Lecture 6: Git/GitHub, Corpus Linguistics

LING 1340/2340: Data Science for Linguists

Na-Rae Han

Objectives

- Twitter mining! How did it go?
- Your term project
- Git/GitHub
  - Collaborating on the same repo – push access
- Data standards, sharing data
  - Review of standard data formats
  - Your own data plans for your project
- Corpus linguistics, linguistic annotation
  - Types of linguistic annotation
  - Annotation formats

Your term project

- Your project is now on GitHub
  - https://github.com/Data-Science-for-Linguists-2021
- First progress report is due next week
  - Focus on data: sourcing, curation and cleaning
- Managing your data
  - You will be manipulating and processing your data.
  - Should you include your data set in your GitHub repo?
    - Depends! For now, include an appropriate amount of samples.

Licensing, public vs. private

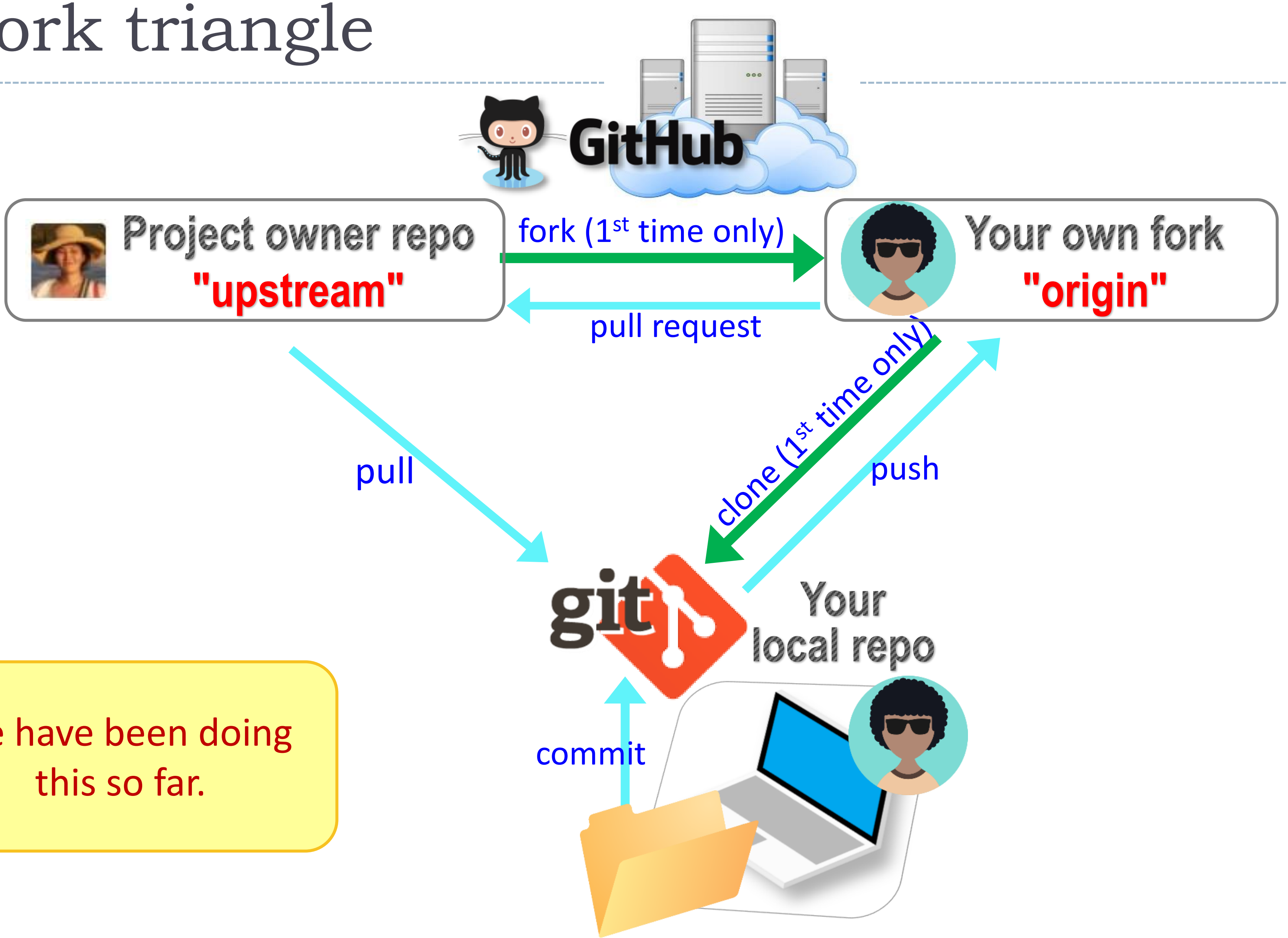
- Your data:
  - Your original data source: what kind of license does it come with?
  - Can you re-distribute the data?
  - How about samples? "Derivative" data?
  - Your own "value-add" (annotation, etc.). What license will you attach?
  - How to best *present* the outcome and ensure *reproducibility* if you cannot share your data in full?
- Your code:
  - Will you allow other people to use your code? Re-distribute?
  - Will you allow other people to turn your code into a commercial product? Patent it?
- `LICENSE.md`
  - This is about YOUR OWN LICENSE that you choose, FOR YOUR ENTIRE PROJECT
  - This is separate from the license that came with your "source data"!!!

Licensing, public vs. private

- As a principle, your term project -- including code and data -- should be **as public and open as possible**.
  - Your repo should be **public**.
  - For now, store your data files in a directory that's ignored through `.gitignore` . Suggestion: private/ or data/.
- Do your research on copyright and licensing.
  - Dr. Lauren Collister's guest lecture
- Document, document, document!
  - You should **document and justify** your sharing and licensing decisions. It is an important part of your project.

The fork triangle

*Avatar icons by FLATICON

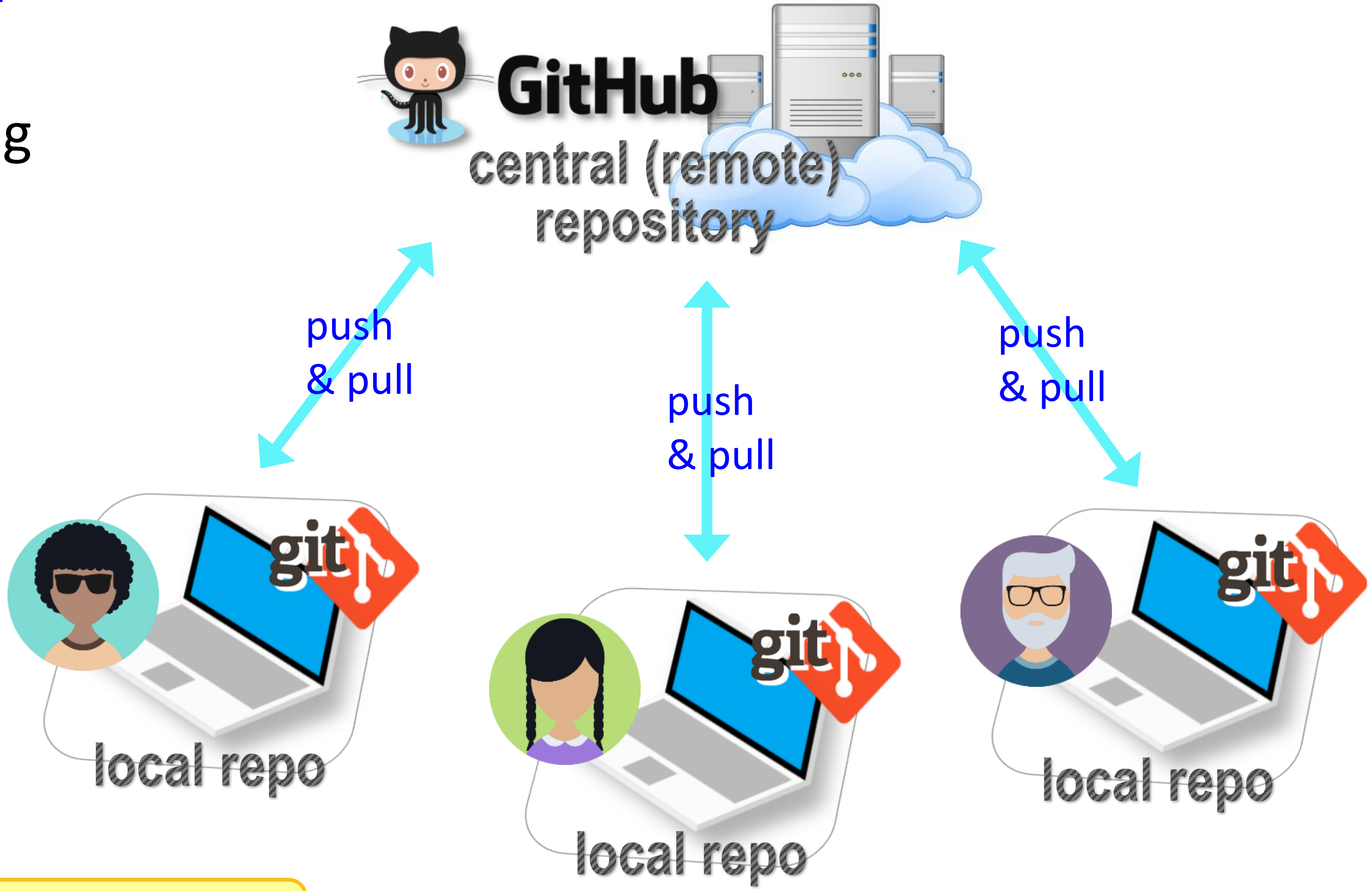


We have been doing this so far.

GitHub: a *social*, remote repository

*Avatar icons by FLATICON

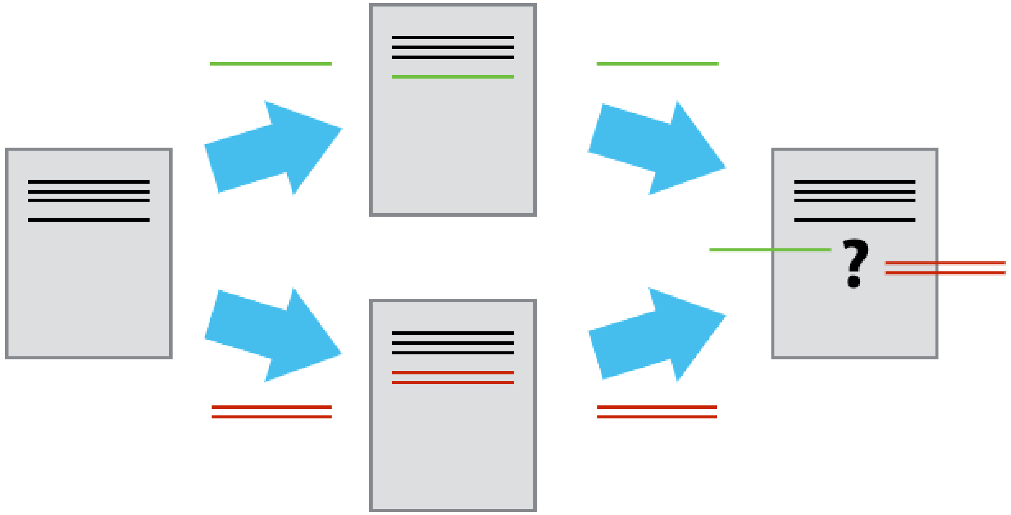
- GitHub also works as a central remote repository among a group of **collaborators** working on a shared project.
  - Everyone works on their own *local* copy of the repository, making changes.
  - Git is able to keep track and merge changes submitted by everyone.
  - Everyone is an **equal collaborator** with push (=write) access.



We are now ready!

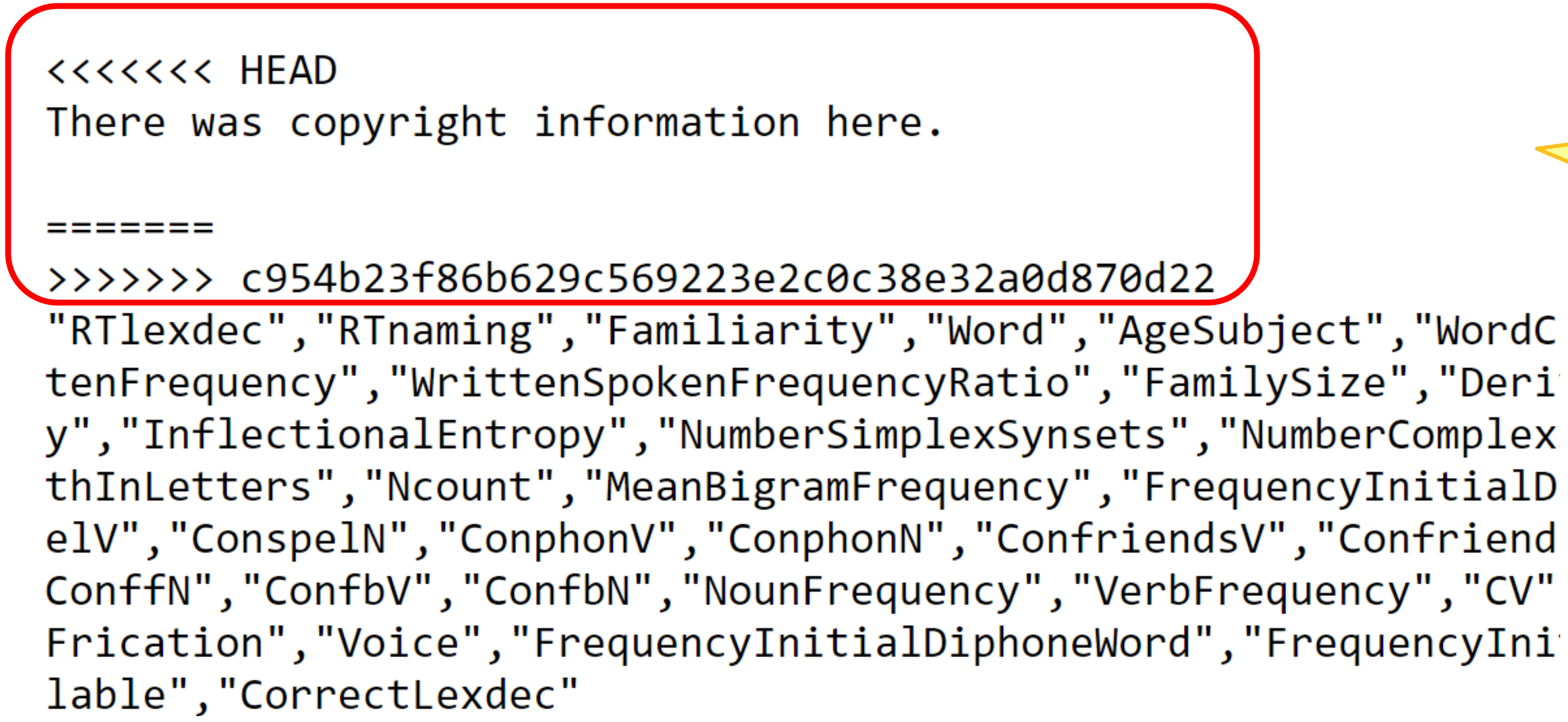
Introducing... "Class-Lounge"

- Public.
- Everyone is listed as a "collaborator".
  - Meaning, everyone has push access.
  - No need to fork: pull and push directly.
  - We will also truly collaborate: **edit shared files**.
- This means: **CONFLICTS**
  - Na-Rae's tutorial on Git conflicts:
    - https://github.com/mcdonn/LSA2019-Reproducible-Research/blob/master/linking_git_and_github.md#conflicts



When there is a conflict

- After you pull, Git changes your file, which then looks like:



```
<<<<<<< HEAD
There was copyright information here.

=======
>>>>>>> c954b23f86b629c569223e2c0c38e32a0d870d22
"RTlexdec","RTnaming","Familiarity","Word","AgeSubject","WordC
tenFrequency","WrittenSpokenFrequencyRatio","FamilySize","Deri
y","InflectionalEntropy","NumberSimplexSynsets","NumberComplex
thInLetters","Ncount","MeanBigramFrequency","FrequencyInitialD
elV","ConspelN","ConphonV","ConphonN","ConfriendsV","Confriend
ConffN","ConfbV","ConfbN","NounFrequency","VerbFrequency","CV"
Frication","Voice","FrequencyInitialDiphoneWord","FrequencyIni
lable","CorrectLexdec"
```

You must **manually edit this file** and tidy it up. (== resolve conflict)

A GitHub race: our favorite animals

Activity
10 minutes

1. Everyone was already added to the repo as a collaborator.
2. Clone the repo to your laptop.
3. Edit "animals.md", add your line.
4. Do your usual local git routine: adding, committing.
5. Try pushing. It is likely you have a conflict (someone else pushed in the meantime) and Git tells you to pull first.
6. Pull to receive the new updates.
7. Open "animals.md". Resolve conflict.
8. Go back to step 4. Hope you were quick enough this time!

Data standards & exchange formats

	What	Notes, reference
CSV	Comma-separated values	Compatible with Excel
TSV	Tab-separated values	
HTML	Web pages	Not meant as data format
XML	For markup and text encoding	A Gentle Introduction to XML by TEI
JSON	JavaScript Object Notation (Twitter, Jupyter Notebook)	Introducing JSON JSON example (vs. XML)

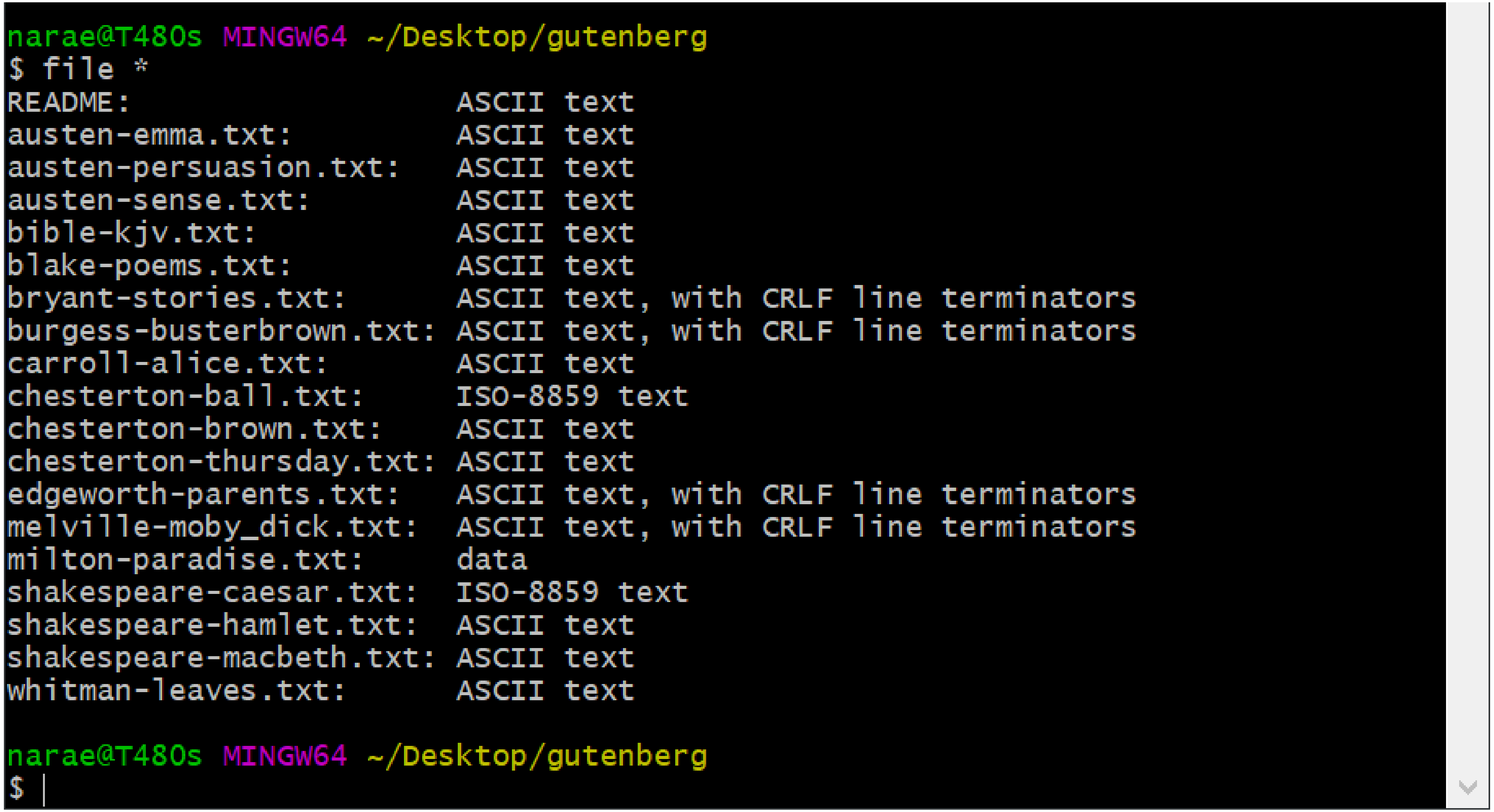
These are all TEXT files!

They are all TEXT files.

- Encoding: Latin-1, ASCII, UTF-8, UTF-16, CP1252, ...
- Line endings:
  - LF (`'\n'`: OS X & Linux) , CRLF (`'\r\n'`: Windows)
- But underneath it all, these files are all TEXT files with **special formatting syntax** and **special characters** designated for formatting purposes.
  - In command line, you can `cat` and `less` through the files.
  - You can open them up in a **text editor** (Atom, Notepad++) and edit.
  - Some editors/applications are aware of the format-specific syntax and will highlight/render accordingly.
    - Unlike, say, PDF files, style attributes are NOT part of the files themselves. (e.g., markdown file)

File formats and conversion

- Download "Project Gutenberg Selections" from the NLTK Corpora page (http://www.nltk.org/nltk_data/).
  - Unzip and examine the included text files ('austen-emma.txt', 'shakespeare-caesar.txt', ...).
  - What **encoding scheme** do the files have? Is every file UTF-8?
  - What about **line ending**? Do you see Windows style "CRLF" line ending?
  - The file command reports 'milton-paradise.txt' as a **'data' file**, not a plain text file. Is this correct?
  - Let's bring some **consistency** to this corpus: every file should have UTF-8 encoding with the Unix-style LF line ending. Apply conversion to appropriate files using either: (1) command-line tools, (2) text editor programs such as Notepad++ and Atom.



```
narae@T480s MINGW64 ~/Desktop/gutenberg
$ file *
README:                     ASCII text
austen-emma.txt:            ASCII text
austen-persuasion.txt:      ASCII text
austen-sense.txt:           ASCII text
bible-kjv.txt:              ASCII text
blake-poems.txt:            ASCII text
bryant-stories.txt:         ASCII text, with CRLF line terminators
burgess-busterbrown.txt:    ASCII text, with CRLF line terminators
carroll-alice.txt:          ASCII text
chesterton-ball.txt:        ISO-8859 text
chesterton-brown.txt:       ASCII text
chesterton-thursday.txt:    ASCII text
edgeworth-parents.txt:      ASCII text, with CRLF line terminators
melville-moby_dick.txt:     ASCII text, with CRLF line terminators
milton-paradise.txt:        data
shakespeare-caesar.txt:     ISO-8859 text
shakespeare-hamlet.txt:     ASCII text
shakespeare-macbeth.txt:    ASCII text
whitman-leaves.txt:         ASCII text

narae@T480s MINGW64 ~/Desktop/gutenberg
$ 
```

Format conversion

- When dealing with corpora, you may need to convert 100+ files at once.
  - On-line services are too cumbersome.
  - Try batch-processing through command line.
- Automatic tools available on command line.
  - Finding out file text file encoding, line ending: `file` command
  - Encoding conversion: `iconv` (Linux, OS X, on Git Bash)
  - Line ending conversion: `unix2dos`, `dos2unix`
  - Pandoc http://www.pandoc.org/
    - Universal document coverter
    - HTML, XML, PDF, LaTeX, Markdown, Epub, MS Doc, ...
    - After installation, you can use it via command line

Resource-specific (ad-hoc) formats

- Brown corpus

```
The/at Fulton/np-tl County/nn-tl Grand/jj-tl Jury/nn-tl said/vbd
Friday/nr an/at investigation/nn of/in Atlanta's/np$ recent/jj
primary/nn election/nn produced/vbd ``/`` no/at evidence/nn ''/''
that/cs any/dti irregularities/nns took/vbd place/nn ./.
```

- Korean Treebank corpus:

```
;;05:127: 저는 그 일을 할 수 있는 한 빨리 하겠습니다 .

(S (NP-SBJ 저/NPN+는/PAU)
   (VP (NP-OBJ-LV 그/DAN
                  일/NNC+을/PCA)
       (VP (NP-ADV (S (NP-SBJ (S (NP-SBJ *pro*)
                                 (VP 하/VV+ㄹ/EAN))
                              (NP 수/NNX))
                      (ADJP 있/VJ+는/EAN))
                   (NP 한/NNX))
           (ADVP 빨리/ADV)
           (VP (LV 하/VV+겠/EPF+습니다/EFN))))
   ./SFN)
```

NOT standard
(cf. XML, JSON).
Project-dependent.

It is up to end users to understand the data format, then write code to parse data files.

Refer to documentation!

Do not re-invent the wheel.

- Don't try and parse them manually.
- There are Python libraries. Import and use them.
  - CSV & TSV: `pandas`
  - HTML & XML: Beautiful Soup (`bs4`)
  - JSON:
    - `json` library
    - `pandas.read_json`
- NLP-specific formats (Treebank, Universal Dependency, CoNLL):
  - Look at NLTK, see if it has reader
  - If not, chances are there is parser library written by someone somewhere (likely on GitHub)

Wrapping up

- To-do #7 out: corpus resources
  - Make sure to properly handle conflicts!

- Your project
  - Feedback will be forthcoming.
  - 1st progress report due next week! Go work on your data.
  - Copyright & licensing issues – you should have a good plan.