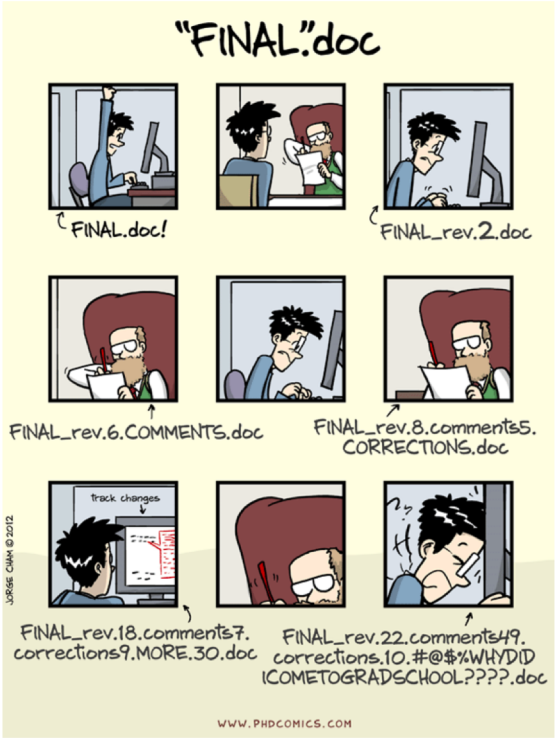
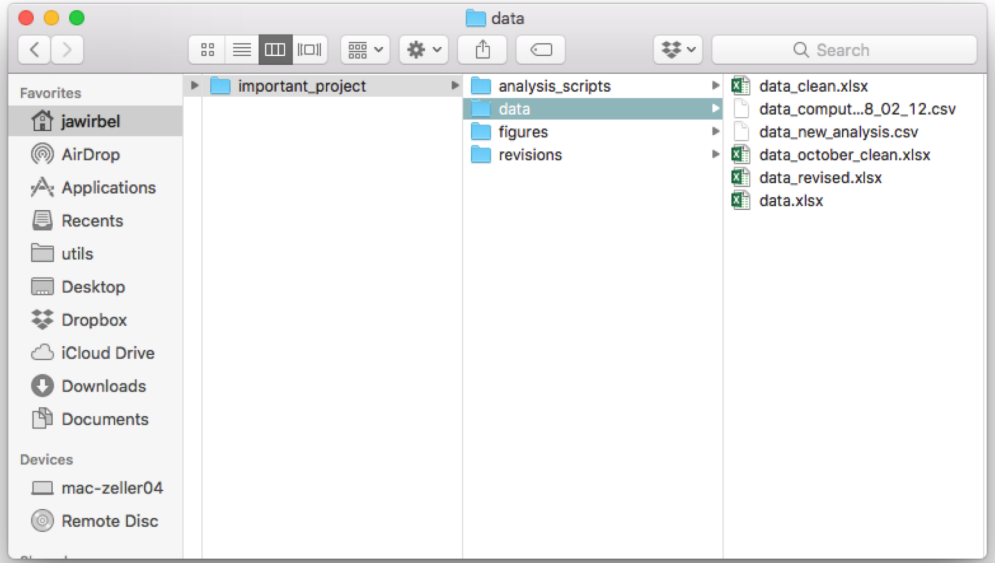
Version control with Git

BTM 2018

What is the problem?





What is the solution?

When to use git?

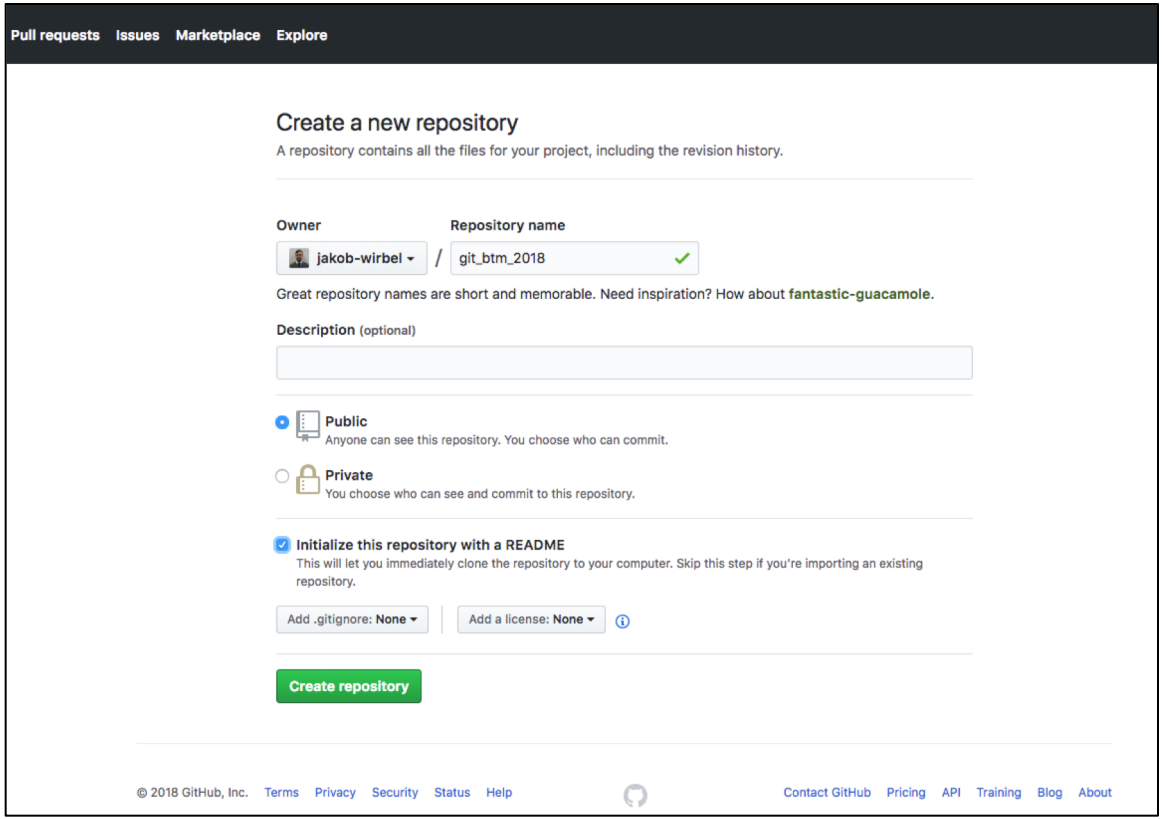
- Code. Version control was made for code.
- Manuscripts
- Collaborative projects
- Everything

When **not** to use git?

- Short-term stuff

How to start?

- Can also be done in the command line
- You can also start a new git repository with already existing files



git clone

```
-bash-4.2$ git clone https://github.com/jakob-wirbel/git_btm_2018.git
Cloning into 'git_btm_2018'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
-bash-4.2$
```

git status

```
-bash-4.2$ git status
# On branch master
nothing to commit, working directory clean
-bash-4.2$
```

Let’s change something

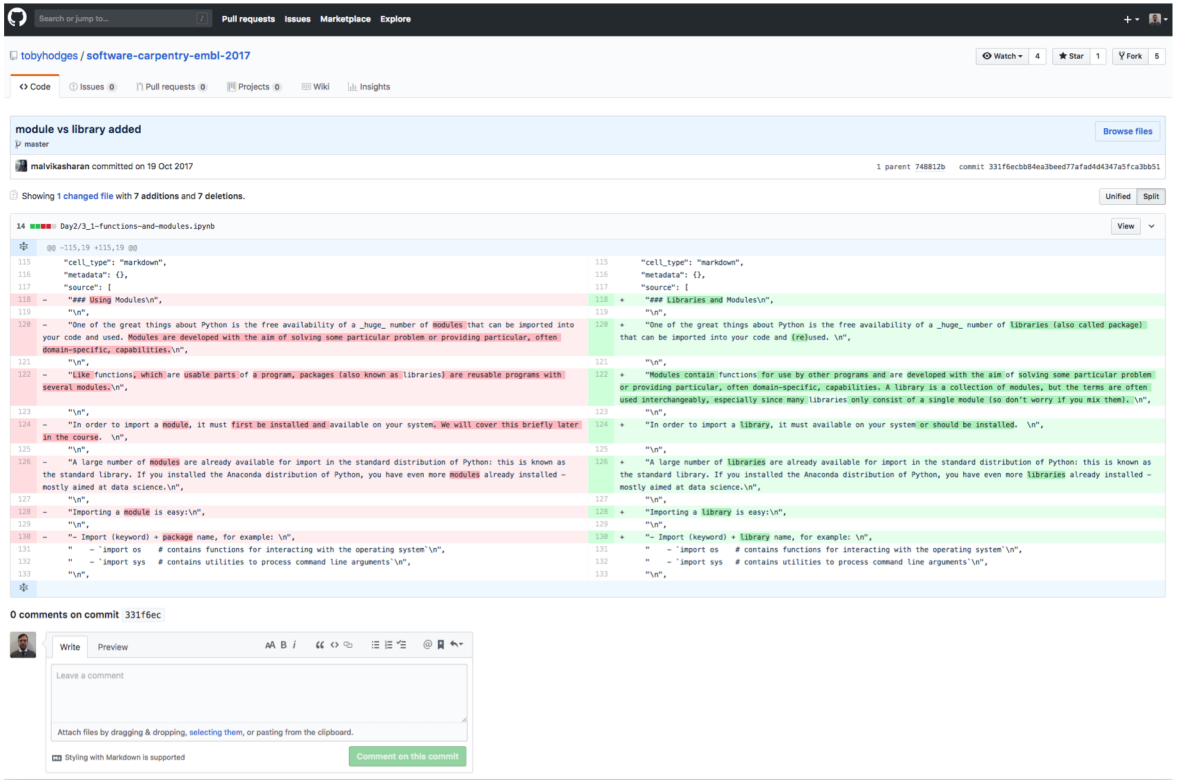
```
[-bash-4.2$ nano README.md
[-bash-4.2$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   README.md
#
no changes added to commit (use "git add" and/or "git commit -a")
```

git diff

```
[-bash-4.2$ git diff README.md
diff --git a/README.md b/README.md
index 6086c6e..2bce39c 100644
--- a/README.md
+++ b/README.md
@@ -1,3 +1,6 @@
 # git_btm_2018

 This repository contains the presentation introducing `git` in the **B**as
+
+If you know `git` already, please go and check out the [this
+guide](https://swcarpentry.github.io/git-novice/).
[-bash-4.2$
```

git diff


Search or jump to...
Pull requests
Issues
Marketplace
Explore
tobyhodges / software-carpentry-embl-2017
Watch 4
Star 1
Fork 5
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Insights
module vs library added
Browse files
master
malvikasharan committed on 19 Oct 2017
1 parent 748812b
commit 331f6ecbb84ea3beed77afad4d4347a5fca3bb51
Showing 1 changed file with 7 additions and 7 deletions.
Unified
Split
14
Day2/3_1-functions-and-modules.ipynb
View
@@ -115,19 +115,19 @@
"cell_type": "markdown",
"metadata": {},
"source": [
- "### Using Modules\n",
+ "### Libraries and Modules\n",
"\n",
- "One of the great things about Python is the free availability of a _huge_ number of modules that can be imported into your code and used. Modules are developed with the aim of solving some particular problem or providing particular, often domain-specific, capabilities.\n",
+ "One of the great things about Python is the free availability of a _huge_ number of libraries (also called package) that can be imported into your code and (re)used. \n",
"\n",
- "Like functions, which are usable parts of a program, packages (also known as libraries) are reusable programs with several modules.\n",
+ "Modules contain functions for use by other programs and are developed with the aim of solving some particular problem or providing particular, often domain-specific, capabilities. A library is a collection of modules, but the terms are often used interchangeably, especially since many libraries only consist of a single module (so don't worry if you mix them). \n",
"\n",
- "In order to import a module, it must first be installed and available on your system. We will cover this briefly later in the course. \n",
+ "In order to import a library, it must available on your system or should be installed. \n",
"\n",
- "A large number of modules are already available for import in the standard distribution of Python: this is known as the standard library. If you installed the Anaconda distribution of Python, you have even more modules already installed - mostly aimed at data science.\n",
+ "A large number of libraries are already available for import in the standard distribution of Python: this is known as the standard library. If you installed the Anaconda distribution of Python, you have even more libraries already installed - mostly aimed at data science.\n",
"\n",
- "Importing a module is easy:\n",
+ "Importing a library is easy:\n",
"\n",
- "- Import (keyword) + package name, for example: \n",
+ "- Import (keyword) + library name, for example: \n",
" - `import os # contains functions for interacting with the operating system`\n",
" - `import sys # contains utilities to process command line arguments`\n",
"\n",
0 comments on commit 331f6ec
Write
Preview
Leave a comment
Attach files by dragging & dropping, selecting them, or pasting from the clipboard.
Styling with Markdown is supported
Comment on this commit

git add

```
-bash-4.2$ git add README.md
-bash-4.2$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   README.md
#
-bash-4.2$
```

git commit

```
-bash-4.2$ git commit -m 'Updated README.md'
[master 98dc7aa] Updated README.md
 1 file changed, 3 insertions(+)
```

- git commit –m ‘Message’
- The message should be informative to others (including you in four months!)

git commit

```
-bash-4.2$ git commit -m 'Updated README.md'
[master 98dc7aa] Updated README.md
 1 file changed, 3 insertions(+)
-bash-4.2$ 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 directory clean
-bash-4.2$
```

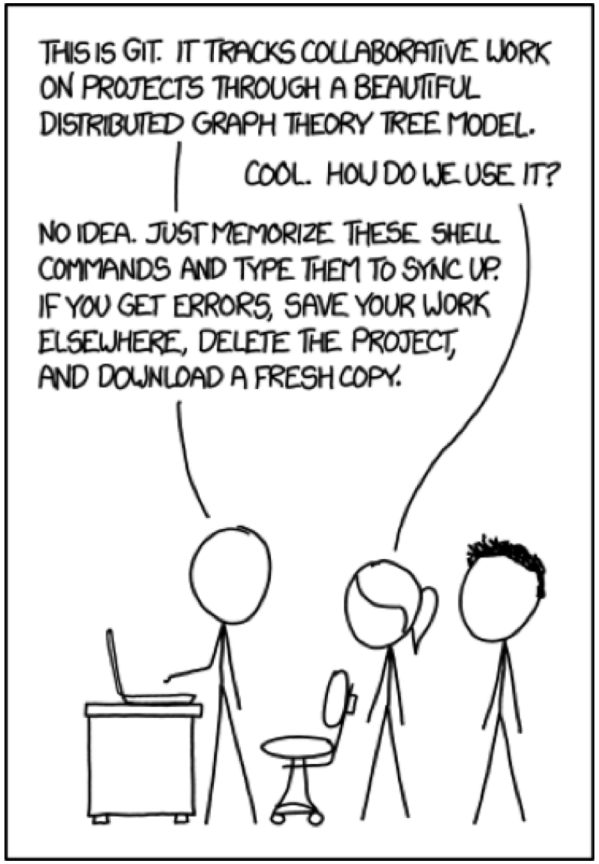
Pushing & Pulling

```
-bash-4.2$ git push
Username for 'https://github.com': jakob-wirbel
Password for 'https://jakob-wirbel@github.com':
Counting objects: 5, done.
Delta compression using up to 64 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 452 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/jakob-wirbel/git_btm_2018.git
   e25edd9..98dc7aa  master -> master
-bash-4.2$
```

Command	What does it do?
git clone *url*	Download a project and its entire history
git status	Lists all new modifications (not yet committed)
git diff *filename*	Show what has been changed in this file
git add *filename*	Snapshot this file
git commit –m ‘Message’	Record the snapshot **forever**
git push	Uploads all commits to the remote repo
git pull	Download all changes since last pull

Where to go from here?

- Software Carpentry Course on Git
- Resources to learn GitHub
- GitHub Documentation



The 99%

- git is very powerful, but hard to learn
- You can do so much more:
  - ignore files in your repository
  - create and merge branches
  - rewrite history, undo mistakes
  - work collaboratively on projects

Hands on

- Organize in pairs
- One of you initiate a repository in Github or git.embl.de
- Add the other person as contributor
- One of you
  - add a file
  - commit it
  - push it
- The other one
  - pull the changes
  - edit the file
  - push it