## INTRODUCING…

# VERSION CONTROL SYSTEMS

DR IAN CORNELIUS

## HELLO

- Learning Outcomes:
    1. Understanding what is Version Control System are
    2. Understand the version control system of choice for this module, and the relevant commands
    3. The ability to use a version control system for any projects/tasks

## INTRODUCTION TO VERSION CONTROL

- A process to manage files and directories over a period of time
- Provides an ability to recall a previous version of a working application

## VERSION CONTROL SYSTEMS

### 1. LOCAL

- Maintains the tracking of files on a local system
- Prone to errors, meaning the chances of writing accidentally to the wrong file is higher

### 2. CENTRALISED

- Maintains the tracking of files with a centralised server
- The server contains information of all versioned files
- **Example**: SVN

### 3. DISTRIBUTED

- Users can clone a repository, including its full history of tracking
- If the server goes down, or is no longer available, users can copy their repositories to the server to restore it
- Each cloned repository is a full backup of all the data
- **Examples**: Git and Mercurial

## WHAT IS GIT?

- Created by Linus Torvalds in April 2005
- A command-line based version control program
- Cross-platform, works on Windows, Linux and macOS
- Open-source and free to use
- Track **changes** in code, and not versions

## WHY USE GIT?

- Primarily for people who work with source-code
  - i.e. programmers/software engineers
- Useful for tracking changes to source-code
  - ability to review historic changes
  - can share and merge changes with other people

## TERMINOLOGIES OF GIT (1)

- `repo`
  - means a repository that organises a project
  - can contain folders, files, images, videos etc.
- `branch`
  - a version of the repository that is separate to the main working version
- `checkout`
  - used for switching between different versions of the project
  - i.e. switching the branches
- `clone`
  - used to make a copy of the repository
- `commit`
  - the process of saving your changes
- `fork`
  - a copy of a repository, enables you to test and debug without affecting the original project

## TERMINOLOGIES OF GIT (2)

- `master`
  - the default branch of a repository
  - often considered to be the main working branch
- `merge`
  - the process of putting a forked history back together
  - takes the data in one branch and merges it with another branch
- `origin`
  - a reference to the remote repository from where the project has been cloned
- `pull`
  - refers to data pulled from a Git service
  - fetches and merges changes from the remote server to the local directory
- `push`
  - refers to the data uploaded to the remote repository from the local version

## TERMINOLOGIES OF GIT (3)

- `remote`
  - refers to the remote repository
  - i.e. the version stored on an online Git service
- `revert`
  - used to revert a commit -`reset`
  - used for undoing changes
- `ignore`
  - used to specifically denote files or folders to ignore and not track changes

## COVENTRY GITHUB (1)

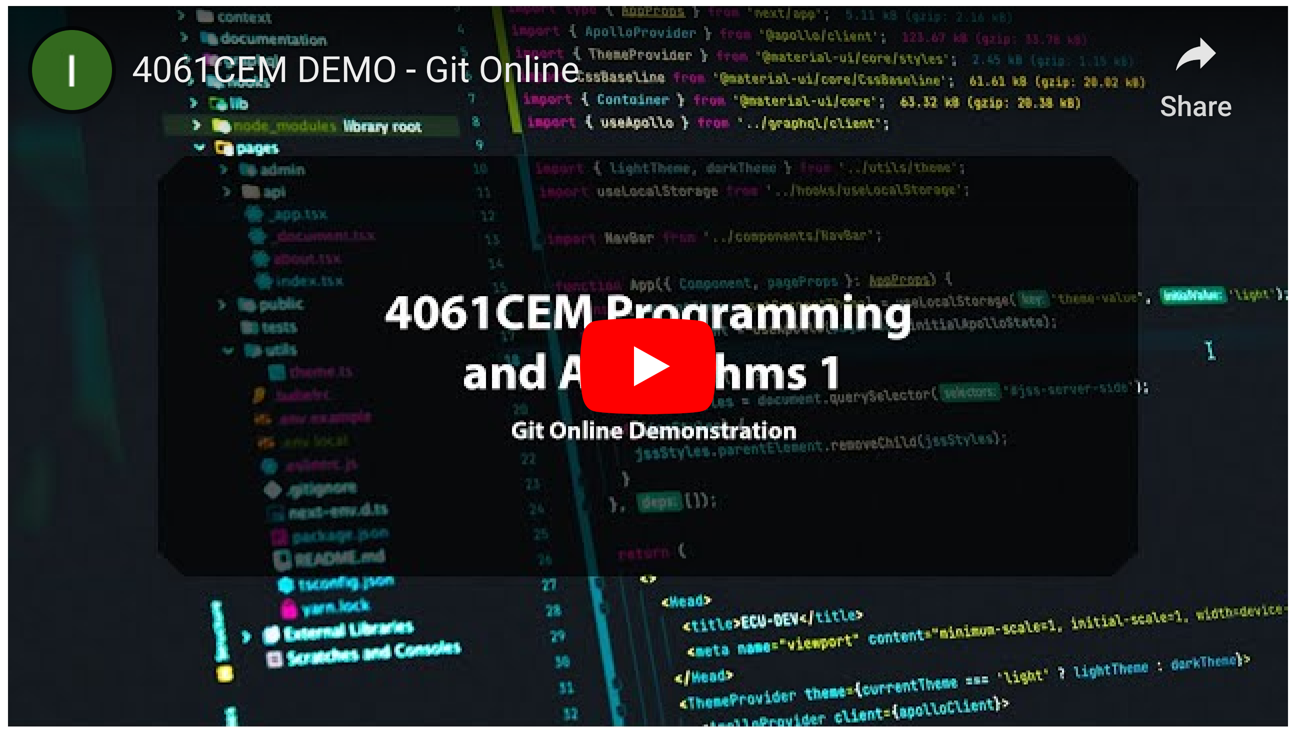
- Coventry University’s very own Git service
  - Accessible via: https://github.coventry.ac.uk
- Login using your university credentials, without the suffix
  - i.e. `username@uni.coventry.ac.uk -> username`

### NO ACCOUNT?

- Contact IT Services:
  - Telephone: 02477 657 777
  - Or visit them at the library on the ground-floor

## COVENTRY UNIVERSITY GITHUB (2)

- Demonstration of Coventry University GitHub Site
  - Refer to the pre-recorded video for a demonstration


4061CEM DEMO - Git Online
Share
4061CEM Programming and Algorithms 1
Git Online Demonstration

## GIT VIA THE COMMAND-LINE

- Demonstration of Git via the Command-Line
  - Refer to the pre-recorded video for a demonstration

4061CEM DEMO - Git via Terminal
Share
4061CEM Programming and Algorithms 1
Git via Terminal
Watch on YouTube

## GIT VIA THE IDE

- Demonstration of Git via the IDE
    - Refer to the pre-recorded video for a demonstration

4061CEM DEMO - Git via the IDE
Share
4061CEM Programming and Algorithms 1
Git via the IDE
Watch on YouTube

## GOODBYE

- Questions?
  - Post them in the **Community Page** on Aula
- Contact Details:
  - Dr Ian Cornelius, ab6459@coventry.ac.uk