Git & GitHub

Created	@July 11, 2021 12:41 AM
Created By	Luke Landau
Last Edited By	Luke Landau
Last Edited Time	@July 12, 2021 10:06 AM
Note	Created for and by a pack of fighting mongooses (take with mucho salt)
Stakeholders	
Status	
Type	

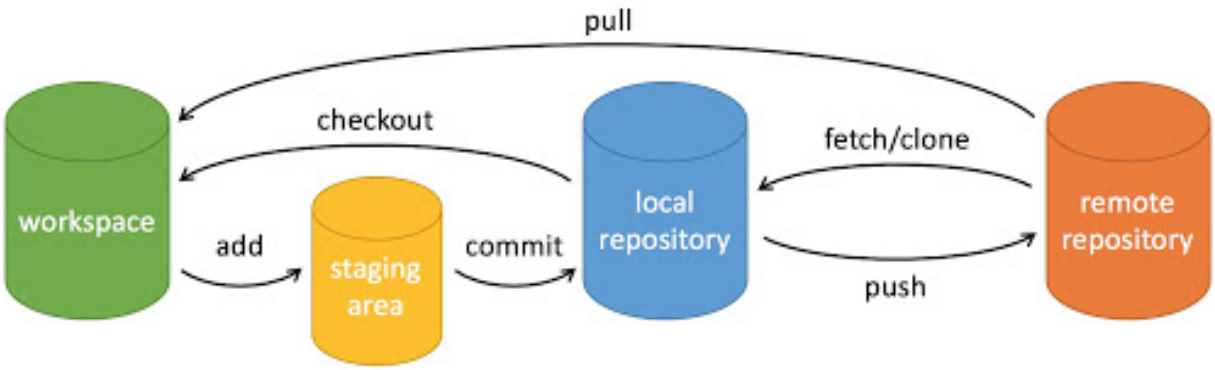
Version Control

- a version control system (VCS) is software designed to keep track of changes to code
- good VCS's maintains a full change log with authorship data
  - this ensures all changes are recorded, transparent and can be rolled back
- git is the big boy, used by pretty much everyone in industry

Git and Github

- **git** is a local program, able to pull, push and commit to a server like **github**, which can;
  - host repositories
  - facilitate code reviews

    - provide file, commit and version browsing
    - clone reposoitories
    - manage hierarchy and permissions
- **gitlab** is similar to **github**, but offers their software to download onto private servers

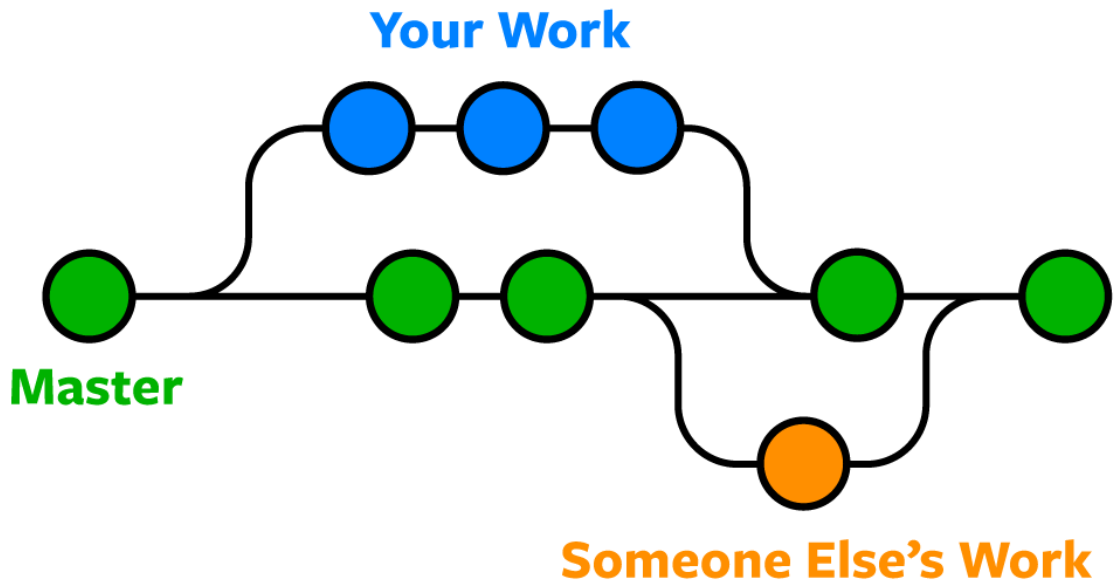


- **repo** - repository storing all versions of a project
    - local - repo downloaded to a device
    - remote - repo hosted on a server
- `git config —-global user.name "Your Name"` - set username
- `git config --global user.email "your-email-address@server.extension"`
- `git config --global core.editor nano` - change default terminal text editor
- **creating** a repo
    - create the repo on github (readme)
        - `git clone [SSH address of the repo]` -
    - `git init` - initialise new repo from project directory
- **updating** a repo
    - Save all files relevant to the commit
    - `git add` all the files to the staging area
    - `git status` checks the status of staging area etc

    - `git commit -m "Summary of Change"` - commits changes to local repo, with text
    - `git push origin [branch]` - push to remote repo for branch of a project
- **cloning** a repo
    - `git clone github-address` - clone and create a folder for repo
    - `git clone github-address .` - clone repo files into current folder
    - to find a repo's `github-address` go to GitHub → Code → Clone → HTTPS / SSH
        - HTTPS - every push requires GitHub username and password
        - SSH - push without login details - requires SSH key setup beforehand
- **deleting** a repo
    - find the `.git` folder you want to delete with `ls -a`
    - `rm -rf .git` to delete repo
    - `git status` to check it has been deleted
- **merge conflicts**
    - git has "auto-merging" where only changes are updated to the repo
    - sometimes multiple devs have changed the same line, creating a merge conflict
        - git cannot solve this issue, so a dev is required to sanitise the changes of the files
        - when conflicts are resolved, a new commit is sent with an appropriate message
- **.gitignore**
    - a file we can create in a repo to specify files for git to ignore
    - useful for ignoring automatically generated files e.g .DS_Store, .idea etc
    - each file named must start on a new line
    - either .gitignore must be added to itself, or the repo

```
file-i-want-to-ignore.txt
image-to-ignore.png
messy-testing-area.html
hacky-or-stolen.css
```

- **branching**
  - branching ensures devs don't delete each others work when pushing to a remote repo



  - the default, and only branch available in a new repo, is the main / master branch
  - `git branch` - display all branches for current project
  - `git checkout -b [new-branch-name]` - create new branch and switch to it it
    - MUST use lowercase branch name, else you will be PUNISHED
  - `git checkout [branch-name]` - switch to an existing branch
  - best practice is to branch once for each feature of a project
    - main branch is reserved for 'perfect' code
    - feature branches are merged into main only after extensive testing
- **pull request**

    - a formal request to merge a less important branch into a more important one
    - this is usually created from the github website, rather than by command line
        - find the branch we want to merge into main
        - Open Pull Request → Enter Description → Submit Request
- **code review**
    - Pull Request → Files Changed
    - add comments to any bits that need improvement (diplomatically)
    - either Make Comments, Approve Request or Request Changes