

# The Unix Shell

Managing your code: quietly introducing *Git* - a friend for life - Part 2

Thanks to all contributors:

Alison Pamment, Sam Pepler, Ag Stephens, Stephen Pascoe, Kevin Marsh, Anabelle Guillory, Graham Parton, Esther Conway, Eduardo Damasio Da Costa, Wendy Garland, Alan Iwi, Matt Pritchard and Tommy Godfrey.

# Managing code in the olden days

- Create "*working\_dir*"...add some code
- Write some outputs...change the code
- Publish a paper...change the code
- Copy "*working\_dir*" to "*working\_dir2*"
- Change the code
- Copy a version to a CD

*...now which version is current? Is it "*working\_dir*" or "*working\_dir2*"? And which one relates to that paper?*

# But those days are gone!

- Scientists are typically **required to publish data and code** (by their funders/institutions).
- Collaboration between scientists requires data-sharing; this implicitly relies on **code-sharing**.
- There are **tools that make it easy** to record our changes, document our workflow and "fix" releases of our code at important steps along the way.



# Introducing Git

There are many different Version Control tools:

- **SVN** (Subversion) is very popular and (relatively) easy to grasp; eclipsed by...
- **Git**, which is also:
  - More useful for collaboration
  - Distributed and *fast*
  - Very well supported in terms of tooling
  - Has free repository hosts on the web (GitHub, BitBucket etc.,).



# More about Git

Git is a *distributed* Version Control System (VCS):

- you store a complete copy of a repository within your working copy.
- this means you can work offline:
  - there is no default 'central' server - if you want one, you (and your team) just nominate where it is - typically GitHub!

# What is a Git repository?

- A directory tree containing files and subdirectories.
- Old and different versions of those files and subdirectories.
- A set of information to enable you to navigate across versions.

# Not Introducing GitHub

<https://github.com>

A service for hosting git repositories.



The screenshot shows the GitHub homepage. At the top, there is a navigation bar with links for Personal, Open source, Business, and Explore. To the right of these links are links for Pricing, Blog, and Support. Further right is a search bar labeled 'Search GitHub' and two buttons: 'Sign in' and 'Sign up'. The main content area features a large heading 'How people build software' on the left, with a subtext 'Millions of developers use GitHub to build personal projects, support their businesses, and work together on open source technologies.' Below this text is a small illustration of the GitHub mascot, Octocat. On the right side of the main content area, there is a sign-up form with three input fields: 'Pick a username', 'Your email address', and 'Create a password'. Below the password field is a note: 'Use at least one letter, one numeral, and seven characters.' A large green button labeled 'Sign up for GitHub' is positioned below the form. At the bottom of the sign-up section, there is a small disclaimer: 'By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy. We'll occasionally send you account related emails.'



# GitHub: repositories (public or private)

The screenshot displays the GitHub interface for the repository **cedadev / crepp**, which is marked as **Private**. The repository has 12 watchers, 0 stars, and 2 forks. The main navigation bar includes links for **Code**, **Issues** (27), **Pull requests** (1), **Projects** (0), **Wiki**, **Pulse**, **Graphs**, and **Settings**.

The repository description is **CEDA REceive-to-Publish Pipeline (CREPP)**. It shows **81 commits**, **6 branches**, **0 releases**, and **2 contributors**. The current branch is **master**, and there is a **New pull request** button. Action buttons include **Create new file**, **Upload files**, **Find file**, and **Clone or download**.

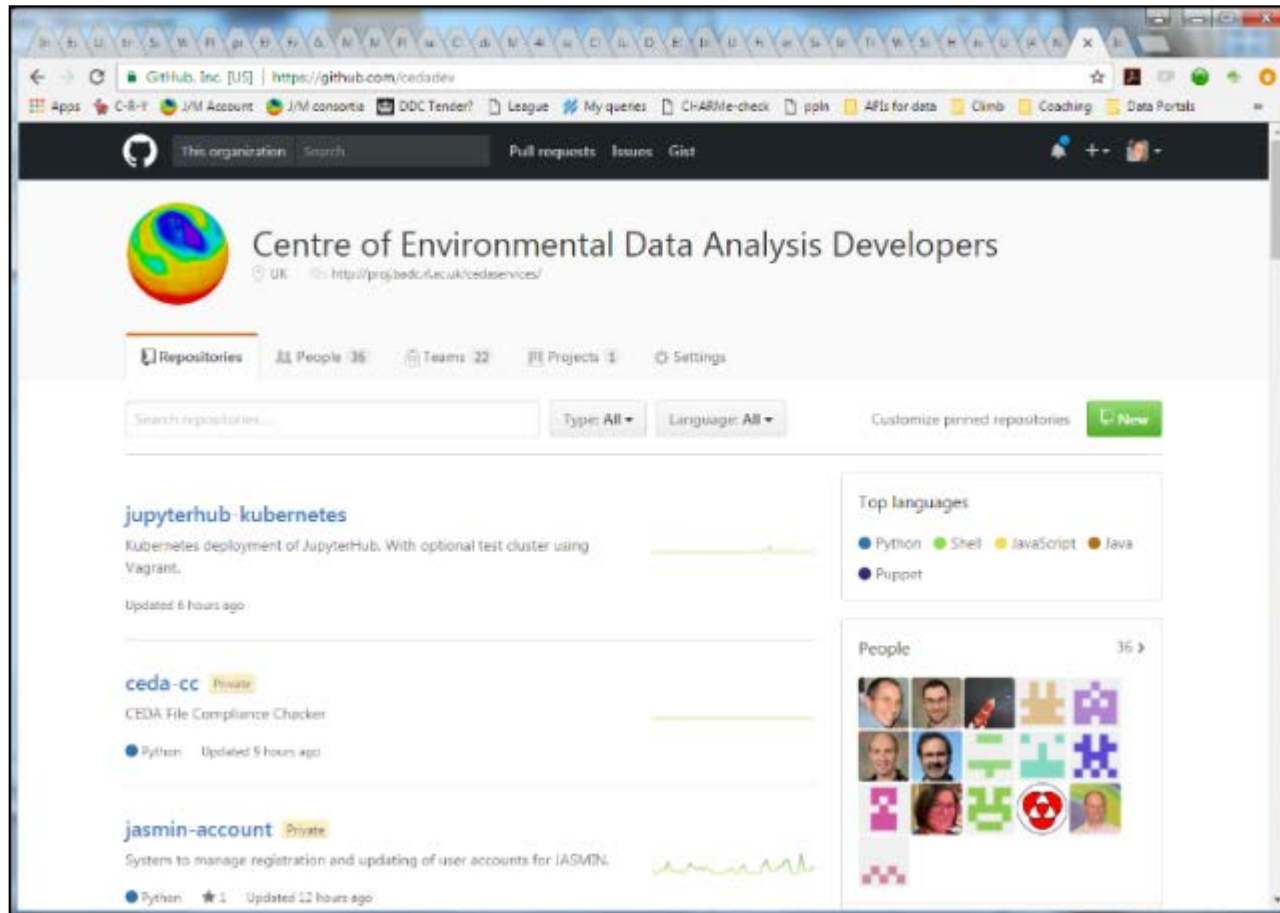
The file list shows the following structure and commit history:

| File/Folder        | Commit Message   | Time Ago     |
|--------------------|--|--------------|
| apps               | rename crepe to crepp throughout                                   | 6 months ago |
| cmd_line_interface | General development.   | 2 months ago |
| crepp_app          | bugfixes re daemonise code   | a month ago  |
| crepp_site         | Moved 'static' directory into 'crepp_app' so it is easy to deploy. | 6 months ago |
| creplib            | Merge branch 'master' of https://github.com/cedadev/crepp          | a month ago  |
| scripts            | bugfixes re daemonise code   | a month ago  |
| templates          | rename crepe to crepp throughout                                   | 6 months ago |
| test               | Fixed test_workflow.py - added test_11 for parallel test.          | a month ago  |
| .gitignore         | rename crepe to crepp throughout                                   | 6 months ago |

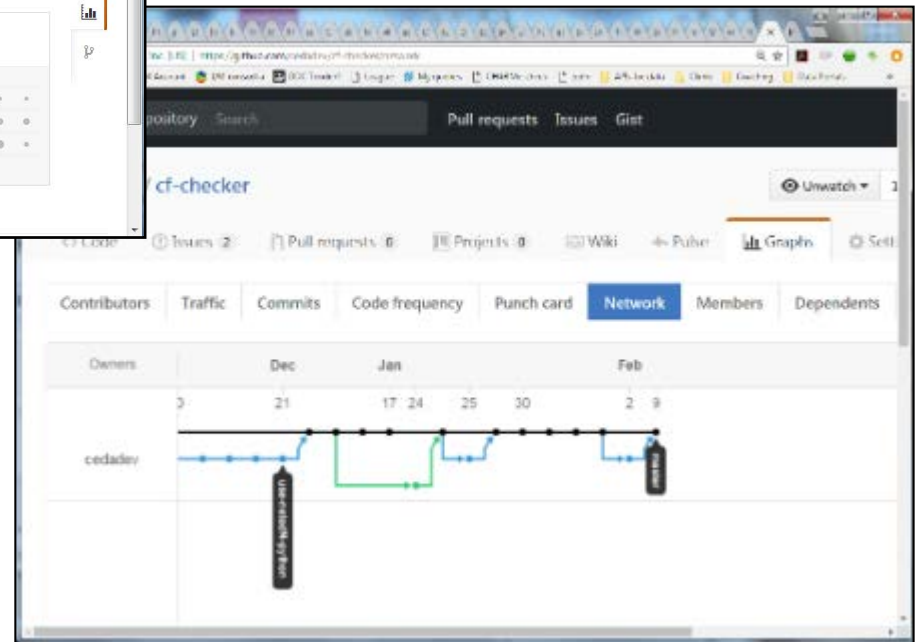
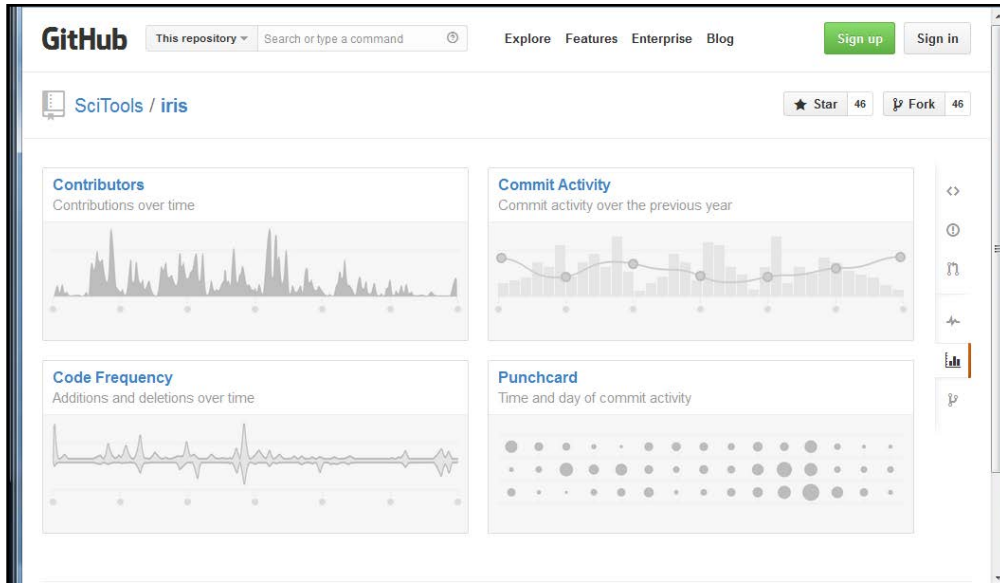




# GitHub: organisations



# GitHub: collaboration (branch/fork)



# GitHub: Issue tracking

The screenshot displays the GitHub interface for the repository `cedadev / ceda_moles_django`. The repository is private and has 11 watchers, 0 stars, and 0 forks. The 'Issues' tab is selected, showing 104 open issues. A search filter is applied: `is:issue is:open`. The issues are listed with their titles, IDs, and creation dates. The first issue is '#147 Can we embed schema.org tags into MOLES returned content to improve indexing by search engines', opened 5 days ago by philipkershaw. The second is '#145 Fix database connection problem in parallel connections', opened 27 days ago by agstephens. The third is '#143 Update MOLES PyDAP link from dap.ceda.ac.uk to data.ceda.ac.uk', opened on 14 Nov 2016 by gap736uk, with labels 'Quick item', 'urgent', and 'User View'. The fourth is '#142 Add cci-tagger to dependencies in MOLES deploy on ingest1', opened on 8 Nov 2016 by agstephens. The fifth is '#141 Create tests: Validation of DRSDataset properties', opened on 7 Nov 2016 by agstephens. The sixth is '#140 Export: templating issues to resolve', opened on 3 Nov 2016 by gap736uk, with labels 'Export' and 'high priority', and a progress bar showing 0 of 4 items. The seventh is '#139 Adapt Obs Col and Obs templates to display selected GEMET theme(s) for the record', with label 'high priority' and a 'User View' button.

GitHub repository: `cedadev / ceda_moles_django` (Private)

Unwatch 11 | Star 0 | Fork 0

Code | Issues 104 | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs | Settings

Filters  | Labels | Milestones | [New issue](#)

104 Open | 42 Closed | Author | Labels | Milestones | Assignee | Sort

- [Can we embed schema.org tags into MOLES returned content to improve indexing by search engines](#) #147 opened 5 days ago by philipkershaw
- [Fix database connection problem in parallel connections](#) #145 opened 27 days ago by agstephens
- [Update MOLES PyDAP link from dap.ceda.ac.uk to data.ceda.ac.uk](#) #143 opened on 14 Nov 2016 by gap736uk [Quick item](#) [urgent](#) [User View](#)
- [Add cci-tagger to dependencies in MOLES deploy on ingest1](#) #142 opened on 8 Nov 2016 by agstephens
- [Create tests: Validation of DRSDataset properties](#) #141 opened on 7 Nov 2016 by agstephens
- [Export: templating issues to resolve](#) #140 opened on 3 Nov 2016 by gap736uk [Export](#) [high priority](#) 0 of 4 DCS checks: reco...
- [Adapt Obs Col and Obs templates to display selected GEMET theme\(s\) for the record](#) [high priority](#) [User View](#)



# GitHub: history and change

```
6 ■■■ cedamoles_app/admin_tools/integrity/routine_checks.py

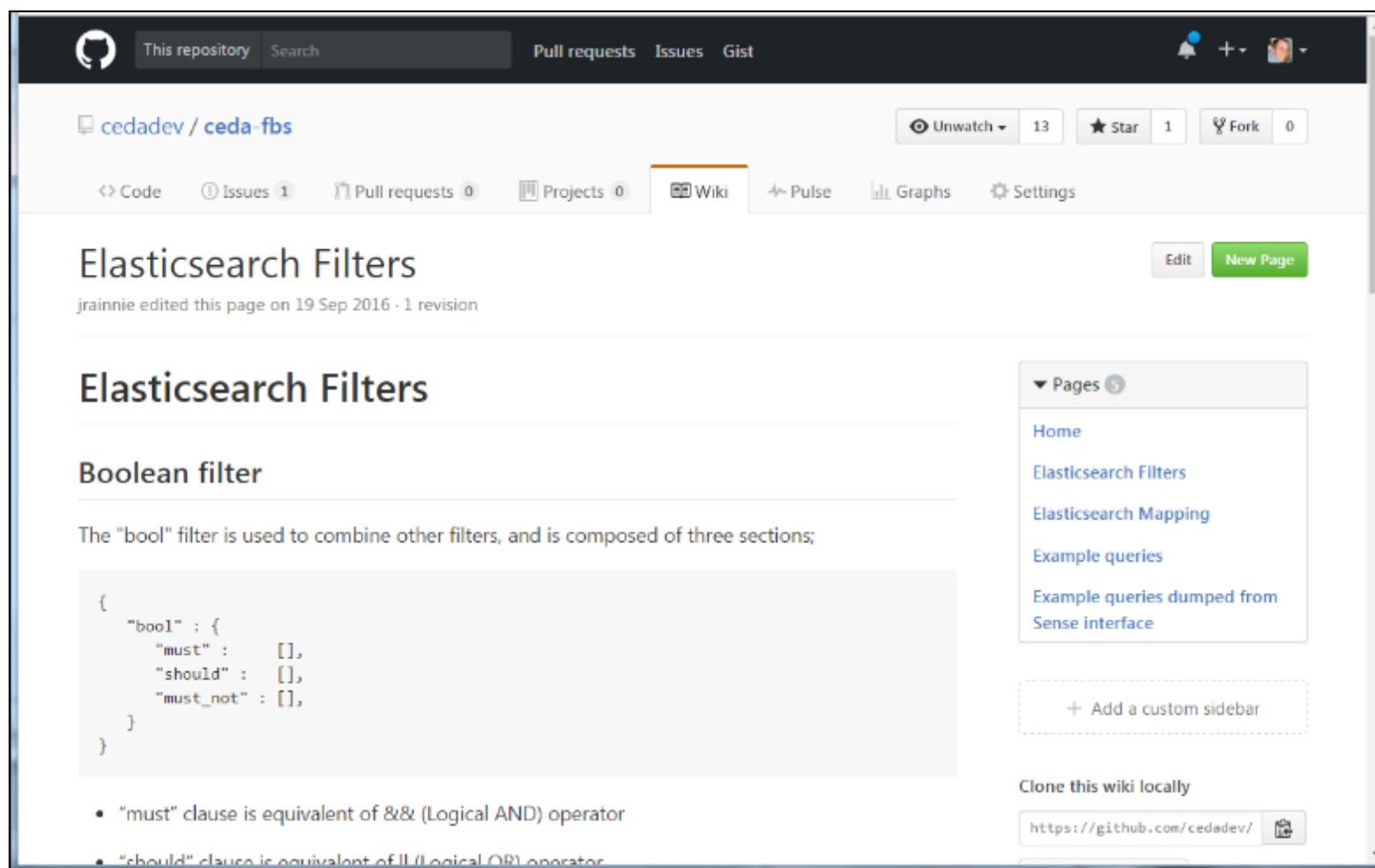
@@ -41,10 +41,10 @@ def run_checks(self):

41 41
42 42     class ResultChecks(ChecksBase):
43 43
44 -     def check_internalPath(self):
44 +     def check_dataPath(self):
45 45         found = Counter()
46 46         for result in Result.objects.all():
47 -         path = result.internalPath
47 +         path = result.dataPath
48 48         found.update([path])
49 49
50 50         dupes = [(path, count) for (path, count) in found.items() if count > 1]
@@ -53,7 +53,7 @@ def check_internalPath(self):

53 53         for path, count in dupes:
54 54             print path, count
```



# GitHub: wikis



The screenshot shows a GitHub repository page for 'cedadev / ceda-fbs'. The repository has 13 pulls, 1 star, and 0 forks. The 'Wiki' tab is selected, showing a page titled 'Elasticsearch Filters' edited by 'jrainnie' on 19 Sep 2016. The page content includes a section for 'Boolean filter' and a code block for a JSON filter definition. A sidebar on the right lists other wiki pages: Home, Elasticsearch Filters, Elasticsearch Mapping, Example queries, and Example queries dumped from Sense interface. At the bottom, there is a section to 'Clone this wiki locally' with the URL 'https://github.com/cedadev/'.

cedadev / ceda-fbs

Unwatch 13 Star 1 Fork 0

Code Issues 1 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

## Elasticsearch Filters

jrainnie edited this page on 19 Sep 2016 · 1 revision

### Elasticsearch Filters

#### Boolean filter

The "bool" filter is used to combine other filters, and is composed of three sections;

```
{
  "bool" : {
    "must" : [],
    "should" : [],
    "must_not" : [],
  }
}
```

- "must" clause is equivalent of && (Logical AND) operator
- "should" clause is equivalent of || (Logical OR) operator

Pages 5

- Home
- Elasticsearch Filters
- Elasticsearch Mapping
- Example queries
- Example queries dumped from Sense interface

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/cedadev/>

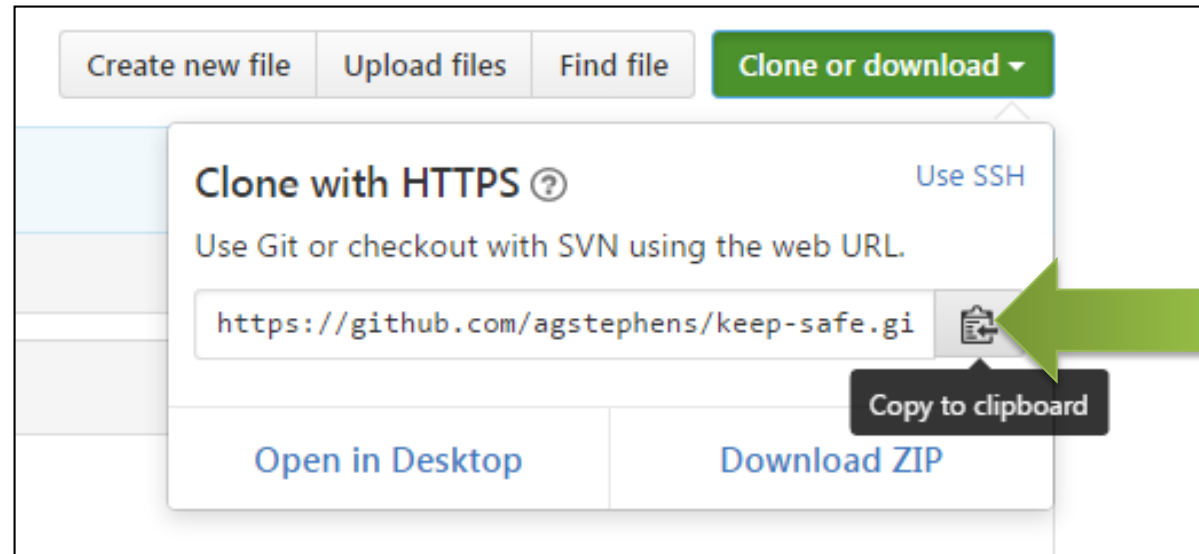


# GitHub does lots of funky things, but...

- On this course we are going only using it as a remote repository.
- We are going to concentrate on simply using git.

# Where to start 1: `git clone`

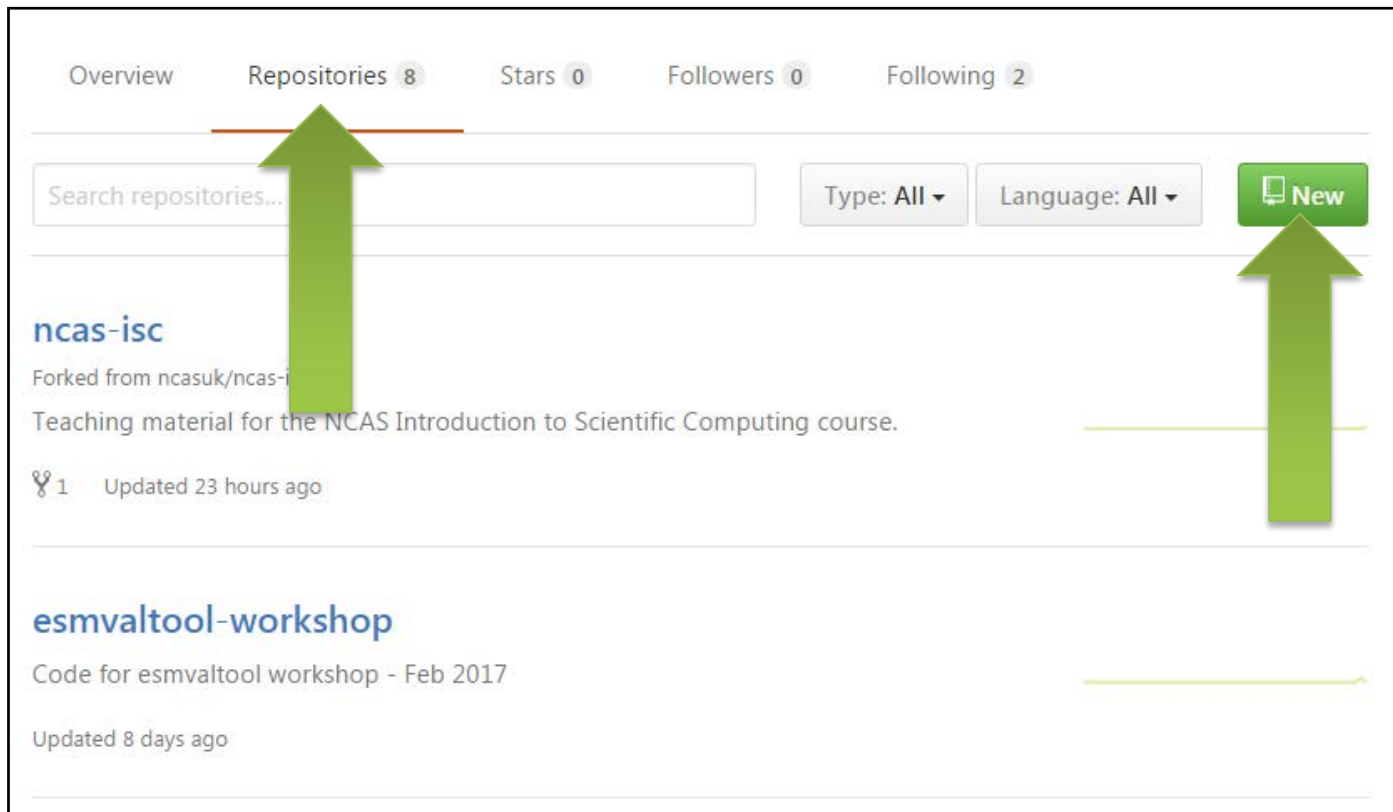
This makes a copy of a repository locally. We did this at the start of the course.



```
$ git clone  
https://github.com/agstephens/keep-safe
```

# Where to start 2: Create a repository on GitHub

Navigate to "Repositories" and click "New".





# Where to start 3: start a new repository from existing files

```
$ ls
x      y      z
$ git init
Initialized empty Git repository in
/Users/sjp23/play/york_workshop_shell/test-package/.git/
$ git add .
$ git commit -m'Initial commit from existing files'
[master (root-commit) 71ecfcf] Initial commit from
existing files
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 x
create mode 100644 y
create mode 100644 z
```

# Add a file to your local repo

1. Enter the repository directory:

```
$ cd ncas-isc
```

2. Create a new file:

```
$ echo "hello world" > hello.txt
```

3. Tell Git about the file:

```
$ git add hello.txt
```

4. Commit the file to the **local** Git repository:

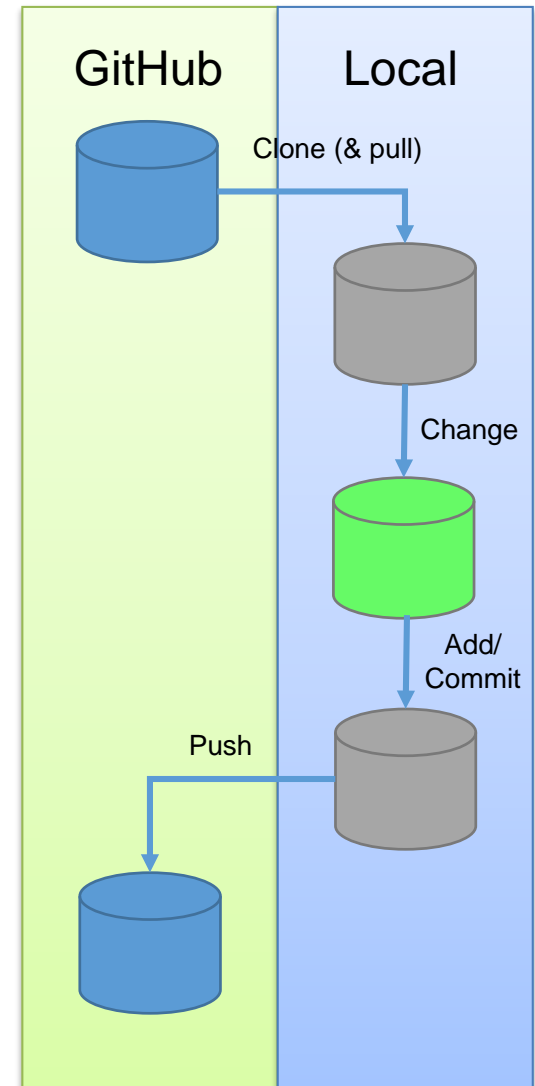
```
$ git commit -m "added hello"
```

5. Push any updates to the **remote** GitHub repo:

```
$ git push
```

# So, what just happened?

- We *cloned* the remote repository to our file system.
  - Now there are two identical copies of one repo.
- We *created* a new text file.
- We *added* and *committed* that new file to the local version of the repo.
- We used *push* to update the remote repo.






# Let's look on GitHub

## Before...

Branch: master ▾ New pull request

Create new file Upload files Find file Clone or download ▾

This branch is 2 commits behind ncasuk:master. [Pull request](#) [Compare](#)





|   |                          |                                  |
|---|--------------------------|----------------------------------|
|  spepler                 | added shell example data | Latest commit 72b7c75 7 days ago |
|  shell_example_data/pain | added shell example data | 7 days ago                       |
|  README.md               | Initial commit           | 10 days ago                      |

## After...

Branch: master ▾ New pull request

Create new file Upload files Find file Clone or download ▾

This branch is 1 commit ahead, 2 commits behind ncasuk:master. [Pull request](#) [Compare](#)

|   |                          |                                      |
|---|--------------------------|--------------------------------------|
|  spepler                  | added hello              | Latest commit fdd3c9e 22 seconds ago |
|  shell_example_data/pain | added shell example data | 7 days ago                           |
|  README.md               | Initial commit           | 10 days ago                          |
|  hello.txt               | added hello              | 22 seconds ago                       |

# The Plan: Use git / GitHub all week

- This stuff is hard to learn - we know that from experience.
- A presentation is quickly forgotten.
- So, we propose that you use Git/GitHub for every exercise.
- You are going to create and update your own Github repository with files from exercises throughout the course.

# Lets make some test files

```
$ mkdir mydir
$ echo "hi" > hi.txt
$ echo "testing..." > mydir/t1.txt
$ ls
hi.txt      hello.txt  mydir
```

# git Status

Use to see what stage files are at

```
$ git status
```

```
On branch master
```

```
Your branch is up-to-date with 'origin/master'.
```

```
Untracked files:
```

```
  (use "git add <file>..." to include in what will be  
  committed)
```

```
hi.txt
```

```
mydir/
```

```
nothing added to commit but untracked files present (use  
"git add" to track)
```

# git add

Adding files tells git to start looking after them or add a new version if it already knows about it.

```
$ git add .
```

```
$ git status
```

On branch master

Your branch is up-to-date with 'origin/master'.

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: hi.txt

new file: mydir/t1.txt



# Add another file

These files are all staged to go into the repository, but are not committed yet.

```
$ echo "testing..." > mydir/t2.txt
```

```
$ git add mydir/t2.txt
```

```
$ git status
```

On branch master

Your branch is up-to-date with 'origin/master'.

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

```
new file:   hi.txt
```

```
new file:   mydir/t1.txt
```

```
new file:   mydir/t2.txt
```

# Lets commit

Now the files are in the local repository. The working tree is the same as repository.

```
$ git commit -m 'Adding my new greetings files'
```

```
[master fe70026] Adding my new greetings files
```

```
3 files changed, 3 insertions(+)
```

```
create mode 100644 hi.txt
```

```
create mode 100644 mydir/t1.txt
```

```
create mode 100644 mydir/t2.txt
```

```
$ git status
```

```
On branch master
```

```
Your branch is ahead of 'origin/master' by 1 commit.
```

```
(use "git push" to publish your local commits)
```

```
nothing to commit, working tree clean
```

# Push the new version back to GitHub

Make the repo on GitHub match the local repo.

```
$ git push
```

```
Counting objects: 5, done.
```

```
Delta compression using up to 4 threads.
```







```
Compressing objects: 100% (3/3), done.
```

```
Writing objects: 100% (5/5), 465 bytes | 0 bytes/s, done.
```

```
Total 5 (delta 0), reused 0 (delta 0)
```

```
To github.com:spepler/ncas-isc.git
```

```
fdd3c9e..fe70026 master -> master
```

|   |                               |                                    |                              |                         |
|---|-------------------------------|------------------------------------|------------------------------|-------------------------|
| This branch is 2 commits ahead, 2 commits behind ncasuk:master.   |                               |                                    | <a href="#">Pull request</a> | <a href="#">Compare</a> |
|  spepler                  | Adding my new greetings files | Latest commit fe70026 22 hours ago |                              |                         |
|  mydir                   | Adding my new greetings files | 22 hours ago                       |                              |                         |
|  shell_example_data/pain | added shell example data      | 8 days ago                         |                              |                         |
|  README.md               | Initial commit                | 11 days ago                        |                              |                         |
|  hello.txt               | added hello                   | a day ago                          |                              |                         |
|  hi.txt                  | Adding my new greetings files | 22 hours ago                       |                              |                         |

# Enough?

- If you are working on your own then that is all you need to know.
- You can keep track of changes in your code, you know its safe and you can share it with people.

# Working with other people

```
$ git clone git@github.com:spepler/ncas-isc.git ncas-isc2
Cloning into 'ncas-isc2'...
remote: Counting objects: 17, done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 17 (delta 1), reused 16 (delta 0), pack-reused 0
Receiving objects: 100% (17/17), done.
Resolving deltas: 100% (1/1), done.
$ cd ncas-isc2
$ ls
hello.txt      hi.txt  mydir
$ emacs hello.txt
$ cat hello.txt
hello world
New line
```

Red Fred clones a copy of the repository  
and changes a file

# They commit their changes and push back to GitHub

```
$ git add hello.txt
$ git commit -m 'added new line'
[master d274491] added new line
 1 file changed, 1 insertion(+)
$ git push
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 283 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local
objects.
To github.com:spepler/ncas-isc.git
   fe70026..d274491  master -> master
```

# Black Bob downloads changes using git pull

```
$ git pull
```

```
remote: Counting objects: 3, done.
```

```
remote: Compressing objects: 100% (1/1), done.
```

```
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0
```

```
Unpacking objects: 100% (3/3), done.
```

```
From github.com:spepler/ncas-isc
```

```
    fe70026..d274491  master      -> origin/master
```

```
Updating fe70026..d274491
```

```
Fast-forward
```

```
hello.txt | 1 +
```

```
1 file changed, 1 insertion(+)
```

# Black Bob looks at change log

```
$ git log hello.txt
```

```
commit d274491d34d96aa92eb110e472006070e537dda0
```

```
Author: Sam Pepler <sam.pepler@stfc.ac.uk>
```

```
Date: Fri Feb 24 12:26:47 2017 +0000
```

```
added new line
```

```
commit fdd3c9eb7cbea69cce46ea22326ed5c801bb75f8
```

```
Author: Sam Pepler <sam.pepler@stfc.ac.uk>
```

```
Date: Thu Feb 23 11:13:13 2017 +0000
```

```
added hello
```



# Exercise

- Clone the repository we made yesterday:

```
$ cd  
$ git clone  
https://github.com/<username>/my-isc-work
```

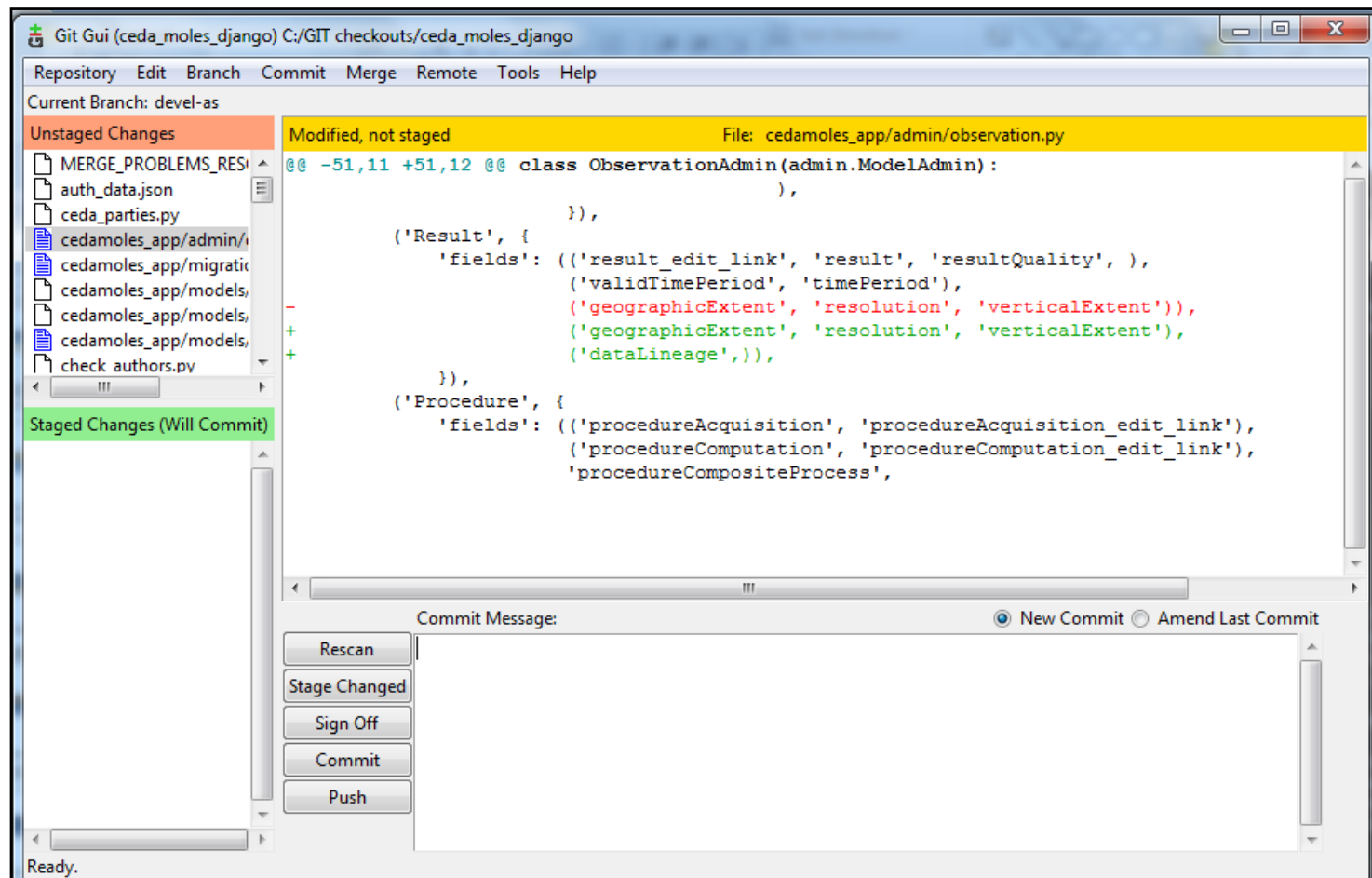
- Make a new directory in your cloned repo and a new file containing a few lines of text.
- Add the file and directory to your local repo: `$ git add <newfile> <newdir>`
- Commit the changes: `$ git commit -m 'Add some test files'`
- Update Github repo: `$ git push`



# Other tools in the Git ecosystem

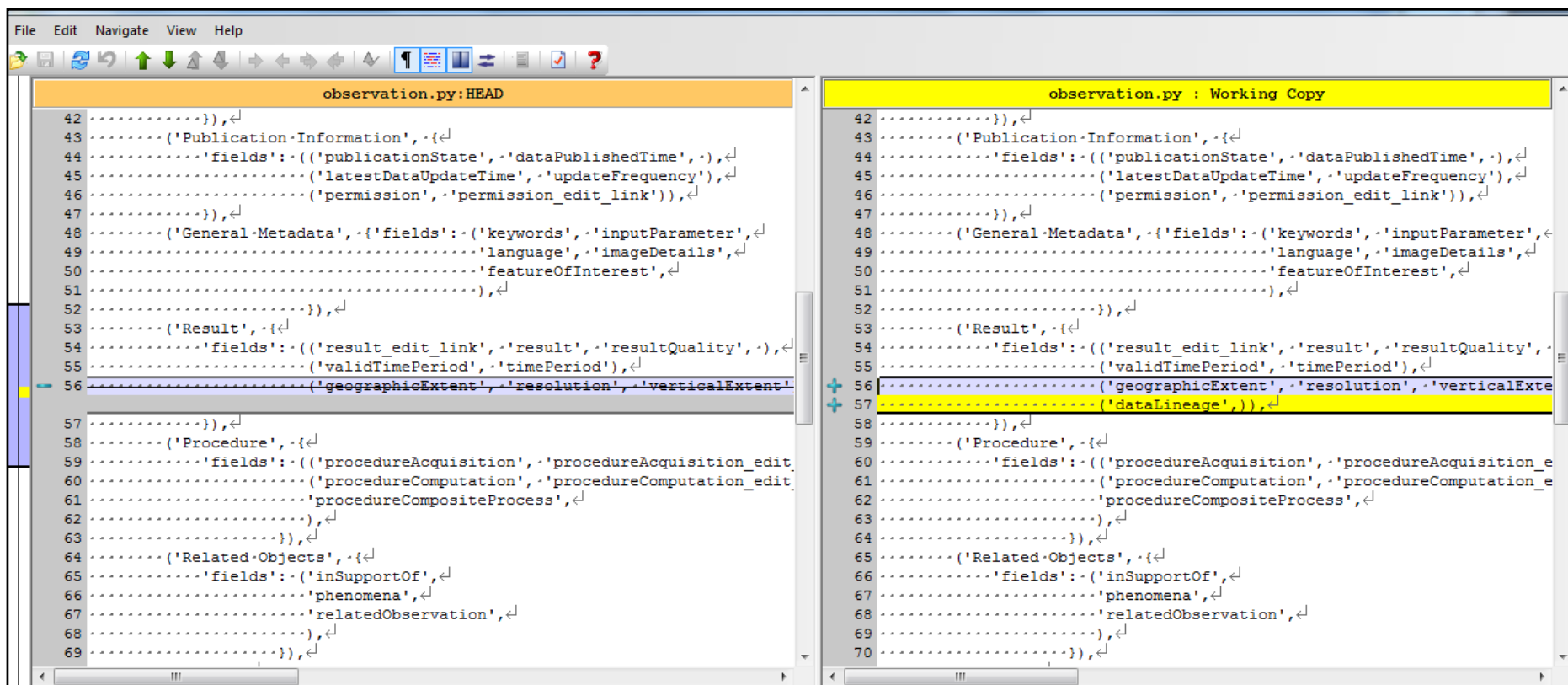
# git gui

Full set of GUIs for interacting with local and remote repos.



# TortoiseGIT (for Windows)

Provides GUIs for adding/committing/changing  
- including a side-by-side diff...



The screenshot displays the TortoiseGIT application interface with a side-by-side diff view. The left pane, titled 'observation.py:HEAD', shows the original code. The right pane, titled 'observation.py : Working Copy', shows the modified code. A yellow highlight in the right pane indicates a new line of code added at line 57: `.....('dataLineage',)),`. The code in both panes is a JSON-like structure representing metadata for an observation, including fields for publication information, general metadata, result details, procedure information, and related objects.





# The *why* - one more time

- Using version control will **save you time** – *No more accidentally deleting your workspace, or working on the wrong version of a file.*
- It will **make you a better programmer** – *It encourages good working practices: such as documenting change.*
- It will help you **collaborate more effectively** – *Others can access tagged releases of your code.*
- It will **boost your scientific integrity** – *Helping you document your work; aiding reproducibility.*
- It will **make you feel safe** – *No more waking up at 3 a.m. wondering if you backed up your work!*

# The NCAS GitHub organisation

An NCAS GitHub organisation has been set up.

This allows repositories to be set up that where users could share code when it has become a more formal collaboration.

If you want to become part of the NCAS GitHub please contact Ag, James or Dan and send them your GitHub account ID.

# Further information

Git documentation:

<http://git-scm.com/documentation>

Nice Git reference:

<http://gitref.org/>

GitHub:

<http://github.com/>

# Acknowledgements

We would like to Acknowledge the following authors for some of the content presented here:

*"Introduction to GIT"*. Lukas Fittl (<http://fittl.com>).

*"Git and GitHub"*. Darren Oakley.