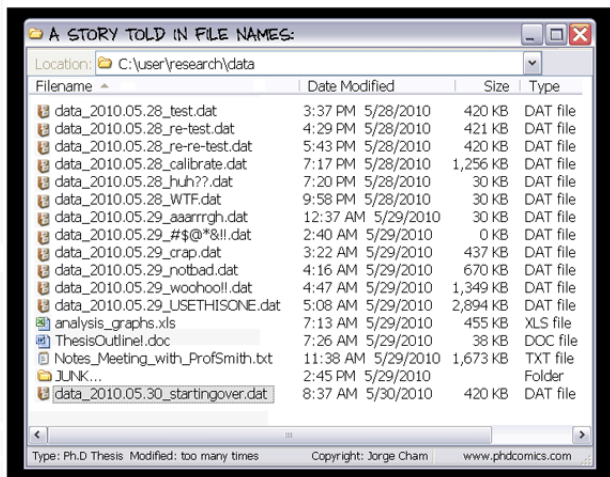


# Introduction to Version Control with Git

Molly Gibson  
@gibsmk

# Why do I need version control?

# Why do I need version control?



# Why do I need version control?

- ▶ Make changes to code with confidence - *can always be reverted if necessary*

# Why do I need version control?

- ▶ Make changes to code with confidence - *can always be reverted if necessary*
- ▶ Reproducibility - *version control can complement your lab notebook*

# Why do I need version control?

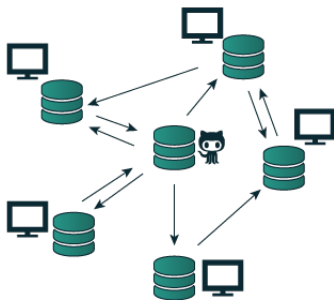
- ▶ Make changes to code with confidence - *can always be reverted if necessary*
- ▶ Reproducibility - *version control can complement your lab notebook*
- ▶ Work as a team - *file names and directory structures are consistent for all team members*

# Why do I need version control?

- ▶ Make changes to code with confidence - *can always be reverted if necessary*
- ▶ Reproducibility - *version control can complement your lab notebook*
- ▶ Work as a team - *file names and directory structures are consistent for all team members*
- ▶ The list goes on...

# Git

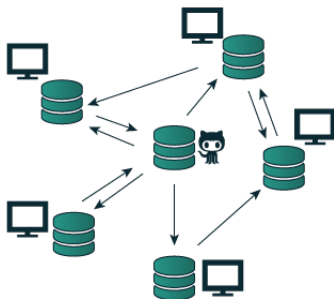
In the scientific world, Git (and Github) is the most widely used version control system.





# Git

In the scientific world, Git (and Github) is the most widely used version control system.



*repository*: A central storage area where a version control system stores old revisions of files and information about who changed what, when.

# How do you get your own repository?

Let's configure Git first:

```
$ git config --global user.name "Your name goes here"  
$ git config --global user.email you@yourdomain.com  
$ git config --global core.editor vim  
$ git config --global color.ui auto
```

Then initialize your first repository:

```
$ git init
```

# How do you get your own repository?

Let's configure Git first:

```
$ git config --global user.name "Your name goes here"  
$ git config --global user.email you@yourdomain.com  
$ git config --global core.editor vim  
$ git config --global color.ui auto
```

Then initialize your first repository:

```
$ git init
```

**You Try (10 minutes):**  
**Exercises (1) - 2**

# Git allows you to save snapshots of your directory

*commit*: snapshots of your directory.

# Git allows you to save snapshots of your directory

*commit*: snapshots of your directory.

- ▶ There is metadata associated with each commit (snapshot):
  - ▶ the date the snapshot was taken
  - ▶ who took it
  - ▶ what files were modified
  - ▶ the changes made on those files
  - ▶ etc.

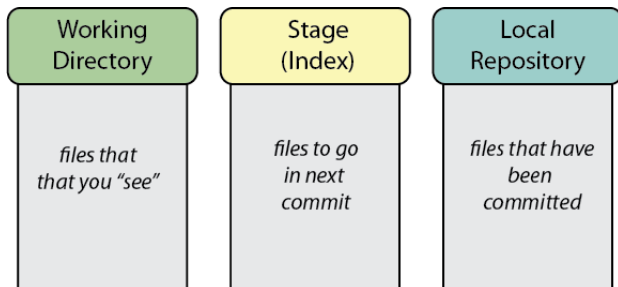
# Git allows you to save snapshots of your directory

*commit*: snapshots of your directory.

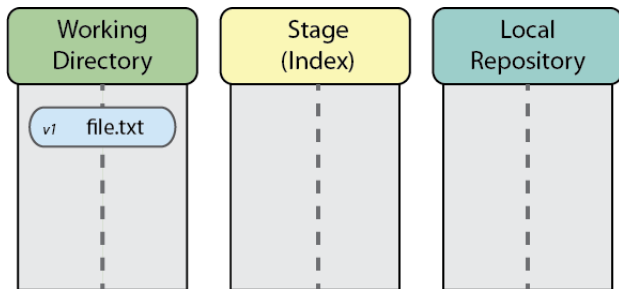
- ▶ There is metadata associated with each commit (snapshot):
  - ▶ the date the snapshot was taken
  - ▶ who took it
  - ▶ what files were modified
  - ▶ the changes made on those files
  - ▶ etc.
- ▶ Git will enable you to:
  - ▶ track the changes made to files in your directory
  - ▶ revert the entire project to a previous snapshot
  - ▶ review changes made over time
  - ▶ view who modified a file
  - ▶ etc.

## A little more vocabulary:

There are three main *trees* or *collections of files (and metadata)* in Git:

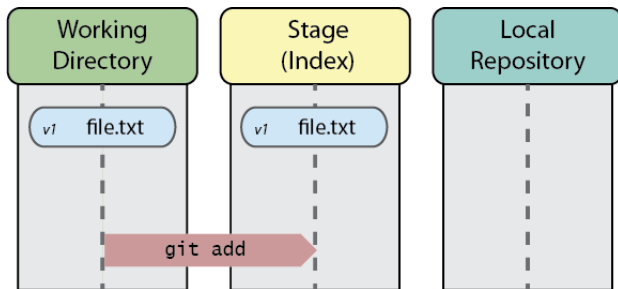


# How to save snapshots with Git

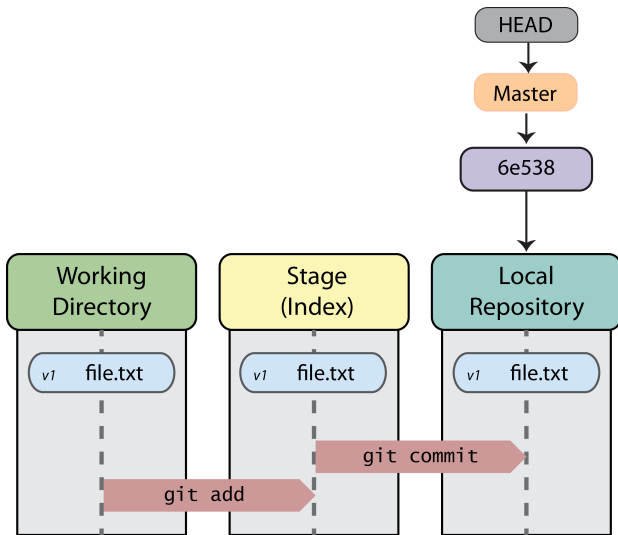




# How to save snapshots with Git

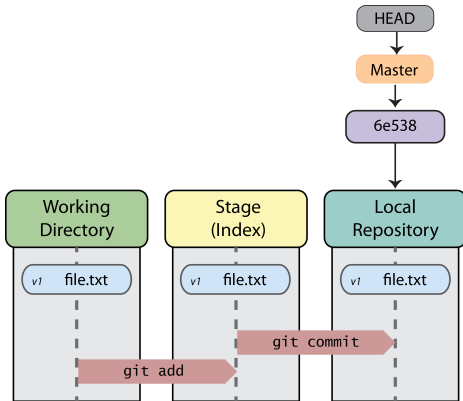


# How to save snapshots with Git



# How to save snapshots with Git

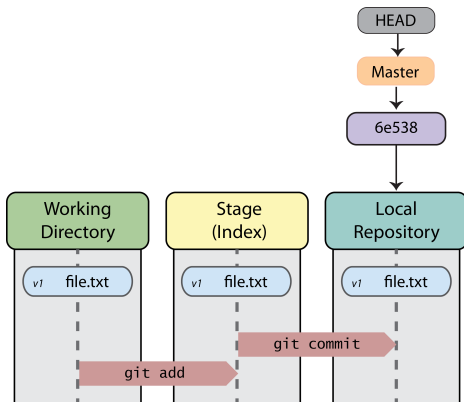
*SHA-1 hash*: unique 40-digit computer-generated identifier for each revision (or commit)



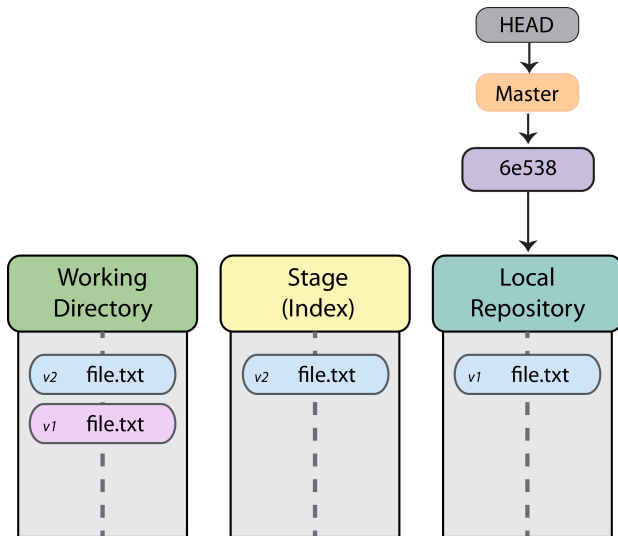
# How to save snapshots with Git

*SHA-1 hash*: unique 40-digit computer-generated identifier for each revision (or commit)

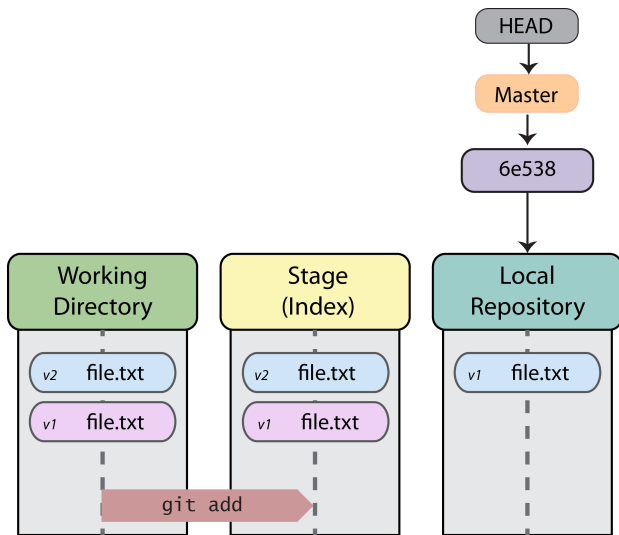
*HEAD*: reference to the current branch or commit



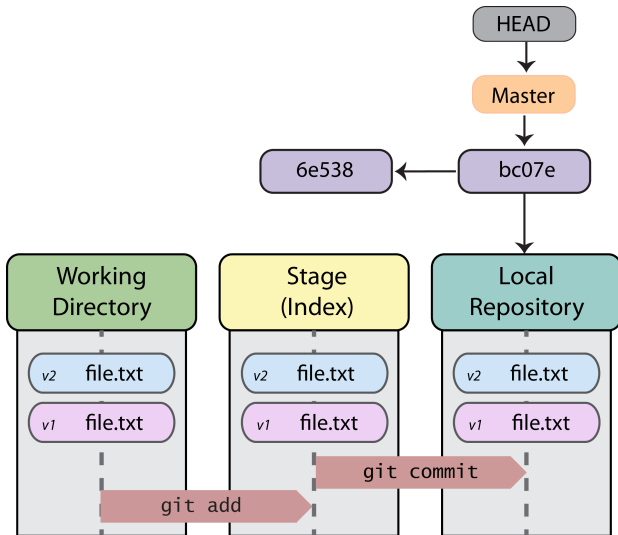
# How to save snapshots with Git: Keep working!



# How to save snapshots with Git: Keep working!



# How to save snapshots with Git: Keep working!



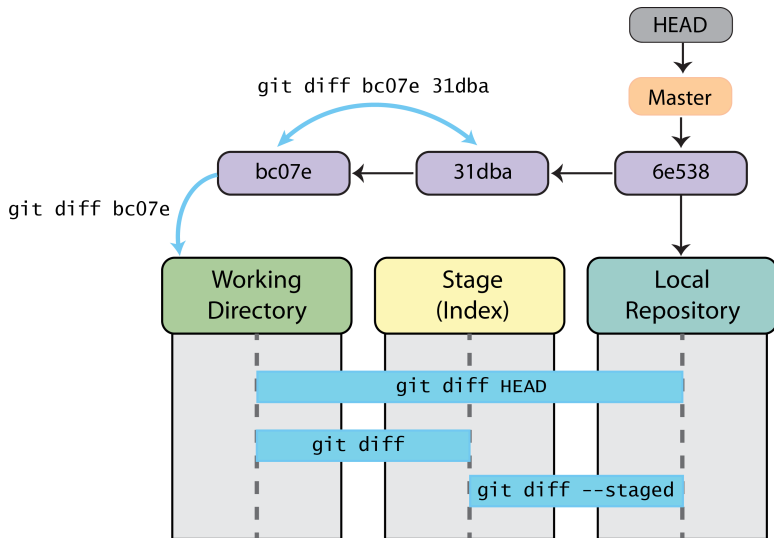
## A simple story so far: what else can we do?!

`git log`: view the change history (commits) of the current repository.



# A simple story so far: what else can we do?!

`git diff`: view changes between files and commits



How do we do this for real?

*An Example*

Now it's your turn.

**Questions?**

Now it's your turn.

**Questions?**

**You Try (15 minutes):**

**Exercises 3**