

# Welcome to Software Carpentry Week 2!

If you missed Week 1, chat a helper to download the workshop data and check your software installs.

Otherwise, locate your workshop data from last week.

Open the Etherpad shared document at  
<https://pad.carpentries.org/2020-07-24-ku-online>

Sign in starting ~ line 455 and open the cheatsheet links if you want them

Say hi, ask questions, and get help by unmuting yourself or using the chat.

We'll get started at 9:00am.

# What we will do today

- Develop a Bash script
- Using Git, apply version control to our script
- Save a copy of our script on GitHub

Breaks: 5 minutes at approx. the start of every hour

# Why Version Control?

## **Without Version Control:**

- Multiple copies get confusing
- Undo limited by software
- Must keep records separately

`thesis-ch1-v1.3.docx`

`thesis-ch1-v1.3-bak.docx`

`thesis-ch1-v1.8.docx`

`thesis-ch2-v3-tables.docx`

## **With Version Control:**

- One copy of a tracked file
- Unlimited undo for tracked files
- Automatic project records

# Why Git?

- Free
- Widely used
- Lots of online documentation and help
- De facto version control for software development
- No server or internet connection required

# Essential Vocabulary

- **Commit:**

The set of changes that make up a version of an item

- **Repository (Repo):**

All the commits (changes) and metadata (descriptions, contributors, timestamps, etc.) for a project

# Git First Time Setup

## Identity:

```
$ git config --global user.name "My Name"
```

```
$ git config --global user.email "my@email.address"
```

(use the email address that matches your GitHub account)

## Line Endings:

MacOS or Linux: 

```
$ git config --global core.autocrlf input
```

Windows: 

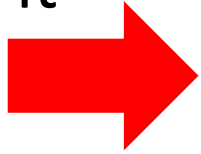
```
$ git config --global core.autocrlf true
```

## Text Editor:

```
$ git config --global core.editor "nano -w"
```

## Exercise: Improve Script Documentation

Edit your copy of  
middle.sh so it  
looks like this:



When you are  
finished, add and  
commit the  
changes to your  
Git repository.

```
# Comments on script development:
# Add ability to run on any file
# Add ability to set line numbers

# Instructions for use:
# bash middle.sh $1 $2 $3 where:
# $1 is the filename
# $2 is the number of lines to send to head
# $3 is the number of lines to send to tail

head -n "$2" "$1" | tail -n "$3"
```

# Exercise: File management and `.gitignore`

1. Make a new directory inside `molecules/` for temporary files
2. Move the existing `*.txt` files from `molecules/` into this new directory
3. Check `git status` - why might it be the same?
4. Update `.gitignore` to:
  1. Not ignore all `.txt` files
  2. Ignore all temporary files
5. Add and commit the changes to `.gitignore`