

[aps]

{abertay /\*PROGRAMMING\*/ society;}

# GitHub Source Control Workshop

**Abertay Programming Society**

**01-10-2019**

Have you got a GitHub  
account yet?!

# Announcements

- Please give us your membership payments – members will be rewarded ;)
- All the examples / code from last week is now on our GitHub page

**<https://github.com/AbertayProgrammers>**

IEEExtreme



# What is 'Version Control', and why should you care?

- A system that means you can keep track of changes to a file/files over time and revert to specific versions later on
- Especially useful for group projects (*3rd year*) as it allows you to 'merge' your version of something with another version of it
- **You can:**
  - **Revert selected files / entire project back to a previous state**
  - **Compare changes over time**
  - **See modification / problem history**
  - **And more!**

# Installing Git

Already installed on lab computers.

## **Windows:**

<http://git-scm.com/download/win>

## **Linux:**

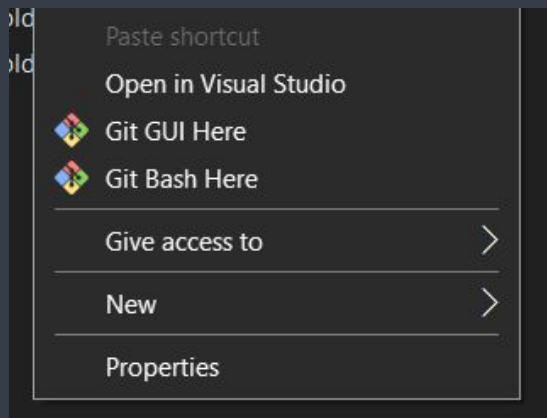
You will **probably** already have git installed!

# Bash

Bash stands for “Bourne-Again Shell”

Basically it’s a command line interface we’ll use for Git.

Right click in a directory and click “Git Bash Here”



# Useful bash commands

cd - change directory

touch - create a new file

mkdir - make directory

ls - display a list of the contents of the current directory



# Create a project!

Just make a folder and create a .txt text document

# Create a Github repository

Create a Github account

Click “New repository”

# Upload project to GitHub

- Open your project directory in Bash
- `git init`
- `git add .`
- `git commit -m "<descriptive message>"`
- `git remote add origin <your github repo>.git`
- `git push -u origin master`

# Also...

- Need to set identity (i.e. login)
- Do both of these commands

```
C:\Users\Jessica\md5crack>git commit -m "first commit"

*** Please tell me who you are.

Run

    git config --global user.email "you@example.com"
    git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'Jessica@Jessica-PC.(none)')

C:\Users\Jessica\md5crack>_
```

# Git pull

- Git pull incorporates changes from the remote repository to local repository
- Git pull “<repository>”

# Branches

Good for adding new features separately before combining them with your project

Good for when you have multiple people working on a project

## Creating a new branch

- `git checkout -b branch_name`

## Switching to another branch

- `git checkout branch_name`

# Git merge

- Git merge incorporates changes from selected branches to master
- Git merge "<branch>"

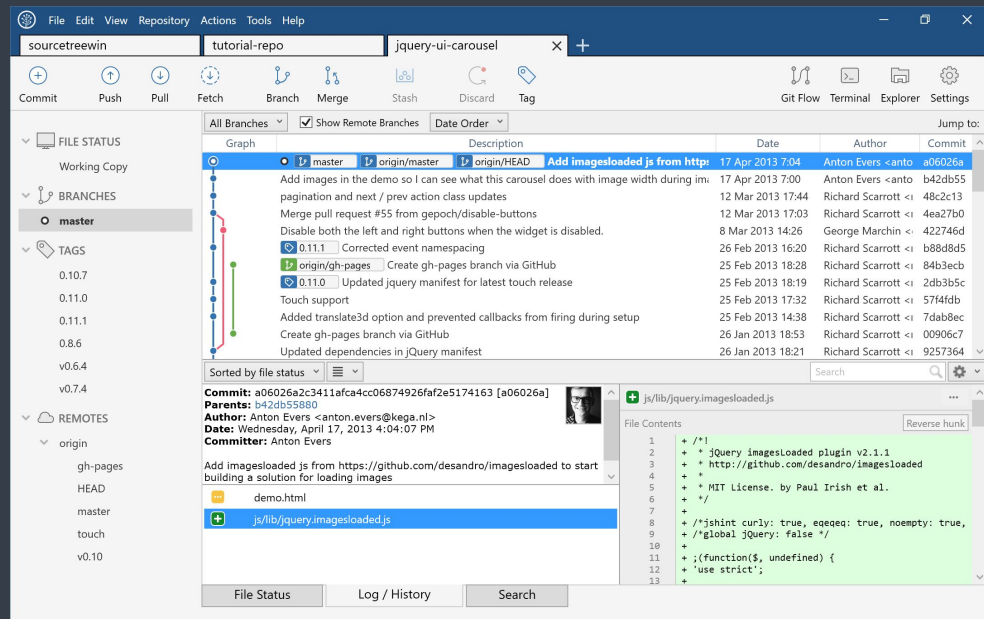
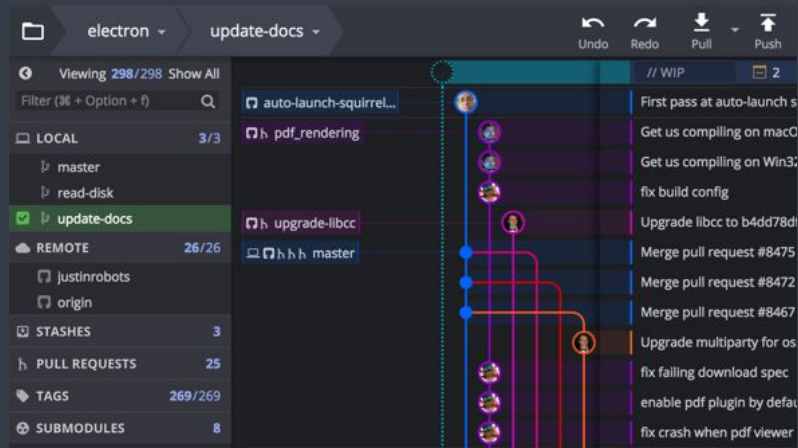


# Diff

- `git diff`
- `git diff master:./file branch-name:./file`

# Git GUIs

Eg GitKraken, Sourcetree



# IDE plugins

- Plugins are built-in for majority of IDEs
  - Visual Studio
  - JetBrains
  - Sublime
  - Atom
  - Etc etc
- Some others have plugins to download
  - (*e.g. EGit*)



SOURCE CONTROL: GIT

Message (press Ctrl+Enter to commit)

CHANGES

hepfac2

main.cpp



main.cpp

main.cpp (Working Tree) x

```
91- int option_index = 0;
92- bool exit_condition = true;
93
94- try {
95-     while ((opt = getopt_long(argc, argv, "hp:t:r:", long_options, &option_index)) != -1) {
96-         // if input is unexpected or doesn't meet requirements
97-         exit_condition = false;
98-
99-         if (optarg == "?") {
100-             return print_usage();
101-         }
102-
103-         // parse options
104-         switch(opt){
105-
106-             case 'p':
107-                 filename_patterns = optarg;
108-                 break;
109-             case 't':
110-                 filename_target = optarg;
111-                 break;
112-             case 'r':
113-                 reduction_levels = (optarg) ? atoi(optarg) : COUNT_PATTERN_CHANGES;
114-                 break;
115-             default:
116-                 break;
117-         }
118-     } catch (std::exception Ex){
119-
120-         std::cout << Ex.what() << std::endl;
121-         return print_usage();
122-     }
123-     if (exit_condition){
124-         return print_usage();
125-     }
126-
127-     // data entry validation
128-     try{
129-         file_patterns = get_file_content(filename_patterns, COUNT_PATTERNS);
130-         file_target = get_file_content(filename_target, COUNT_SOURCE_LINES);
131-     }
132-     catch (myException Ex){
133-         std::cout << "While attempting to read file '" << Ex.what() << "' an error was
```

```
72-     try {
73-         int option_index = 0;
74-         while ((opt = getopt_long(argc, argv, "hp:t:r:", options, &option_index)) != -1) {
75-             // if input is unexpected or doesn't meet requirements
76-             if (optarg == "?" || argc < 2){
77-                 usage();
78-                 return 1;
79-             }
80-
81-             // parse options
82-             switch(opt){
83-
84-                 case 'h':
85-                     usage();
86-                     return 0;
87-                 case 'p':
88-                     filename_patterns = optarg;
89-                     break;
90-                 case 't':
91-                     filename_target = optarg;
92-                     break;
93-                 case 'r':
94-                     reduction_levels = atoi(optarg);
95-                     break;
96-             }
97-         } catch (std::exception Ex){
98-
99-             std::cout << "Something went wrong :(" << std::endl;
100-             std::cout << Ex.what() << std::endl;
101-             usage();
102-             return 1;
103-         }
104-
105-         std::string *file_patterns = get_file_content(filename_patterns, COUNT_PATTERNS);
106-         std::string *file_target = get_file_content(filename_target, COUNT_SOURCE_LINES);
```

# Hacktoberfest



# Hacktoberfest

Registration is open!

<https://hacktoberfest.digitalocean.com>

You are now ready to take part - have a look:

<https://github.com/search?q=label%3Ahacktoberfest+state%3Aopen&type=Issues>

More on this next week!

But if you want to start then get going.

First 50,000 participants get a T-Shirt!

# Thanks for listening!

```
if (Questions)
{
    return answer;
}
else
{
    print( 'See you next week!' );
}
```