Git

-Collaboration

-the 2nd best time to learn git is today

-living and breathing git

-intro to git, merging, fetching and pulling, git tags, installation, diffing, github odds and ends, git behind the scenes, git basi

Intro to git

Installation

Git basics

Committing in detail

Branching

Merging

Diffing

Stashing

Undoin changes

Git hub intro

Fetching and pulling

Github odds and ends

Collaborative workflows

Rebasing

Interactive rebasing

Git tags

Git behind the scenes

Reflogs

Custom aliases

????

Note on the exercises

Exercises are at the end

Accessing the slides and diagrams

Introducing git

1. What is git- critical
2. Understanding the diff between git and github-critical
3. The history of git
4. Who uses git
5. What does git do for us-critical

What exactly is git

* Version control system (VCS)
* Version control is software that tracks and manages changes to files over time
* Revisit earlier versions of files
* Compare changes between versions
* Undo changes

There are other VCSs – subversion, Mercurial

We only need to care about Git –

1. 88,4% of developers use git compared to other VCS

Some games allow u to create checkpoints

U can screw things up 🡪 but u can always go to those checkpoints

A screenshot of a computer

Description automatically generated

Optional history of git

Who uses git

Git vs github

* Git is a software that runs locally on ur machine. Can use git without touching github
* Github is a service – webservice – its an online place to share work that is done using git

Github connects git projects

* Do a git project
* Take the git project and all its history and put on github
* Then two ppl can take ur work and put in on their machine
* They can play around with it

Installation and setup for git

1. How to use git – command line or gui
2. Installing git
3. Gitkraken – git gui
4. Configuring ur git name and email
5. Terminal crash course

Git is primarily a terminal tool

The rise of guis – eg tortoisegit, gitkraken

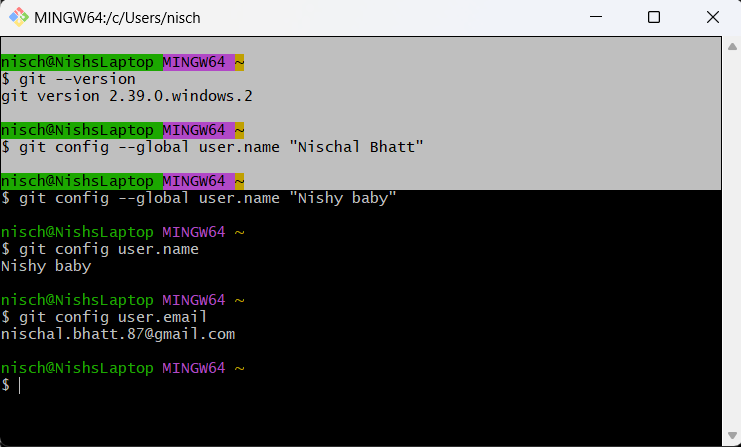
Better not to use gui – coz user can become dependant on 1 piece of software

Command line –

Some advances git features are only available on the command line

Windows git installation

1. Git was designed to run on unix based interface, so it was designed for linux
2. In windows machine, they dun come with a unix based prompt
3. Bash – we have to download git bash on windows – so that the windows machine can emulate a unix like environment
4. Go to git website and download git for windows
5. Git-scm.com
6. C;/ program files / git
7. Use vs code as gits default editor
8. A screenshot of a computer

   Description automatically generated
9. Need to configure name and email
10. So that when u commit u know who committed it etc
11. 
12. GitKraken

Core git commands

1. Git repos
2. Git init
3. Git status
4. Understanding the .git folder
5. Committing workflow
6. Git add
7. Git commit
8. Git log

What is a git repo

A git repo is a workspace which tracks and manages files within a folder

A picture containing text, electronics, screenshot, software

Description automatically generated

A screenshot of a computer

Description automatically generated

The .git folder is where all the git stuff occurs

A common early git mistake

.git will watch even the subdirectories

Eg. Below

A screenshot of a computer

Description automatically generated

Even the inner Intro folder is tracked by git

Do not init a repo inside of a repo !

Always git status first before running git init

If u remove a .git folder, then that directory automatically loses its git capabilities and it becomes a normal folder like any other folder u have on ur useless boring machine

1. Work on stuff
2. Add - group specific stuff together
3. Commit

Working directory 🡪 git add 🡪 staging area 🡪 git commit 🡪 repository

A screenshot of a computer program

Description automatically generated with medium confidence

When we make a commit , we are updating the .git folder

---

Working directory

Staging area

Repository

----

A screenshot of a computer program

Description automatically generated with medium confidence

A picture containing text, screenshot, software, font

Description automatically generated

If u use git commit alone, there will be a popup forcing u to put the message in

So might as well run git commit -m “msg”

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screen shot of a computer program

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated

Writing atomic commits

Working with a gui

Amending commits

Git ignore

Writing good commit messages

Navigating git documentation

Git-scm.com/docs – documentation

Keeping your commits atomic

-keep each commit focused on a single thing (single feature)

Don’t add everything to 1 single commit

Git add some files

Git commit -m “some commits”

Commit messages : : Make xyzzy do frotz (present tense)

A picture containing text, screenshot, font

Description automatically generated

A screen shot of a computer screen

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated with medium confidence

Lets say we want to change the commit message add a gitkraken.txt file 🡪 we want to remove the .txt

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

Lets say we want to ignore the wordDoc.dox

A screenshot of a computer program

Description automatically generated with medium confidence

We don’t want this to happen

When we type git status, we don’t want wordDoc.docx to appear

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated with medium confidence

If there is a .git file, there should be a .gitignore file too

Branching

What is branching

Understanding git HEAD

Git branch

Git switch

Git checkout

HEAD and refs behind the scenes

Deleting and renaming branches

HEAD & REF behind the scenes

Master vs main branch

Introducing branches

Each commit references 1 parent commit

Branches – alternatives timelines for a project

What we do on one branch – does not impact the other branches

In git u are always working on a branch

When u make a new git repo 🡪 automatically the branch that u start on is called master

HEAD is the thing that is currently open – that im checking out in my repository