Lecture 30:

WORKING WITH FILES

CSC111: Introduction to CS through Programming

R. Jordan Crouser

Assistant Professor of Computer Science

Smith College

Announcements: Smithies in CS events

Alumni in CS Discuss Identity & Tech

6 pm - 7pm

REGISTER WITH QR CODE OR LINK BELOW

BLACK SMITHIES IN CS
NOV 5TH, 2020

HTTPS://TINYURL.COM/Y5NJQTDL

1ST GEN SMITHIES IN CS
NOV 6TH, 2020

HTTPS://TINYURL.COM/Y5ELYQ5X

ASIAN SMITHIES IN CS
NOV 12TH, 2020

HTTPS://TINYURL.COM/Y5Y72HDR

LGBTQ+ SMITHIES IN CS
NOV 13TH, 2020

HTTPS://TINYURL.COM/Y56VVHRJ

LATINX SMITHIES IN CS
NOV 19TH, 2020

HTTPS://TINYURL.COM/Y6GHULMO

INTERNATIONAL SMITHIES IN CS
NOV 20TH, 2020

HTTPS://TINYURL.COM/Y397Y2ZW

NON-TRADITIONAL SMITHIES IN CS
NOV 23RD, 2020

HTTPS://TINYURL.COM/Y5UVOJK7

Announcements: auto-extension on A6

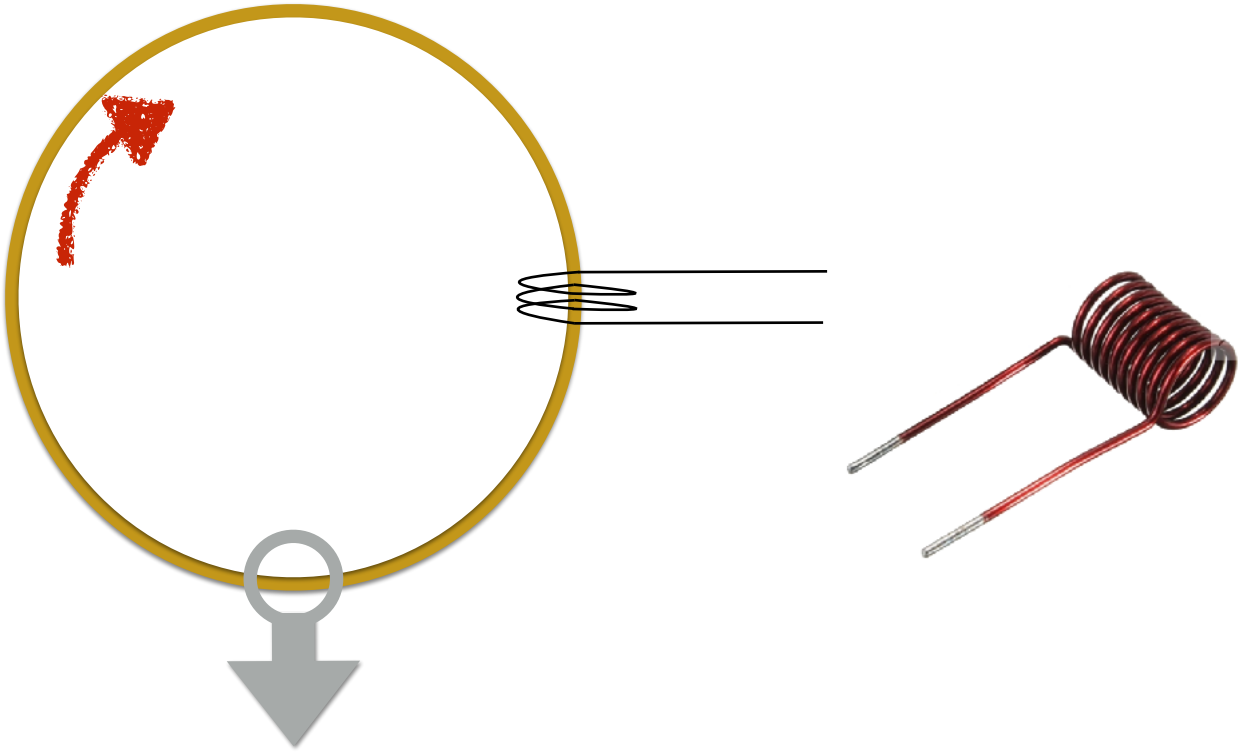
- This week is a rough one for a lot of us
- A6 now due **Monday 11/9** for everyone (+3 days)
- Take care of yourselves: remember that you have a free from of one lab and one assignment

Outline

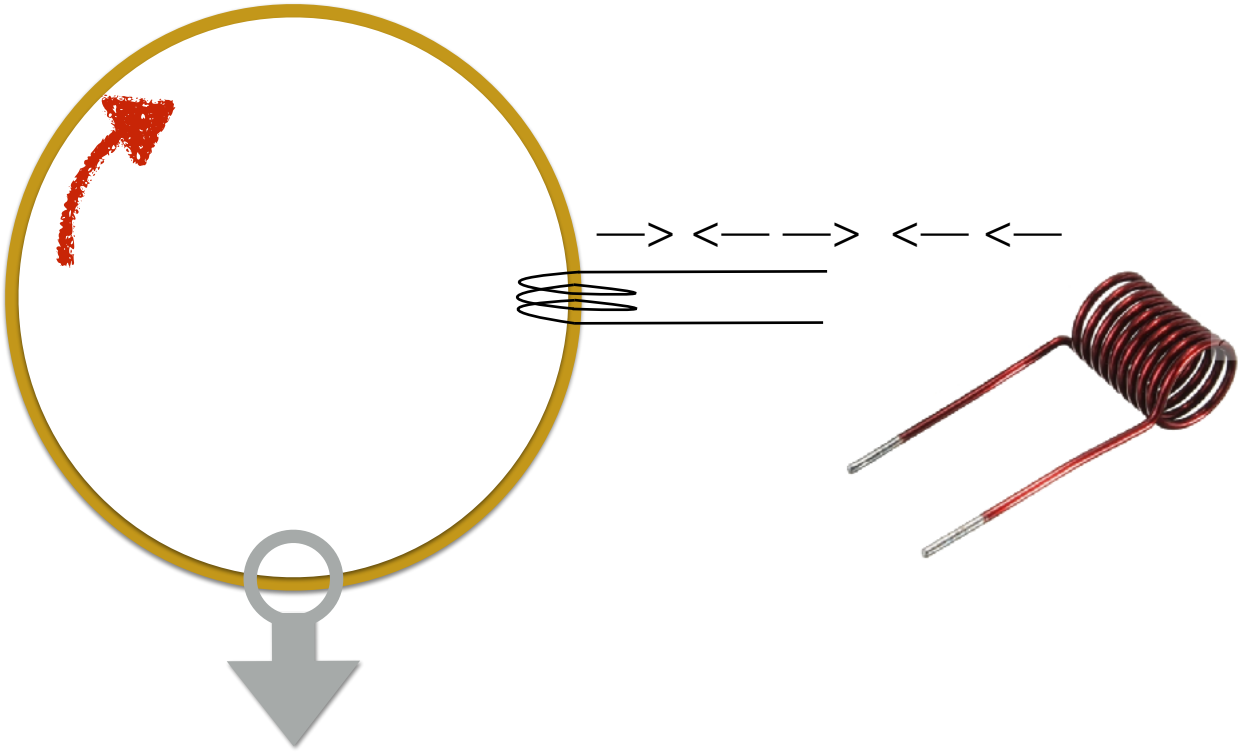
- ✓Announcements
- Working with files
  - what is a file?
  - reading data in
  - writing data out
  - *some details about the final project*
- Lab: Sorting
- Life Skill #5: Code Diagrams
- Useful python packages

RECAP: how does a hard drive work?

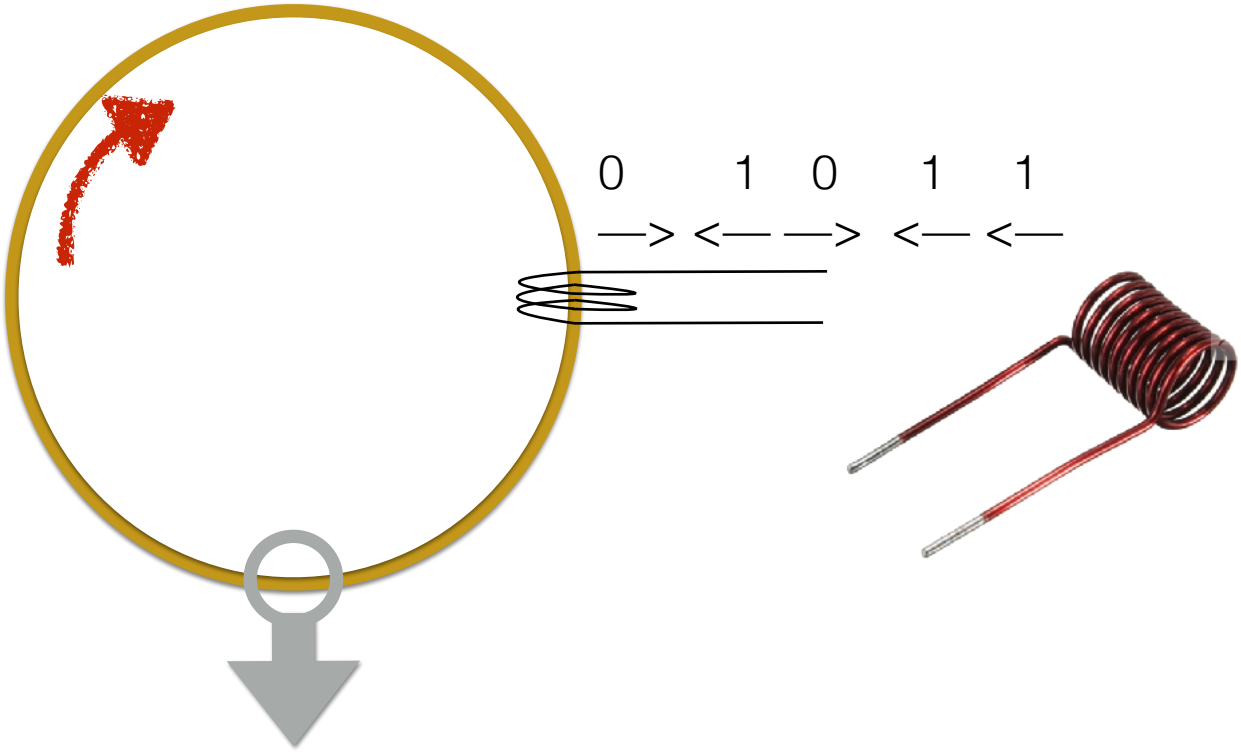
RECAP: how does a hard drive work?



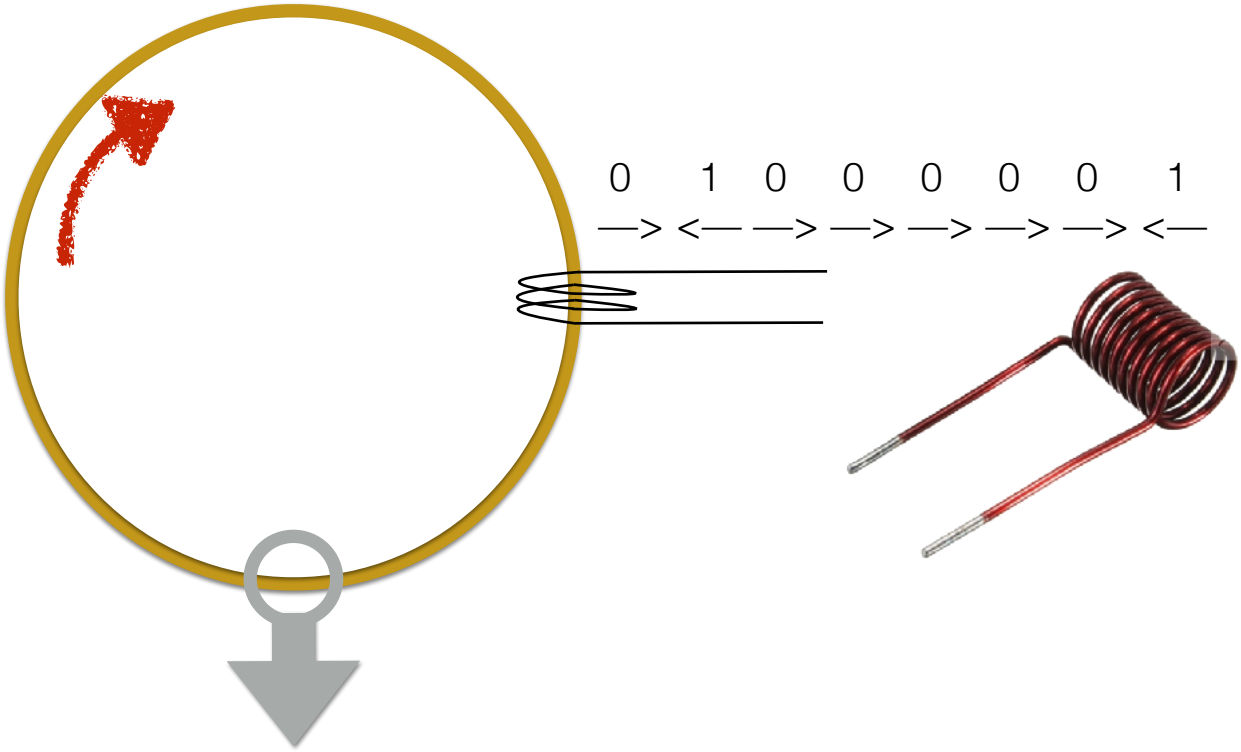
RECAP: how does a hard drive work?



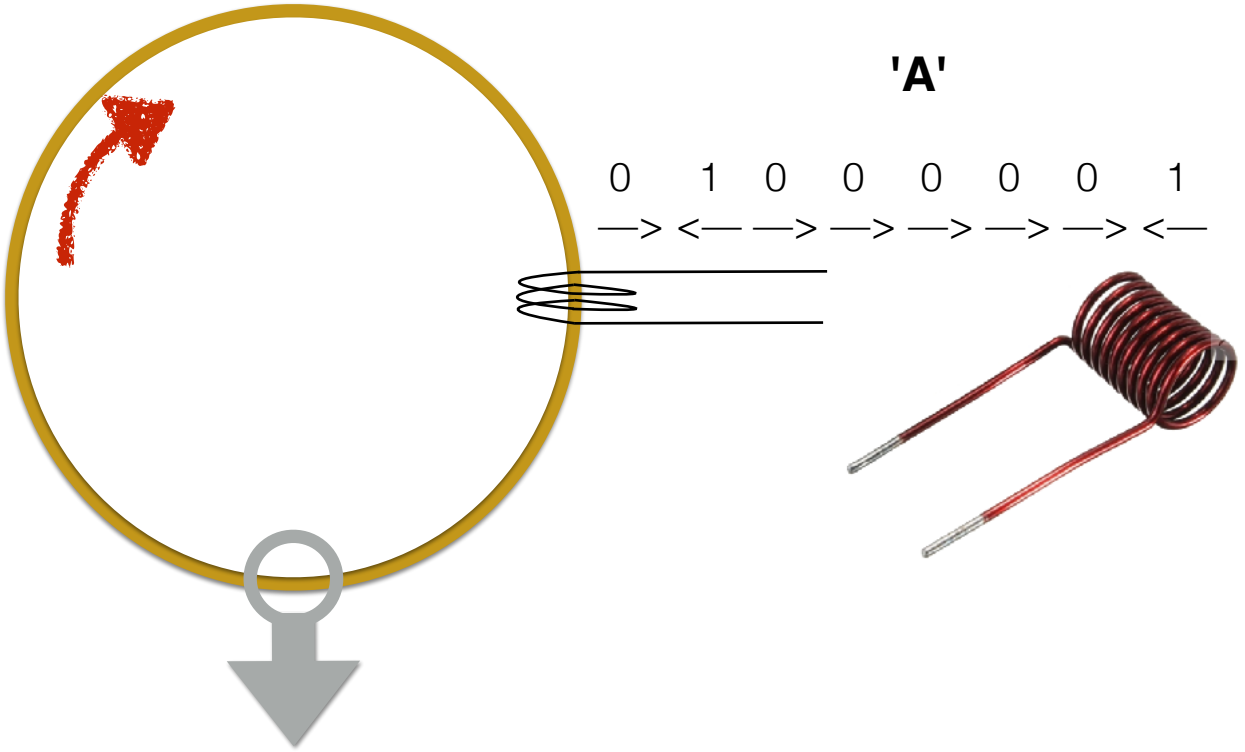
RECAP: how does a hard drive work?



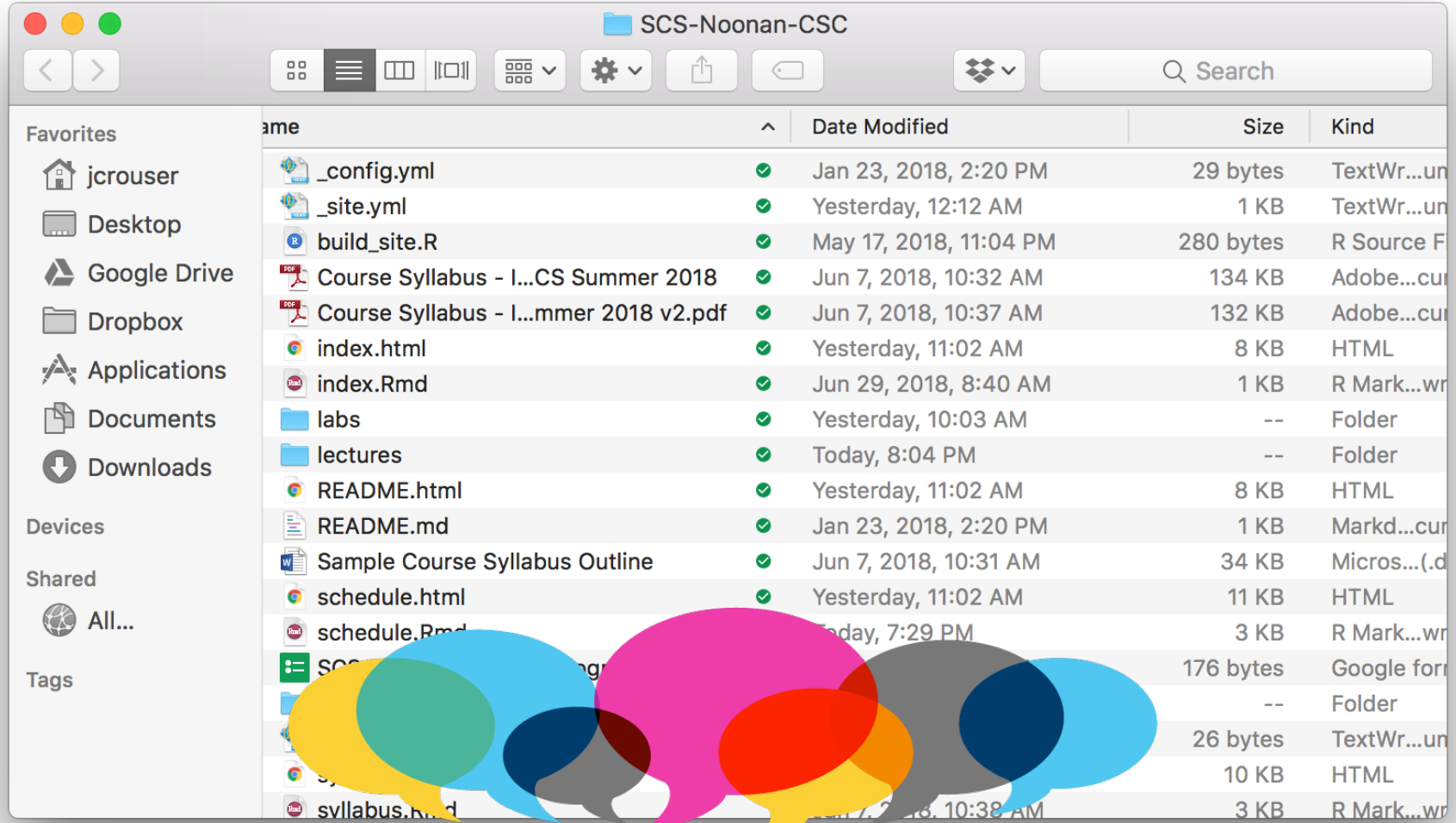
RECAP: how does a hard drive work?



RECAP: how does a hard drive work?

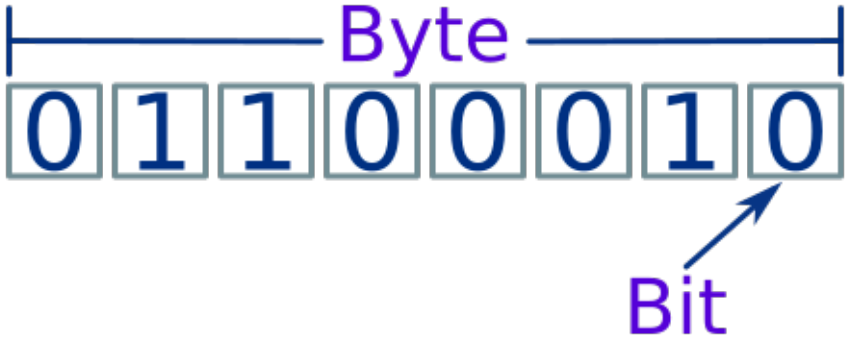


What is a **file**?



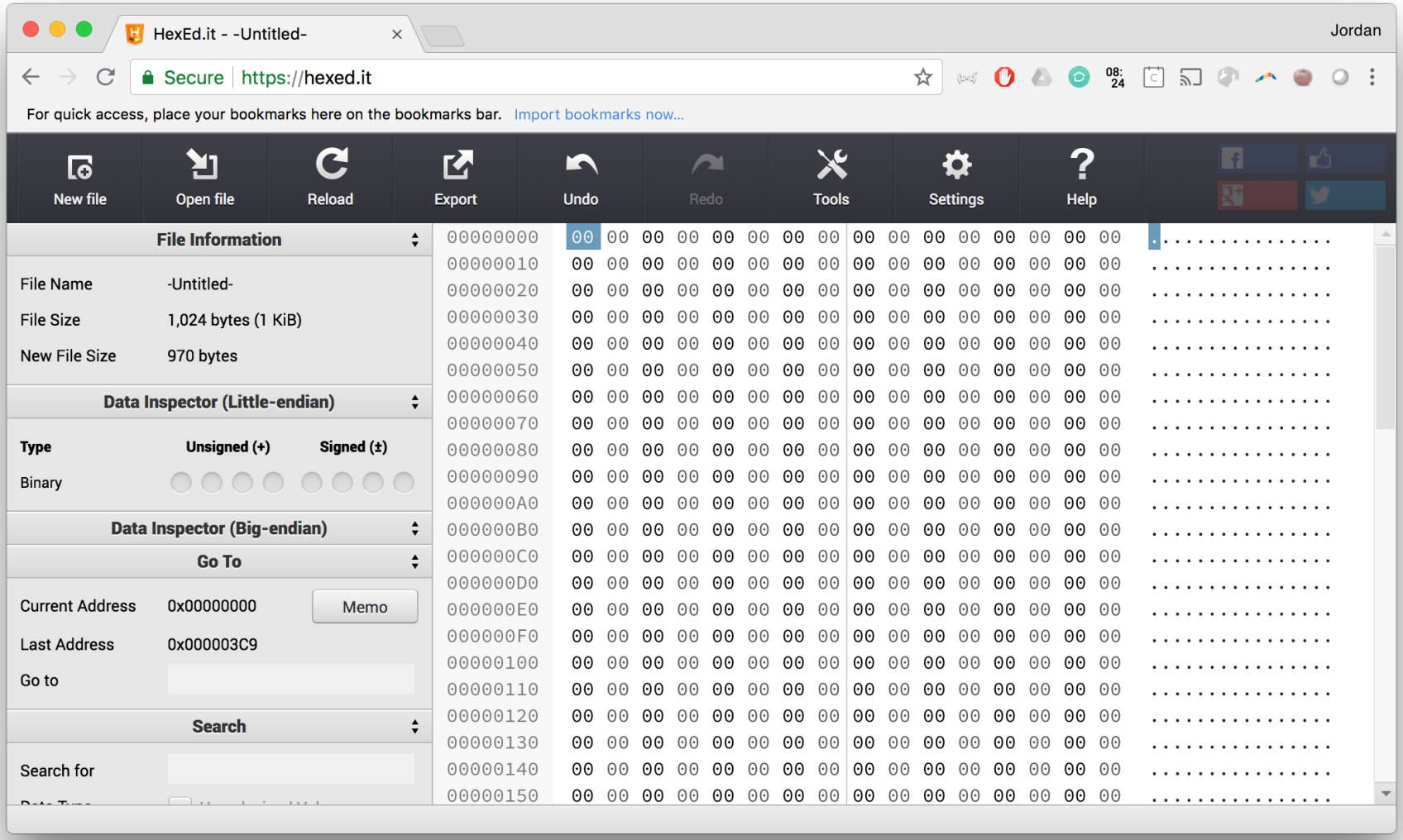
Working with **files**

- Containers of **bits**, organized into **bytes**



- Could represent text, images, music, movies, programs, applications, list of files (folders)… but underneath, they're all the same: 0s and 1s
- We'll start by playing with text files

Demo: looking inside a text file



Reading a text file

- In order to bring data stored in a text file into a Python program, we need to **read** it:

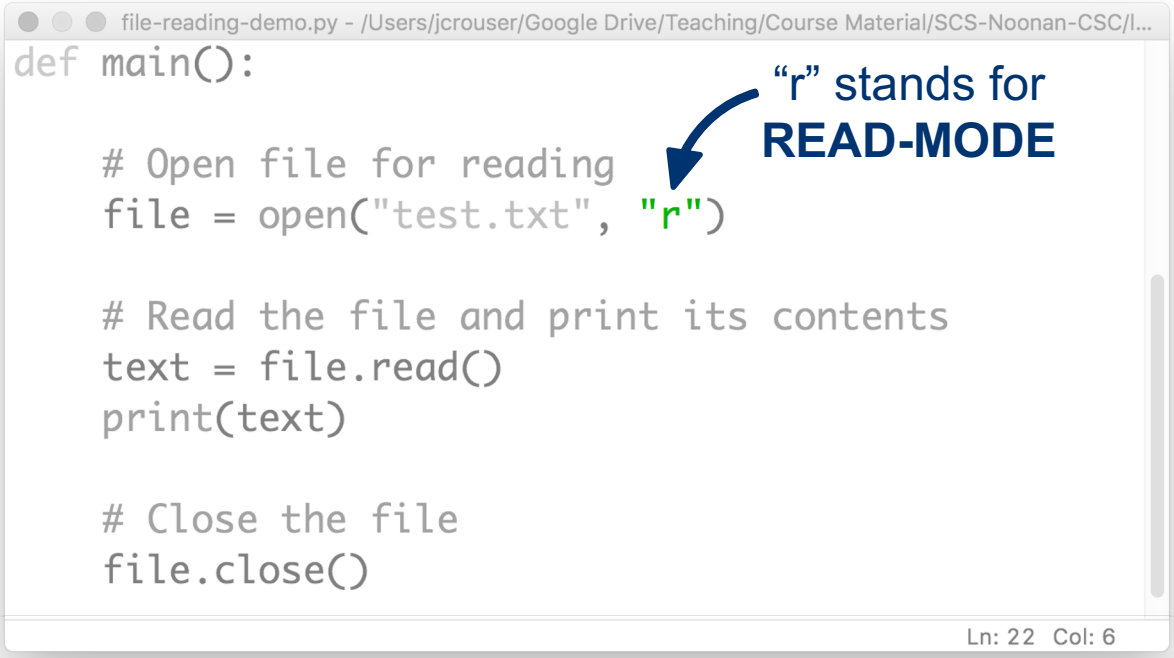
file-reading-demo.py - /Users/jcrouser/Google Drive/Teaching/Course Material/SCS-Noonan-CSC/l...

```
def main():

    # Open file for reading
    file = open("test.txt", "r")

    # Read the file and print its contents
    text = file.read()
    print(text)

    # Close the file
    file.close()
```

Ln: 22 Col: 6

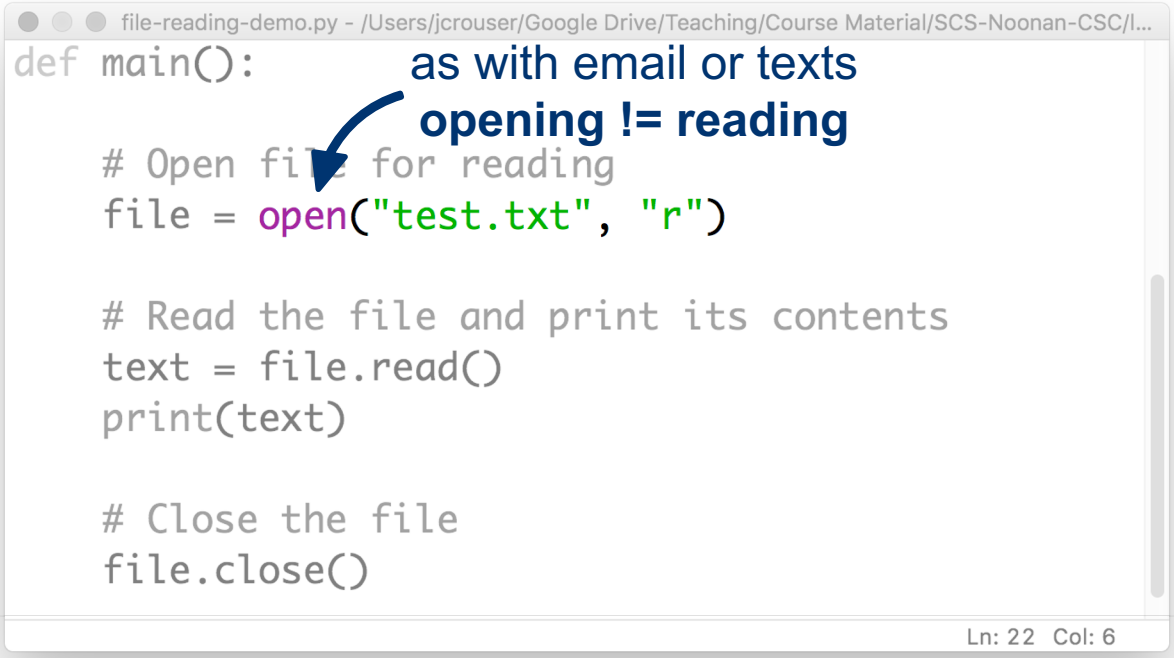
Reading a text file

- In order to bring data stored in a text file into a Python program, we need to **read** it:



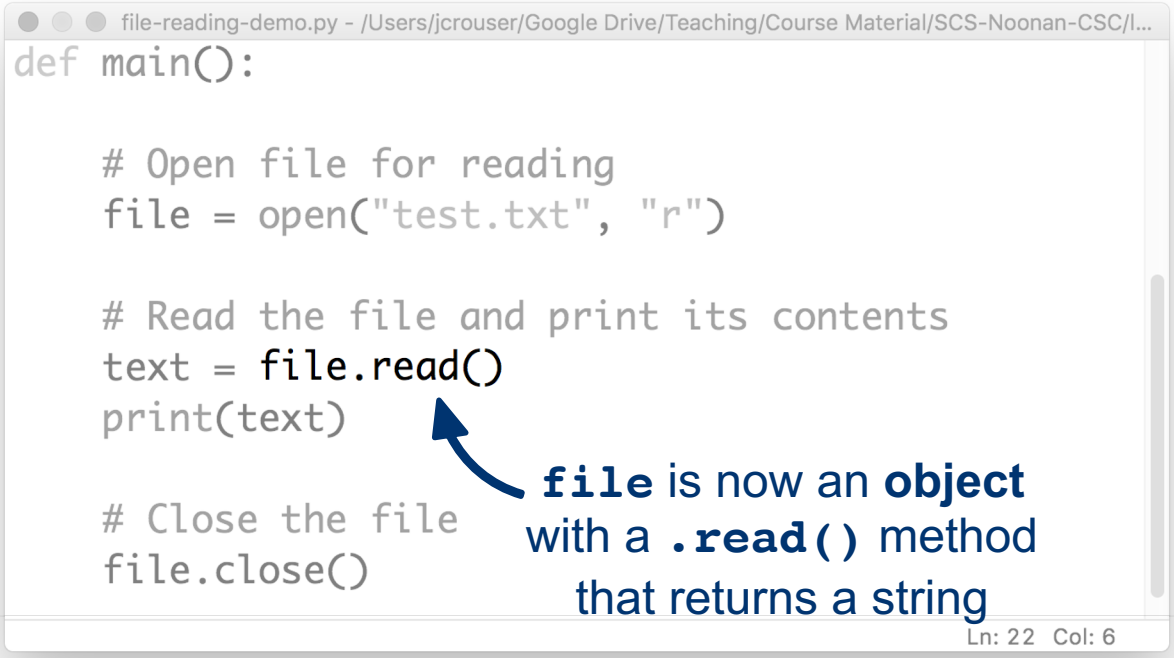
Reading a text file

- In order to bring data stored in a text file into a Python program, we need to **read** it:



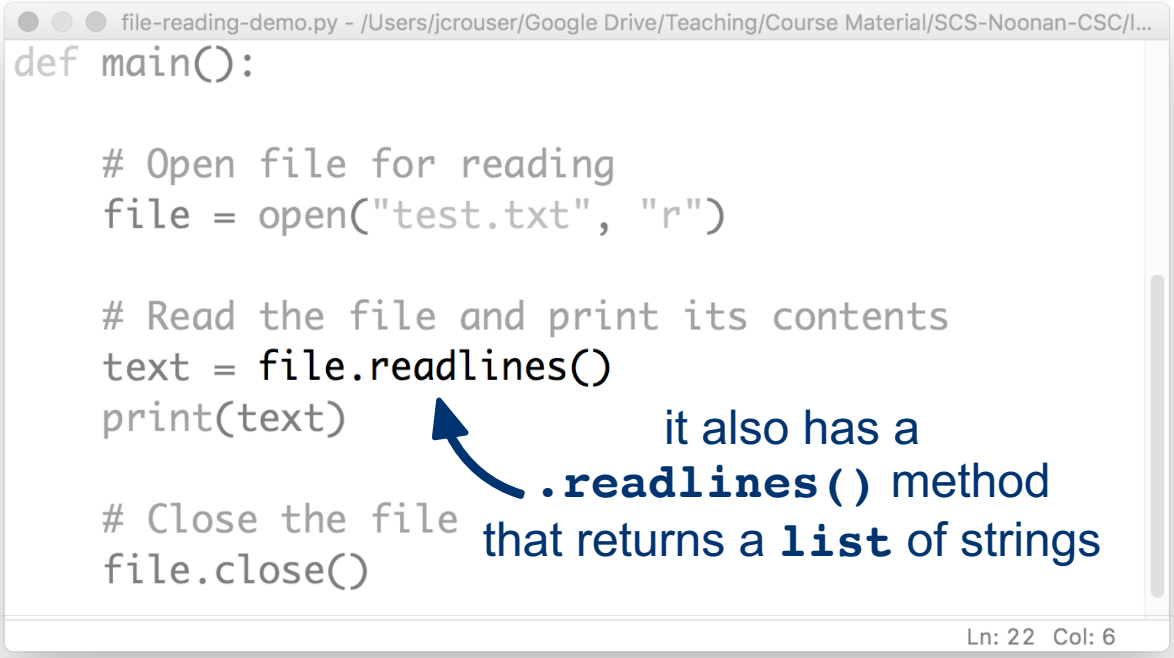
Reading a text file

- In order to bring data stored in a text file into a Python program, we need to **read** it:



Reading a text file

- In order to bring data stored in a text file into a Python program, we need to **read** it:



Reading a text file

- In order to bring data stored in a text file into a Python program, we need to **read** it:

file-reading-demo.py - /Users/jcrouser/Google Drive/Teaching/Course Material/SCS-Noonan-CSC/l...

```
def main():

    # Open file for reading
    file = open("test.txt", "r")

    # Read the file and print its contents
    text = file.read()
    print(text)

    # Close the file
    file.close()
```

Ln: 22 Col: 6

after you **`.read()`** or **`.readlines()`** a file, remember to **`.close()`** it

Key points for **reading** files

- Three-step process:
  1. **`.open()`**
  2. **`.read()`** or **`.readlines()`**
  3. **`.close()`**
- All three steps, always in that order
- If you want to **`.read()`** a file multiple times, you have to repeat the whole process

```
file = open("test.txt", "r")
text1 = file.read()
text2 = file.read() # Empty!
file.close()
```

Writing data to a text file

- The process looks very similar when we want to **write** data to a file:

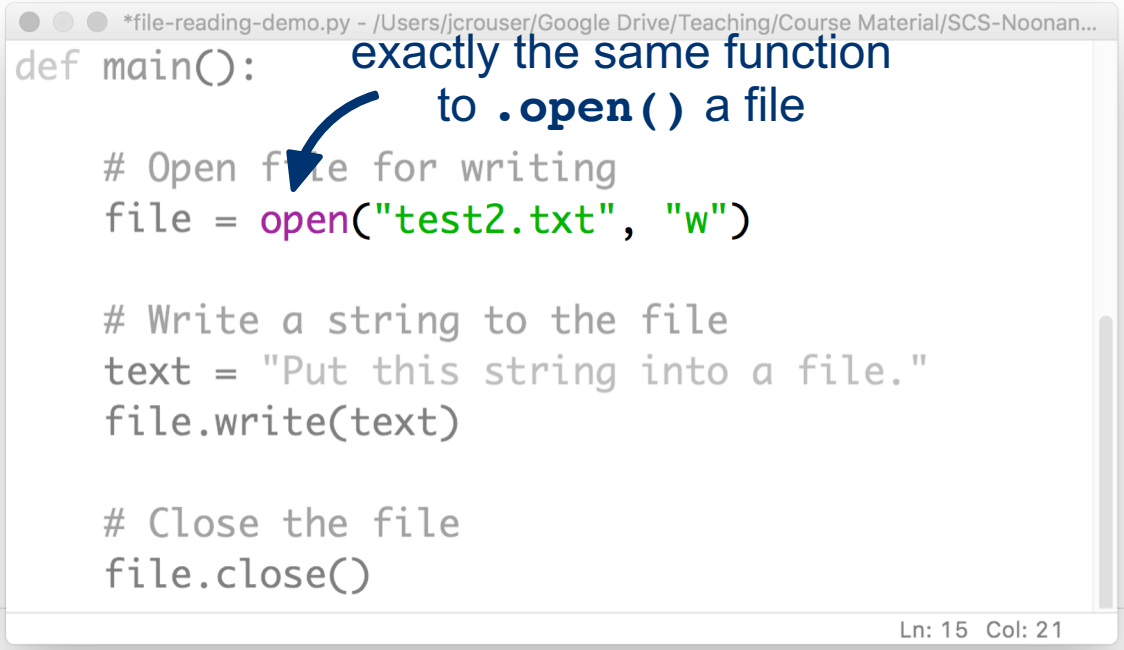
*file-reading-demo.py - /Users/jcrouser/Google Drive/Teaching/Course Material/SCS-Noonan...

```
def main():

    # Open file for writing
    file = open("test2.txt", "w")

    # Write a string to the file
    text = "Put this string into a file."
    file.write(text)

    # Close the file
    file.close()
```

Ln: 15 Col: 21

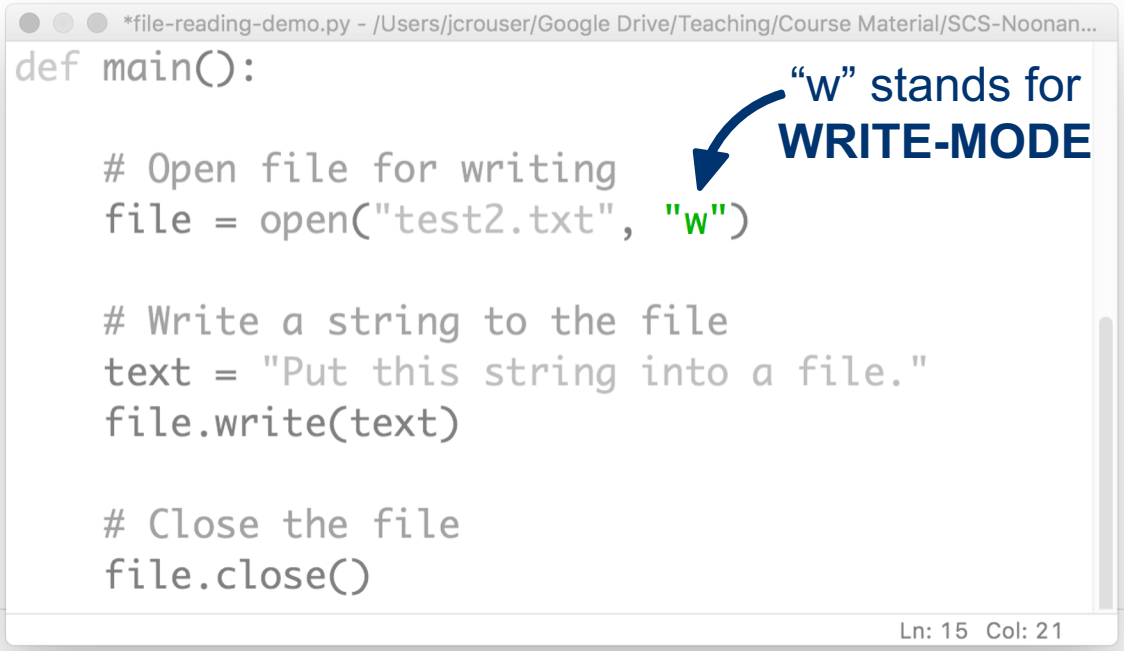
Writing data to a text file

- The process looks very similar when we want to **write** data to a file:



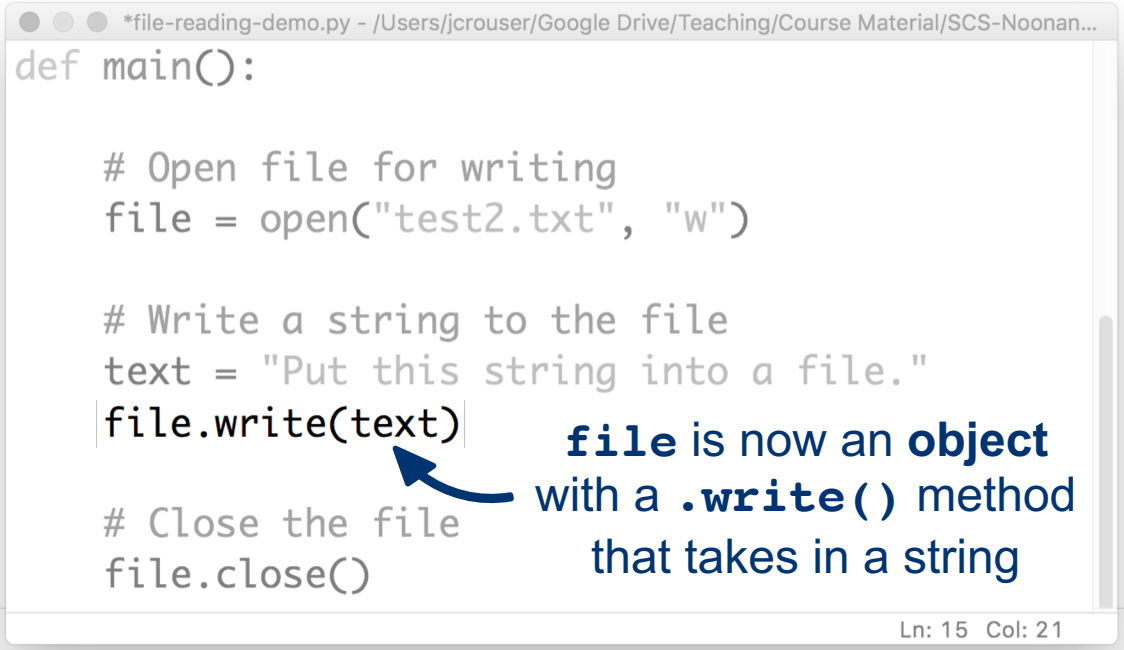
Writing data to a text file

- The process looks very similar when we want to **write** data to a file:



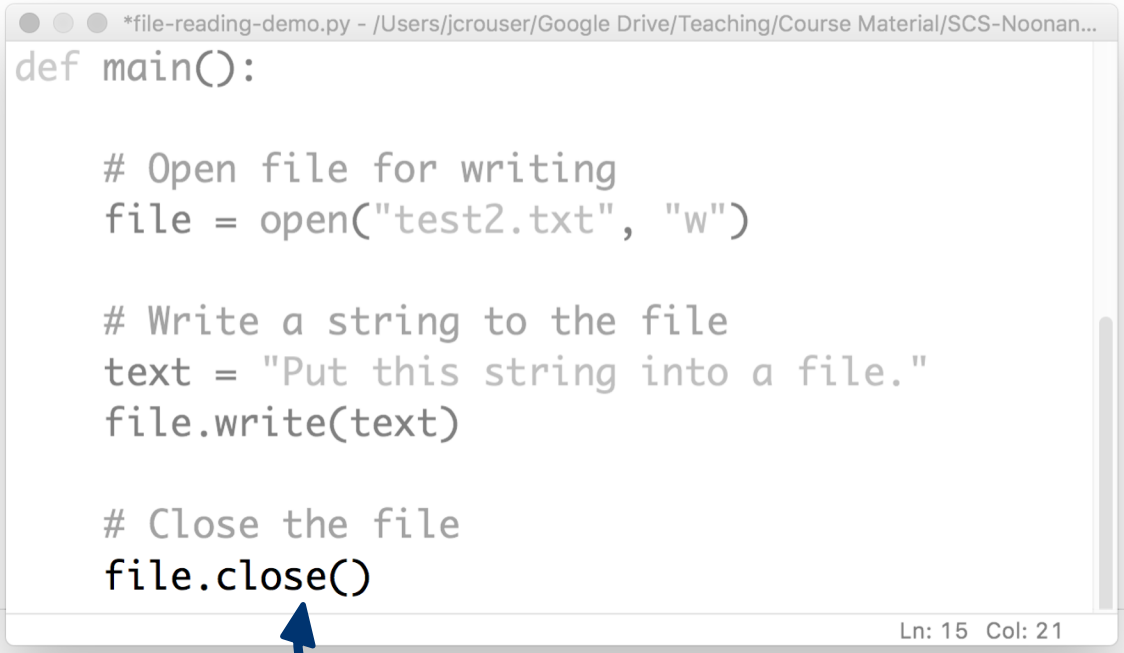
Writing data to a text file

- The process looks very similar when we want to **write** data to a file:



Writing data to a text file

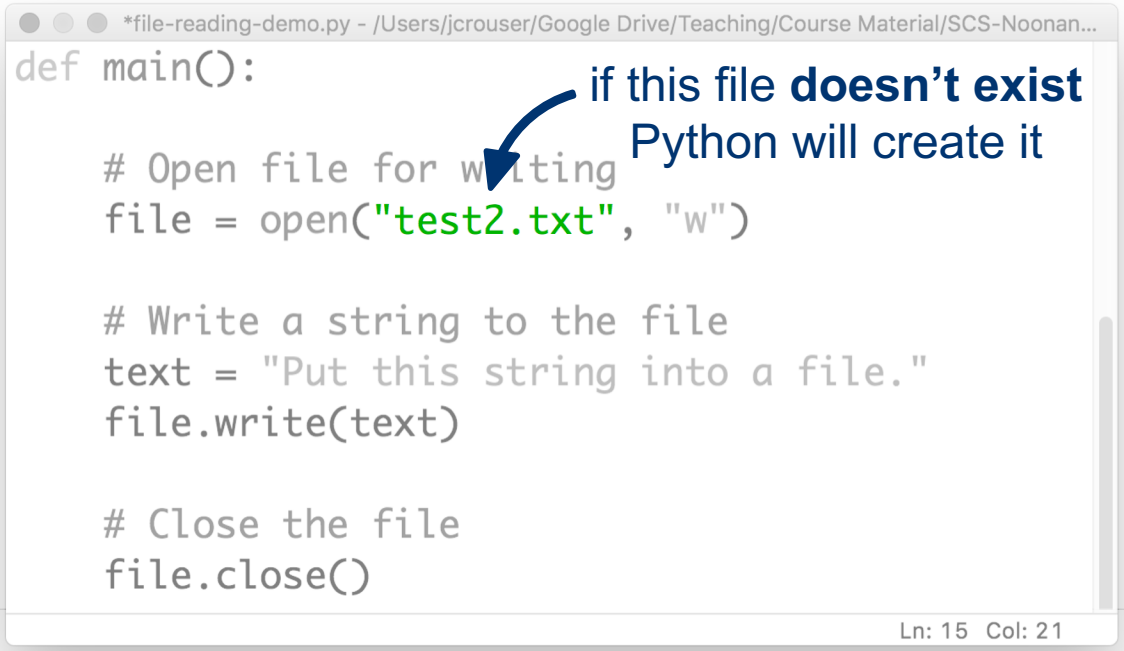
- The process looks very similar when we want to **write** data to a file:



as before, after you **`.write()`** to a file remember to **`.close()`** it

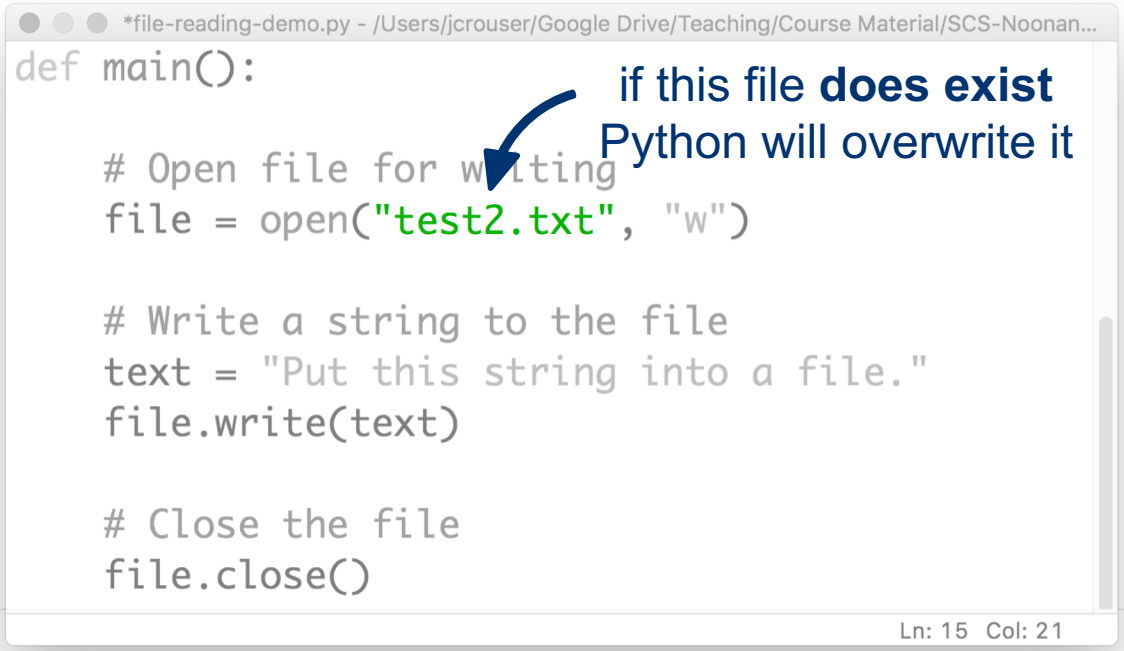
Writing data to a text file

- Some quirks to be aware of when **writing** to files:



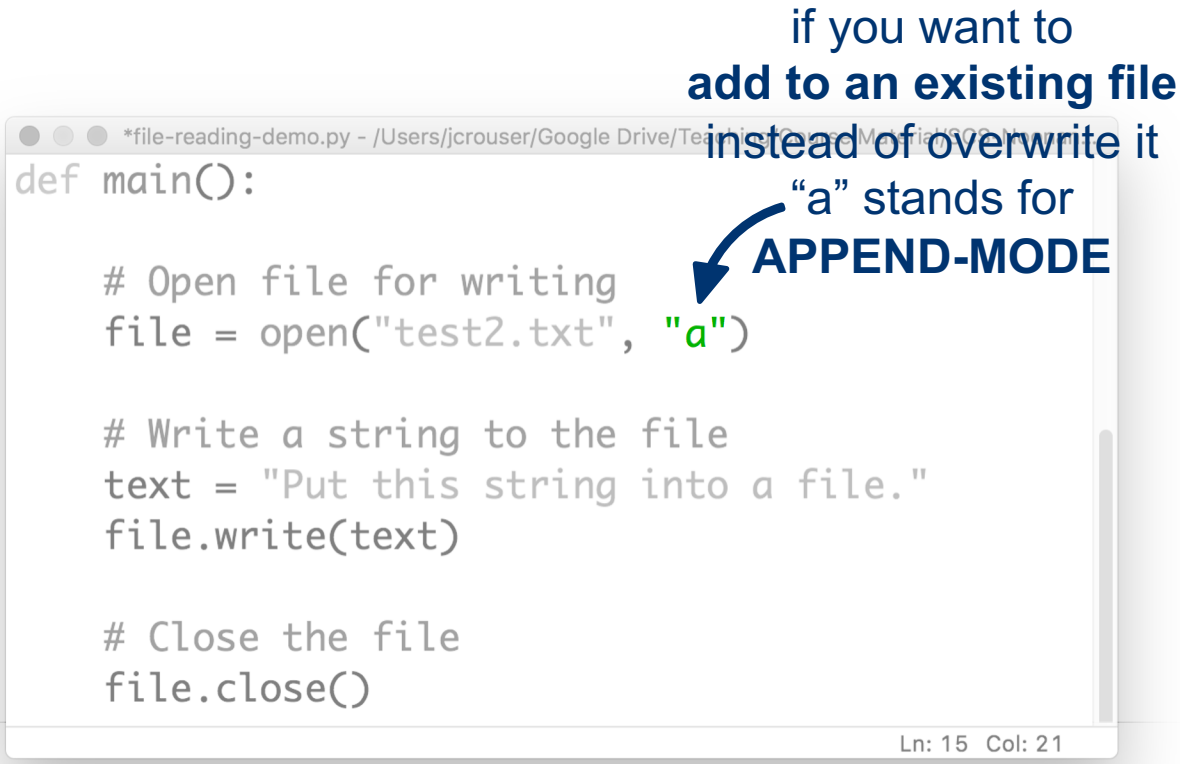
Writing data to a text file

- Some quirks to be aware of when **writing** to files:



Writing data to a text file

- Some quirks to be aware of when **writing** to files:



Key points for **writing** to files

- Three-step process:
  1. **`.open()`**
  2. **`.write()`**
  3. **`.close()`**
- Unlike **`.read()`**, you can **`.write()`** to an **`.open()`** file as many times as you want (appending each time)

```
file = open("test2.txt", "w")
file.write("Hello")
file.write("there!")
file.close()
```

- If you want a new line, you have to add it yourself! (**`\n`**)

15-minute exercise: read and write

Fork: https://repl.it/@JordanCrouser/horizontal-vertical

Write a program that:

1. reads the file **`horizontal.txt`**
2. breaks it into individual words
3. and writes the words to a new file **`vertical.txt`**, each one on its own line

Discussion

What did you come up with?

About the final project

- **Four weeks of class left!** (time to talk about the final)
- **Goal of the project**: apply the techniques we've learned in this class to something **you care about**
- **Ideas**:
  - an computer-generated animation
  - a custom game
  - a tool to help plan your path through the major
  - a automatic poem generator
  - a graphing calculator program
  - **anything else you can think of!**

Final project details and deliverables

- **Recommendation:** work in teams of 2-5 people
- **Deliverables:**
  - Friday Nov 13th: **Final Project Proposal**
  - Friday November 20th: **Prototype I**
  - Monday November 30th: **Prototype II**
  - Thursday December 17th: **Final Write-Up** (solo or group)

FP1: **final project proposal**

1. Names of **people** working on this project
2. What's the **big idea** behind this project?
3. What are the (major) **building blocks** the project will need to be successful?
4. Which do **you know how to build already**, and which ones **do you still need to figure out**?
5. Are there any **potential roadblocks**?

Coming up

- ✓Announcements
- ✓Working with files
  - ✓what is a file?
  - ✓reading data in
  - ✓writing data out
  - ✓*some details about the final project*
- Lab: Sorting
- Life Skill #5: Code Diagrams
- Useful python packages