MANCHESTER 1824

Git/GitHub tutorial

GitHub

Alan Stokes

HBP code jam #7
Jan 2016

Human Brain Project

SpiNNaker

Contents

1. **Getting ready**
   - What is Git / GitHub?
   - Setting Git / GitHub account.
   - Setting up a Git / GitHub Repository.
   - Command vs IDE's.

2. Basic Abilities

- Add.
- status.
- .gitignore.
- Revert.
- Commit.
- Diff.
- Push.

3. Collaboration Abilities

- Branches.
- Forks.
- Fetch / Pull / Pull Requests.
- Merges / **Conflicts!**
- **Rebase!**
- **Stash.**
- Snapshots / releases.

What is Git/ GitHub?

What Is it?

- A version control system (SVN, CVS, RCS etc).

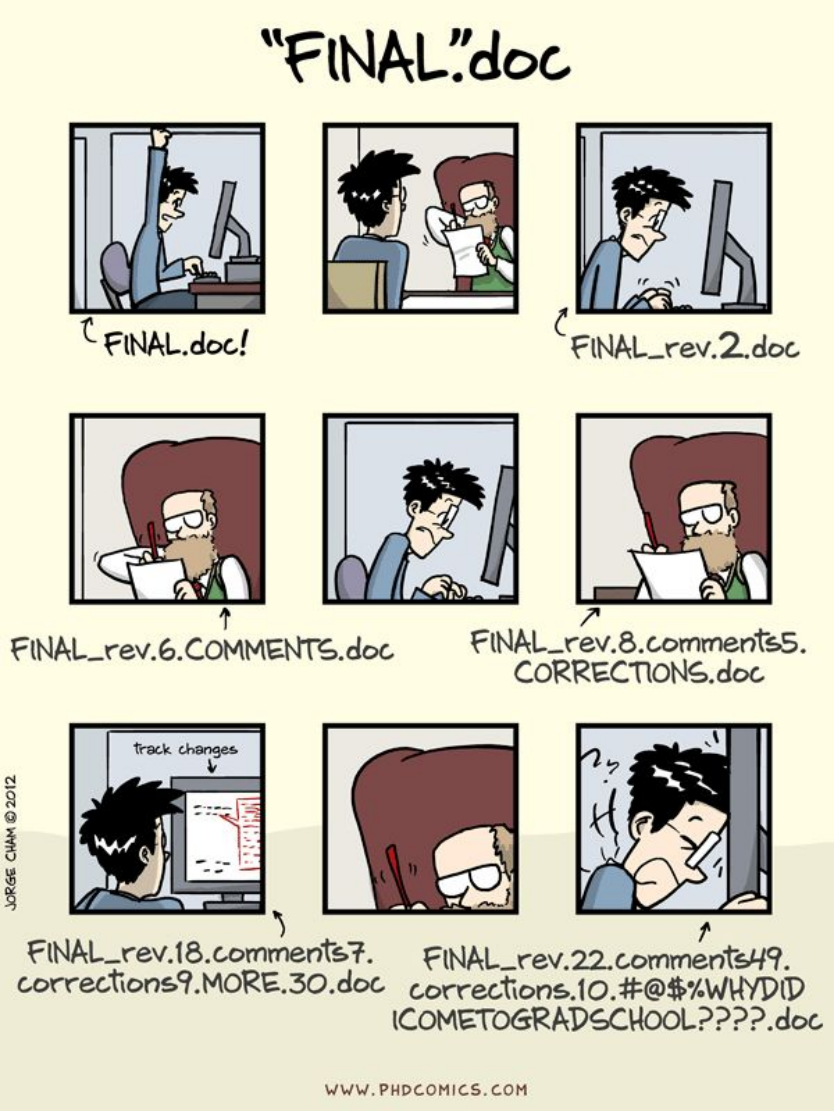
Why do we care?

1. Keep track of changes to your work.
2. Allows to backup work securely.
3. Allows easier organization of pieces of work.
4. Make collaboration easier (GitHub).

Work Place 1 with Git

Work Place 2 with Git

Version control



Thanks to Phd comics and software carpentry for this figure.

Why Git / GitHub?

- Old version controls were limited in their capabilities.
- Git/Github and Mercurial are distributed systems.
- With Git/GitHub everyone can be a “Git server”.
- GitHub is becoming a standard place to store code.

https://github.com/numpy/numpy

https://github.com/scipy/scipy

https://github.com/SpiNNakerManchester/sPyNNaker

https://github.com/NeuralEnsemble/PyNN

Who uses GitHub???? https://government.github.com/community/

Setting up a Git/GitHub Account

local machine command line

```
$ git config --global user.name "YOUR_NAME"
$ git config --global user.email "YOUR_EMAIL"
$ git config --global color.ui "auto"
$ git config --global credential.helper cache
```

Register on GitHub

1. Go to https://github.com/
2. fill in details.
3. choose package (recommend the free one!).
4. confirm email address via email.

Creating a Git Repository

Command Line

1. make a folder and enter it.
2. $ git init.

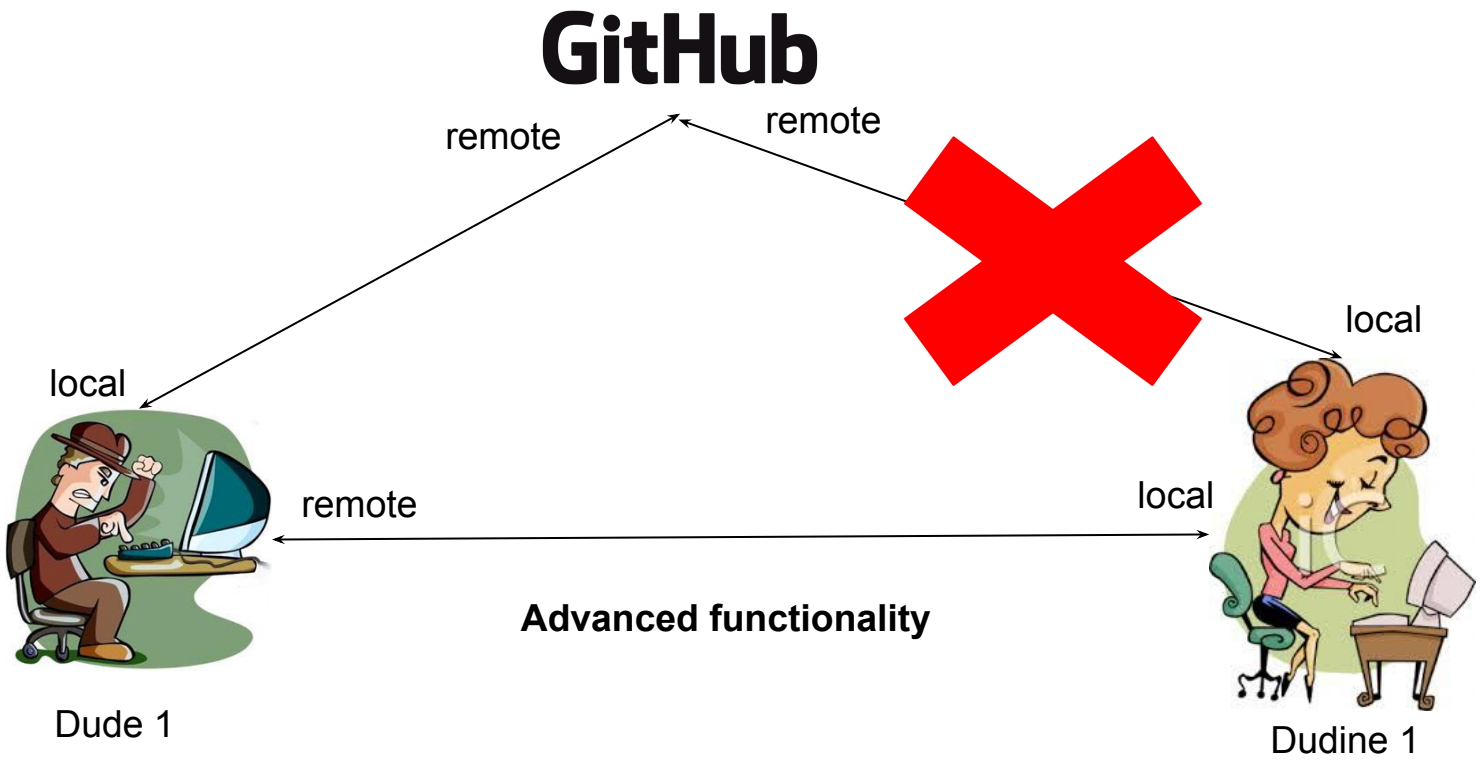
Register on GitHub

1. Log in to https://github.com/
2. Click on new repository in middle right side of screen.
3. Make name of repository.
4. Click create repository.
5. Follow instructions.

IDE

1. click on vcs->checkout_from_version_control->github
2. give http address for the repository on github.

Local / remote repositories



Command Line vs IDE's

Command Line

- Need to remember every command.
- More powerful.
- Can be painful to handle the more difficult bits of Git.

IDE's (Eclipse, PyCharm, Emacs etc... zzzzz)

- Usually have built in Git/GitHub support.
- Usually are much more intuitive to use.
- Hides complexity and thus can be less powerful.
- Can help more with the more difficult bits of Git.

https://www.jetbrains.com/pycharm/

https://www.gnu.org/software/emacs/

https://eclipse.org/downloads/

And Many many many more.

Basic Abilities: Add

- Tells git that you want to put this File in the local repository at some point.

Command Line

- create a file.
- $ git add **FILENAME**

IDE

- Create a file.
- will be asked if the file should be added to the repository.
- Some IDE's may not prompt you, and you'll need to find the add command.

FileName

remote repository **local repository** **Uncommited**

Basic Abilities: Status

- Lets you see uncommitted files.

Command Line

```
$ git status
```

```
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#   new file:   filename.txt
#
```

IDE

- Not normally exposed.
- Can be found by asking to commit.

Basic Abilities: .gitignore

- File added to the top level folder of a git repository.
- Informs git of types of files / folders to not commit to git.
- contains a list of names and folders.

example .gitignore file

```
/.project
/.pydevproject
/.idea
/.settings
/doc/build
*.pyc
*reports/
*application_generated_data_files/
*build/
*dist/
*egg-info/
```

Basic Abilities: Revert

Allows you to remove all uncommitted changes

Command Line

$ git revert

IDE

1. Right Click on the repository in question.
2. From the pop up menu git->revert
3. Approve the list of files that are going to be reverted.

Basic Abilities: commit

- Stores files into your local repository.

Command Line

- $ git commit -m "SOME MESSAGE EXPLAINING COMMIT"

IDE

- Right Click on repository in listings.
- In popup list go to:
    - git->commit_directory
- select files.
- click commit.

GitHub

remote repository

Basic Abilities: Diff (1 of 2)

Allows you to see the difference between local repository and uncommitted files.

Command Line

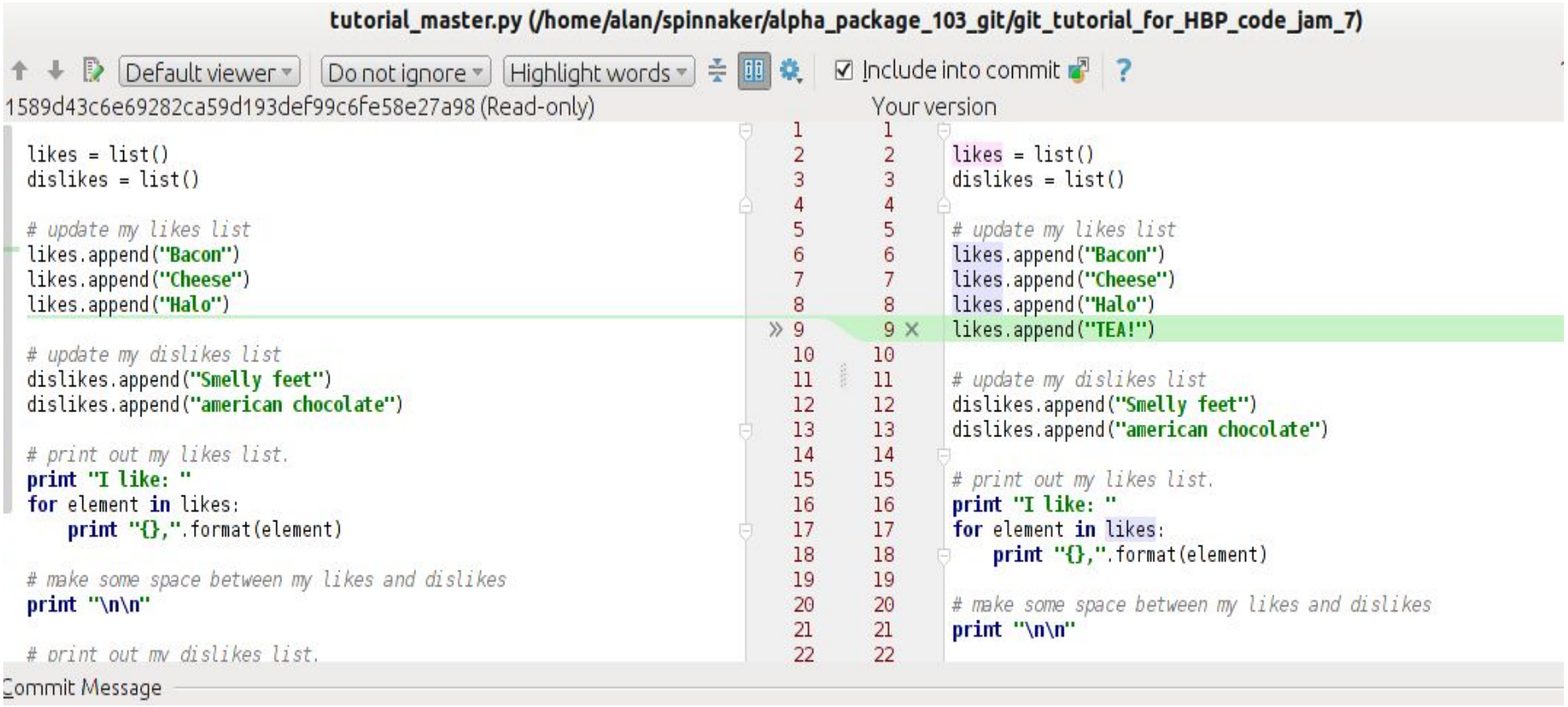
```
$ git diff
```

```
diff --git a/tutorial_master.py b/tutorial_master.py
index df0654a..315bf3a 100644
--- a/tutorial_master.py
+++ b/tutorial_master.py
@@ -1 +1,2 @@
likes.append("Halo")
+likes.append("TEA!")
```

IDE

- Get to the commit window.
- Double click on a file.
- Gets a comparision screen.

Basic Abilities: Diff (2 of 2)



Basic Abilities: push

Moves files between the local repository and the remote repository.

Command Line

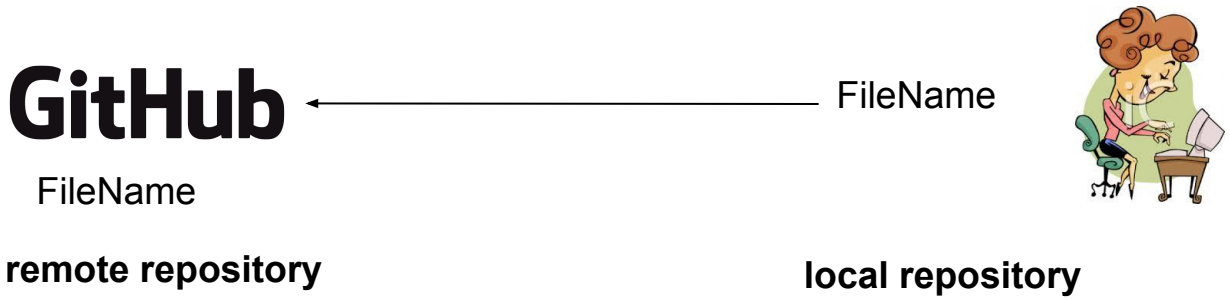
```
$ git push
```

IDE

- Get to the commit window.
- Click commit and push.

OR

- Right Click on repository in listings.
- In popup list go to:
  - git->repository->push



Collaboration

- Where all the exciting stuff happens.
- All with the use of remote servers.
- GitHub, BitBucket, GitHosting etc.

Rest of these slides are about GitHub

Collaboration Abilities: Branches

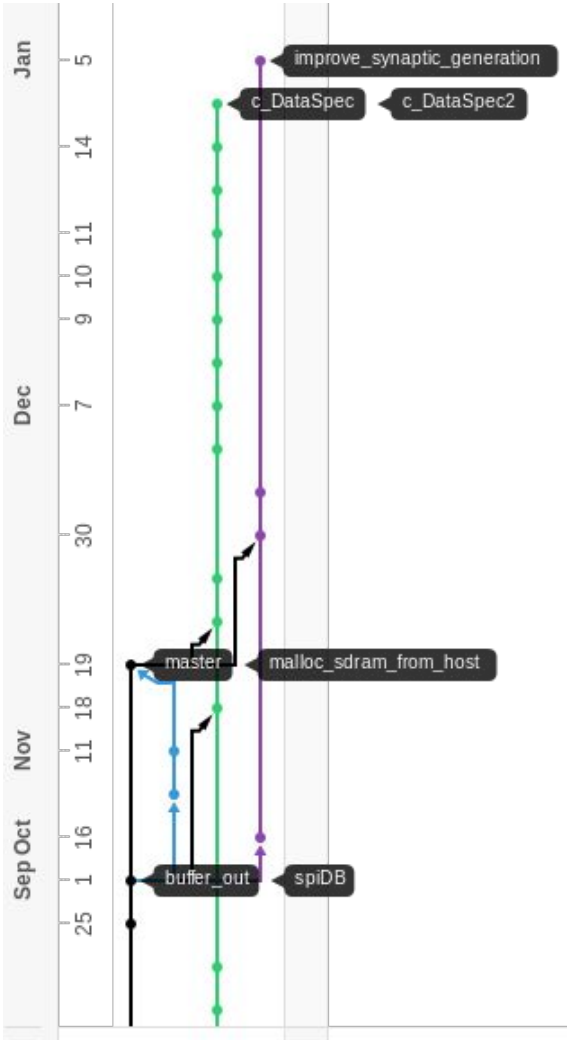
Supports doing changes on your repository without changing the main repository.

Command Line

$ git checkout -b **NEW_BRANCH_NAME**

IDE

1. Click on VCS on top level menu.
2. From the pop up menu go to git->branches
3. Select repository.
4. Select “new branch”.
5. Enter **NEW_BRANCH_NAME**.



Collaboration Abilities: Forks

Supports making branches of other people's code without needing authorisation to commit to their repositories.

Command Line
Not supported.
IDE
Not supported.

GitHub via browser

- Log onto github.com
- Locate the repository you want to fork.
- Click fork in the top right corner.
- Select where to store the fork.

Collaboration Abilities: Branches ACTIVITY!!!!

- Now we're going to build a few branches together.

Code

```
likes = list()
dislikes = list()
# update my likes list
likes.append("Bacon")
likes.append("Cheese")
likes.append("Halo")
# update my dislikes list
dislikes.append("Smelly feet")
dislikes.append("american chocolate")
# print out my likes list.
print "I like: "
for element in likes:
   print "{},".format(element)
# make some space between my likes and dislikes
print "\n\n"
# print out my dislikes list.
print "I dislike: "
for element in dislikes:
   print "{}".format(element)
```

https://github.com/alan-stokes/git_tutorial_for_HBP_code_jam_7

Collaboration Abilities: Fetch / Pull / Pull requests (1 of 4)

Fetch

updates records in local about what's changed in remote repositories.

Command Line

$ git fetch

IDE

1. click on VCS menu
2. click git->fetch.

Collaboration Abilities: Fetch / Pull / Pull requests (2 of 4)

Pull

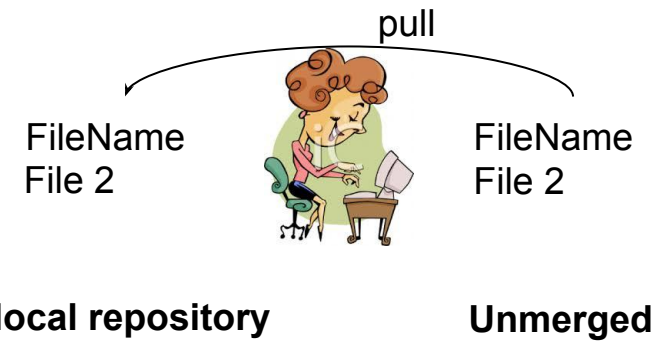
Merges changes from remote to your local repository.

Command Line

$ git pull

IDE

1. Right Click on the repository in the listings.
2. on the pop up menu go to git->repository->pull



Collaboration Abilities: Fetch / Pull / Pull requests (3 of 4)

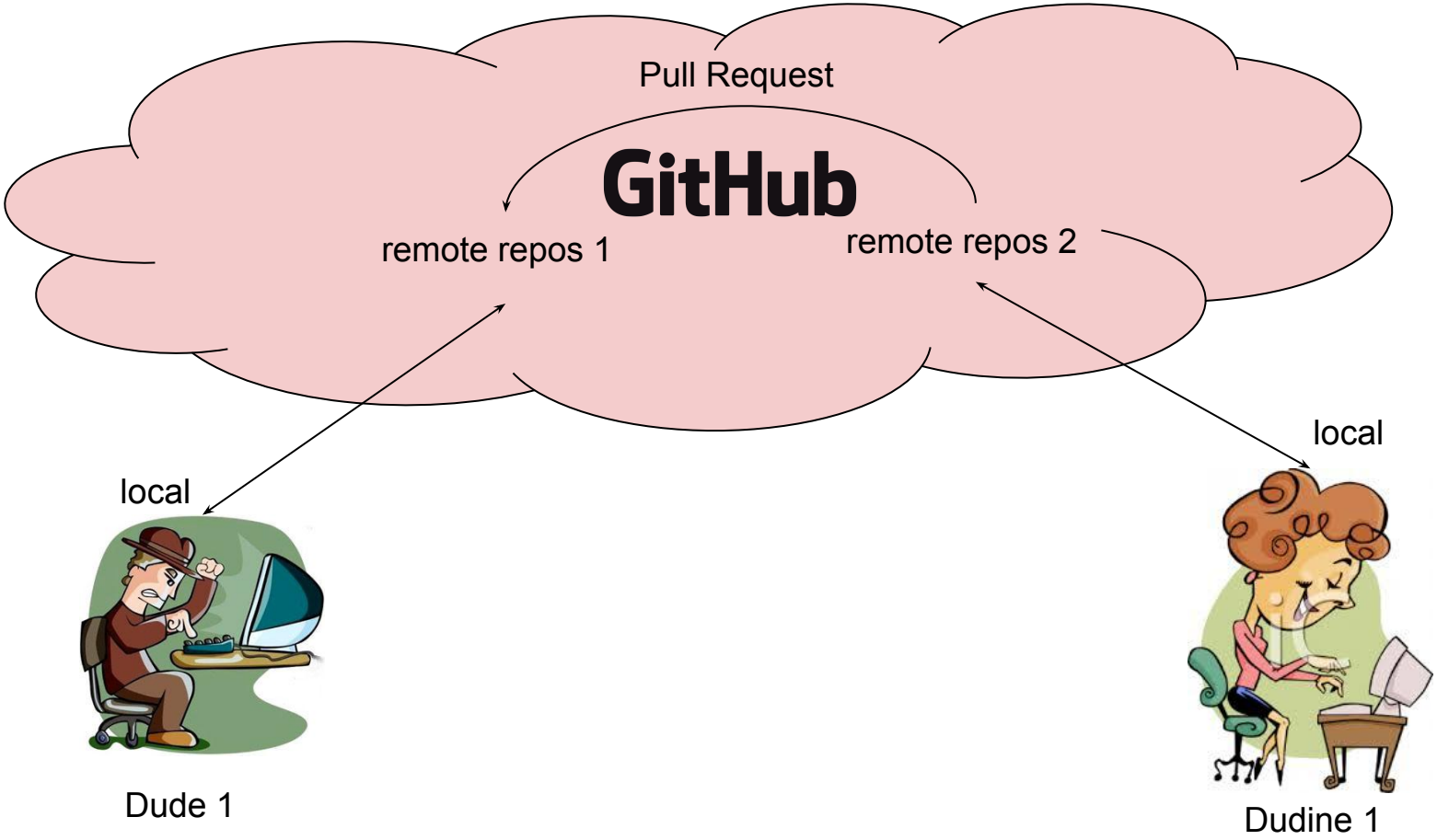
Pull Requests

requests that someone else merges your repository into theirs.

Command Line & IDE
Not supported

GitHub
1. Log into github.com
2. Get onto your repository.
3. Click new pull request.
4. Select **compare** to be your branch/fork, **base** the repository/fork you want to merge your stuff into.

Collaboration Abilities: Fetch / Pull / Pull requests (4 of 4)



Collaboration Abilities: Pull request Example

Example pull request from spinnaker manchester software stack

https://github.com/SpiNNakerManchester/sPyNNaker/pull/182

Collaboration Abilities: Merge and Conflicts (1 of 4)

Merge

1. The act of integrating changes between two repositories.
2. Executed every time you push / pull / commit.
3. Mostly invisible till you are dealing with pull requests.

Conflicts

The act of handling changes between two repositories on the same location.

Collaboration Abilities: Merge and Conflicts (2 of 4)

Just before a Merge

- Ensure you do a fetch to update your cache.
- Ensure that the local repository has no uncommitted changes.

Command Line

1. move into repository you want to merge into.

$git merge NameOfBranch

IDE

1. Right click on repository in listing.
2. From pop up menu git->repository->merge_changes
3. select branch to merge in.
4. follow instructions.

Collaboration Abilities: Merge and Conflicts (3 of 4)

GitHub

1. log onto github.com
2. Select your repository you want to merge into.
3. Select pull requests from the tabs.
4. Select the pull request you want to deal with.
5. Select **merge pull request**

If there are conflicts within the merge, you need to download the repository and merge it using command line of IDE or diss reader functionalities.

Collaboration Abilities: Merge and Conflicts (4 of 4)

Comparison of the command line and IDE support for conflicts.

Collaboration Abilities: Merge and Conflict Activity

Now we're going to merge all our branches back into master.

Collaboration Abilities: Rebase!!!!!

1. The act of changing the history of a repository to make changes disappear.
2. Really really really tabooed.
3. Makes tracking your changes much more difficult.

Alternative

Do a commit which changes the changes back to the state you desire.

Collaboration Abilities: Stash (1 of 2)

Stash

Allows you to cache uncommitted changes into a temporary repository.

Command Line

$ git stash

IDE

1. Right Click on repository in listings.
2. in pop up menu go to git->repository->stash_changes
3. name the stash

Collaboration Abilities: Stash (2 of 2)

Unstash

Allows you to merge these uncommitted changes back into your repository.

Command Line

$ git stash list

```
stash@{0}: WIP on master: 049d078 added the index file
stash@{1}: WIP on master: c264051 Revert "added file_size"
stash@{2}: WIP on master: 21d80a5 added number to log
```

$ git stash apply **or** git stash apply stash@{id}

IDE

1. Right Click on repository in listings.
2. In pop up menu go to git->repository->unstash_changes
3. Select the stash
4. Click apply stash

Collaboration Abilities: Snapshots / releases

Snapshots

Allows you to tag a commit specially as a commit which is important for some reason.

- A specific release version:
  - Just Testing, Little Rascal, Arbitrary, Another Fine Product From The Nonsense Factory.
- Used for finding a certain version of code.

GitHub

1. log onto github.com
2. To go repository you want to make a release for.
3. click on releases
4. click **draft new release**
5. follow instructions.

WorkBook

- GO to https://github.com/alan-stokes/git_tutorial_for_HBP_code_jam_7/blob/master/HBPCodeJam7GitTutorialworkbook.pdf

- Follow tasks