



# DSML BOOTCAMP

GIT RECAP

# WHY?

## ADVANTAGES



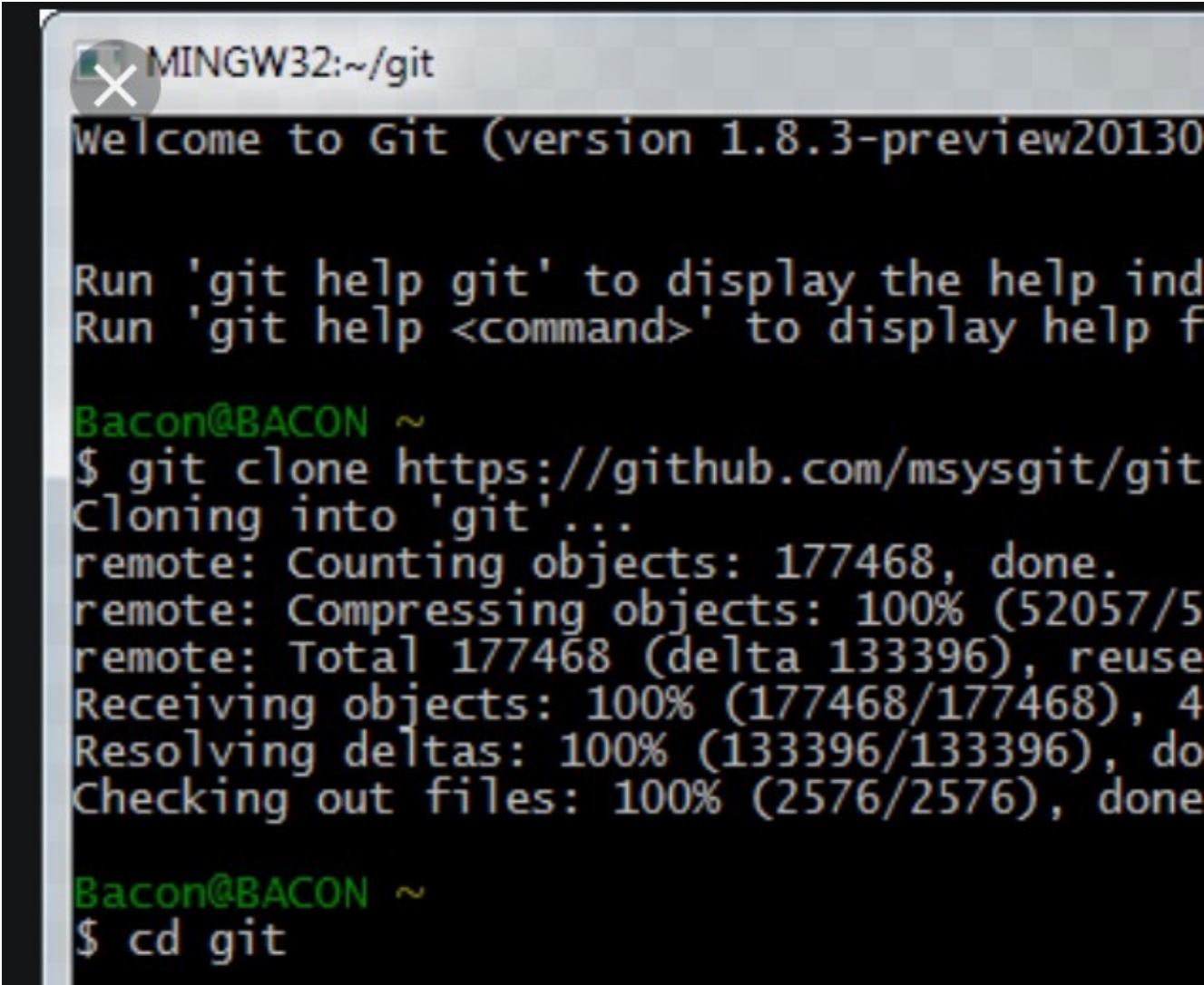
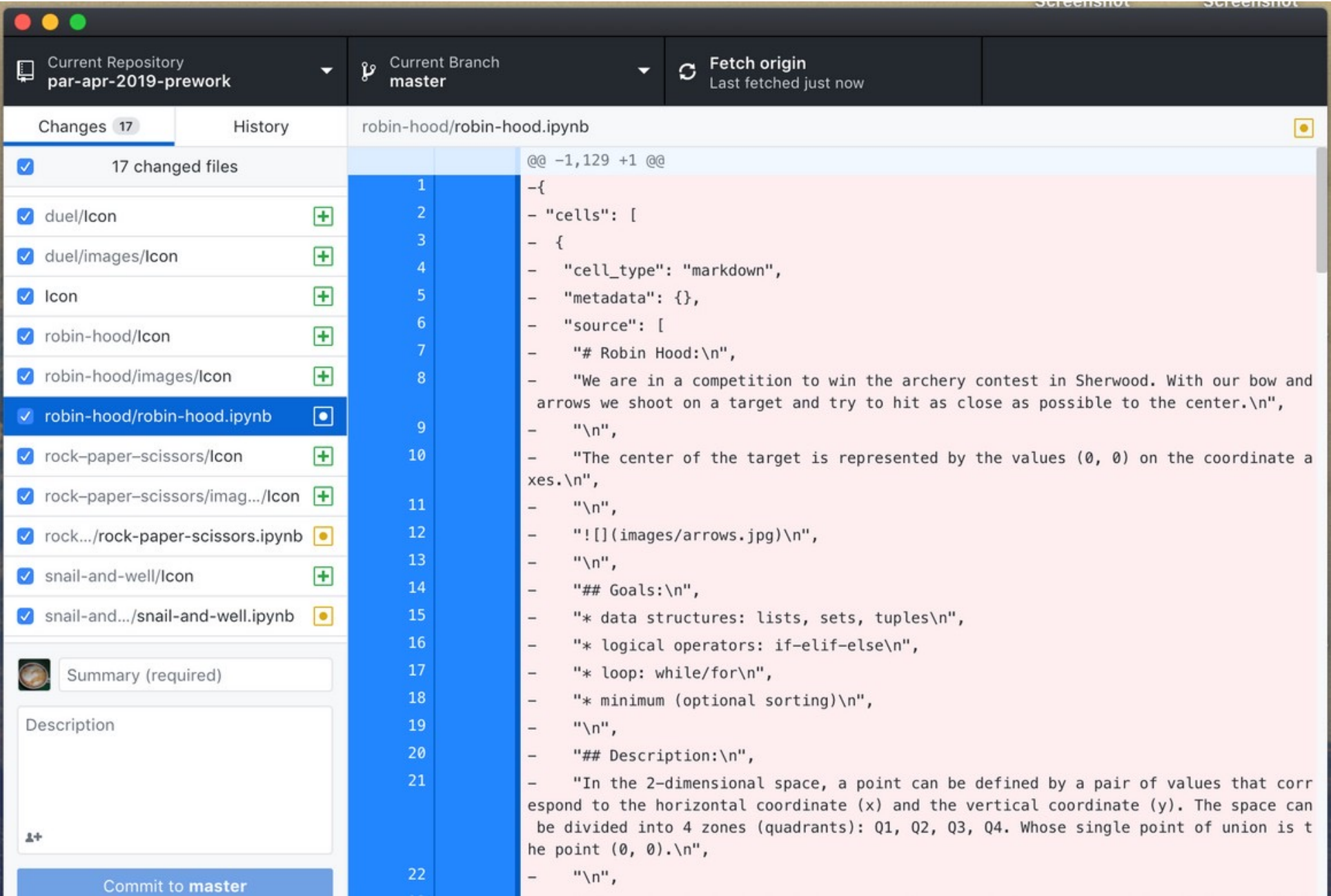
**Version control with branching** - When a developer wants to start working on something—no matter how big or small—they create a new branch. This ensures that the master branch always contains production-quality code.

**Local repositories** - each developer gets the whole repository complete with history. Whatever everyone else does, the others can just keep working on their own repositories.

**Community** - distributed development.

# POSSIBLE INTERFACES

## APP OR CODE



# WHAT CAN YOU DO IN GIT

AMONG OTHER THINGS



- **Create a repository**
- **Add files to a repository**
- **Examine the history of your project**
- **Jump to an earlier state and back**
- **Publish the code on a public repository**
- **Manage multiple code branches in parallel**
- **Merge branches**

# CREATE A SOURCE REPOSITORY

## WHAT TO DO



### **Terminal**

#### **Go in the folder**

>git clone 'url'

### **App**

**Go on File >Clone repository >choose url >paste url**

# CHANGES FILES TO REPOSITORY

## WHAT TO DO



### **Terminal**

#### **Go in the folder**

> git status

> git pull

> git add

> git commit -m "comprehensible short explanation"

> git push

#### **CONFLICTS?**

### **App**

**Same thing (no status), only visual**

# BRANCHING

WHAT IS IT, WHEN IS IT, HOW IS IT

## Terminal

### Go in the folder

> git brunch 'meaningful\_name'  
> git checkout origin new\_branch

## App

### Same thing, only visual

<https://learngitbranching.js.org/>

# BRANCHING VS FORKING

## FORKING IS REALLY JUST GITHUB

**Forking** - Cloning the repo on your user account. You clone every branch it had at the time you forked. Use this for personal work on someone else's repo.

**Branching** - used to work on a version, some improvement or an additional feature. It could be used for experimental work. Use this for team work.

**Pull Request** - one pull one task.

**Commit** - set of changes in the code. You transfer logs of change



# CHANGES FILES TO REPOSITORY

## WHAT TO DO



**Create a branch**

**Change the file**

**Commit the changes**

**Push the changes into the branch**

**Checkout master**

**Do other changes and push (so now master is not updated)**

**Merge the branch into master**

ANY  
QUESTIONS ?