

Introduction to



What is git?

- *Decentralised* version control system
- Easy to use
- Easy to use with external services, e.g. GitHub



What is git?

- *Decentralised* version control system
- Easy to use
- Easy to use with external services, e.g. GitHub

Linus Torvalds:

"I'm an egotistical bastard, and I name all my projects after myself. First 'Linux', now 'git'".



Definition of GIT



British

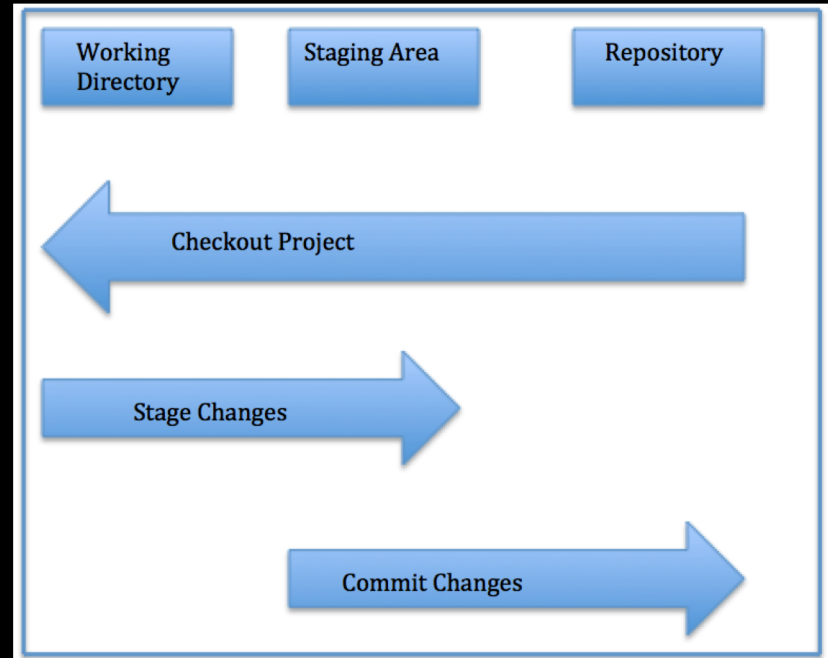
: a foolish or worthless person

Git workflow

- Tell git a file is interesting
- Tell git about the file
- Update git to include the new file

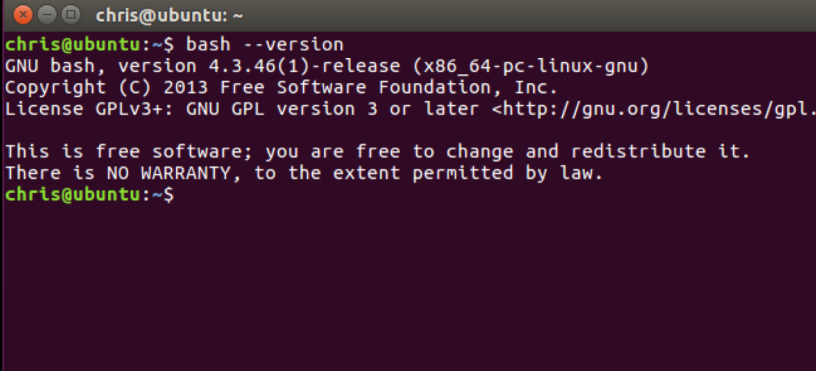
Git workflow

- Tell git a file is interesting
- Tell git about the file
- Update git to include the new file



Starting from scratch

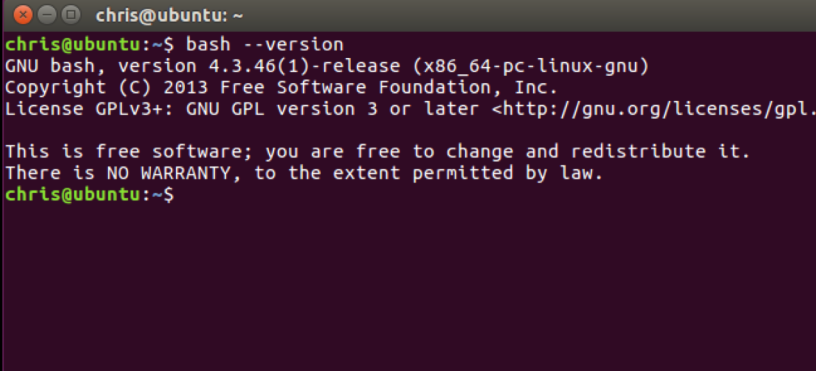
- \$ git init

A terminal window with a dark background and light text. The window title is 'chris@ubuntu: ~'. The prompt is 'chris@ubuntu:~\$'. The user has entered 'bash --version'. The output shows 'GNU bash, version 4.3.46(1)-release (x86_64-pc-linux-gnu)', copyright information for 2013, and the GPL license. It also includes a disclaimer: 'This is free software; you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.' The prompt returns to 'chris@ubuntu:~\$'.

```
chris@ubuntu: ~  
chris@ubuntu:~$ bash --version  
GNU bash, version 4.3.46(1)-release (x86_64-pc-linux-gnu)  
Copyright (C) 2013 Free Software Foundation, Inc.  
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.  
  
This is free software; you are free to change and redistribute it.  
There is NO WARRANTY, to the extent permitted by law.  
chris@ubuntu:~$
```

\$ git status

- git status is your friend!

A terminal window with a dark purple background and a grey title bar. The title bar contains window control icons and the text 'chris@ubuntu: ~'. The terminal shows the command 'bash --version' and its output: 'GNU bash, version 4.3.46(1)-release (x86_64-pc-linux-gnu)', 'Copyright (C) 2013 Free Software Foundation, Inc.', 'License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.', 'This is free software; you are free to change and redistribute it.', and 'There is NO WARRANTY, to the extent permitted by law.' The prompt 'chris@ubuntu:~\$' is visible at the bottom.

```
chris@ubuntu: ~  
chris@ubuntu:~$ bash --version  
GNU bash, version 4.3.46(1)-release (x86_64-pc-linux-gnu)  
Copyright (C) 2013 Free Software Foundation, Inc.  
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.  
  
This is free software; you are free to change and redistribute it.  
There is NO WARRANTY, to the extent permitted by law.  
chris@ubuntu:~$
```

Adding files

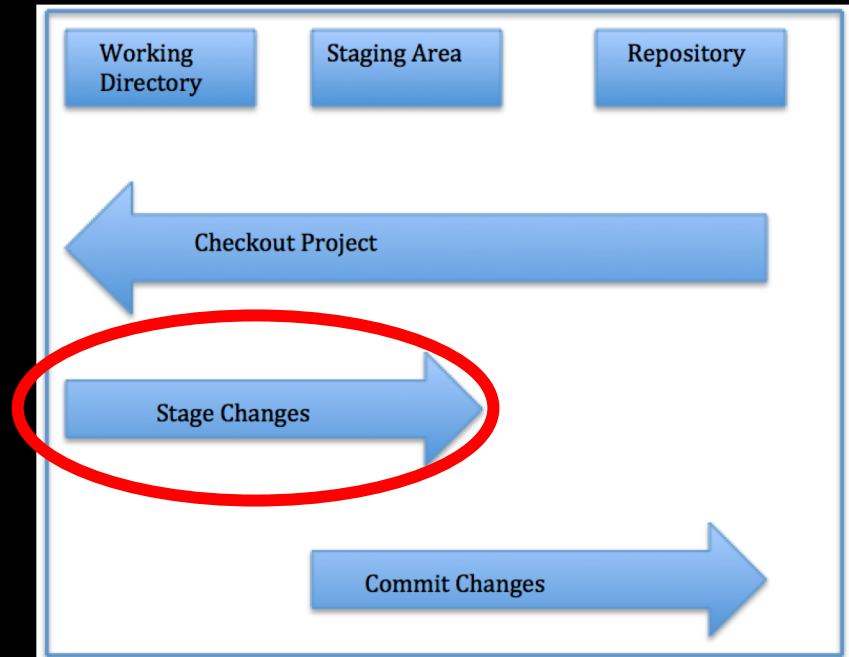
- `git add`

\$ Copy
simple_python_script.py to
your folder.

\$ `git status`

\$ `git add`
simple_python_script.py

\$ `git status`

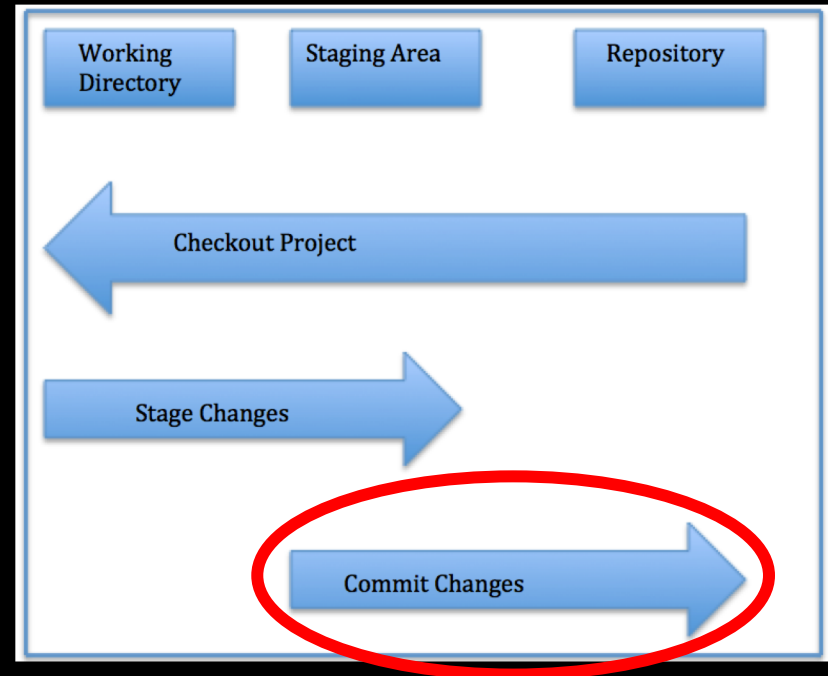


git commit

- git commit: a short description of what has changed.

\$ git commit -m "Added file"

\$ git status



What is a good commit message?

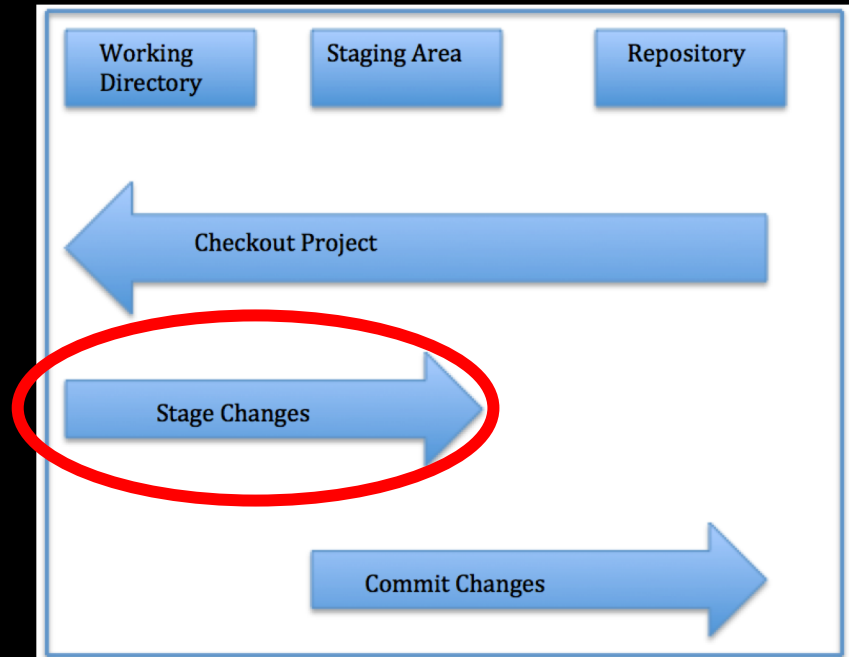
- Descriptive of the change, not the change itself, people can read the code.
- Blog on writing git commit message:
<https://chris.beams.io/posts/git-commit/>

	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
○	AAAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO
AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.		

<https://xkcd.com/1296/>

git log: show the history of the git repo

\$ git log

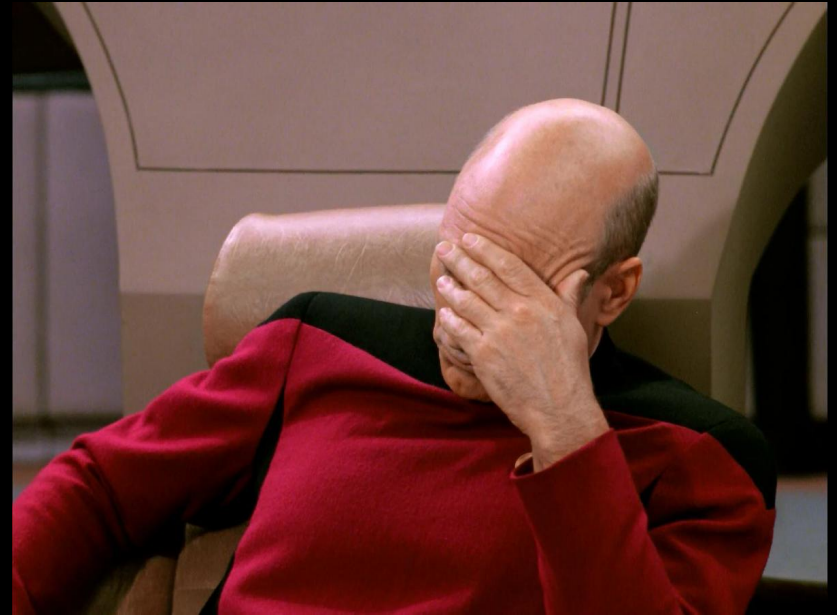


Correct wrong import name in python script.

- Numpy is imported as "nmp" not "np" as it should be.

\$ Use an editor and change
nmp to np in line 9

\$ git status



git diff: what has changed?

- git diff will show the difference(s) between the latest committed file(s) and the changed file(s)

\$ git diff

Change print statement and commit file

- The python file in line 15 does not have "()" around "random_numbers" change that.
- Add, commit etc

Change the number of random numbers generated with the "numbers" parameter

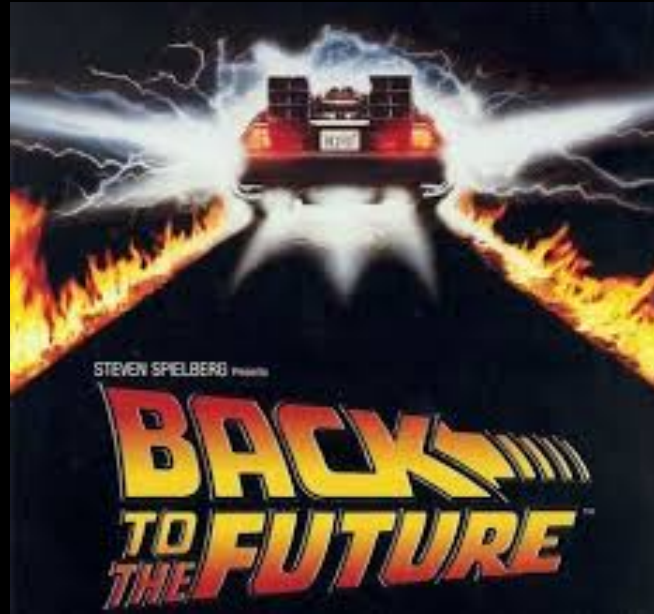
- The parameter "numbers" line 11, set the number of random numbers generated. Change that to a new number.
- Add, commit etc

Revert a change

- git tracks the changes for each "commit"
- Revert to a previous stage of file

```
$ git log --oneline
```

```
$ git checkout 4b095cb  
simple_python_script.py
```



Whitespaces: yes they matter!

- Whitespaces are tracked by git!
- **Definition:**
white space is any character or series of characters that represent horizontal or vertical space in typography

Whitespaces: yes they matter!

- Whitespace are tracked by git!
- In line 17 there are trailing whitespaces, remove them

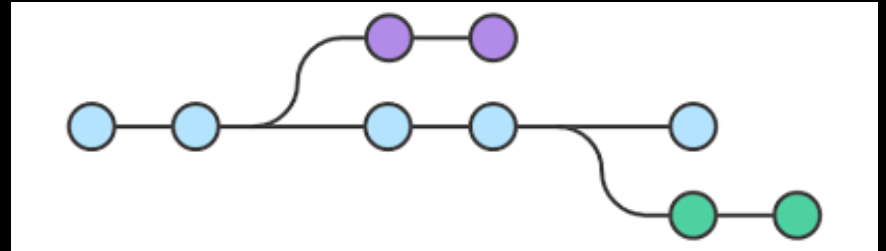
\$ git status

\$ git diff

- **Definition:**
white space is any character or series of characters that represent horizontal or vertical space in typography

Branches

- Branches are one of git's strengths.
- Branching means you diverge from the main line of development and continue to do work without messing with that main line git status



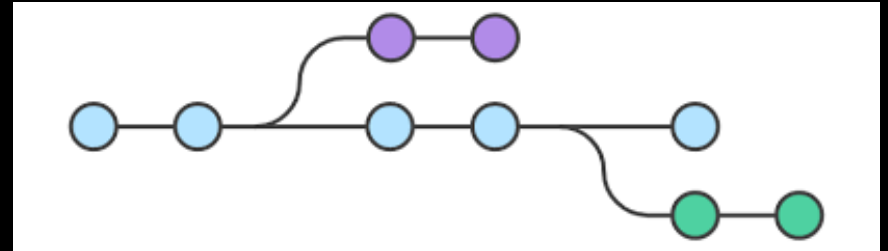
Branches

- Branches are one of git's strengths.
- Branching means you diverge from the main line of development and continue to do work without messing with that main line git status

Make a new branch

```
$ git checkout -B n_100
```

```
$ git branch
```



- "master" is special!

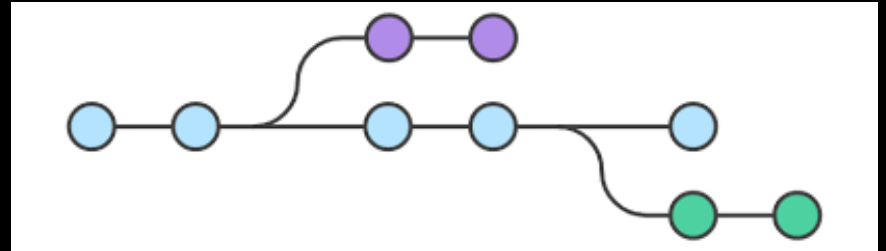
Switching between branches

```
$ git checkout n_100
```

```
$ git status
```

```
$ git checkout master
```

```
$ git status
```



Switching between branches: files

```
$ git checkout n_100  
  simple_python_script.py
```

```
$ git status
```

```
$ git merge n_100
```

```
$ git status
```

