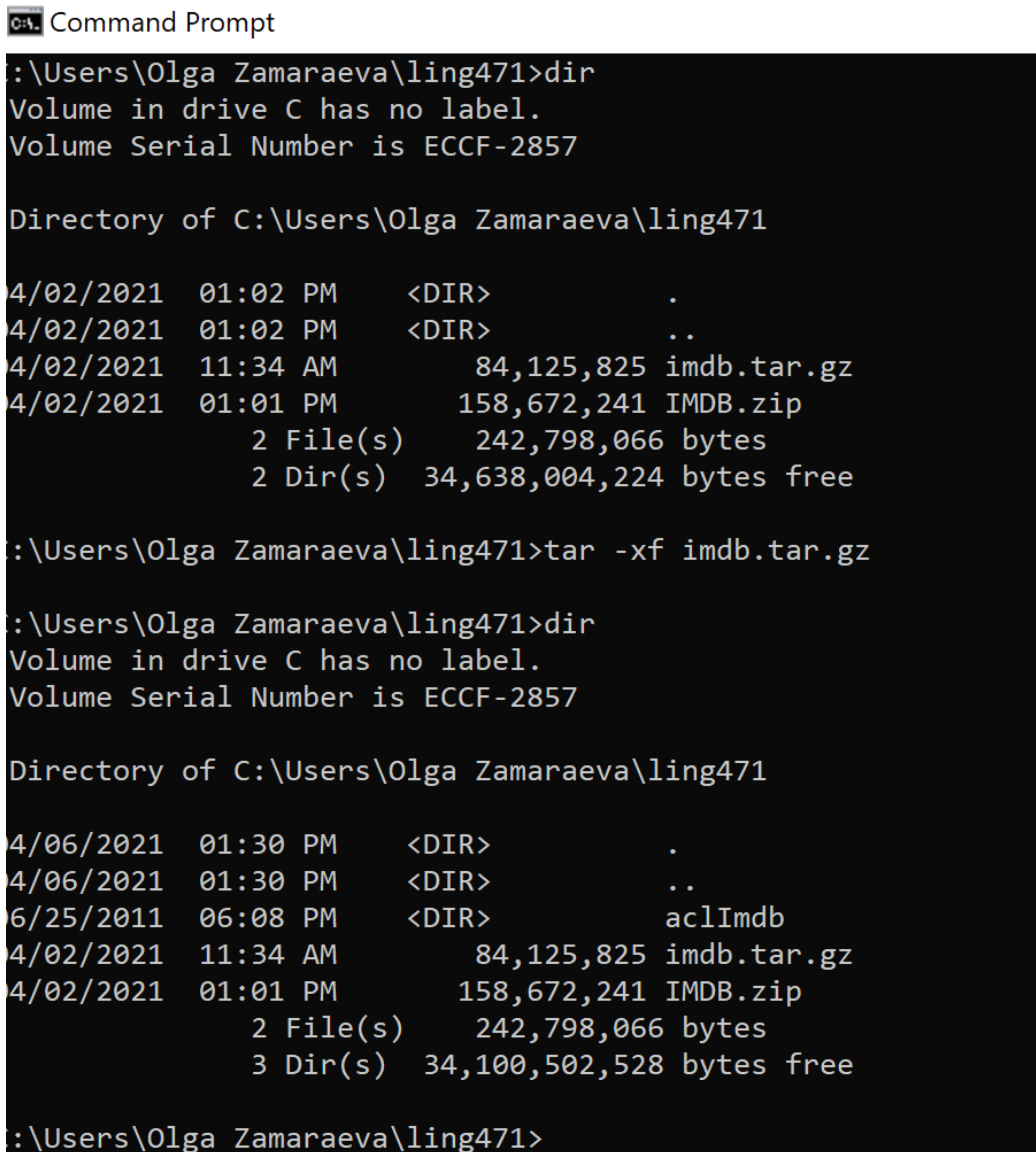
- A tool to give operating system instructions
  - e.g.:
  - ***Run** python version 2.6 **on the code** stored in a file located in Documents/program.py, and **pass** that program a folder located in ~/data and a file located in ~/models/neural-model.h5 **as the arguments**; **store the output** in a file called ~/results-h5.txt. Then **sort** the results in decreasing order and **store** in file called ~/results-h5-sorted.txt*
  - Need **special language** to pass that sequence of commands



```
(base) Murkin16:~ olzama$ ls
Applications                    Tools
Calibre Library                 Virtual Machines.localized
Desktop                         ali
Documents                       anaconda3
Downloads                       data
Graduate                        nltk_data
Library                         pixiedust.db
Mini-Conf                       public_html
Movies                          sorted.txt
Music                           uniq.txt
Pictures                        xigt
Public                          xigt.wiki
Research                        yux
Sites                           yzlui
Teaching
(base) Murkin16:~ olzama$
```

Command line

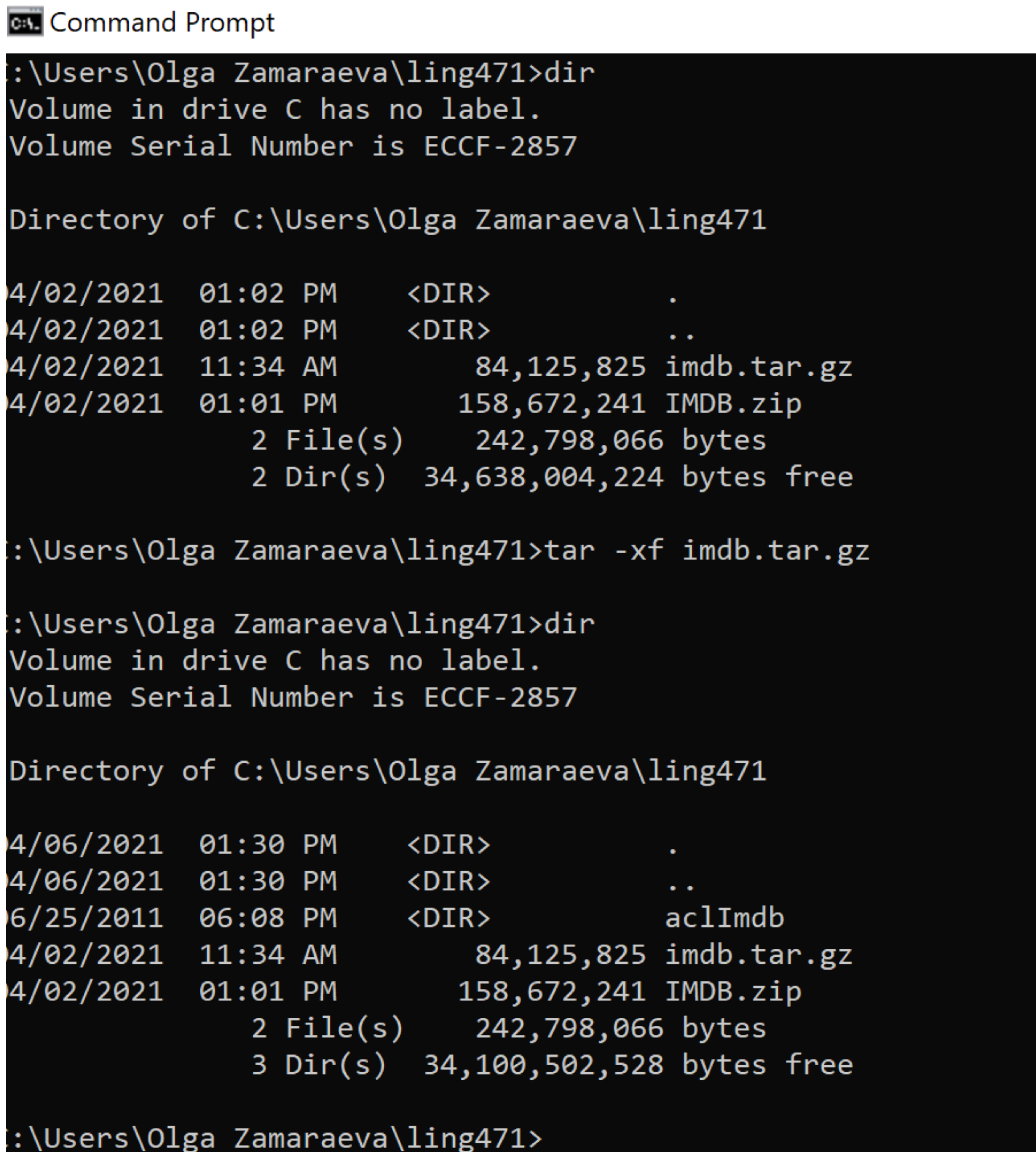
- A tool to give operating system instructions
  - e.g. *Extract contents from ~/Download/ imdb.tar.gz*
  - Windows cmd prompt:
    - dir: **list** current directory
    - tar -xf filename: **extract** files from archive
    - dir: list current directory again
      - The aclImdb folder is **new**



```
:\Users\Olga Zamaraeva\ling471>dir
Volume in drive C has no label.
Volume Serial Number is ECCF-2857

Directory of C:\Users\Olga Zamaraeva\ling471

4/02/2021  01:02 PM    <DIR>          .
4/02/2021  01:02 PM    <DIR>          ..
4/02/2021  11:34 AM        84,125,825 imdb.tar.gz
4/02/2021  01:01 PM       158,672,241 IMDB.zip
              2 File(s)    242,798,066 bytes
              2 Dir(s)  34,638,004,224 bytes free

:\Users\Olga Zamaraeva\ling471>tar -xf imdb.tar.gz

:\Users\Olga Zamaraeva\ling471>dir
Volume in drive C has no label.
Volume Serial Number is ECCF-2857

Directory of C:\Users\Olga Zamaraeva\ling471

4/06/2021  01:30 PM    <DIR>          .
4/06/2021  01:30 PM    <DIR>          ..
6/25/2011  06:08 PM    <DIR>          aclImdb
4/02/2021  11:34 AM        84,125,825 imdb.tar.gz
4/02/2021  01:01 PM       158,672,241 IMDB.zip
              2 File(s)    242,798,066 bytes
              3 Dir(s)  34,100,502,528 bytes free

:\Users\Olga Zamaraeva\ling471>
```

Command line
Unix-based vs Windows

- Unix-based:
  - "Terminal" for command line; *bash* language
    - e.g. "ls" to **list** a directory
    - "cd" to **change** directory
- Windows:
  - "Command prompt"; *batch* language
    - e.g. "dir" to **list** a directory
    - Still "cd" to **change** directory
    - Linux *bash* also available as an additional feature
- Demo

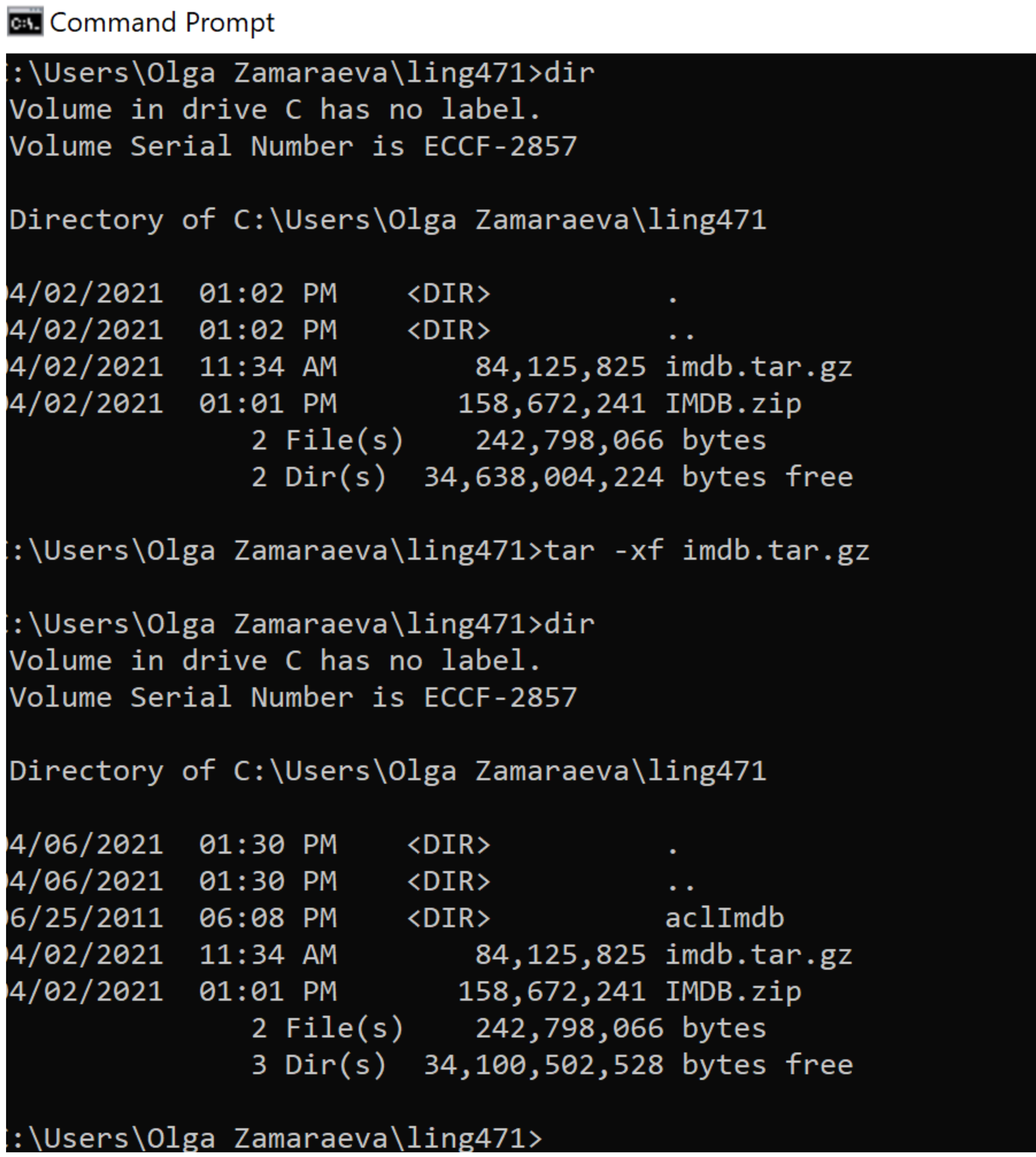


```
:\Users\Olga Zamaraeva\ling471>dir
Volume in drive C has no label.
Volume Serial Number is ECCF-2857

Directory of C:\Users\Olga Zamaraeva\ling471

4/02/2021  01:02 PM    <DIR>          .
4/02/2021  01:02 PM    <DIR>          ..
4/02/2021  11:34 AM        84,125,825 imdb.tar.gz
4/02/2021  01:01 PM       158,672,241 IMDB.zip
              2 File(s)    242,798,066 bytes
              2 Dir(s)  34,638,004,224 bytes free

:\Users\Olga Zamaraeva\ling471>tar -xf imdb.tar.gz

:\Users\Olga Zamaraeva\ling471>dir
Volume in drive C has no label.
Volume Serial Number is ECCF-2857

Directory of C:\Users\Olga Zamaraeva\ling471

4/06/2021  01:30 PM    <DIR>          .
4/06/2021  01:30 PM    <DIR>          ..
6/25/2011  06:08 PM    <DIR>          aclImdb
4/02/2021  11:34 AM        84,125,825 imdb.tar.gz
4/02/2021  01:01 PM       158,672,241 IMDB.zip
              2 File(s)    242,798,066 bytes
              3 Dir(s)  34,100,502,528 bytes free

:\Users\Olga Zamaraeva\ling471>
```

Command line

Common commands Linux-Mac/Windows

- List directory: ls/dir
- Change directory: cd
- Create directory: mkdir
- Copy file(directory): cp (cp -r)/copy (xcopy)
- Move (rename!) file or directory: mv/move
- Run a program (with arguments)
  - e..g "python" to run python
- Ask the system, what the current path is: pwd/echo %cd%
- More advanced:
  - Connect to a remote machine: ssh
  - Open a text file in a command-line editor
    - And then try to exit it :)

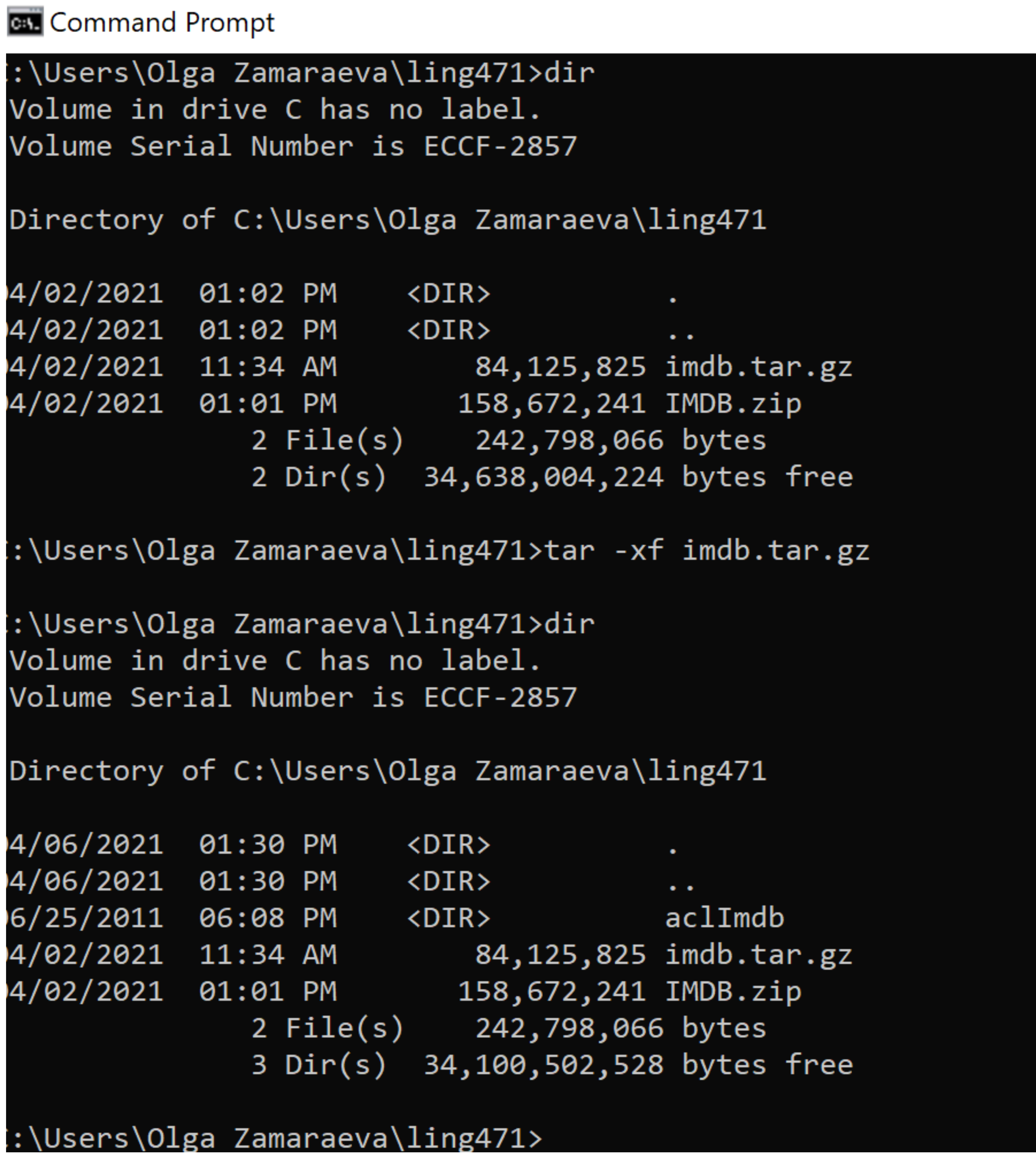

Command Prompt

```
:\Users\Olga Zamaraeva\ling471>dir
Volume in drive C has no label.
Volume Serial Number is ECCF-2857

Directory of C:\Users\Olga Zamaraeva\ling471

4/02/2021  01:02 PM    <DIR>          .
4/02/2021  01:02 PM    <DIR>          ..
4/02/2021  11:34 AM        84,125,825 imdb.tar.gz
4/02/2021  01:01 PM       158,672,241 IMDB.zip
              2 File(s)    242,798,066 bytes
              2 Dir(s)  34,638,004,224 bytes free

:\Users\Olga Zamaraeva\ling471>tar -xf imdb.tar.gz

:\Users\Olga Zamaraeva\ling471>dir
Volume in drive C has no label.
Volume Serial Number is ECCF-2857

Directory of C:\Users\Olga Zamaraeva\ling471

4/06/2021  01:30 PM    <DIR>          .
4/06/2021  01:30 PM    <DIR>          ..
6/25/2011  06:08 PM    <DIR>          aclImdb
4/02/2021  11:34 AM        84,125,825 imdb.tar.gz
4/02/2021  01:01 PM       158,672,241 IMDB.zip
              2 File(s)    242,798,066 bytes
              3 Dir(s)  34,100,502,528 bytes free

:\Users\Olga Zamaraeva\ling471>
```

The great power of command line

…the up-arrow and the tab

- The terminal stores previous commands
  - Repeat a complex command simply by hitting the up-arrow
- The terminal knows which paths are available from the current directory
  - Autocomplete paths by hitting TAB

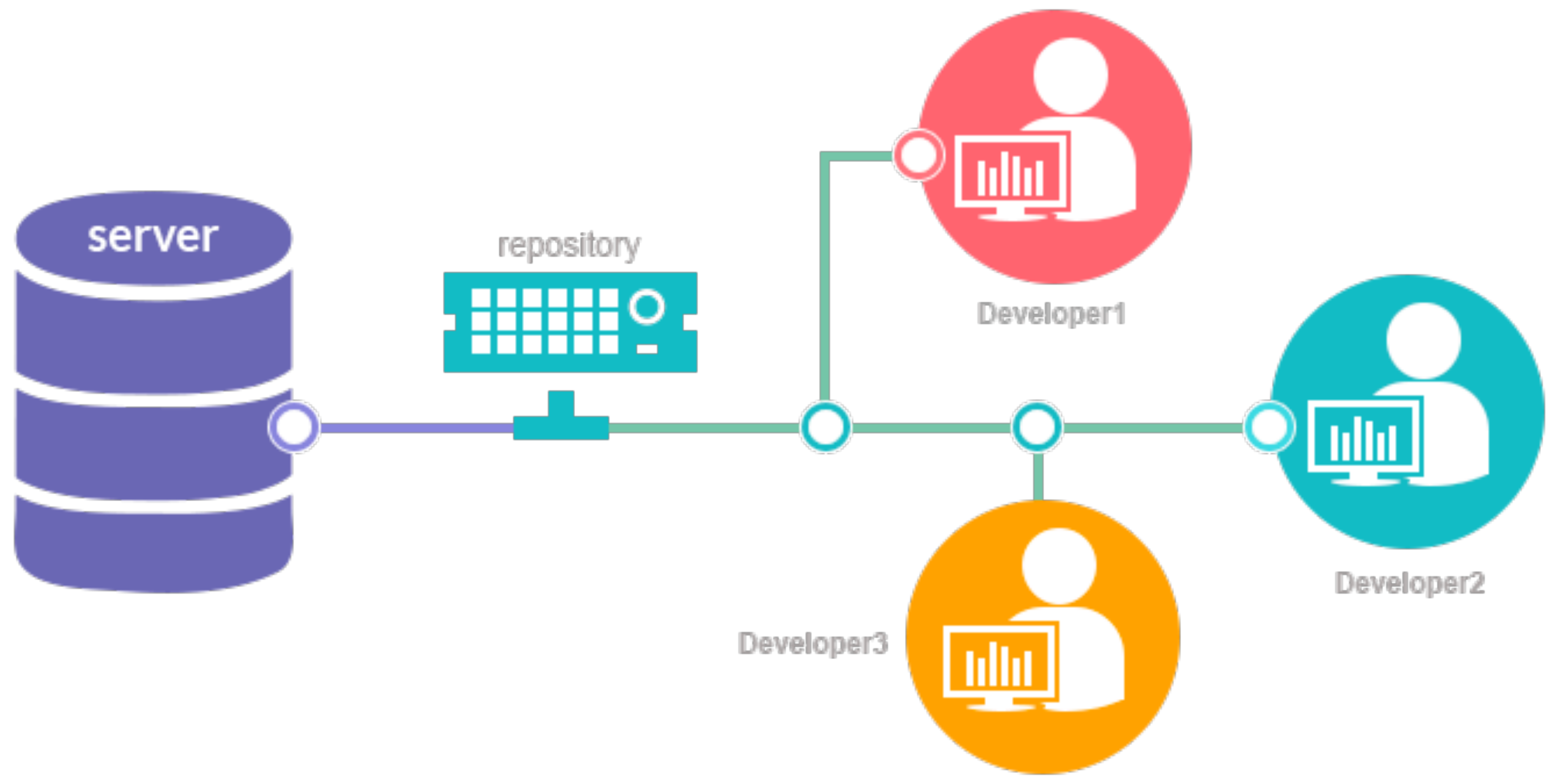


```
:\Users\Olga Zamaraeva\ling471>dir
Volume in drive C has no label.
Volume Serial Number is ECCF-2857

Directory of C:\Users\Olga Zamaraeva\ling471

4/02/2021  01:02 PM    <DIR>          .
4/02/2021  01:02 PM    <DIR>          ..
4/02/2021  11:34 AM        84,125,825 imdb.tar.gz
4/02/2021  01:01 PM       158,672,241 IMDB.zip
              2 File(s)    242,798,066 bytes
              2 Dir(s)  34,638,004,224 bytes free

:\Users\Olga Zamaraeva\ling471>tar -xf imdb.tar.gz

:\Users\Olga Zamaraeva\ling471>dir
Volume in drive C has no label.
Volume Serial Number is ECCF-2857

Directory of C:\Users\Olga Zamaraeva\ling471

4/06/2021  01:30 PM    <DIR>          .
4/06/2021  01:30 PM    <DIR>          ..
6/25/2011  06:08 PM    <DIR>          aclImdb
4/02/2021  11:34 AM        84,125,825 imdb.tar.gz
4/02/2021  01:01 PM       158,672,241 IMDB.zip
              2 File(s)    242,798,066 bytes
              3 Dir(s)  34,100,502,528 bytes free

:\Users\Olga Zamaraeva\ling471>
```

Version and source control

Git

- Keep snapshots of all the changes
  - Organized in batches, with comments
- Go back to any version any time
- A life-saver!
  - Originally for groups of people working on the same project
  - Essential for sole developers as well
    - As a general back up
    - As a way of having stable versions you can rely on

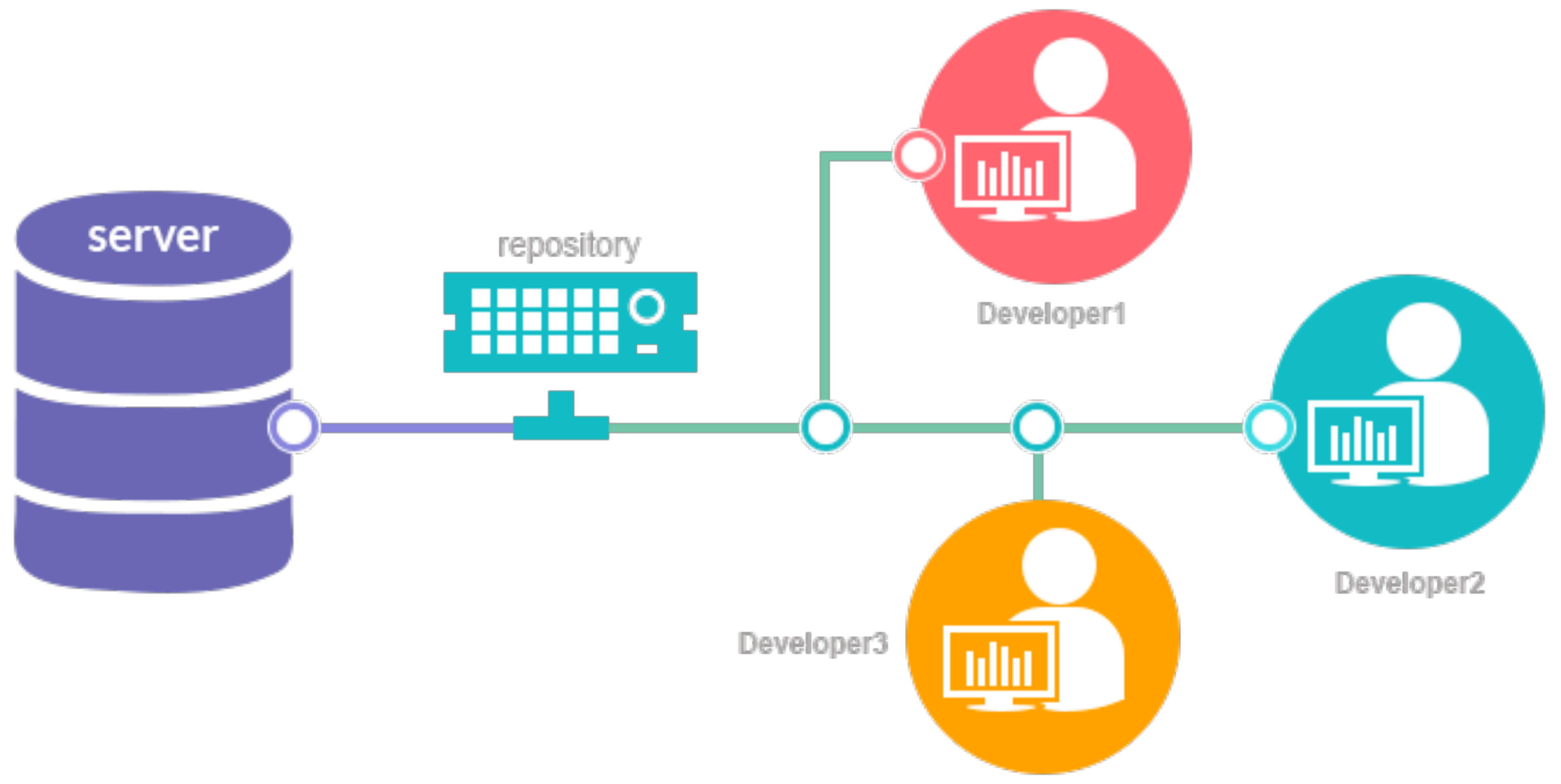


https://medium.com/@vemulasrinivas2505/version-control-systems-74375eb48961

Version and source control

Git repository structure

- Remote repository
  - e.g. on Github
- Local repositories
  - The one you have on your computer
  - Other people's, if they also cloned the same remote

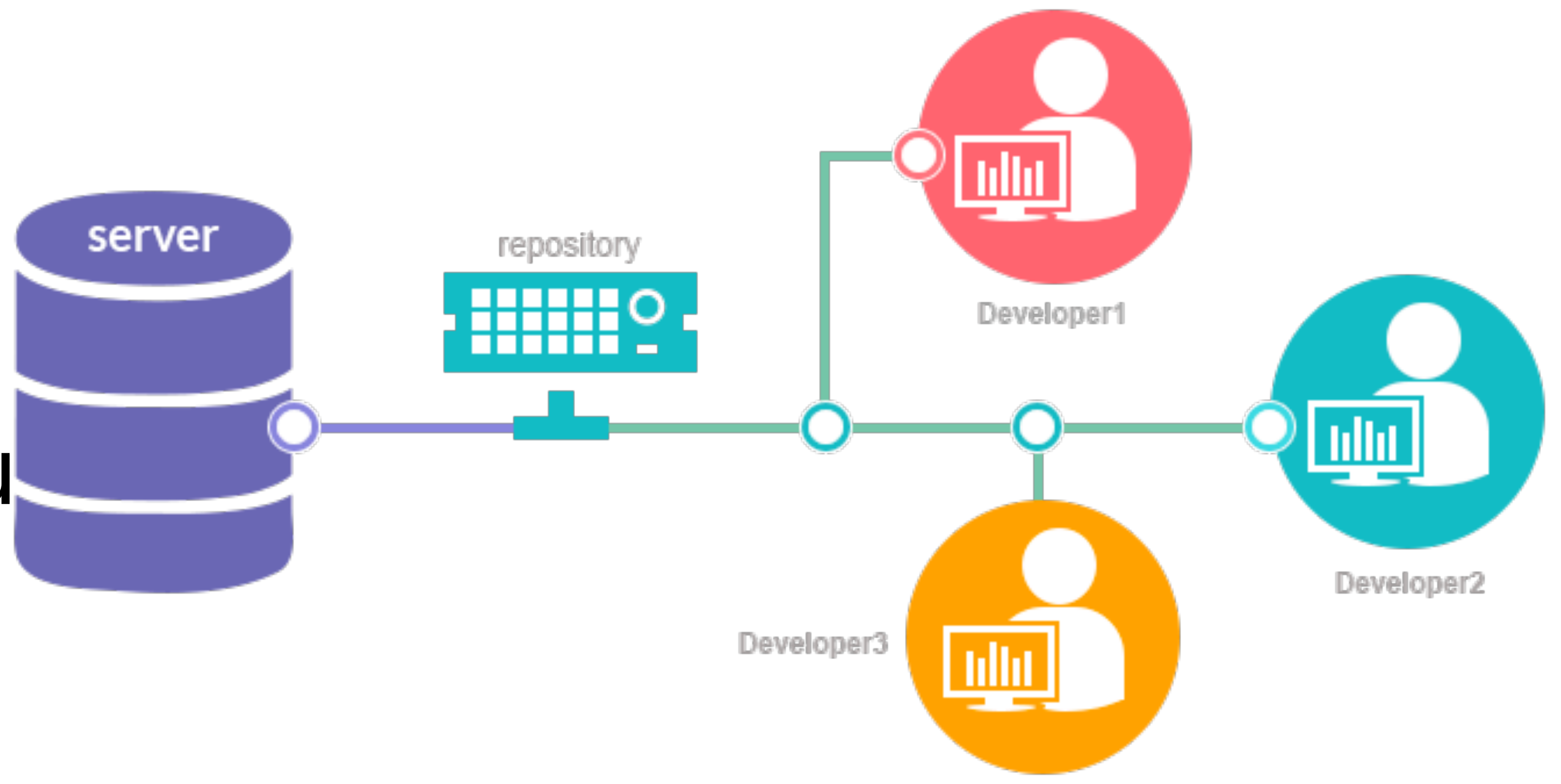


https://medium.com/@vemulasrinivas2505/version-control-systems-74375eb48961

Version and source control

Git repository structure

- You can clone remote repositories
- ...or fork them
    - To get starter code for future assignments, you will fork the "skeleton" repositories we created for you
        - This means you cannot push back into them
        - They become your own repository copies
            - Note: You wont **submit** to the forked copy
            - Always carefully read instructions what to submit how

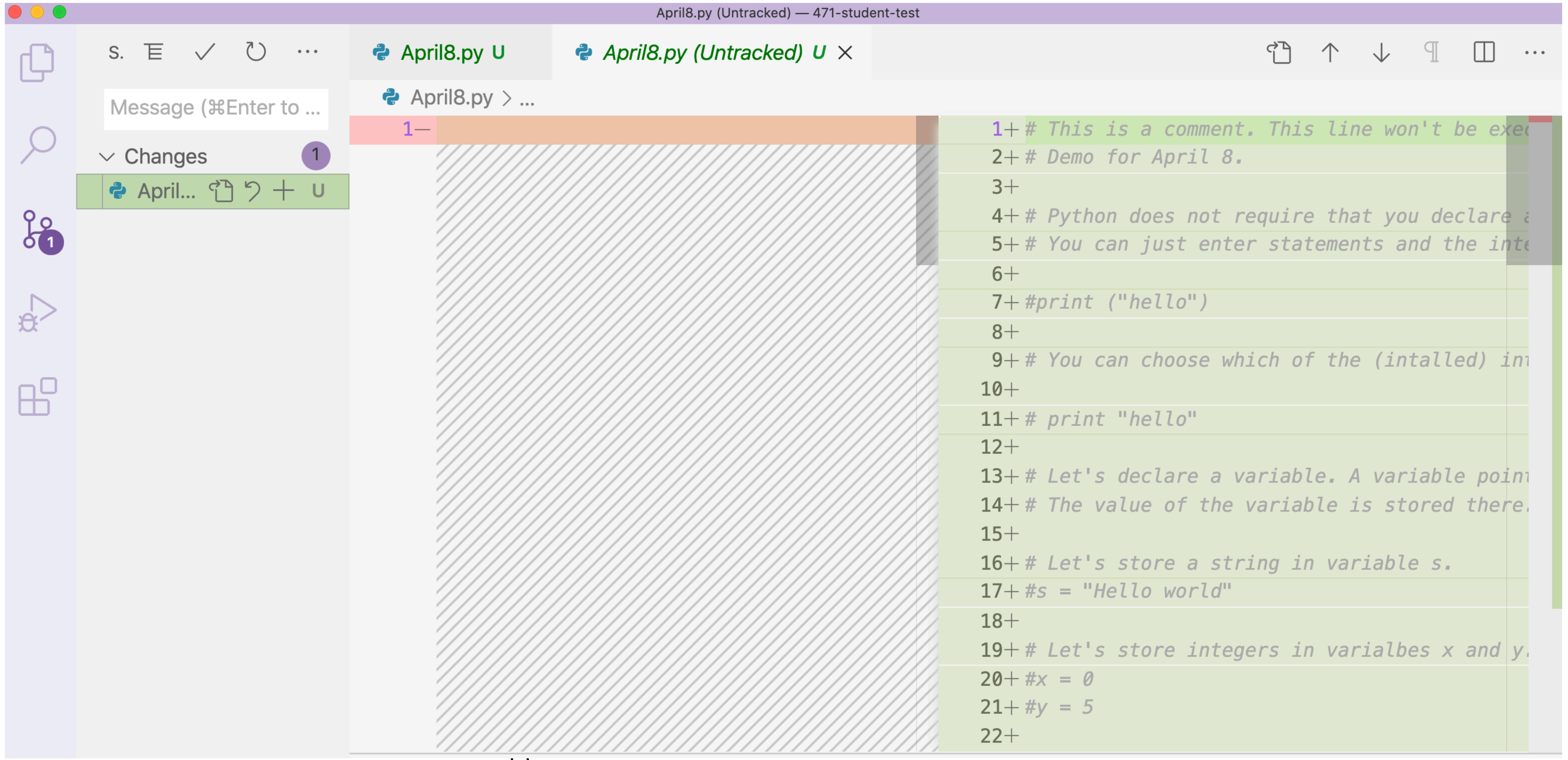


https://medium.com/@vemulasrinivas2505/version-control-systems-74375eb48961

Version control

Staging changes

- Pick which changes to include in a version
  - Note: while some changes are unaccounted for, won't be able to integrate stuff from other versions



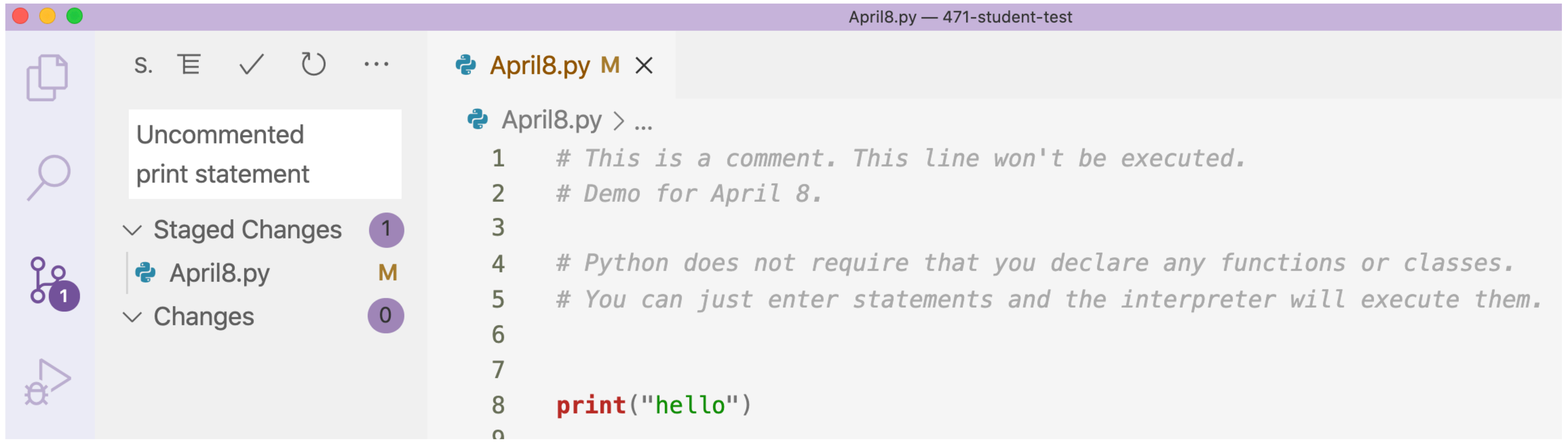
```
-
+# This is a comment. This line won't be exe
+# Demo for April 8.
+
+# Python does not require that you declare a
+# You can just enter statements and the inte
+
+#print ("hello")
+
+# You can choose which of the (intalled) int
+
+# print "hello"
+
+# Let's declare a variable. A variable poin
+# The value of the variable is stored there.
+
+# Let's store a string in variable s.
+#s = "Hello world"
+
+# Let's store integers in varialbes x and y.
+#x = 0
+#y = 5
+
```

Version control

Committing changes

- Once you picked which changes you want to **commit** them to the new version
  - Screenshot: 1 **staged** change ready for commit
- Commit (in git) **does not** mean the changes went to the remote repository!
  - And that is a **good** thing!
  - Pushing to remote affects

other people



Uncommented print statement

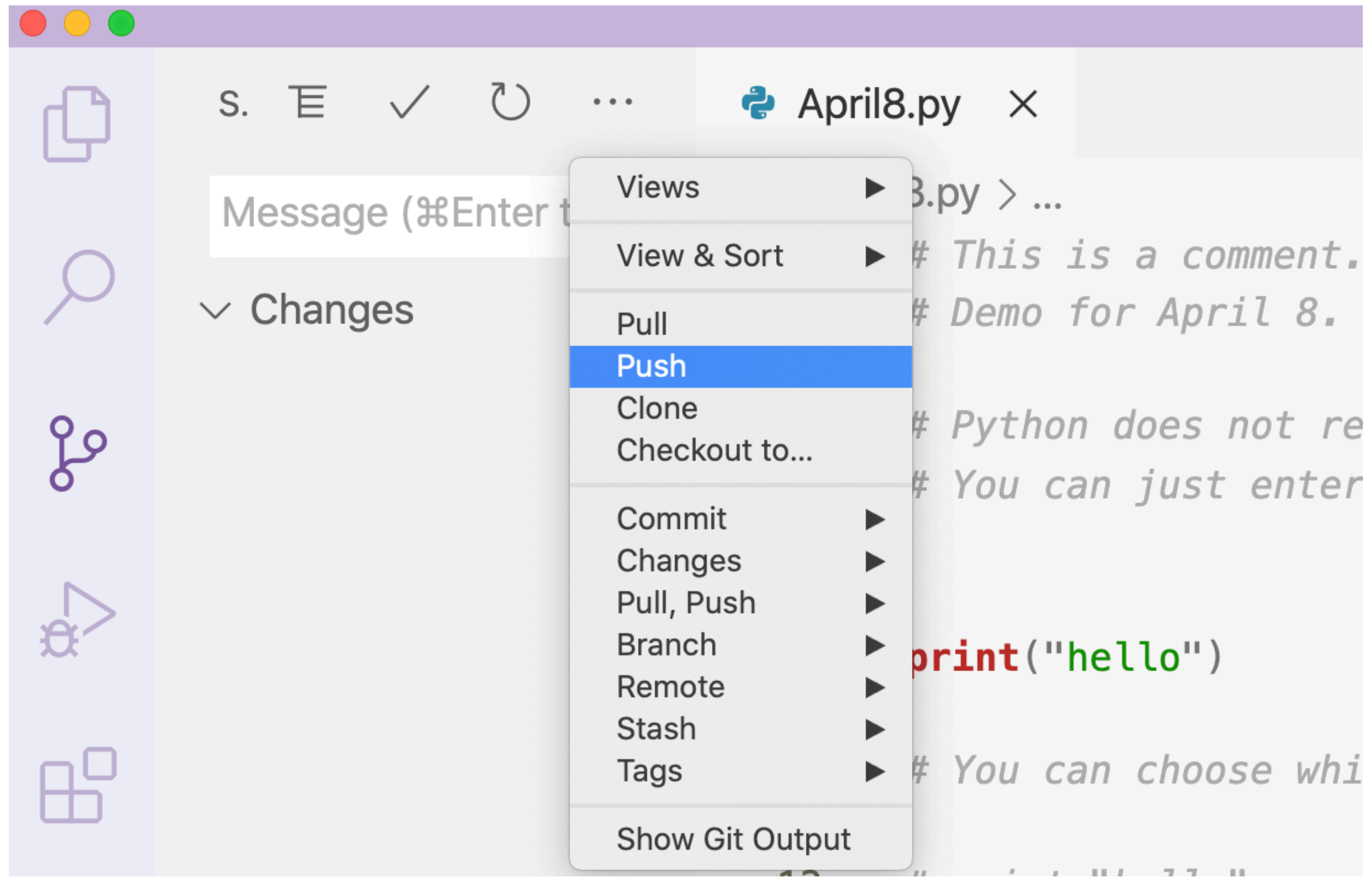
Staged Changes 1
April8.py M
Changes 0

```
# This is a comment. This line won't be executed.
# Demo for April 8.

# Python does not require that you declare any functions or classes.
# You can just enter statements and the interpreter will execute them.


print("hello")
```

Version control

Pushing and pulling changes

- Once you committed some changes, send them to the remote repository
- Now, others will only be able to push their changes to it if there are no conflicts
- Likewise, you will be able to pull others' changes but then may need to deal with conflicts, if you made changes to the same lines
- Do some **demo:**
  - Adding files, staging changes, committing, pushing

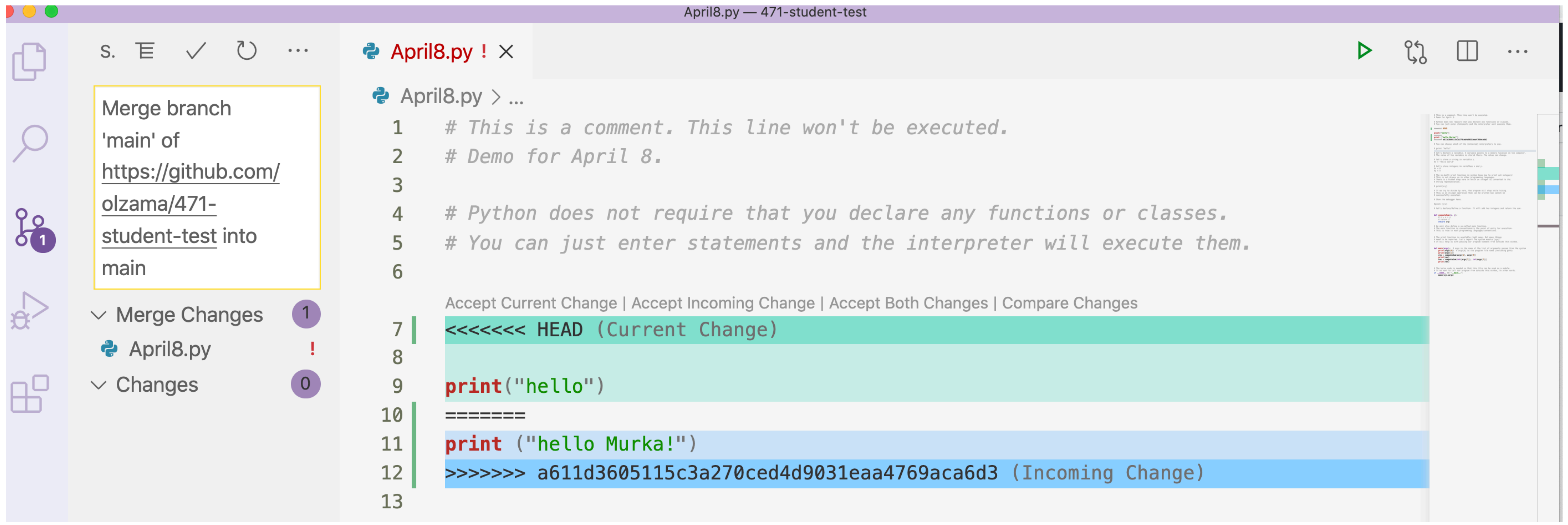


```
# This is a comment.
# Demo for April 8.

# Python does not re
# You can just enter

print("hello")

# You can choose whi
```

Version control

Conflicts and merging

- This can be unpleasant :(
- We won’t have to do much of this
- Idea:
    - Remote has its own opinion about some lines
    - Local has its own opinion
    - You need to go into the file and decide what you want
        - Sometime it’s pretty manual work :/
        - Goal: only leave meaningful python lines
- Do some **demo** here.
- NB: Git GUIs often fail to resolve conflicts
    - Command line to the rescue!


April8.py — 471-student-test
Merge branch 'main' of https://github.com/olzama/471-student-test into main
Merge Changes
April8.py
Changes
April8.py !
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes

```
# This is a comment. This line won't be executed.
# Demo for April 8.

# Python does not require that you declare any functions or classes.
# You can just enter statements and the interpreter will execute them.

<<<<<<< HEAD (Current Change)

print("hello")
=======
print ("hello Murka!")
>>>>>>> a611d3605115c3a270ced4d9031eaa4769aca6d3 (Incoming Change)

```

Questions?