

A QUICK INTRO TO

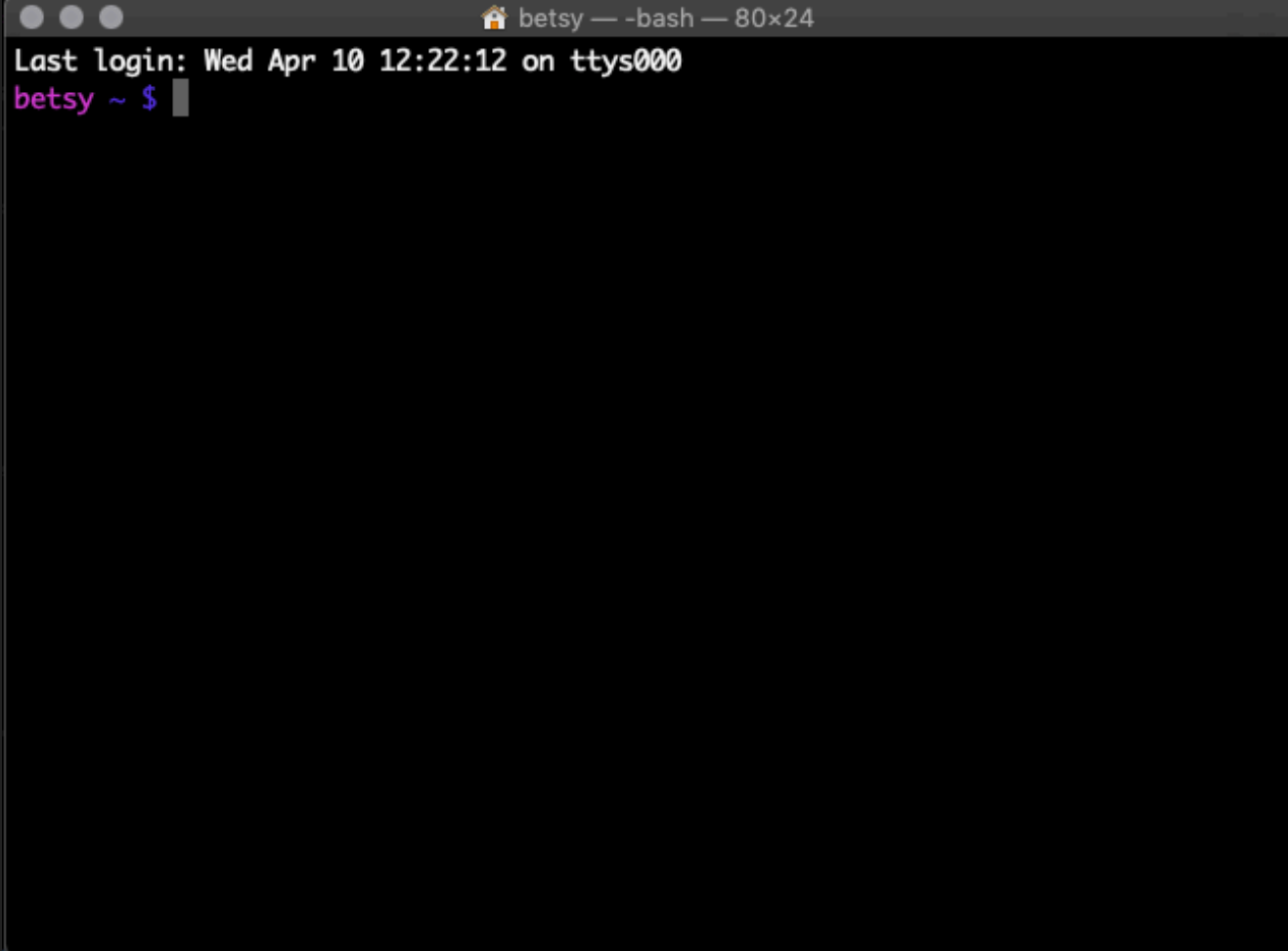
GIT

WHAT ARE WE GOING TO DO TODAY?

- ▶ Set up your Setup
- ▶ What is GIT?
- ▶ Why should you use GIT?
- ▶ How to use GIT?
- ▶ Want more?

TERMINAL/SHELL

- ▶ For Mac: open the terminal
check if git is installed:
git --version
- ▶ If not install git:
<https://git-scm.com/downloads>
- ▶ For PC: download git bash
<https://gitforwindows.org/>

A screenshot of a terminal window. The title bar at the top shows a home icon, the name 'betsy', and the command '-bash' followed by the window size '80x24'. The terminal content shows 'Last login: Wed Apr 10 12:22:12 on ttys000' in white text. Below that, the prompt 'betsy ~ \$' is displayed in pink text, with a black cursor block following the dollar sign.

```
betsy ~ $
```

VERSION CONTROL

- ▶ Without version control: one file, only the latest version, impossible to go back to later versions
- ▶ Hand-made version control: multiple files, all saved versions available, changes are traceable
- ▶ git: one repository, entire history of your project in one place, changes are traceable, undoable, easy to find

WHAT GIT CAN DO

- ▶ Show the history of your project - step by step
- ▶ Show you the changes you've made on every step
- ▶ Let you try new stuff without risking anything
- ▶ Let you search for specific changes in your project history

3 STAGES OF GIT

▶ Working directory

where you work on
your project

▶ Staging Area

where you decide
what is committed
to your repo

▶ Repository

where the
history of your
project is
stored

INIT —————> **ADD** —————> **COMMIT**

WHAT YOU HAVE TO DO

- ▶ Show the history of your project - step by step
commit often
- ▶ Show you the changes you've made on every step
commit in logical steps
- ▶ Let you try new stuff without risking anything
make a new branch for every feature
- ▶ Let you search for specific changes in your project history
add meaningful commit messages

BASIC SHELL COMMANDS

- ▶ `mkdir` - make directory - creates a new folder
- ▶ `cd folder-name` - change directory - goes inside the folder with folder-name
- ▶ `cd ..` - goes up one folder-level
- ▶ `ls` - shows you a list with all the items in the folder you're in
- ▶ `pwd` - print working directory - shows you the path your in on your folder structure

BASIC GIT COMMANDS

- ▶ `git init` - transforms the folder you're in into a repository
- ▶ `git status` - shows you the status of the files in your project (is something added or waiting to be committed?)
- ▶ `git add filename` - adds file with filename to the staging area
- ▶ `git add .` - adds all uncommitted files to the staging area
- ▶ `git commit` - commits all files from the staging area to the repo
- ▶ `git commit -m „meaningful message“` - commits all files from the staging area to the repo with commit-message

MORE BASIC GIT COMMANDS

- ▶ `git log` - shows the history of your project
- ▶ `git log --oneline` - shows short history of your project
- ▶ `git branch branch-name` - makes new branch
- ▶ `git checkout branchname or SHA` - checks out branch or commit
- ▶ `git diff` - shows changes from last commit to working directory
- ▶ `git clone link` - clones repo from link (e.g. GitHub)

WANT MORE?

THE HELP

- ▶ Git

<https://git-scm.com/>

- ▶ Git Cheat Sheet

<https://www.git-tower.com/blog/git-cheat-sheet>

- ▶ Git Pro Book

<https://git-scm.com/book/en/v2>

- ▶ Google & YouTube!!!

THANK YOU!