



# Introduction to Programming Utilities

Lecture 3: Basics of Git and GitLab



# Course Syllabus

1. **Linux** and **The Command Line**
2. **Text editor** and **Compiler**
3. Basics of **Git** and **GitLab** (*This Lecture*)
4. Integrated Development Environment (**IDE**)
5. **Advanced Git** for Group Projects



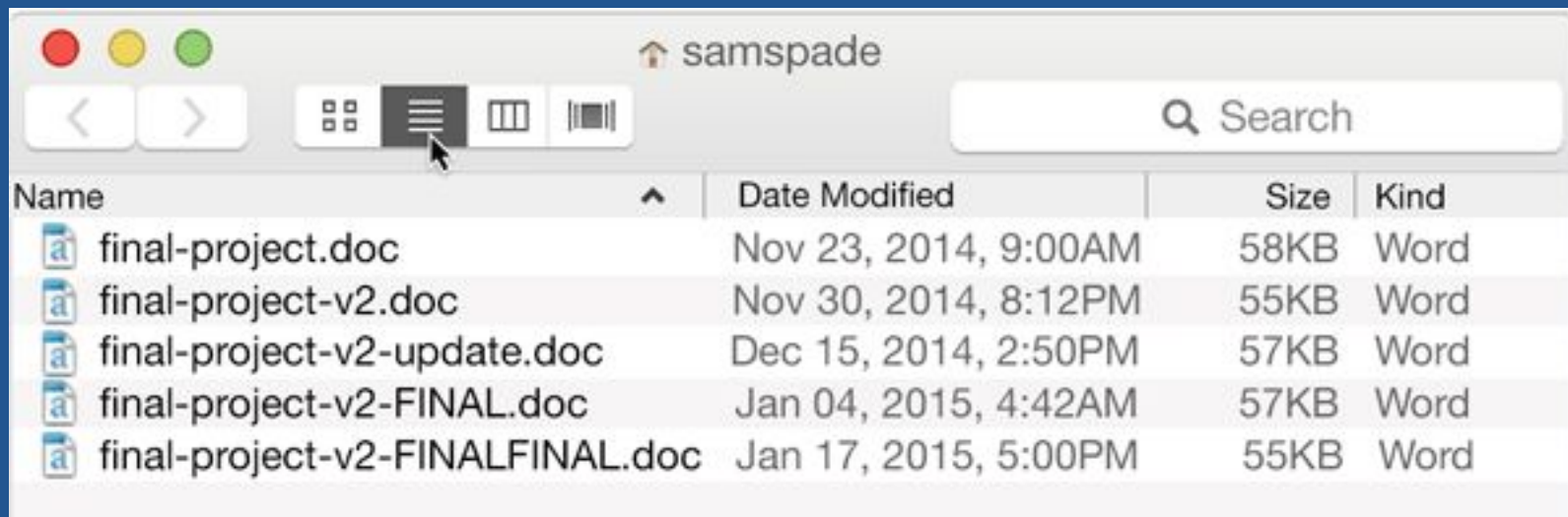
# This Lecture: Basics of **Git** and **GitLab**

- The problem with version control
  - Solution - a Version Control System
  - Is Centralized Version Control System (CVCS) the solution?
  - Git - a Distributed Version Control System (DVCS)
- Basic concepts in Git
  - Separation of local and remote repository
  - Commits
  - Synchronizing local and remote repositories
- Using GitLab

Warning: headache  
incoming



# The Problem

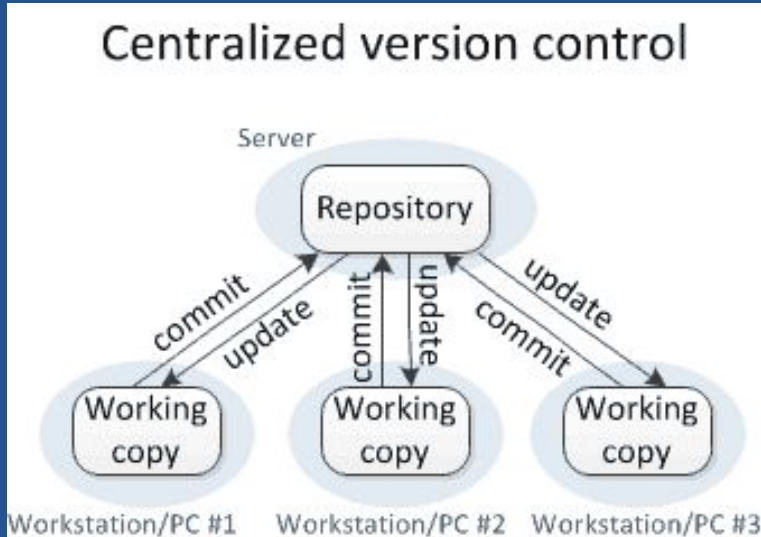




# Solution: a Version Control System

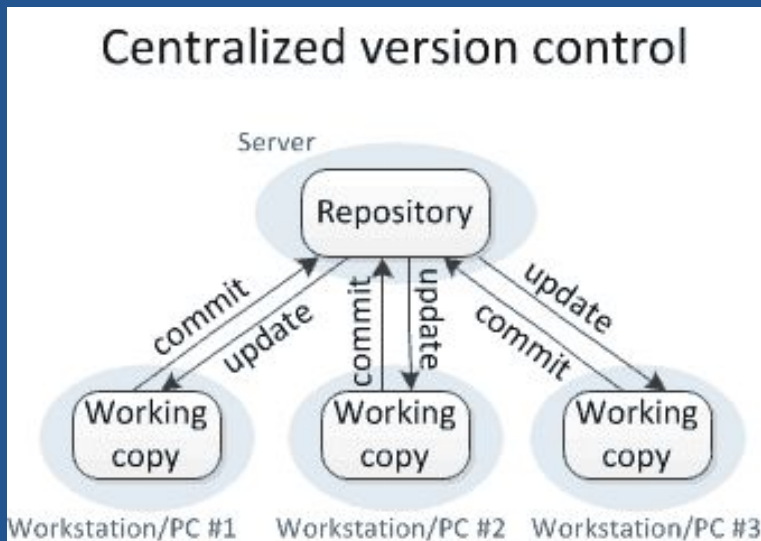
- “*System that records changes to a file or set of files over time so that you can recall specific versions later*”
- Will work with *any* files
  - We will focus on source code
- Record changes to a version database
- Keep track of multiple versions of a project (and its files)
- Revert selected files or entire project to any previous state
- Compare changes over time and who made those changes
- Collaboration by multiple people on the same project
  - Will not be covered in this session -> look forward to lecture 5!

# Centralized Version Control System



- Central server (repository) with all files
- Individual users “**checkout**” a file from server
- Once complete, “**commit**” the change to the central server
- Like a library
  - Borrow a book by signing it out
  - Work on the book and return it to the library
- Was popular in industry
  - Subversion, CVS

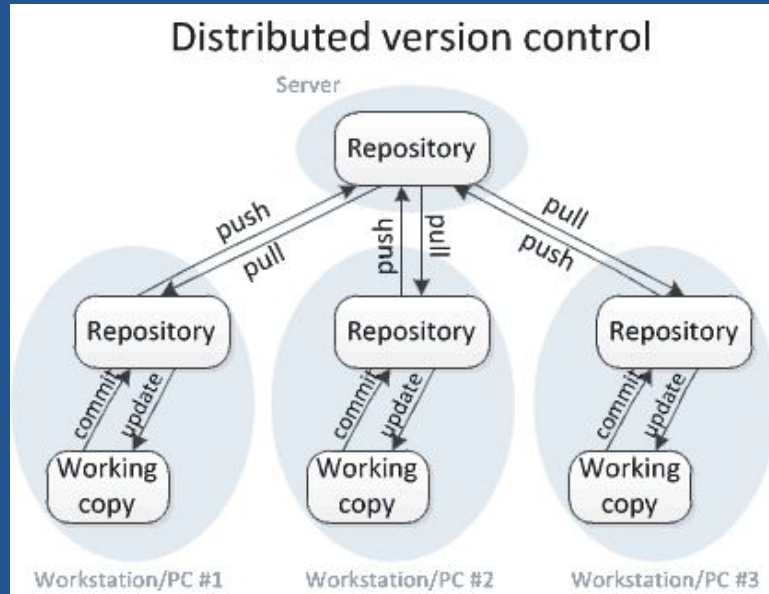
# Centralized Version Control System - Problem



- Only **one person** can checkout a file
  - Multiple people cannot work on the same file at a time
  - Multiple people working on the same feature?
  - File checked out for a long time?
- A **single point of failure** (central server)
  - Server goes down?
    - No changes/updates possible
  - Data corruption with no proper backup?
    - EVERYTHING is lost



# Distributed Version Control System



- Each individual user now also gets its own repository on their computer as well
  - Through “**cloning**” the remote repository
- Committing your change will only update your own local repository
- Synchronization with the remote repository needed
  - You “**push**” your changes (commits) to remote repo
  - You “**pull**” from the updated remote repository with changes made by other users

# How Git works

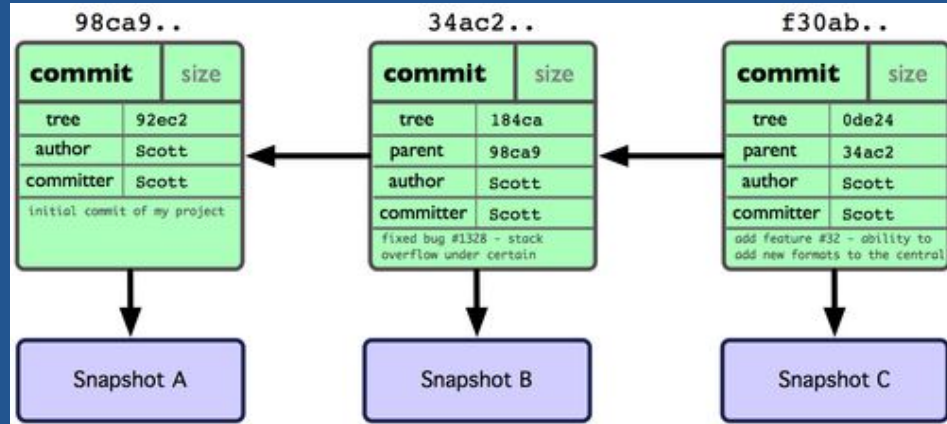




# Repository - local vs remote

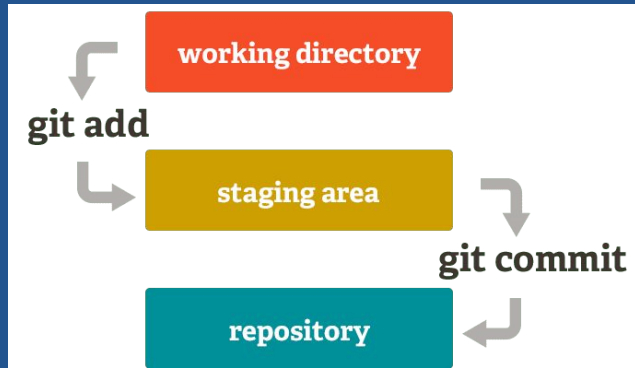
- File location where all files are stored
  - Think of it as a “server” where files are stored and changes are tracked
- **Local** repository
  - A folder on your computer (that you get by `init` or `clone`)
  - `.git/` directory inside the top folder contains the metadata for the project
    - DO NOT TOUCH *unless you know what you are doing*
- **Remote** repository
  - A server somewhere else that hosts a “shared” version of the project
    - For DoC: `gitlab.doc.ic.ac.uk`
  - (Eventually) where everyone in a project shares the code
  - (For now) just a server where you “store” a copy of your code

# Commit - a unit snapshot of a project's version



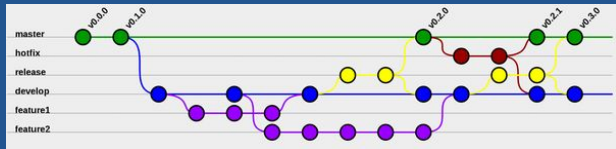
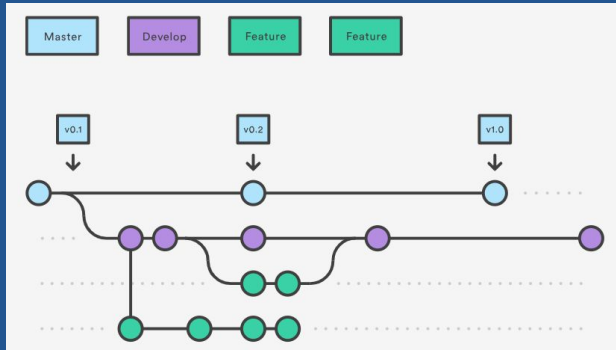
- **Commit is a unit of work in Git**
  - Every action you do is in terms of commits
  - Each commit represents a different 'state' of your project
    - Contains changes to certain modified file(s) since the previous commit
    - Commits don't contain the modified files - just records the changes
  - "Commit messages" - description of what changed in this commit

# How you “commit”



- Local repository can be split into three areas
- **Working directory**
  - Where all of your files are (including all files you modified but not committed yet)
- **Staging area**
  - Where files you want to include in the next commit goes
  - You may not want to add all modified files in a commit
- **Repository**
  - Where commits are stored
  - It may contain commits that are ONLY on this local repository

# How commits are stored and organized



- As a **tree** (git tree)
  - Supports multiple branches
  - Joining and diverging branches
  - The central **master branch**
    - Only commits that are deployable
    - For now: the only branch you'll work on
- For now: ignore branches
  - Treat it like the diagram (right)
  - Just a sequence of commits as a **list** in a chronological order

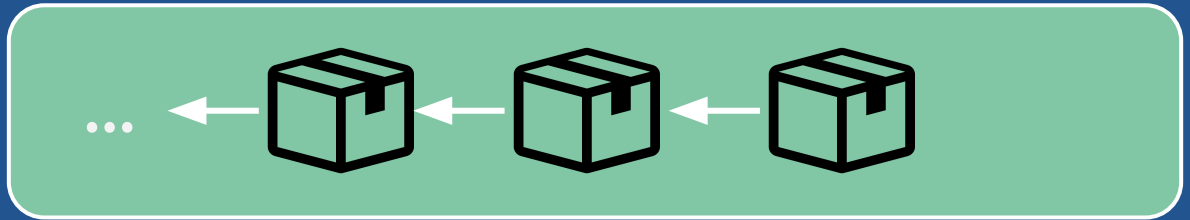




Working Directory



Staging Area



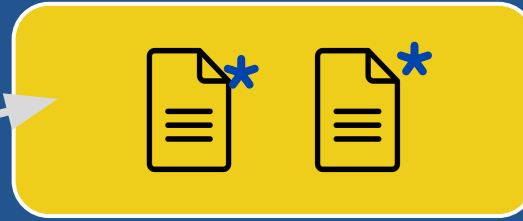
Repository

Local Repository - How it works (1)

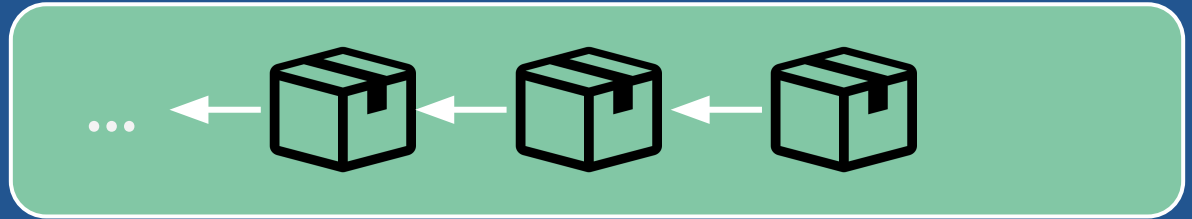


Working Directory

`git add`



Staging Area



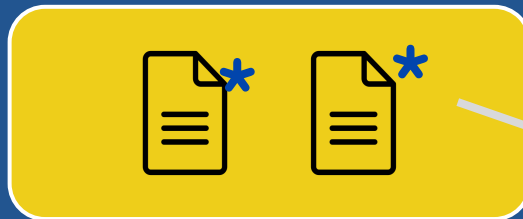
Repository

Local Repository - How it works (2)





Working Directory

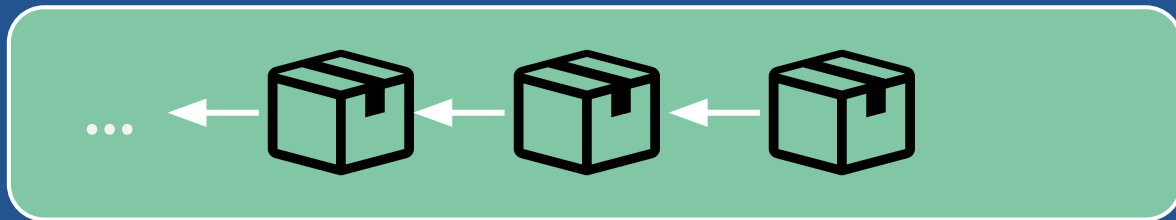


Staging Area

`git  
commit`



commit



Repository

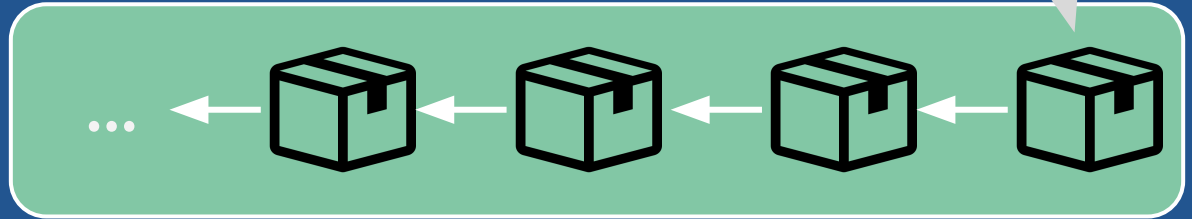
Local Repository - How it works (3)



Working Directory

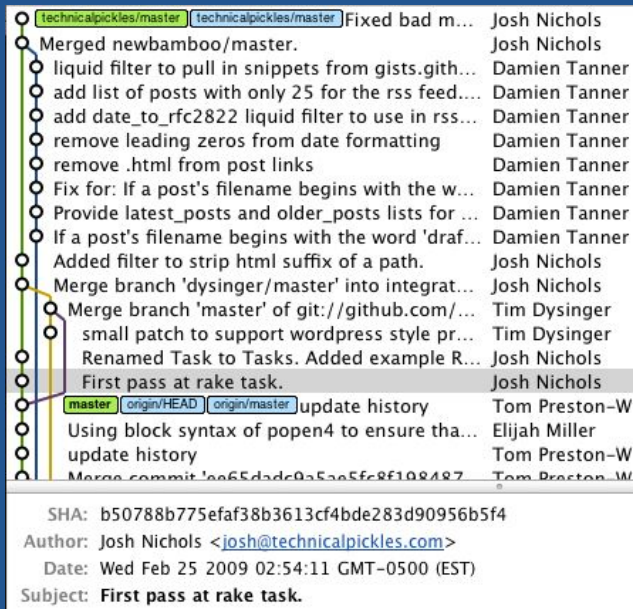


Staging Area



Repository

# Making “good” commits



The screenshot shows a Git commit history on the left and the details of a specific commit on the right. The commit history lists various commits with their messages and authors. The selected commit is 'First pass at rake task.' by Josh Nichols. The commit details on the right show the SHA, author, date, and subject of the commit.

```
technicalpickles/master technicalpickles/master Fixed bad m... Josh Nichols
Merged newbamboo/master. Josh Nichols
liquid filter to pull in snippets from gists.gith... Damien Tanner
add list of posts with only 25 for the rss feed.... Damien Tanner
add date_to_rfc2822 liquid filter to use in rss... Damien Tanner
remove leading zeros from date formatting Damien Tanner
remove .html from post links Damien Tanner
Fix for: If a post's filename begins with the w... Damien Tanner
Provide latest_posts and older_posts lists for ... Damien Tanner
If a post's filename begins with the word 'draf... Damien Tanner
Added filter to strip html suffix of a path. Josh Nichols
Merge branch 'dysinger/master' into integrat... Josh Nichols
Merge branch 'master' of git://github.com/... Tim Dysinger
small patch to support wordpress style pr... Tim Dysinger
Renamed Task to Tasks. Added example R... Josh Nichols
First pass at rake task. Josh Nichols
master origin/HEAD update history Tom Preston-W
Using block syntax of popen4 to ensure tha... Elijah Miller
update history Tom Preston-W
Merge commit '0a65d3dc9a5a05fc8f108487' Tom Preston-W

SHA: b50788b775efaf38b3613cf4bde283d90956b5f4
Author: Josh Nichols <josh@technicalpickles.com>
Date: Wed Feb 25 2009 02:54:11 GMT-0500 (EST)
Subject: First pass at rake task.
```

- Making **good commits** are **very important**
  - **Small** but **meaningful**
  - Should implement a **single feature**
  - Good commit messages
    - **Descriptive**
    - **Concise**
- A rule of thumb: commit at LEAST once a day


# What “bad” commits look like

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAANDS	2 HOURS AGO


AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.


master ▾ javathreads\_hk619


06 Feb, 2020 1 commit


 **Done concurrency actually works thank god**  
hk619 authored 8 months ago


03 Feb, 2020 5 commits

 **Core done?**  
hk619 authored 8 months ago


 **Core done**  
hk619 authored 8 months ago

 **Revert "Core done"** ...  
hk619 authored 8 months ago

 **Core done**  
hk619 authored 8 months ago

 **Core done**  
hk619 authored 8 months ago

01 Feb, 2020 1 commit

 **Original skeleton**  
Konstantinos Gkoutzis authored 8 months ago

# What “good” commits look like



**[hk619][jqp18][ry819][ag2719] First draft of Final Report Finished**  
Koo, Hyunhoi authored 3 months ago

18 Jun, 2020 2 commits



**Merge branch 'backend\_debugging' into 'master'** ...  
Poon, Jia authored 3 months ago



**[hk619][ry819][ag2719][jqp18] Final refactoring of assembler, emulator and extension**  
Poon, Jia authored 3 months ago

17 Jun, 2020 10 commits



**Merge branch 'backend' into 'master'** ...  
Koo, Hyunhoi authored 3 months ago



**[jqp18] Remove main function**  
Jia Qi Poon authored 3 months ago



**[ry819] Add optimisation to backend math**  
ry819 authored 3 months ago



**[jqp18] Implement more functions for javascript to call** ...  
Poon, Jia authored 3 months ago



**[ry819] Refactor backend math**  
Yang, Richard authored 3 months ago



**Merge branch 'frontend' into 'master'** ...  
Koo, Hyunhoi authored 3 months ago



**[hk619] Implemented skeleton for frontend website**  
Koo, Hyunhoi authored 3 months ago



**[hk619] added -fsanitize=undefined flag to makefiles for assembler and emulator**  
hk619 authored 3 months ago



**[ag2719] Added test cases for assembled\_to\_bin**  
Goncharov, Alexander authored 3 months ago



**[jqp18] Refactor second\_pass and second\_pass\_branch**  
Jia Qi Poon authored 3 months ago



**[ry819] Second\_pass tests done**  
ry819 authored 3 months ago



**[ry819] Code refactoring**  
ry819 authored 3 months ago



11 Jun, 2020 20 commits



**[hk619] made Makefile for libassembler**  
hk619 authored 3 months ago



**Merge branch 'Branch' into 'master'** ...  
Koo, Hyunhoi authored 3 months ago



**[ry819] added branch for second pass**  
Yang, Richard authored 3 months ago



**Merge branch 'second\_pass\_multiply' into 'master'** ...  
Koo, Hyunhoi authored 3 months ago



**[ry819] Added test cases for second pass multiply**  
Yang, Richard authored 3 months ago

'function pointers



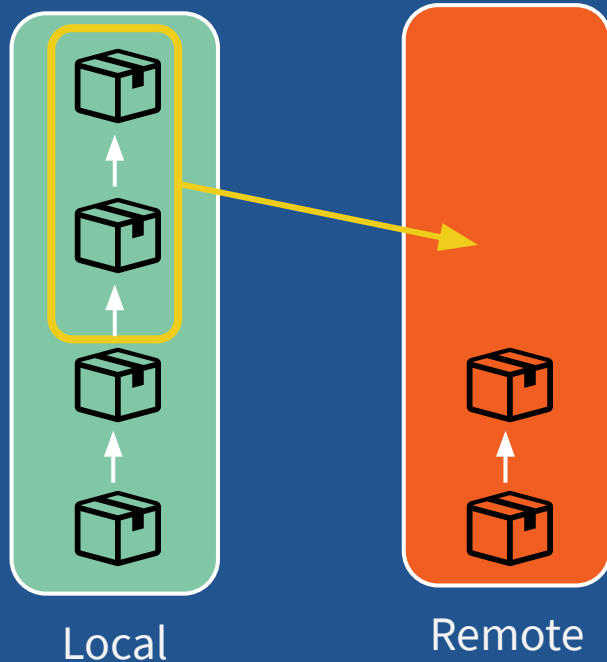
# The remote repository?

- Now you know how the local repository works
- But remote repository?
  - Server that hosts the “backup copy” of the commits in your local repository
- When you commit, you only add the commit to the local repository

**Syncing Local and Remote repository is necessary**



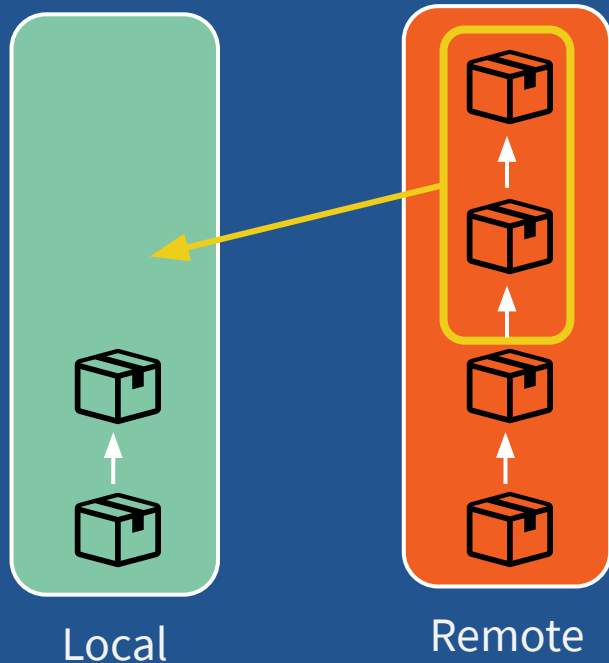
# git push : local -> remote



- Commits in the local repository but NOT in the remote repository
- ‘Push’ the commits in the local repository to the remote repository
- Use
  - Backup the commits to the remote server
  - For DoC: to submit the project’s commit on LabTS



```
git pull :local <- remote
```



- Commits in the remote repository but NOT in the local repository.
- ‘Pull’ the commits from the remote repository to the local repository
- Use
  - When working on multiple computers and you ‘pushed’ commits made on one computer, and then you want to work on another computer, you would ‘pull’ the commits





# Git Commands

- **git init** : initialize a directory to be a git repository
  - Almost never used for DoC lab work
- **git clone** *<repository>* : copy the remote repository over to the local directory to create a local repository
  - *repository* : SSH or HTTPS URL of the remote repository
  - e.g. `git@gitlab.doc.ic.ac.uk:<user>/<repository>.git`
- **git add** *<file(s)>* : add file(s) to the staging area
  - Wildcards can be used for *file(s)* (e.g. `git add *`)
- **git commit** : package all files in the staging area as a commit and put it onto the local repository
  - `-m <message>` : set the commit message as *message*. Without it a nano window will pop up asking for the message
- **git push** : get all commits in the branch and send it to the remote
  - `-f` : force push. Do not use unless you know what you are doing



# A typical workflow

1. **git clone** the lab exercise shared on GitLab
  - (ssh) `git clone git@gitlab.doc.ic.ac.uk:hk619/intro-to-programming-utilities-lecture-3-demo.git`
  - (https) `git clone https://gitlab.doc.ic.ac.uk/hk619/intro-to-programming-utilities-lecture-3-demo.git`
2. Work on the project
3. Once a feature is implemented, **git add** all relevant files
4. **git commit** to create a new commit
5. **git push** to save the commit on the repository
6. Repeat 2-5 until the entire lab exercise is complete
7. Submit the commit of finished version to LabTS

Demo





# Department of Computing Society

Imperial College London

[docsoc.co.uk](http://docsoc.co.uk) | [fb.com/icdocsoc](https://fb.com/icdocsoc) | [twitter.com/icdocsoc](https://twitter.com/icdocsoc)