Git/GitHub: Workshop

15/11/2023 - Filip J. Cierkosz

Introduction - Filip J. Cierkosz

Roles:

- BSc Computer Science student (3rd year)
- Software Engineer @ The Curve
- Student Advisor @ Sheffield TechVision
- Athlete

Fields of interest:

- Machine Learning (LLMs, NLP)
- Entrepreneurship
- Sports

Feel free to visit my **website** at:
filipcierkosz.com

You can find me on **GitHub**:
github.com/chizo4

…or let’s connect at **LinkedIn**:
linkedin.com/in/filip-cierkosz

Aims – Git Workshop

We're gonna explore the following concepts:

- Motivation behind learning **Git/GitHub** tools
- Understand the concept of **Version Control**
- Recap of basic **Git** and **Linux** commands
- **Git vs GitHub**
- **Branch management** in Git
- Collaborative workflows
- Tips & additional resources
- **Notice**: You can find **Installation Guides** here.

Notice:

You can access the workshop materials (and more!) at:

(QR has been removed.)

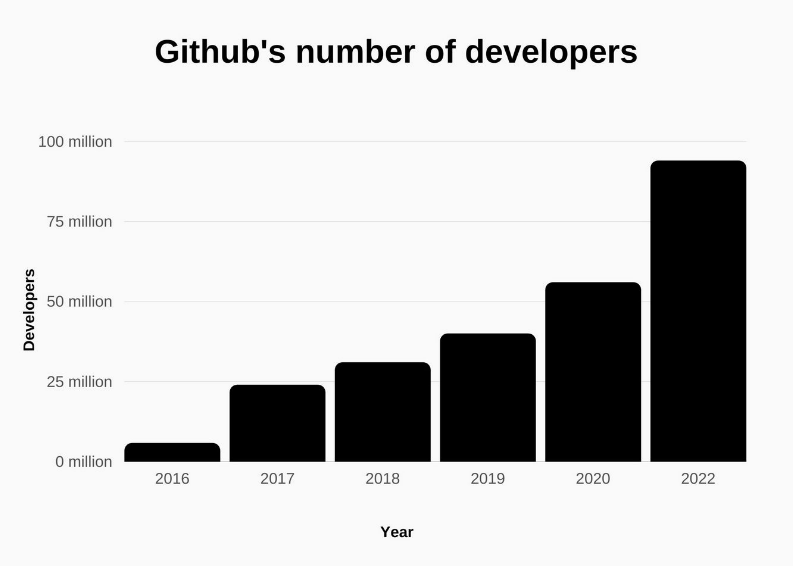
https://github.com/chizo4/git-workshop

Motivation:
Why is it worth learning how to use Git & related tools?

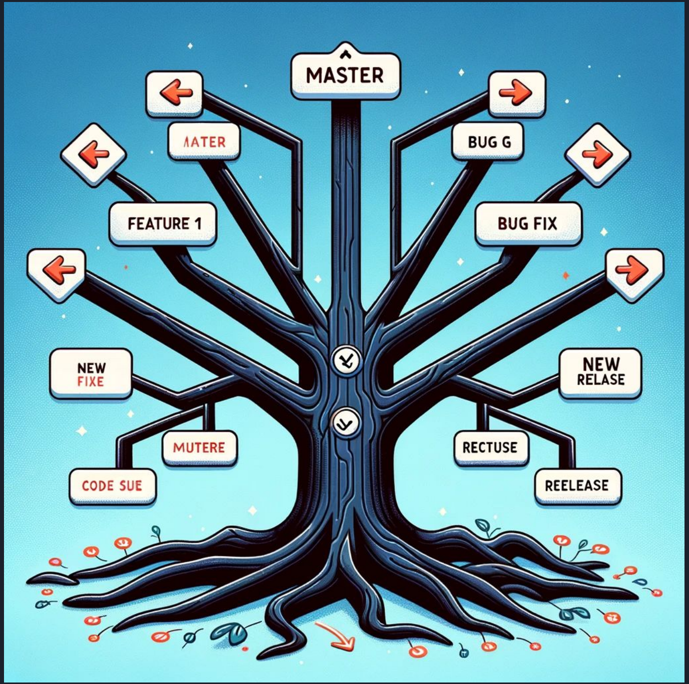
Git – Learning Motivation

Reasons to learn **Git/GitHub** :

- **Better** coding efficiency and software maintenance
- Useful for **collaboration** while developing software
- Relevance in **personal**, **university** and **open-source** projects
- Many companies **require** sufficient knowledge of these technologies
- 100M+ developers use GitHub worldwide
- The number of users just keeps growing!



Understand Version Control



(...thanks DALL-E!)

Version Control:

- Can be defined as a system that records changes to a file or set of files over time
- Useful to track iterations, revert back to previous versions, manage parallel work streams
- **Crucial for complex projects!**
- Git is an instance of a VC

Git ≠ GitHub

- They are not the same!

Git vs. Github - Differences

Git - Overview:

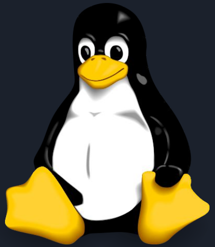
- Distributed version control system (VCS) to record different versions of a file (or set of files)
- Lets users access, compare, update and distribute any of the recorded version at any time

GitHub - Overview:

- Platform that lets you host, share and manage your files on the Internet
- GitHub users use Git underneath to manage their repositories easily
- Other online platforms using Git, e.g. GitLab

Let’s recap fundamentals

LCL – Basics Recap



Command	Description
`cd [directory]`	Change the current directory.
`ls [options]`	List contents of the current working directory.
`mkdir [directory]`	Create a new directory.
`rm [file]`	Delete a file (or directory).
`touch [file]`	Create a new file.
`cp [source] [destination]`	Copies files or directories into desired location.

Git – Basics Recap

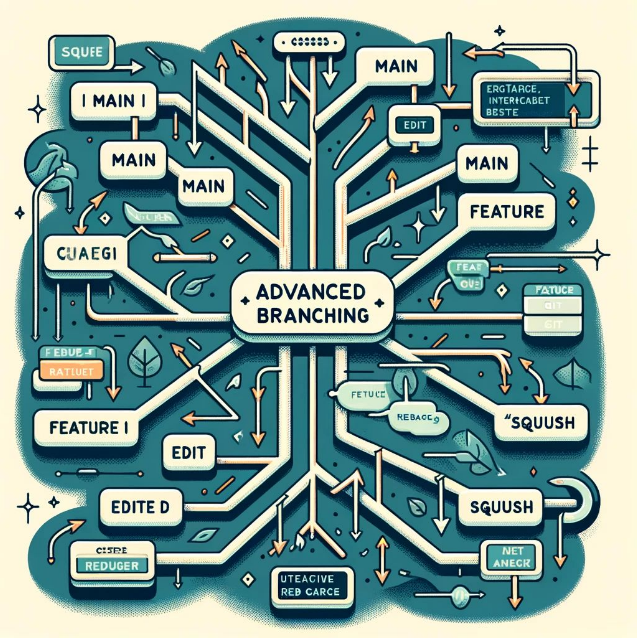
Command	Description
`git init`	Initialize a new Git repository.
`git add [file / dir]`:	Stages a file (or directory), preparing it to be committed to the repository.
`git commit -m "[commit message]"`	Commit the staged files to the repository.
`git status`	Show status of changes as untracked, modified, or staged.
`git push [alias] [branch]`	Upload all local branch commits to the remote.
`git pull`	Fetch and merge changes from remote to working directory.
`git clone [url]`	Clone a repository into a new directory.

Any questions so far?

Demo 1:
Let’s try all this in action!

Now, let's talk about <u>branching</u>...

Git – Branch Management



(Good friend DALL-E…)

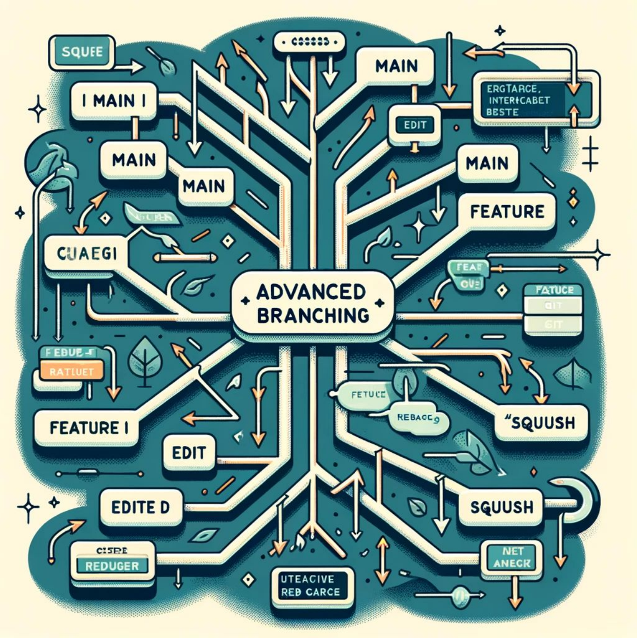
Why would you care about branching?

- Working on one branch is **inefficient** for a team of developers
- Branches for **isolated development** → clean organization & management
- Creating and switching branches
- Integrating branches → **merging** with main version of the software

Git – Branch Management

Command	Description
`git branch -a`	Lists all the local and remote branches of the repository.
`git checkout [branch-name]`	Switches to a specified branch name.
`git checkout -b [branch-name]`	Create a new branch and switch.
`git merge [branch-name]`	Merge a branch into the current branch (rare usage).
`git branch -d [branch-name]`	Delete a branch.
`git switch -c [branch-name]`	Alternative to checkout with passing current changes.
`git log`	Show the commit logs for current branch.

Git – Advanced Branching



(Good friend DALL-E…)

Key concepts associated with the <u>advanced branching</u>:

- Merging vs. rebasing → how to avoid the conflicts
- Interactive rebase → the conflict savior!
- Resolving conflicts - why do they appear?

Git – Advanced Branching

Command	Description
`git rebase [branch-name]`	Move all changes to the base branch to the tip of the branch.
`git rebase -i [branch-name]`	Interactive rebase based in reference to a branch.
`git stash`	Temporary store all the modified tracked files and stages changes.
`git stash pop`	Apply changes from the most recently stashed state and then removes it from the stash stack. It's a quick way to reapply your stashed changes and continue where you left off.

Demo 2:
Let’s try these advancements in action!

Any questions?

Learning Outcomes

Today, you have learned following concepts:

- Version Control concept
- Git fundamentals recap
- Git vs. GitHub
- Branching concepts
- Practical demos
- **Further resources?!**

Thank you for your attention - I hope you found the session useful!

Filip J. Cierkosz – find me at:

- Website: filipcierkosz.com
- LinkedIn: linkedin.com/in/filip-cierkosz
- GitHub: github.com/chizo4

Resources

- S. Ajibola (2022). “How to Use Git and GitHub – Introduction for Beginners”. Available at: https://www.freecodecamp.org/news/introduction-to-git-and-github/#what-are-git-and-github
- O. Sofela (2020). “Git vs GitHub – What is Version Control and How Does it Work?”. Available at: https://www.freecodecamp.org/news/git-and-github-overview/
- K. Bromand. “Your first time with git and github”. Available at: https://kbroman.org/github_tutorial/pages/first_time.html