

Introduction to git and GitHub

Daniel Thaagaard Andreasen
daniel.andreasen@astro.up.pt

CAUP

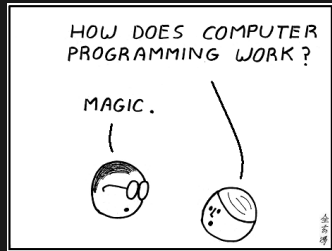
29th of January, 2015

Table of contents

- 1 What is git?
- 2 Collaboration

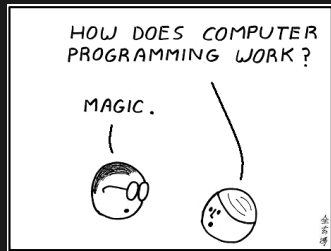
Your job

- **Create things**



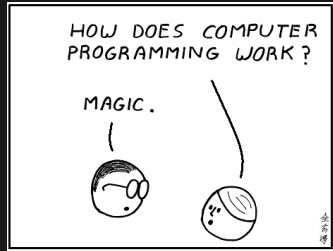
Your job

- **Create** things
- **Save** things



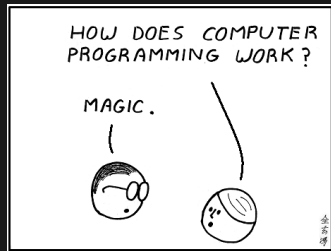
Your job

- **Create** things
- **Save** things
- **Edit** things



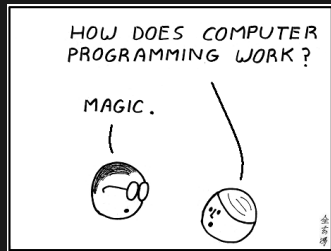
Your job

- **Create** things
- **Save** things
- **Edit** things
- **Save** things again
(overwrite)



Your job

- **Create** things
- **Save** things
- **Edit** things
- **Save** things again
(overwrite)
- The last point here is crucial!



Your job, improved

- Keep track of a file every time it is saved.

Your job, improved

- Keep track of a file every time it is saved.
- This is the step further for backups which just overwrite previous versions of a file.

Your job, improved

- Keep track of a file every time it is saved.
- This is the step further for backups which just overwrite previous versions of a file.
- Be sure to write helpful messages!

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

What is git?

- Git is a modern and fast version control (invented by Linus Torvalds).



What is git?

- Git is a modern and fast version control (invented by Linus Torvalds).
- A tool to keep track of changes in files with a time stamp and a user-written message ("What did I do and why?").



What is git?

- Git is a modern and fast version control (invented by Linus Torvalds).
- A tool to keep track of changes in files with a time stamp and a user-written message ("What did I do and why?").
- Allow the user to "push" files to a server: supernova, GitHub, etc.

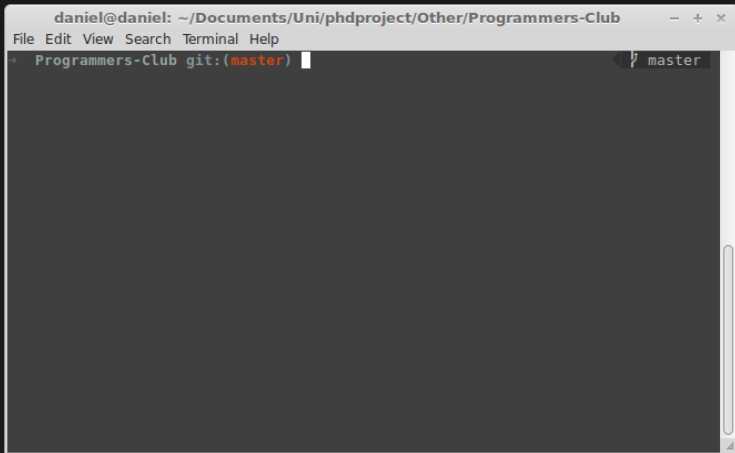


What is git?

- Git is a modern and fast version control (invented by Linus Torvalds).
- A tool to keep track of changes in files with a time stamp and a user-written message ("What did I do and why?").
- Allow the user to "push" files to a server: supernova, GitHub, etc.
- So in the end you have backup *and* all previous versions of files.



Workflow - forward



A terminal window titled "daniel@daniel: ~/Documents/Uni/phdproject/Other/Programmers-Club" with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt shows "Programmers-Club git:(master) |" and the command "git checkout master" has been entered, with the output "Switched to branch 'master'" visible on the next line.

```
daniel@daniel: ~/Documents/Uni/phdproject/Other/Programmers-Club
File Edit View Search Terminal Help
→ Programmers-Club git:(master) | git checkout master
Switched to branch 'master'
```

Workflow - forward

```
daniel@daniel: ~/Documents/Uni/phdproject/Other/Programmers-Club
File Edit View Search Terminal Help
→ Programmers-Club git:(master) ls
LICENSE README.md
→ Programmers-Club git:(master) git status
On branch master
Your branch is up-to-date with 'origin/master'.

nothing to commit, working directory clean
→ Programmers-Club git:(master) █
```


Workflow - forward

```
daniel@daniel: ~/Documents/Uni/phdproject/Other/Programmers-Club
File Edit View Search Terminal Help
→ Programmers-Club git:(master) ls
LICENSE README.md
→ Programmers-Club git:(master) git status
On branch master
Your branch is up-to-date with 'origin/master'.

nothing to commit, working directory clean
→ Programmers-Club git:(master) touch test
→ Programmers-Club git:(master) ls
LICENSE README.md test
→ Programmers-Club git:(master) 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)

    test

nothing added to commit but untracked files present (use "git add" to track)
→ Programmers-Club git:(master) █
```

Workflow - forward

```
daniel@daniel: ~/Documents/Uni/phdproject/Other/Programmers-Club
File Edit View Search Terminal Help
→ Programmers-Club git:(master) touch test
→ Programmers-Club git:(master) ls
LICENSE README.md test
→ Programmers-Club git:(master) 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)

    test

nothing added to commit but untracked files present (use "git add" to track)
→ Programmers-Club git:(master) git add test
→ Programmers-Club git:(master) x 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:   test

→ Programmers-Club git:(master) x
```

Workflow - forward

```
daniel@daniel: ~/Documents/Uni/phdproject/Other/Programmers-Club
File Edit View Search Terminal Help

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)

        test

nothing added to commit but untracked files present (use "git add" to track)
→ Programmers-Club git:(master) git add test
→ Programmers-Club git:(master) x 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:   test

→ Programmers-Club git:(master) x git commit -m "Added new file"
[master a999b69] Added new file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test
→ Programmers-Club git:(master)
```

Workflow - forward

```
daniel@daniel: ~/Documents/Uni/phdproject/Other/Programmers-Club
File Edit View Search Terminal Help

nothing added to commit but untracked files present (use "git add" to track)
→ Programmers-Club git:(master) git add test
→ Programmers-Club git:(master) x 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:   test

→ Programmers-Club git:(master) x git commit -m "Added new file"
[master a999b69] Added new file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test
→ Programmers-Club git:(master) git push
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 309 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/DanielAndreasen/Programmers-Club.git
 1177e53..a999b69  master -> master
→ Programmers-Club git:(master)
```

Workflow - backward

```
daniel@daniel: ~/Documents/Uni/phdproject/Other/Programmers-Club
File Edit View Search Terminal Help
→ Programmers-Club git:(master) ls
LICENSE README.md test
→ Programmers-Club git:(master) cat test
→ Programmers-Club git:(master) 
```

Workflow - backward

```
daniel@daniel: ~/Documents/Uni/phdproject/Other/Programmers-Club
File Edit View Search Terminal Help
→ Programmers-Club git:(master) ls                               ↵ master
LICENSE README.md test
→ Programmers-Club git:(master) cat test                          ↵ master

→ Programmers-Club git:(master) echo "This is some text" >> test ↵ master
→ Programmers-Club git:(master) x cat test                       ↵ master
This is some text
→ Programmers-Club git:(master) x                                ↵ master
```

Workflow - backward

```
daniel@daniel: ~/Documents/Uni/phdproject/Other/Programmers-Club
File Edit View Search Terminal Help
→ Programmers-Club git:(master) ls
LICENSE README.md test
→ Programmers-Club git:(master) cat test

→ Programmers-Club git:(master) echo "This is some text" >> test
→ Programmers-Club git:(master) x cat test
This is some text
→ Programmers-Club git:(master) x git add test
→ Programmers-Club git:(master) x git commit -m "Added some text"
[master dele03e] Added some text
1 file changed, 1 insertion(+)
→ Programmers-Club git:(master) git push
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 274 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To https://github.com/DanielAndreasen/Programmers-Club.git
a999b69..dele03e master -> master
→ Programmers-Club git:(master)
```

Workflow - backward

```
git log
File Edit View Search Terminal Help
commit de1e03eccb41b540a102373227139f068b808608
Author: DanielAndreasen <daniel.andreasen@astro.up.pt>
Date: Tue Jan 27 14:51:23 2015 +0000

    Added some text

commit a999b6957738d3d799b78c25d01e2f6fe964da34
Author: DanielAndreasen <daniel.andreasen@astro.up.pt>
Date: Tue Jan 27 14:39:55 2015 +0000

    Added new file

commit 1177e532519b3ab918e82b8aadcf4d1f283f49f
Author: DanielAndreasen <daniel.andreasen@astro.up.pt>
Date: Mon Jan 26 15:07:51 2015 +0000

    Added wiki

commit a6de123bd044cea3f8b05ff3c3176875deacb538
Author: DanielAndreasen <daniel.andreasen@astro.up.pt>
Date: Mon Jan 26 15:00:34 2015 +0000

    Added description
:
```


Workflow - backward

```
daniel@daniel: ~/Documents/Uni/phdproject/Other/Programmers-Club
File Edit View Search Terminal Help

1 file changed, 1 insertion(+)
→ Programmers-Club git:(master) git push
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 274 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To https://github.com/DanielAndreasen/Programmers-Club.git
 a999b69..dele03e  master -> master
→ Programmers-Club git:(master) git log
→ Programmers-Club git:(master) git checkout a999b6957738
Note: checking out 'a999b6957738'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

    git checkout -b new_branch_name

HEAD is now at a999b69... Added new file
→ Programmers-Club git:(a999b69) █
```

Workflow - backward

```
daniel@daniel: ~/Documents/Uni/phdproject/Other/Programmers-Club
File Edit View Search Terminal Help
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 274 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To https://github.com/DanielAndreasen/Programmers-Club.git
a999b69..dele03e master -> master
→ Programmers-Club git:(master) git log
→ Programmers-Club git:(master) git checkout a999b6957738
Note: checking out 'a999b6957738'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

    git checkout -b new_branch_name

HEAD is now at a999b69... Added new file
→ Programmers-Club git:(a999b69) ls
LICENSE README.md test
→ Programmers-Club git:(a999b69) cat test
→ Programmers-Club git:(a999b69) 
```

Workflow - backward

```
daniel@daniel: ~/Documents/Uni/phdproject/Other/Programmers-Club
File Edit View Search Terminal Help

To https://github.com/DanielAndreasen/Programmers-Club.git
a999b69..dele03e master -> master
→ Programmers-Club git:(master) git log
→ Programmers-Club git:(master) git checkout a999b6957738
Note: checking out 'a999b6957738'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

    git checkout -b new_branch_name

HEAD is now at a999b69... Added new file
→ Programmers-Club git:(a999b69) ls
LICENSE README.md test
→ Programmers-Club git:(a999b69) cat test
→ Programmers-Club git:(a999b69) git branch
* (detached from a999b69)
master
→ Programmers-Club git:(a999b69) 
```

Workflow - backward

```
daniel@daniel: ~/Documents/Uni/phdproject/Other/Programmers-Club
File Edit View Search Terminal Help

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

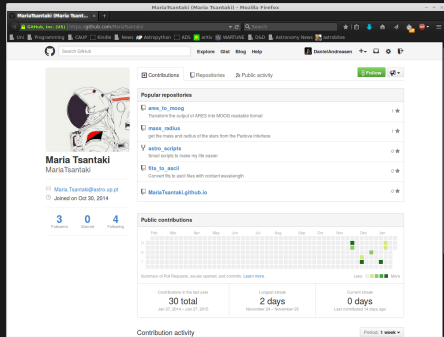
    git checkout -b new_branch_name

HEAD is now at a999b69... Added new file
→ Programmers-Club git:(a999b69) ls
LICENSE README.md test
→ Programmers-Club git:(a999b69) cat test

→ Programmers-Club git:(a999b69) git branch
* (detached from a999b69)
master
→ Programmers-Club git:(a999b69) git checkout master
Previous HEAD position was a999b69... Added new file
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.
→ Programmers-Club git:(master) cat test
This is some text
→ Programmers-Club git:(master)
```

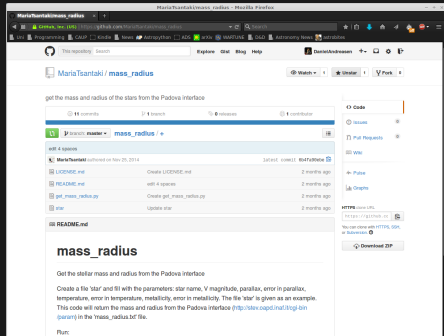
Getting other's code

- Follow people you like (social).



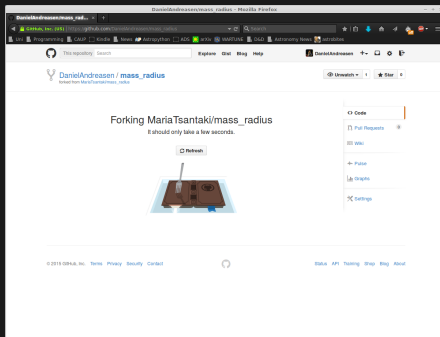
Getting other's code

- Follow people you like (social).
- Find the code you want to use.



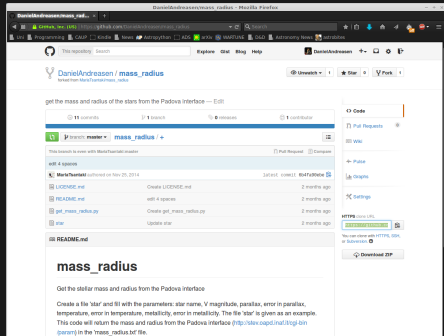
Getting other's code

- Follow people you like (social).
- Find the code you want to use.
- Fork the project.



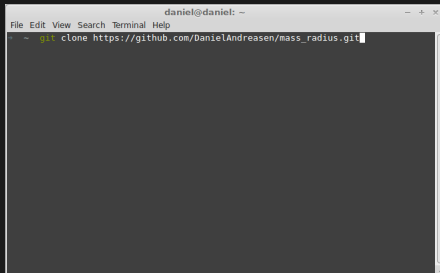
Getting other's code

- Follow people you like (social).
- Find the code you want to use.
- Fork the project.
- Clone to your local computer.



Getting other's code

- Follow people you like (social).
- Find the code you want to use.
- Fork the project.
- Clone to your local computer.

A terminal window titled 'daniel@daniel: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command 'git clone https://github.com/DanielAndreasen/mass_radius.git' is entered at the prompt. The terminal background is dark gray, and the text is white. The window has standard OS window controls (minimize, maximize, close) in the top right corner.

```
daniel@daniel: ~  
File Edit View Search Terminal Help  
~ git clone https://github.com/DanielAndreasen/mass_radius.git
```

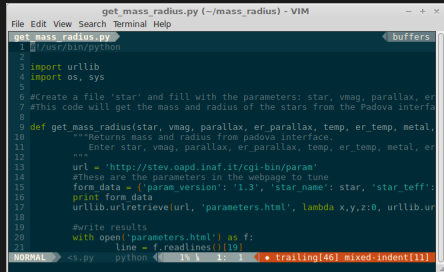
Branching

- New branch
(feature1)

```
daniel@daniel: ~/mass_radius
File Edit View Search Terminal Help
~ git clone https://github.com/DanielAndreasen/mass_radius.git
Cloning into 'mass_radius'...
remote: Counting objects: 33, done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 33 (delta 14), reused 33 (delta 14)
Unpacking objects: 100% (33/33), done.
Checking connectivity... done.
~ cd mass_radius
+ mass_radius git:(master) git branch
+ master
+ mass_radius git:(master) git branch feature1
+ mass_radius git:(master) git branch
+ feature1
+ master
+ mass_radius git:(master) git checkout feature1
Switched to branch 'feature1'
+ mass_radius git:(feature1) |
```

Branching

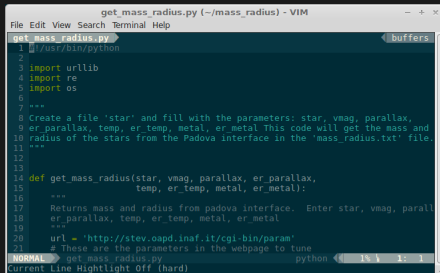
- New branch (feature1)
- Implement your changes



```
get_mass_radius.py (-/mass_radius) - VIM
File Edit View Search Terminal Help
get_mass_radius.py
1 #!/usr/bin/python
2
3 import urllib
4 import os, sys
5
6 #Create a file 'star' and fill with the parameters: star, vmag, parallax, er
7 #This code will get the mass and radius of the stars from the Padova interfa
8
9 def get_mass_radius(star, vmag, parallax, er_parallax, temp, er_temp, metal,
10                     """Returns mass and radius from padova interface.
11                     Enter star, vmag, parallax, er_parallax, temp, er_temp, metal, er
12                     """
13     url = 'http://stev.oapd.inaf.it/cgi-bin/param'
14     #These are the parameters in the webpage to tune
15     form_data = {'param_version': '1.3', 'star_name': star, 'star_teff':
16                 print form_data
17     urllib.urlretrieve(url, 'parameters.html', lambda x,y,z:0, urllib.ur
18
19     #write results
20     with open('parameters.html') as f:
21         line = f.readlines()[19]
22
23 NORMAL <5.py python <14 1 1: 1 <• trailing[46] mixed-indent[11]
```

Branching

- New branch (feature1)
- Implement your changes



```
get_mass_radius.py (-/mass_radius) - VIM
File Edit View Search Terminal Help
get_mass_radius.py buffers
1 11/usr/bin/python
2
3 import urllib
4 import re
5 import os
6
7 """
8 Create a file 'star' and fill with the parameters: star, vmag, parallax,
9 er_parallax, temp, er_temp, metal, er_metal This code will get the mass and
10 radius of the stars from the Padova interface in the 'mass_radius.txt' file.
11 """
12
13
14 def get_mass_radius(star, vmag, parallax, er_parallax,
15                     temp, er_temp, metal, er_metal):
16     """
17     Returns mass and radius from padova interface. Enter star, vmag, parall
18     er_parallax, temp, er_temp, metal, er_metal
19     """
20     url = 'http://stev.oapd.inaf.it/cgi-bin/param'
21     # These are the parameters in the webpage to tune
22     # get_mass_radius.py
23     python
```

Branching

- New branch (feature1)
- Implement your changes
- Push your branch

```
daniel@daniel: ~/mass_radius
File Edit View Search Terminal Help
* mass_radius git:(feature1) ls
get_mass_radius.py LICENSE.md README.md star
* mass_radius git:(feature1) vim get_mass_radius.py
* mass_radius git:(feature1) / git add get_mass_radius.py
* mass_radius git:(feature1) / git commit -m "More pythonic"
(feature1 e7d9f35) More pythonic
1 file changed, 80 insertions(+), 79 deletions(-)
rewrite get_mass_radius.py (75%)
* mass_radius git:(feature1) git push
fatal: The current branch feature1 has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin feature1

* mass_radius git:(feature1) git push --set-upstream origin feature1
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 726 bytes | 0 bytes/s, done.
Total 3 (delta 2), reused 0 (delta 0)
To https://github.com/DanielAndreasen/mass_radius.git
 * [new branch] feature1 -> feature1
Branch feature1 set up to track remote branch feature1 from origin.
* mass_radius git:(feature1)
```

Branching

- New branch (feature1)
- Implement your changes
- Push your branch
- Merge your branch with master

```
daniel@daniel: ~/mass_radius
File Edit View Search Terminal Help
- + x
- mass_radius git:(feature1) git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.
- mass_radius git:(master) git merge feature1
Updating 6b4fa90..e7d9f35
Fast-forward
 1 file changed, 54 insertions(+), 53 deletions(-)
- mass_radius git:(master) git
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
- mass_radius git:(master) git push
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/DanielAndreasen/mass_radius.git
 6b4fa90..e7d9f35 master -> master
- mass_radius git:(master) git status
On branch master
Your branch is up-to-date with 'origin/master'.

nothing to commit, working directory clean
- mass_radius git:(master)
```

Branching

- New branch (feature1)
- Implement your changes
- Push your branch
- Merge your branch with master
- Clean up

```

daniel@daniel: ~/mass_radius
File Edit View Search Terminal Help
* mass_radius git:(master) git branch
feature1
* master
* mass_radius git:(master) git push origin --delete feature1
To https://github.com/DanielAndreasen/mass_radius.git
- [deleted] feature1
* mass_radius git:(master) git branch
feature1
* master
* mass_radius git:(master) git branch -d feature1
Deleted branch feature1 (was 67d9f35).
* mass_radius git:(master) git branch
* master
* mass_radius git:(master)

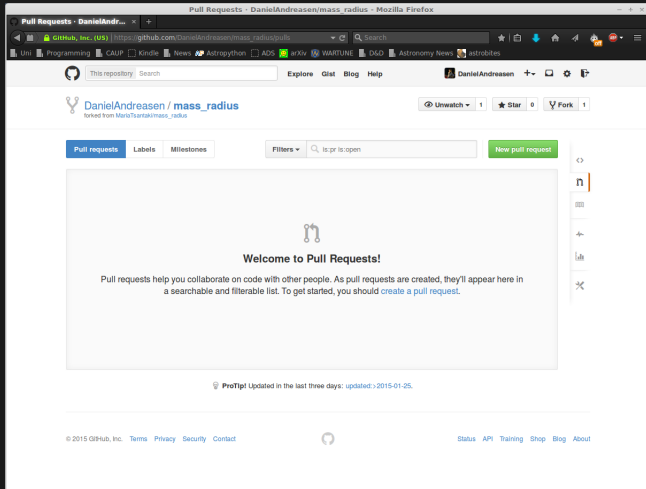
```

Pull request

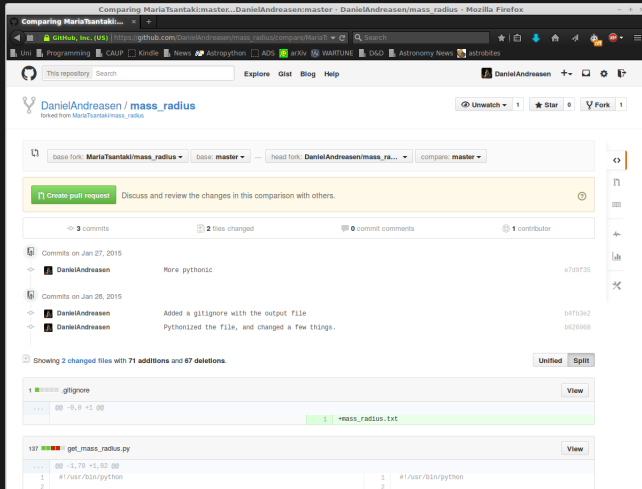
The screenshot shows a web browser window displaying a GitHub pull request for the repository `DanielAndreasen/mass_radius`. The page title is "get the mass and radius of the stars from the Padova interface — Edit". The repository has 14 commits, 1 branch, 0 releases, and 2 contributors. The pull request is titled "branch: master" and "mass_radius / +". The description states: "Pythonized the file, and changed a few things." The commit history shows a table with columns for the file, the change, and the time ago. The files listed are `.gitignore`, `LICENSE.md`, `README.md`, `get_mass_radius.py`, and `star`. The `README.md` file is expanded, showing the title "mass_radius" and the text: "Get the stellar mass and radius from the Padova interface. Create a file 'star' and fill with the parameters: star name, V magnitude, parallax, error in parallax, temperature, error in temperature, metallicity, error in metallicity. The file 'star' is given as an example. This code will return the mass and radius from the Padova interface (<http://stev.oapd.inaf.it/cgi-bin>).

File	Change	Time ago
<code>.gitignore</code>	Added a .gitignore with the output file	44 minutes ago
<code>LICENSE.md</code>	Create LICENSE.md	2 months ago
<code>README.md</code>	edit 4 spaces	2 months ago
<code>get_mass_radius.py</code>	Pythonized the file, and changed a few things.	43 minutes ago
<code>star</code>	Update star	2 months ago

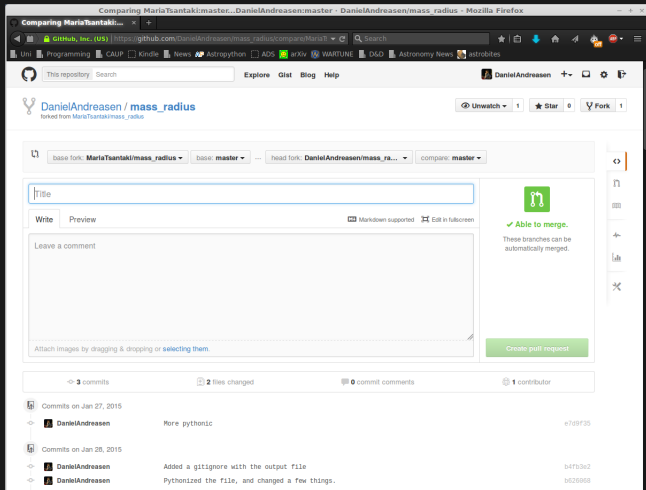
Pull request



Pull request



Pull request



Bonus slides: git cheat sheet

- **Most used commands (all starting with git)**

status: Show status on files in directory

add: Add <files> to be committed/updated

commit: Attach a <message> (what did I do and why) to the added files

push: Push the committed files to a server (GitHub/supernova/etc.)

log: Show a commit log (author, time stamp, message, and a key)

branch: Create a branch of the work with <name>. Good for adding new features.

checkout: Switch to a different branch (<name>) or commit (<key>).

blame: See who did what and when on a file.

clone: Get a repository locally with the link provided on GitHub.

pull: Get the latest updates from a code (if working on multiple computers or with collaborators).