

CogLab: Anatomy of an Experiment

WEEK 3

recap: Sep 7, 2023

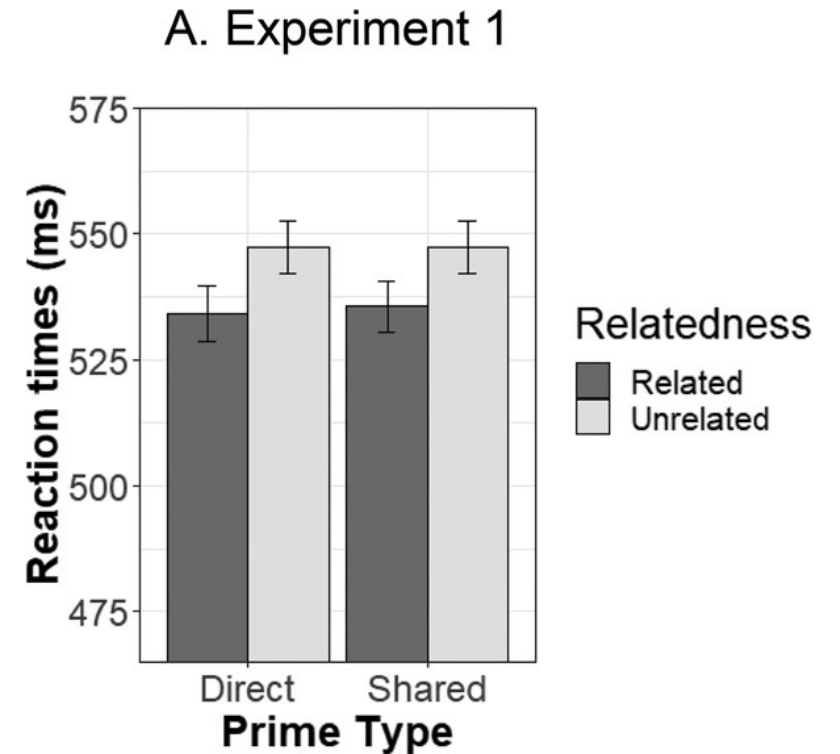
- what we covered:
 - reviewed research methods
 - discussed basic experiment design
- your to-dos were:
 - *prep*: Barnes, N. Publish your computer code: it is good enough. Nature 467, 753 (2010). <https://doi.org/10.1038/467753a>
 - *try*: week 2 quiz
 - *apply*: project milestone 1 (team plan + review article)
 - *apply*: optional Meme

today's agenda

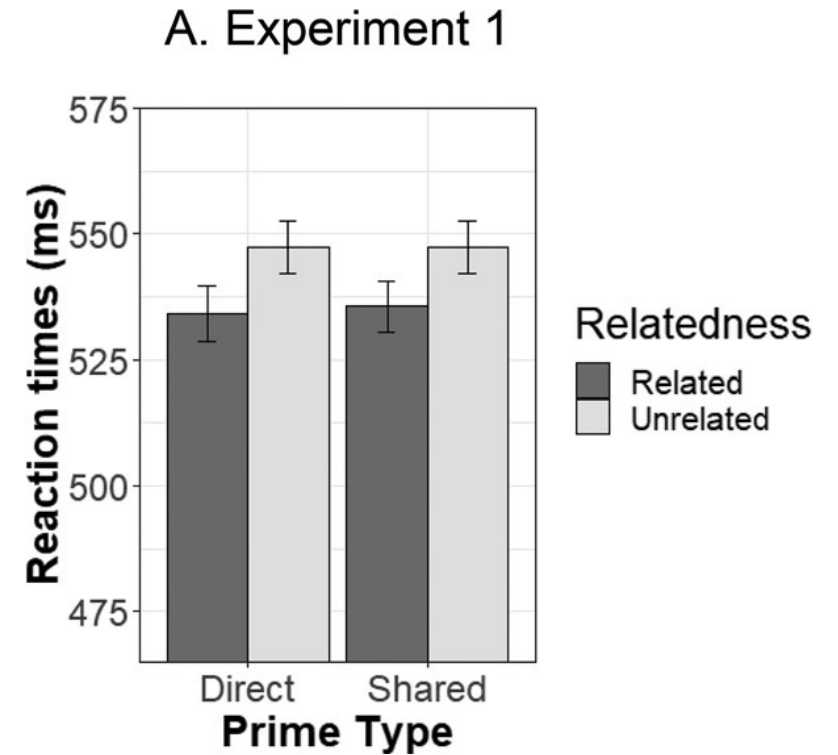
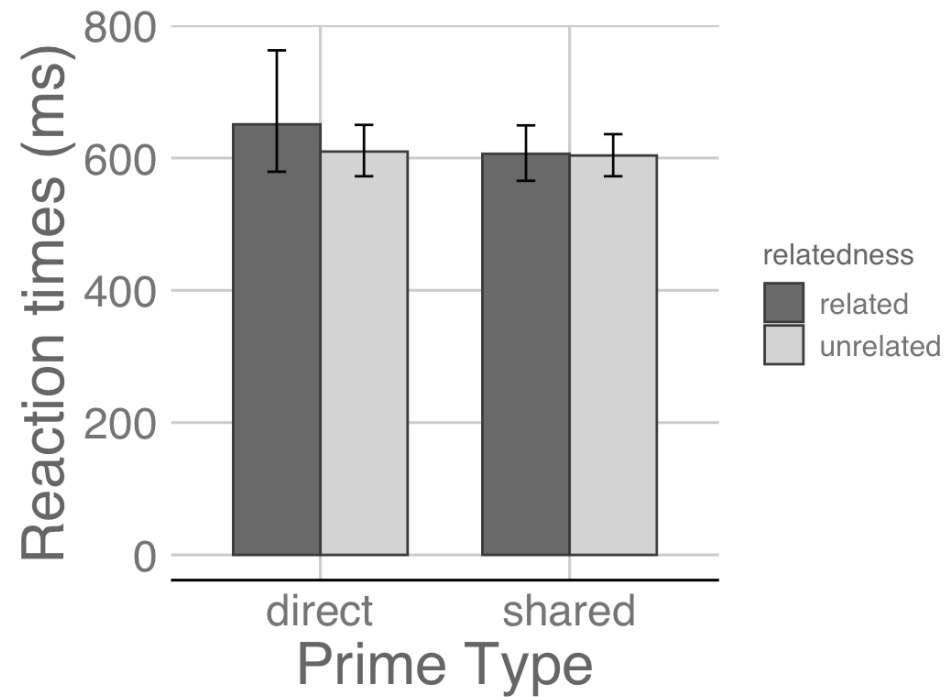
- review Savic et al. fundamentals
- anatomy of an experiment
- introduction to github and git

Savic et al. (2022): key findings

- Savic et al. (2022) had participants read sentences with novel and familiar words
 - novel words co-occurred with familiar words (directly or indirectly)
- **reaction time** to identify targets was faster when they were preceded by novel pseudowords/primes with which they directly co-occurred or shared co-occurrence in training
- pattern did not differ for direct and indirect co-occurrences
- **inference**: co-occurrences in natural language can drive semantic integration of new words



pilot data (N = 19) vs. Savic et al.'s data



possible questions to explore

- what is the **core idea** being tested?
- which **parts of the experiment** test this core idea?
- **how many repetitions** does it take for integration?
- is there something special about the **pairings**?
 - dodish-horse and foobly-apple
- is association the same as meaning?

the experiment

- put on your “researcher” hat
- what does it take to conduct this study?

anatomy of an experiment

the **front** end

- website URL
- links, images
- pages

the **back** end

- a “server”
- **content**: HTML
- **styling**: CSS
- **interactivity**:
Javascript

under the hood

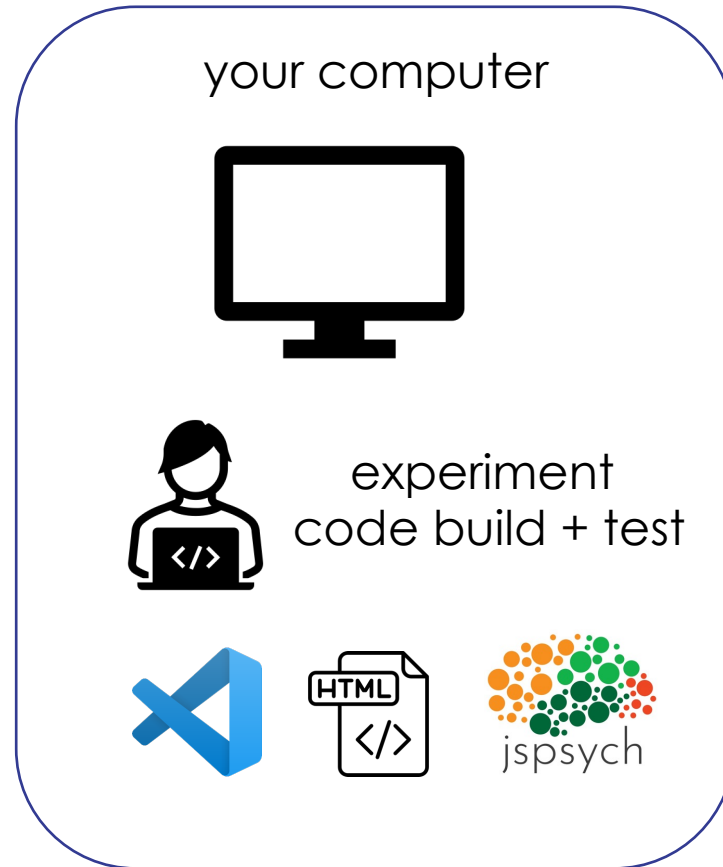
- any “page” you visit on the web has the following:
 - an IP address + URL (that you typically need to purchase)
 - a server hosting the content, styling, and interactivity
- for a typical web-based experiment:
 - we will use [Github](#) to manage our code
 - we will build our experiment in [HTML](#) (on your computer)
 - we will get help from [jsPsych](#) to make it easier to build
 - we will outsource the URL + server stuff ([cognition.run](#))

Github: what is it?

- Github is an **online** code-hosting service
 - uses **version control** to keep track of changes to files
 - Assignment_1_1 and Assignment_1_20 no more!
 - uses **git** (a specific type of version control)
- Git is a **local** version-control system
 - typically installed on your Mac automatically
 - we will connect Github to your local Git



github
keeping
track of
changes



Cognition.

cognition.run
going
online

Git

vs.

GitHub



Git is installed and maintained on your local system (rather than in the cloud)



First developed in 2005



One thing that really sets Git apart is its branching model

Git is a high quality version control system

GitHub is designed as a Git repository hosting service



You can share your code with others, giving them the power to make revisions or edits

GitHub is a cloud-based hosting service



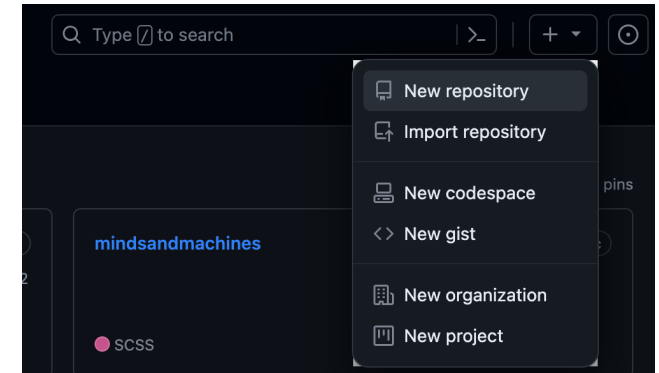
GitHub is exclusively cloud-based



Github: overview

- go to www.github.com and create an account
- you will see your “profile” on github
- repository
 - self-contained unit for a project or assignment
 - keeps track of all your changes over time
 - easy to share with collaborators or instructors
 - can be supplemented with “read-me” documentation

setting up a Github repository



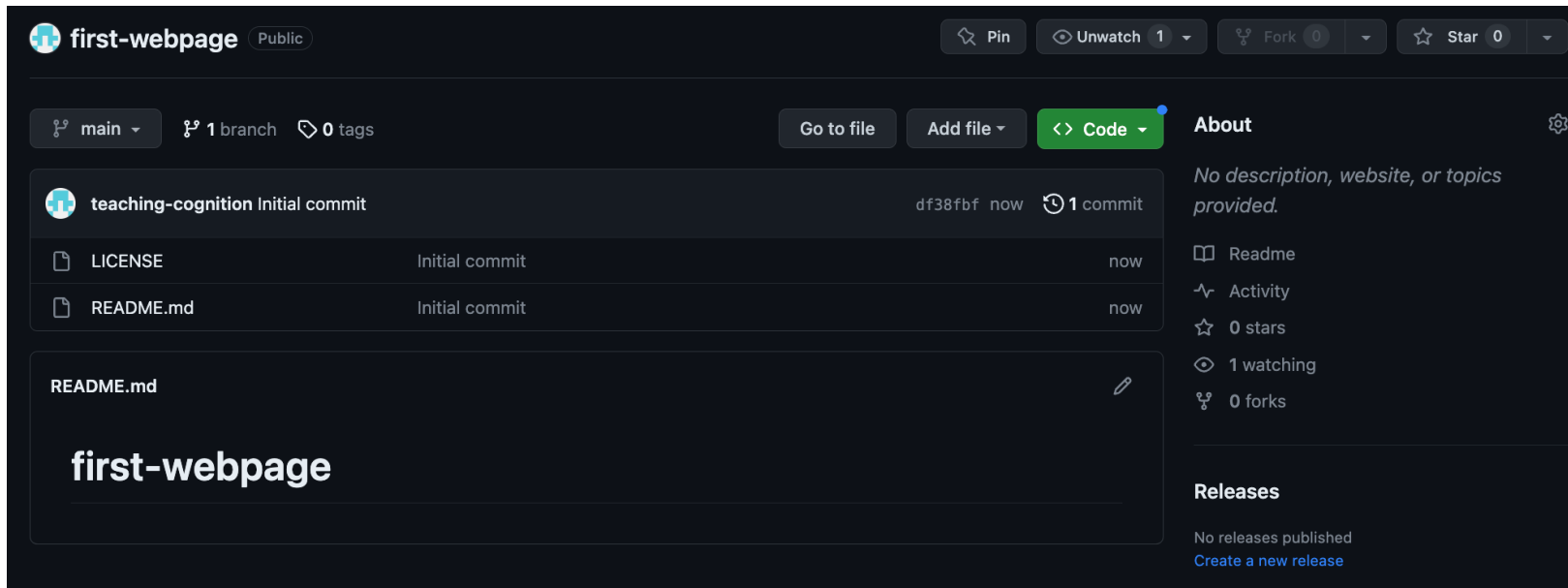
- create a new repository titled “first-webpage”
 - public
 - add a README file
 - MIT License
 - create!

A screenshot of the 'Create a new repository' form in GitHub. The form is titled 'Create a new repository' and includes the following sections:

- Repository template:** A dropdown menu set to 'No template'.
- Owner:** A dropdown menu set to 'teaching-cognition'.
- Repository name:** A text input field containing 'first-webpage', with a green checkmark and the text 'first-webpage is available.' below it.
- Description:** An optional text input field.
- Visibility:** Two radio buttons: 'Public' (selected) and 'Private'.
- Initialize this repository with:** A checked checkbox for 'Add a README file'.
- Add .gitignore:** A dropdown menu set to '.gitignore template: None'.
- Choose a license:** A dropdown menu set to 'License: MIT License'.

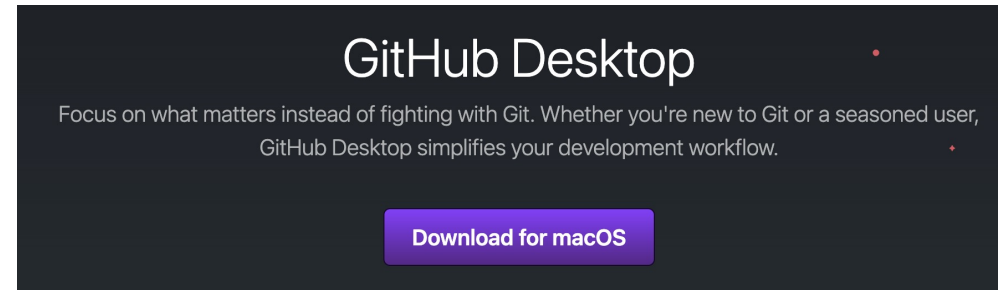
At the bottom right, there is a green 'Create repository' button. A note at the bottom left states: 'You are creating a public repository in your personal account.'

setting up a Github repository



- this “repository” is currently only online/remote
- for an efficient workflow, we want this on our **local** computer
- we will use the “Github Desktop” for this

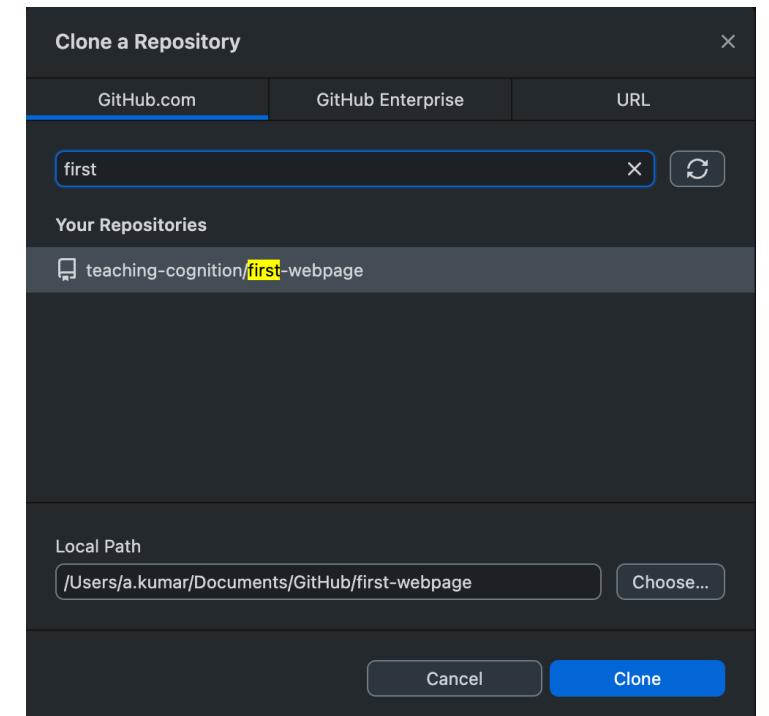
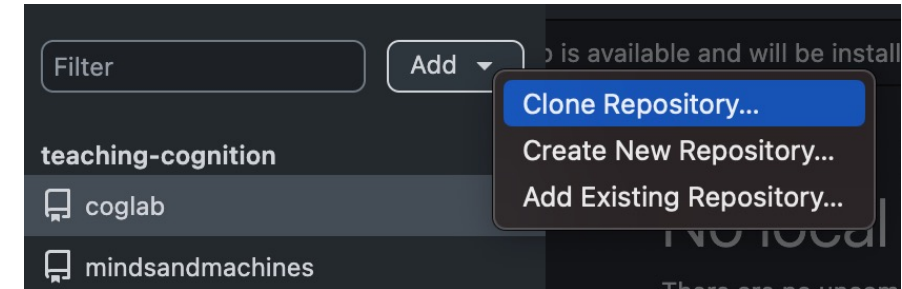
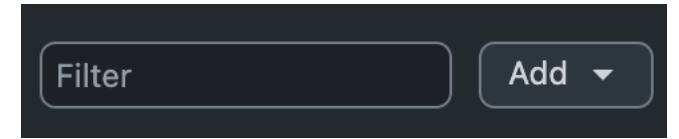
Github desktop



- go to <https://desktop.github.com/> and download
- move github desktop into your Applications folder
- open GitHub desktop and login (if asked to!)

cloning a repository

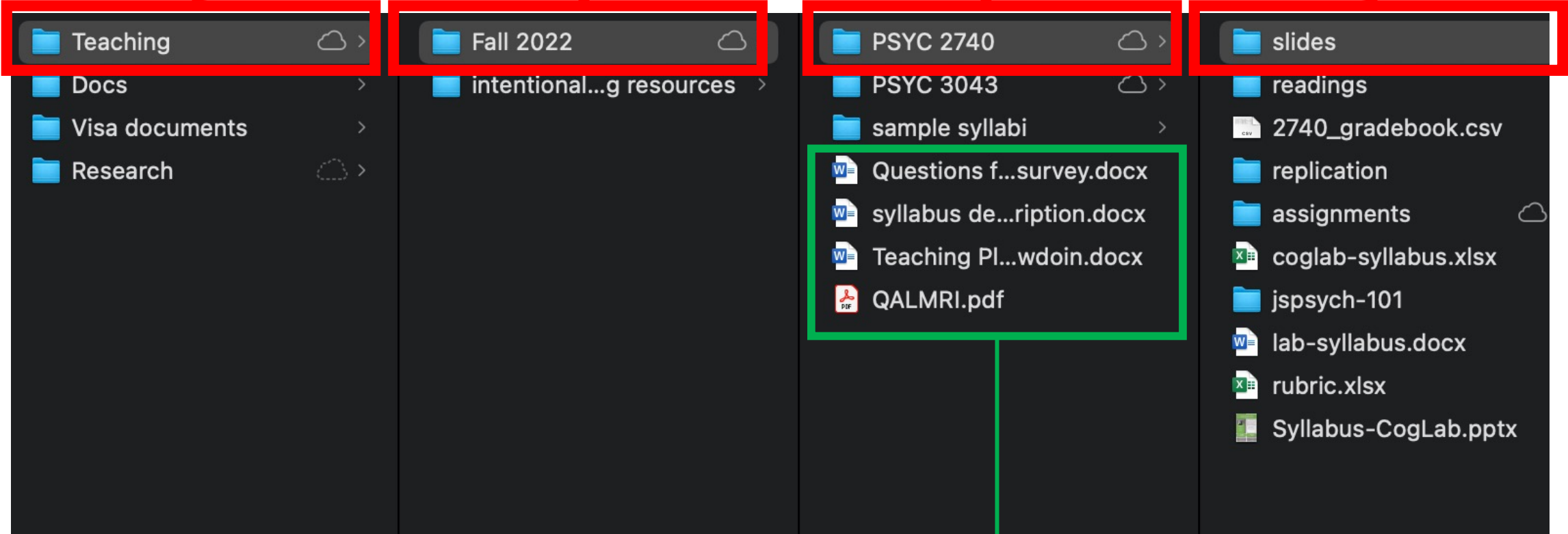
- create a new folder on your Mac titled **PSYC2740**
- clone your **first-webpage** repository
 - choose the location as **PSYC2740**
- open your folder through Finder
 - your Mac should have a first-webpage folder



some terminology: files and directories

- **local** vs. **remote**/online
 - local files are those that are on your Mac
 - nobody else has access to these files
 - remote files are *not* on your Mac: typically cloud-based (e.g., a Google doc)
- a **directory** refers to a folder on your Mac
 - can contain other folders + files
- a **file** is a computer document
 - Usually contained within a directory
 - Usually has an extension (.pdf, .docx, .xlsx)
- a **path** describes the route to access a file or folder
 - /Users/abhilashakumar/Documents: path to my Documents folder
 - /Users/abhilashakumar/Documents/xyz.docx: path to a specific Word document

directories

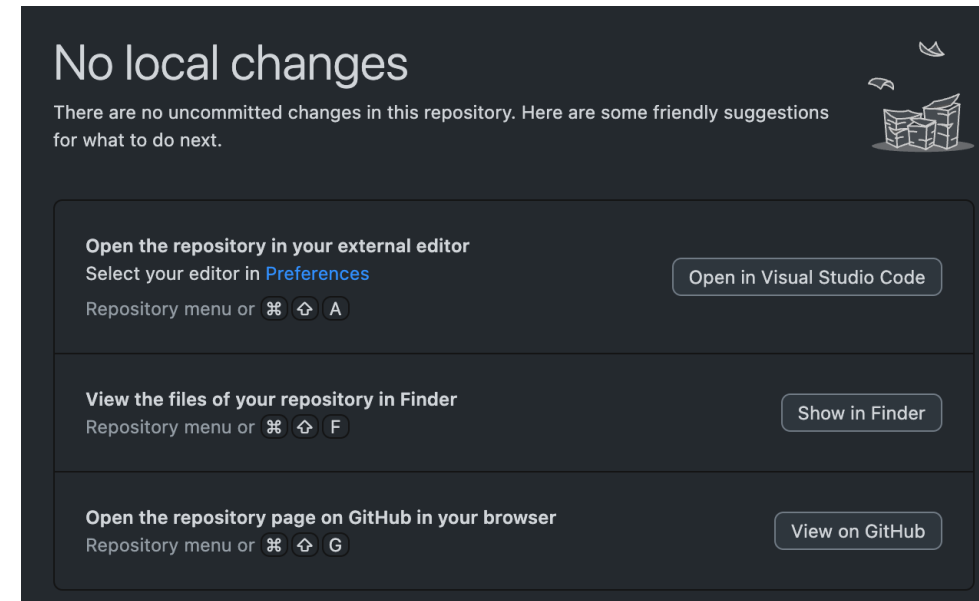


path → Teaching/Fall 2022/QALMRI.pdf

files

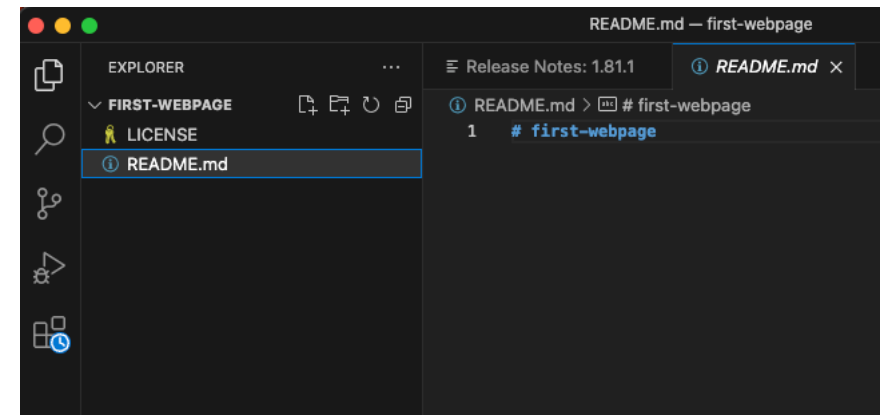
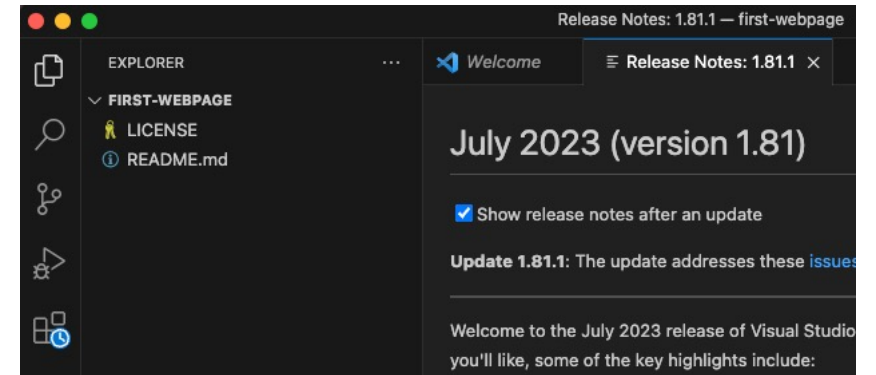
adding to your repository: part 1

- you can now “edit” the contents of your local repository
- but...you need a good code editor
- Visual Studio Code!
 - download for Mac
 - add to Applications
- now open GitHub Desktop
- open your repository in Visual Studio Code



adding to your repository: part 2

- once your repository is open in Visual Studio Code, you can make all kinds of changes!
- for example, click on [README.md](#)
 - this should open up the file in the editor
 - .md is a file extension that stands for Markdown
 - this is a markdown file that can store text and display it online
 - go to your github repo online and see what it looks like right now
- let's try to add some headings and paragraphs to this!

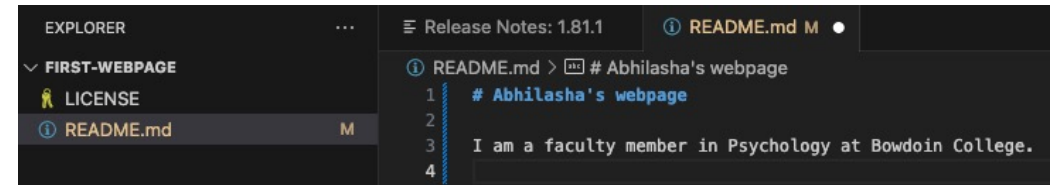


basic Markdown syntax

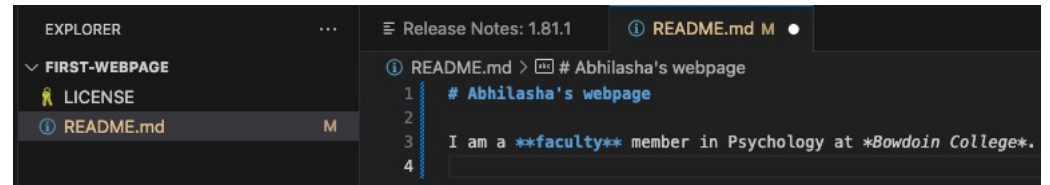
- cheat sheet
- headings are preceded by #, subheadings by ##
- you can **bold** and *italicize* things
- you can add hyperlinks: [coglab](<https://teaching-cognition.github.io/coglab/>)
- you can make lists using – or 1./2./3.

adding to your repository: part 2

- let's try a few things
- change the heading to # your-name's first webpage
- add a one-line description about yourself
- ****bold**** one word in that description and **italicize** another word
- save the file via [File > Save](#) or [Command + S](#)



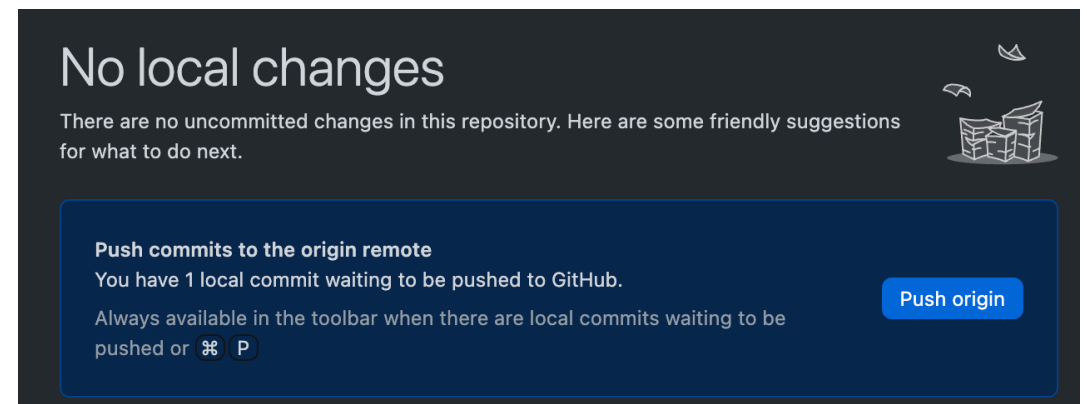
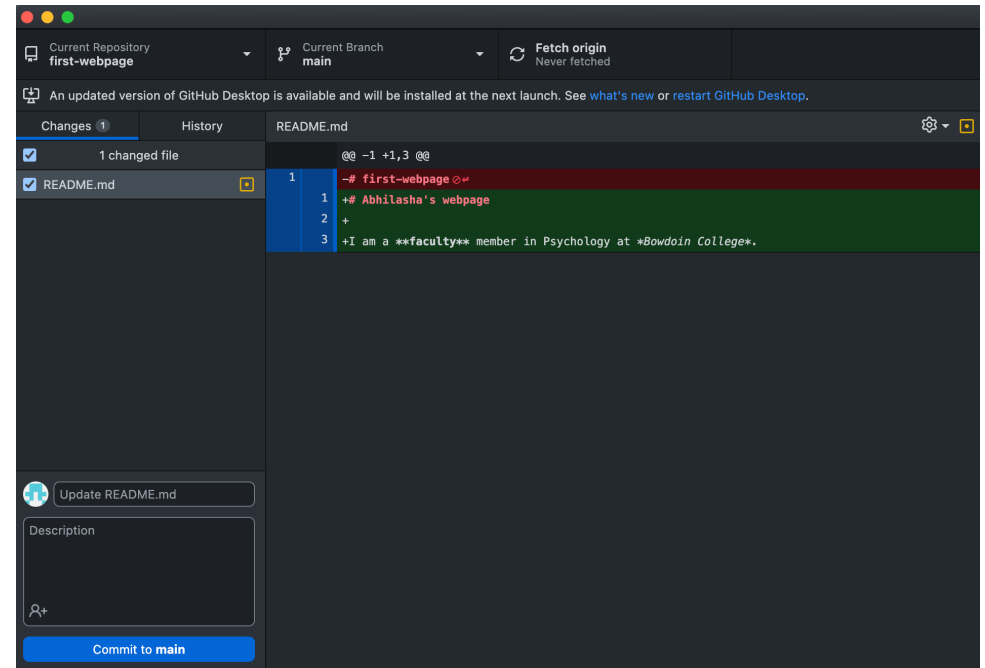
```
EXPLORER  ...  Release Notes: 1.81.1  README.md M ●
└─ FIRST-WEBPAGE
  └─ LICENSE
  └─ README.md M
      1 # Abhilasha's webpage
      2
      3 I am a faculty member in Psychology at Bowdoin College.
      4
```



```
EXPLORER  ...  Release Notes: 1.81.1  README.md M ●
└─ FIRST-WEBPAGE
  └─ LICENSE
  └─ README.md M
      1 # Abhilasha's webpage
      2
      3 I am a faculty member in Psychology at Bowdoin College.
      4
```

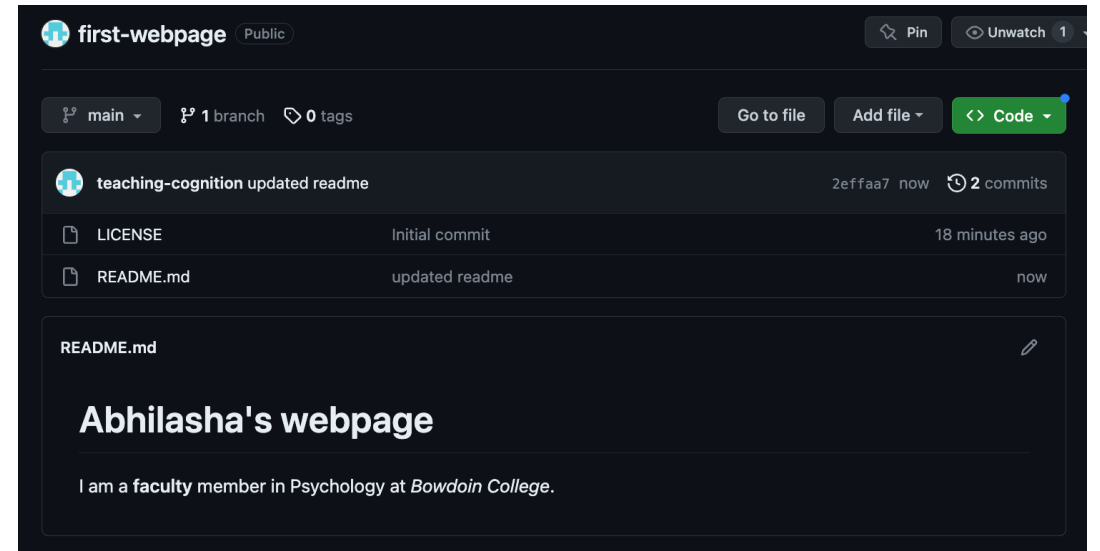
sending local changes online

- the changes you have made are currently local
- these changes need to be sent back to the cloud
- open GitHub Desktop
 - you should now see a record of all changes in your repo
 - add a description of your changes
 - click “commit to main”
 - click “push origin”



viewing updates online

- now go to your repository on github!
- your changes should be reflected online 😊 [might take 5 mins depending on how many changes]

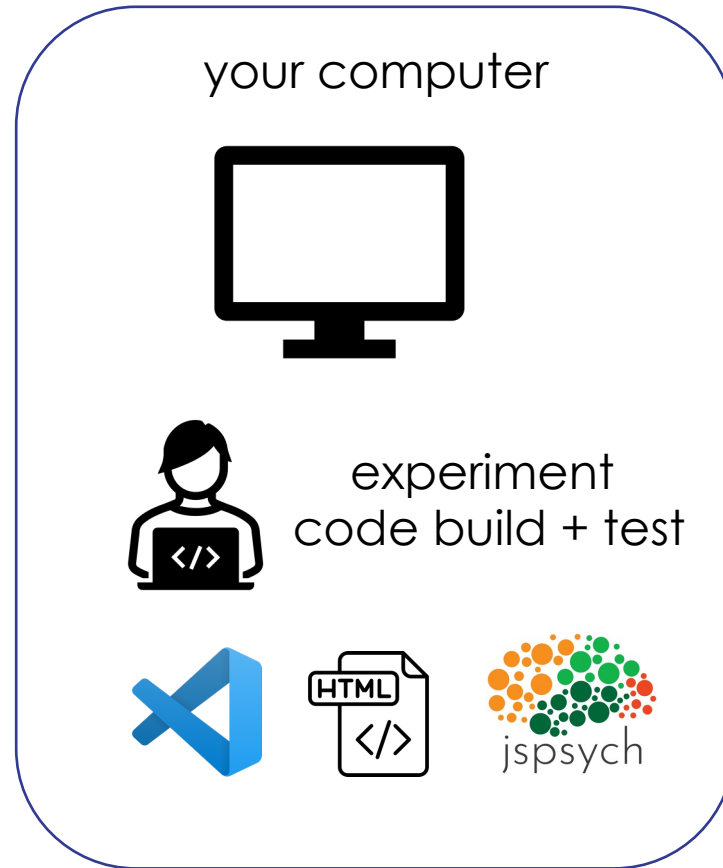


what did we learn today?

- general order of working with github:
 - make changes / editing
 - commit changes / “announce to git”
 - push changes / transfer online!



github
keeping
track of
changes



your computer



experiment
code build + test



jspsych

Cognition.

cognition.run
going
online

next class

- **before** class
 - *prep*: [How does the internet work](#)
 - *prep*: [A non-technical introduction to HTML/CSS/Javascript](#)
 - *download*: Google Chrome (default browser for class)
- **during** class
 - building your first webpage!