

Introduction to Git version control & GitHub

NHS-R Conference 2023

Presented by:
Helen Richardson and Joe Wilson

Who we are:



Helen Richardson:

Joined NHS Digital in 2019 (now England). I have worked on implementing RAP principles with NHS Digital's statistical publication teams on building robust, transparent and automated analytical pipelines. Currently, I'm working on CVD Prevent tool, a pipeline used to link and curate CVD PREVENT audit data to HES and death registration data.



Joe Wilson:

Joined NHS Digital in September 2021 as a Graduate Trainee, after completing a Masters in Data Science. First placement was with NHS Spine. Second placement was working with the Business Intelligence Dashboards team, developing a testing framework for Tableau dashboards. Currently assigned to the RAP Squad, engaging with teams within NHS Digital and helping them learn and utilise RAP techniques and principles.

Schedule

- **9:30 Introduction to version control, Git and GitHub**
- **Two short breaks in between practicals**
- **Practical using Git/GitHub/Codespaces**
- **(roughly) 12:00 – 13:00 Lunch break**
- **13:00 – 16:30 Git Collaboration with allocated breaks**

Workshop requirements

You will need:

- **Access to a browser**
- **Internet connection**
- **GitHub account! (this is free)**



What will we learn today?

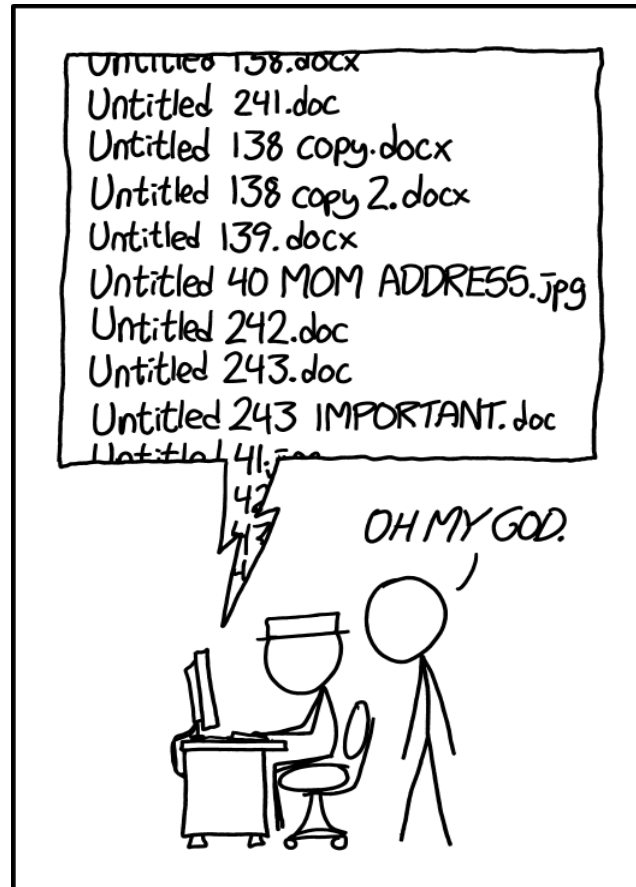
- What is version control, Git and GitHub?
- What is a repository, branches, README, .gitignore etc.?
- How to apply version control to my project using Git commands?
- How to create/fork a repository?
- Use GitHub Codespaces

Afternoon session:

- How to submit a Pull Request on GitHub?
- How to use Git collaboratively (within a team setting)?
- How to handle merge conflicts?



My documents folder



PROTIP: NEVER LOOK IN SOMEONE ELSE'S DOCUMENTS FOLDER.

There must be a better way...

What is version control?

- **Version control is the practice of tracking and managing any changes on project's code, files or folders over time.** This allows you to observe a detailed history of the changes made and enables your team members to collaborate on the same project.

And how does it work?

- We use **Git**, a **version control** software tool, to help us **1)** transform our working folder into a **Git** project folder and **2)** track changes (i.e. adding/removing files, changing files).
- We then use an online version control system aka **Git** hosting platform (e.g. GitHub or GitLab) to upload and store our **Git** project folders, known as **repositories**.



Why should we care about version control?



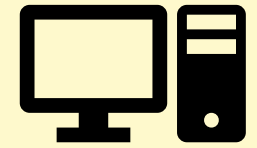
**Tracking changes –
Restoring previous
versions if things
break.**



**The ability to review
someone's changes
and to leave
comments.**



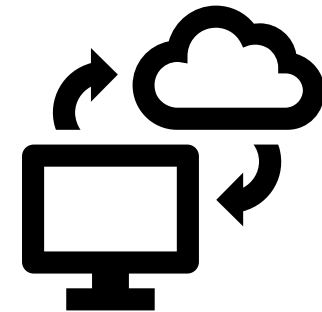
**Trying out
experiments
without the risk of
breaking your main
code.**



**Avoid code being
hidden away on
someone's
machine.**

How do **repositories** work?

- In Git, each user has a **copy** of the entire **repository** (project's working directory/folder) on their computer (**offline**) or a virtual machine, hence, they can work separately offline until they opt to push their updated version of the code or files to the **remote** (**online**) central **repository**.
- **Remote repositories** are versions of your project that are hosted on an online Version Control System. At NHS England, we store our **Git projects** on GitLab site for internal users (GitHub site for the public).



What does a repository look like?

- Example of a remote (online) repository:

https://github.com/nhs-r-community/git_training

- Practical: Create a repository:

https://github.com/nhs-r-community/git_training/blob/main/guides/how_to_create_a_repo.md



10 - minute breather

To recap: Git vs GitHub



- **Git is a software**
- released in 04/2005
- is maintained by Linux
- **manages code and file history**
- **is installed locally on your computer**
- focuses on version control and code sharing
- no user management ability



- **GitHub is a service/platform**
- released in 2008
- is maintained by Microsoft
- **hosting service for Git repositories**
- **available online**
- focuses on centralised source code hosting
- has user management features

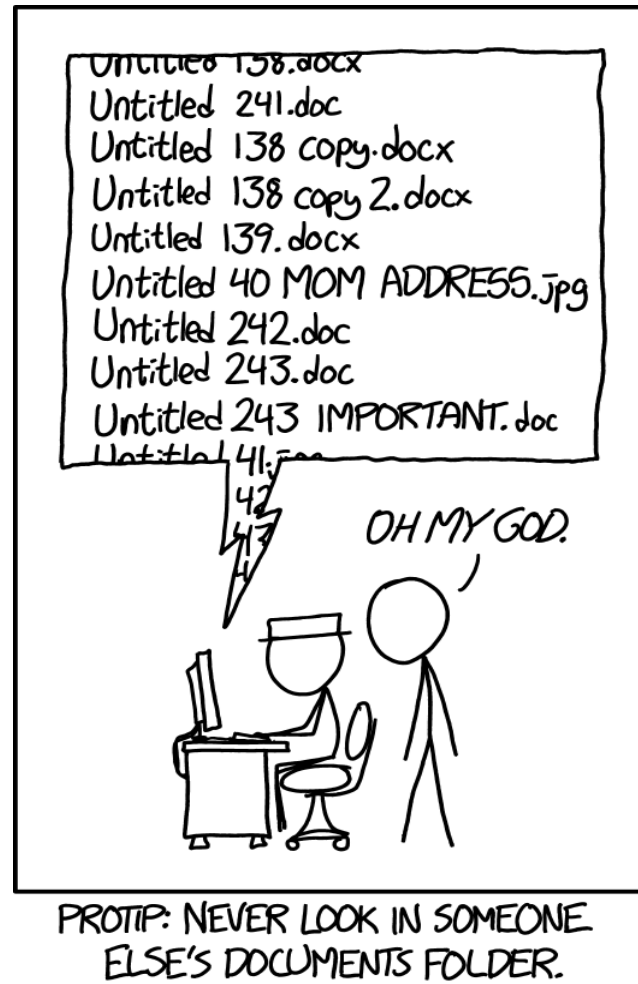
tl;dr

Git is the tool to make things appear on GitHub (basically).

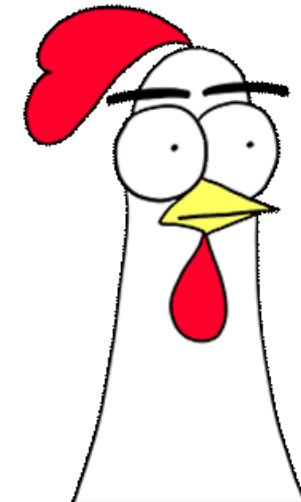
Other Git hosting services:

- GitLab
- Bitbucket
- SourceForge
- GitBucket
- AWS Elastic Beanstalk
- Etc.

Chaos



How can Git help?



What is a branch?



What is a branch?

"**Branch**" is another word for "**version**".

Usually when developing a document or slides for a presentation, we could have numerous versions of the same slides, with small or major differences.

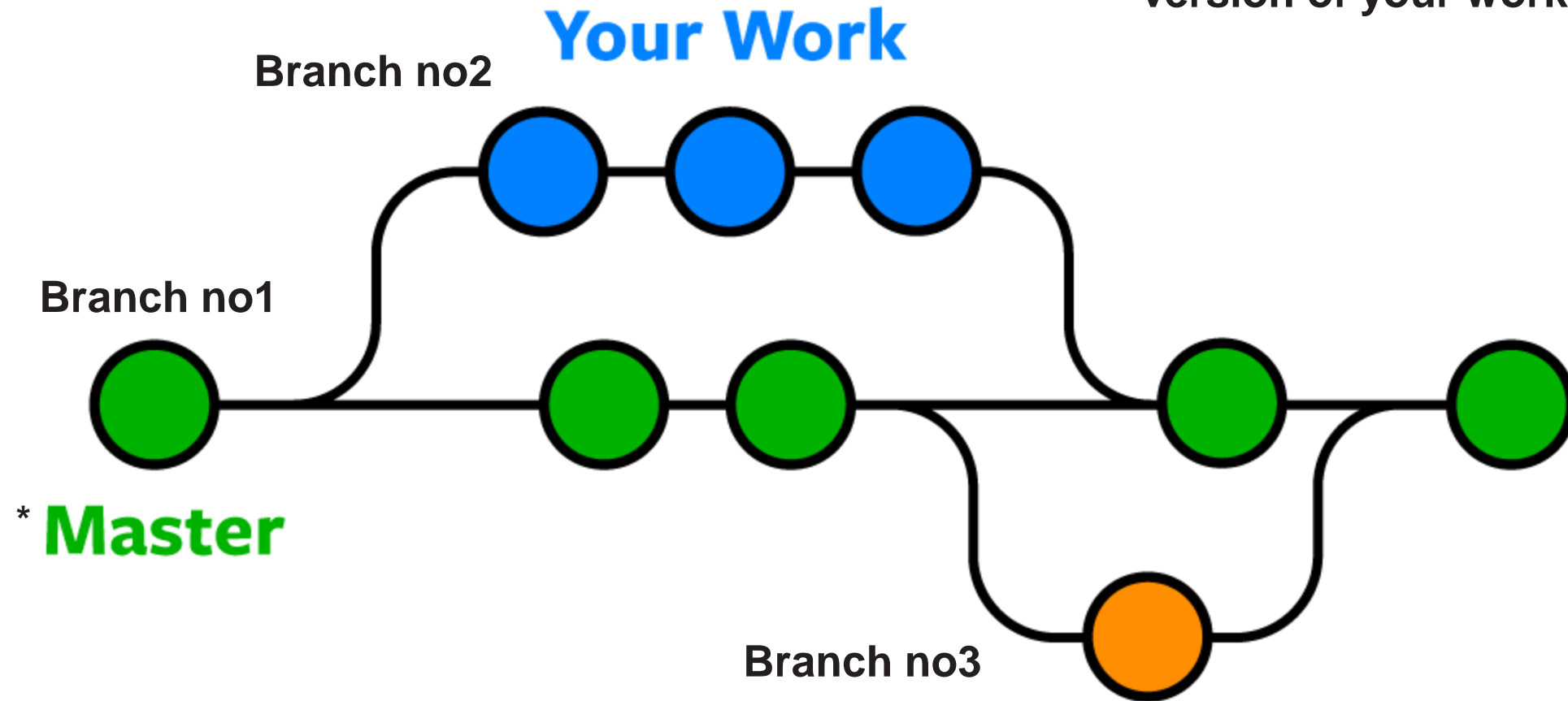
By using different **branches** (or versions) of the same code or document (any type of file), we can safely work and test without breaking our main code or file, that resides in the default main branch of the repository.

In Git, we use the word "**main**" as the name of the default version/branch of our project repository.



How does it work?

- Branch = version
- Branch no1 is the default version of your work



*You will see this also as
“Main” or “Dev” (or even
other names!)

Someone Else's Work

What's next?

- **Log into GitHub.**
- **Practical: creating branches, making changes to files, using Git, push changes back to GitHub.**
- **We are going to use Codespaces, a virtual machine via GitHub for the above.**
- **Further instructions:**
 - https://github.com/nhs-r-community/git_training/blob/main/README.md
- **First, 10-minute breather**



10 - minute breather