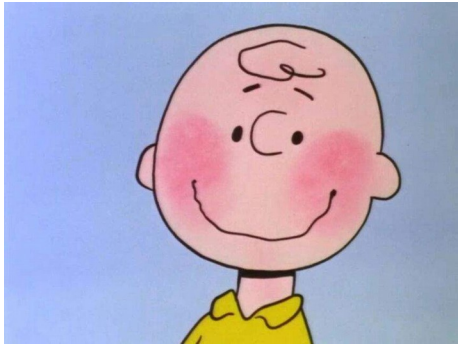
Git Lost

A roadmap for the overcommitted

git commit <> save

- Remember the last time you wrote your cv?
  - Final
  - CorrectedFinal
  - FinalCorrectedFinal
  - ThisTimeThereAreNoTyposFinal

- Which version did you email to that last recruiter?
  - My CV



look familiar…?

https://xkcd.com/1296/

but what do I do when I mess up?

- “I don’t want to lose hours of work by only committing once everything’s working!”

- Absolutely

  git commit as often as you like

  just don’t PUSH until it’s working and your map is drawn

- You draw the map once you reach your destination

Part 1: Safety Net

- What did things look like 3 commits ago?

  git checkout HEAD~3

- Can I try a different approach from back there?

  git checkout -b new-branch HEAD~5

- Oops. Can I keep my changes but undo the commits?

  git reset SHA

- Everything I’ve done is a disaster. Make it go away!

  git reset --hard

Part 2: Map Reading

- How did we get here?

  git log --oneline

- Which files changed in each commit?

  git log --name-only --oneline

- Give me that file!

  git checkout SHA full/path/to/file.ext

Power Search: The Pick Axe

- ***git log -S "thing" --patch --reverse***
- any commit which added or removed the string “thing”
- reverse order means oldest first so you know when it was first created
- use something like a method name which is more likely to be unique
- this won’t work to search for a file name
  - Use ***git log --name-only –oneline directory/to/search***

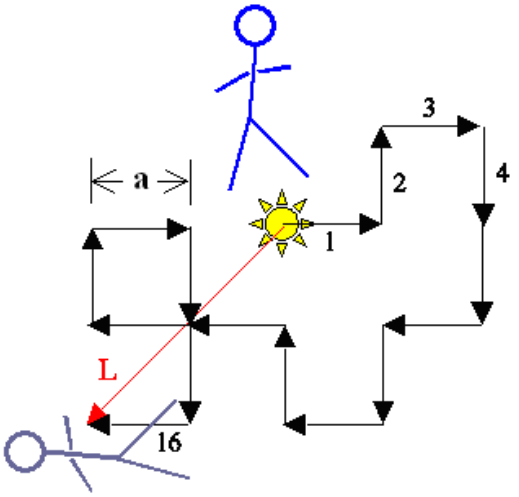
Part 3: Map Making

- Your readers don’t need to know how you got there
- They need to know

  what the outcome was

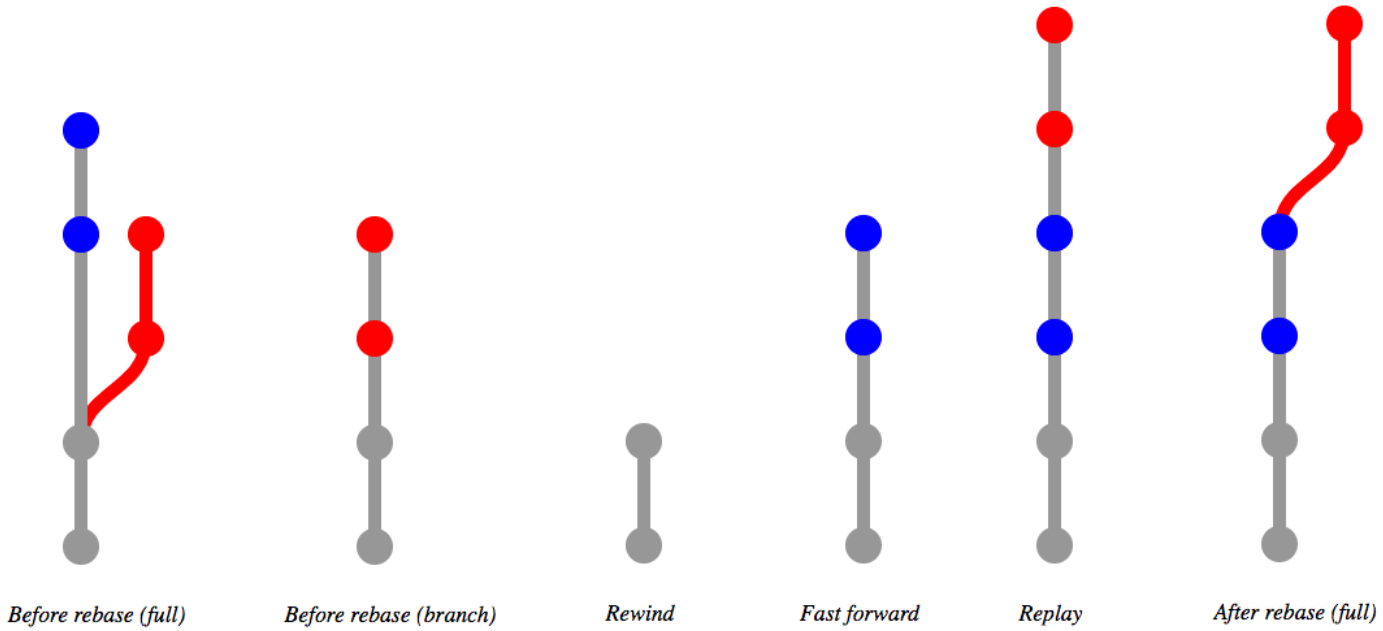
  why you did it
- Enter ***git rebase -i***

  (and ***git rebase --abort*** when it all goes wrong)



git rebase (here be dragons)

- git rebase creates new commits on the destination branch



https://jason.pureconcepts.net/2017/02/git-rebase/

git rebase -i can change anything

git log

```
2018-03-09 aaa984a Weather card complete

2018-03-09 d96e190 flag alt message appears

2018-03-09 4ffa7d8 Weather method extracted into own file

2018-03-09 28a4764 fixed selector bug

2018-03-09 3735d22 API key moved to separate file
```

rebase started

git rebase -i 4ffa7d8~

```
1 git-rebase-todo
pick 4ffa7d8 Weather method extracted into own file
pick 28a4764 fixed selector bug
pick aaa984a Weather card complete
pick d96e190 flag alt message appears

# Rebase e8c65fc..43e1d60 onto e8c65fc (4 commands)
```

you even get instructions

```
# Commands:
 # p, pick = use commit
 # r, reword = use commit, but edit the commit message
 # e, edit = use commit, but stop for amending
 # s, squash = use commit, but meld into previous commit
 # f, fixup = like "squash", but discard this commit's log message
 # x, exec = run command (the rest of the line) using shell
 # d, drop = remove commit
```

decide what to do

```
1 git-rebase-todo +
reword 4ffa7d8 Weather method extracted into own file
fixup aaa984a Weather card complete
edit 28a4764 fixed selector bug
reword d96e190 flag alt message appears

# Rebase e8c65fc..43e1d60 onto e8c65fc (4 commands)
```

edit the commit message only

```
1 COMMIT_EDITMSG
Weather method extracted into own file

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
```

```
1 COMMIT_EDITMSG +
Weather Card

- extract method into separate file
- refactor to display images as well as text
```

edit everything

```
Stopped at 28a4764...  fixed selector bug
You can amend the commit now, with

  git commit --amend

Once you are satisfied with your changes, run

  git rebase --continue
```

git tools work within rebase

git diff head~

```
--- a/client/src/views/countries_select_view.js
+++ b/client/src/views/countries_select_view.js
@@ -5,14 +5,13 @@ const CountrySelectView = function(s
   this.filteredCountries = [];
   this.selectElement.addEventListener("change", funct
     const target = e.target;
-    const index = target.selectedIndex;
+    const index = target.selectedIndex - 1;
     const country = this.countries[index];
```

keep going

git commit --amend

```
1 COMMIT_EDITMSG
fixed selector bug

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
```

```
1 COMMIT_EDITMSG +
Country Select View

- country select list is numbered from 1
- country data is zero indexed
- onChange method now passes selectedIndex-1
```

git rebase --continue

merge conflict!

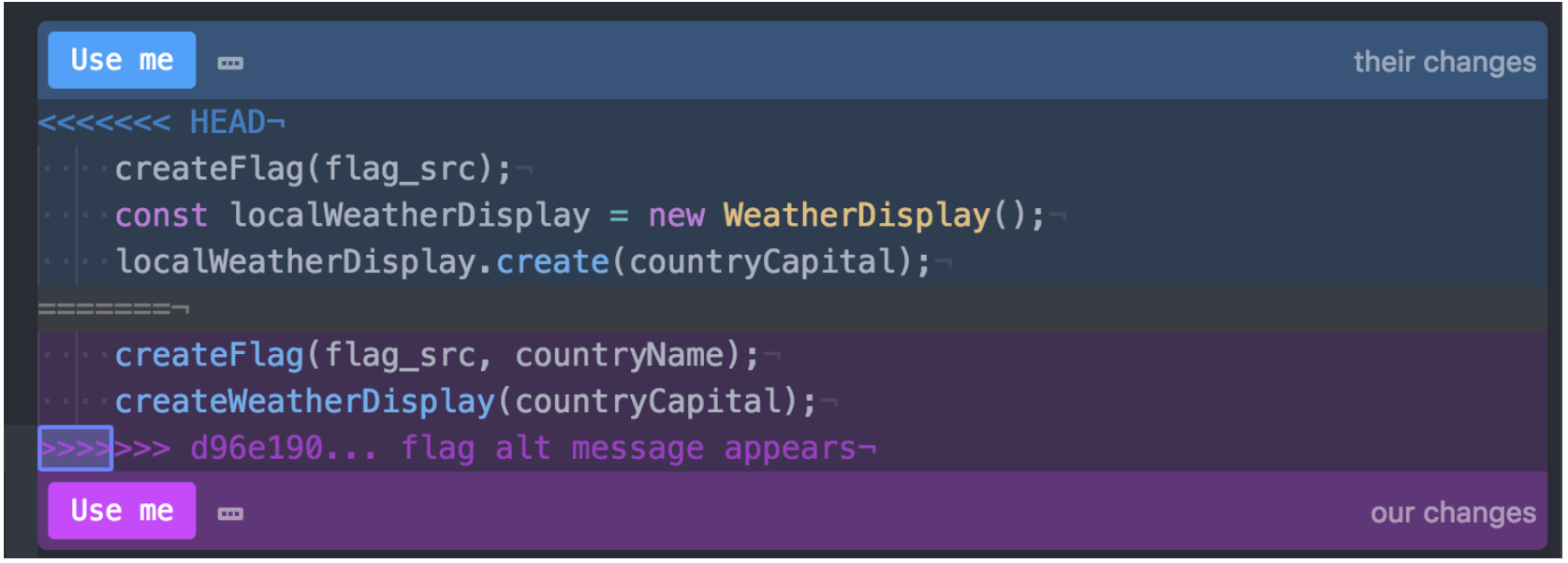
```
Auto-merging client/src/app.js
CONFLICT (content): Merge conflict in client/src/app.js
error: could not apply d96e190... flag alt message appears

Resolve all conflicts manually, mark them as resolved with
"git add/rm <conflicted_files>", then run "git rebase --continue".
```

git status

```
both modified:   client/src/app.js
```

so fix it

atom client/src/app.js



save the file and close atom again

and carry on merrily

git add client/src/app.js

git status

```
modified:   client/src/app.js
```

git rebase --continue

```
1 COMMIT_EDITMSG
flag alt message appears
```

```
1 COMMIT_EDITMSG +
Country Flag alt image text is country name
```

all done

```
Successfully rebased and updated refs/heads/demo.
```

git log

```
2018-03-09 183e638 Country Flag alt image text is country name

2018-03-09 fd190df Weather Card
- extract method into separate file
- refactor to display images as well as text

2018-03-09 3735d22 API key moved to separate file
```

Commit Messages

- Be consistent
- Be concise
- Be clear

Title: AccountName : Purpose/Result of Change (50 characters)

<blank line> (essential)

Body: Explanation in as many or as few words as you need (optional)

Wrap at ~72 characters

Put things you would normally add in code comments here

Consider adding any tracking reference for context

https://chris.beams.io/posts/git-commit/

In Summary

- Learn how to read a map before you’re lost
- Treat git commits as a safety net first
- Rewrite history to provide a clear routemap

    git rebase -i