This is not the git tutorial you are looking for

Git According to Merriam - Webster

- Git: noun; a person who lacks good sense or judgment
- Synonyms
  - Doofus
  - Dingbat
  - Fool
  - Simpleton
  - …

Prerequisites: Hashing

- Hashing: compress lots of bits into a small number of bits
- (Bad) example: add up ascii codes of string and take modulo

"Hello World!"
72+101+108+108+111+32+87+111+114+108+100+33=1085
mod 10 => 5

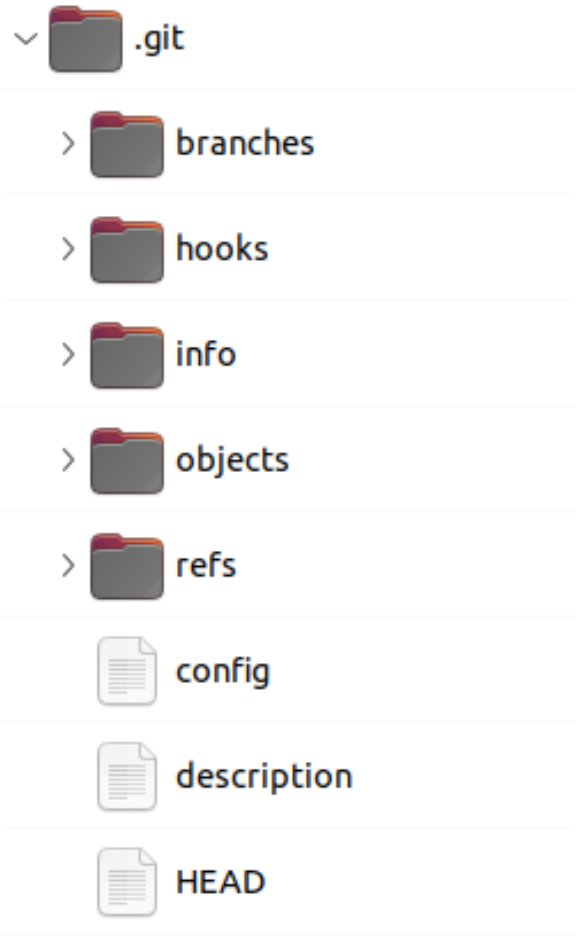
"Kilroy was here."
75+105+108+114+111+121+32+119+97+115+32+104+101+114+101+46=1495
mod 10 => 5

Create empty git repo

mkdir demo

cd demo

git init



Add File to git

echo "Read Me" > README

git add README

git hash-object README

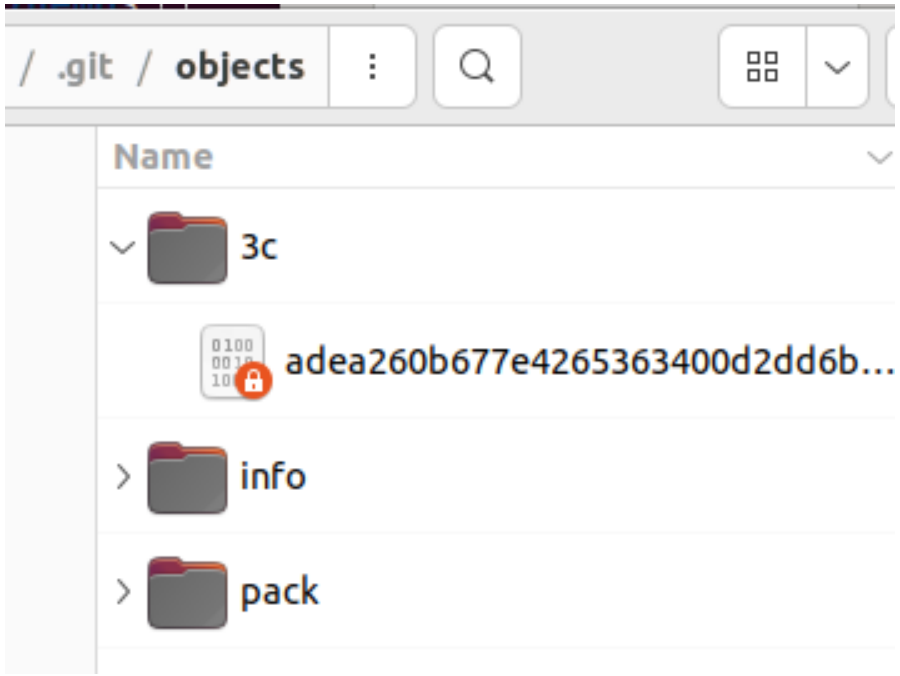
=> 3cadea260...

git cat-file -t 3cadea260

=> blob

git cat-file -p 3cadea260

=> Read Me

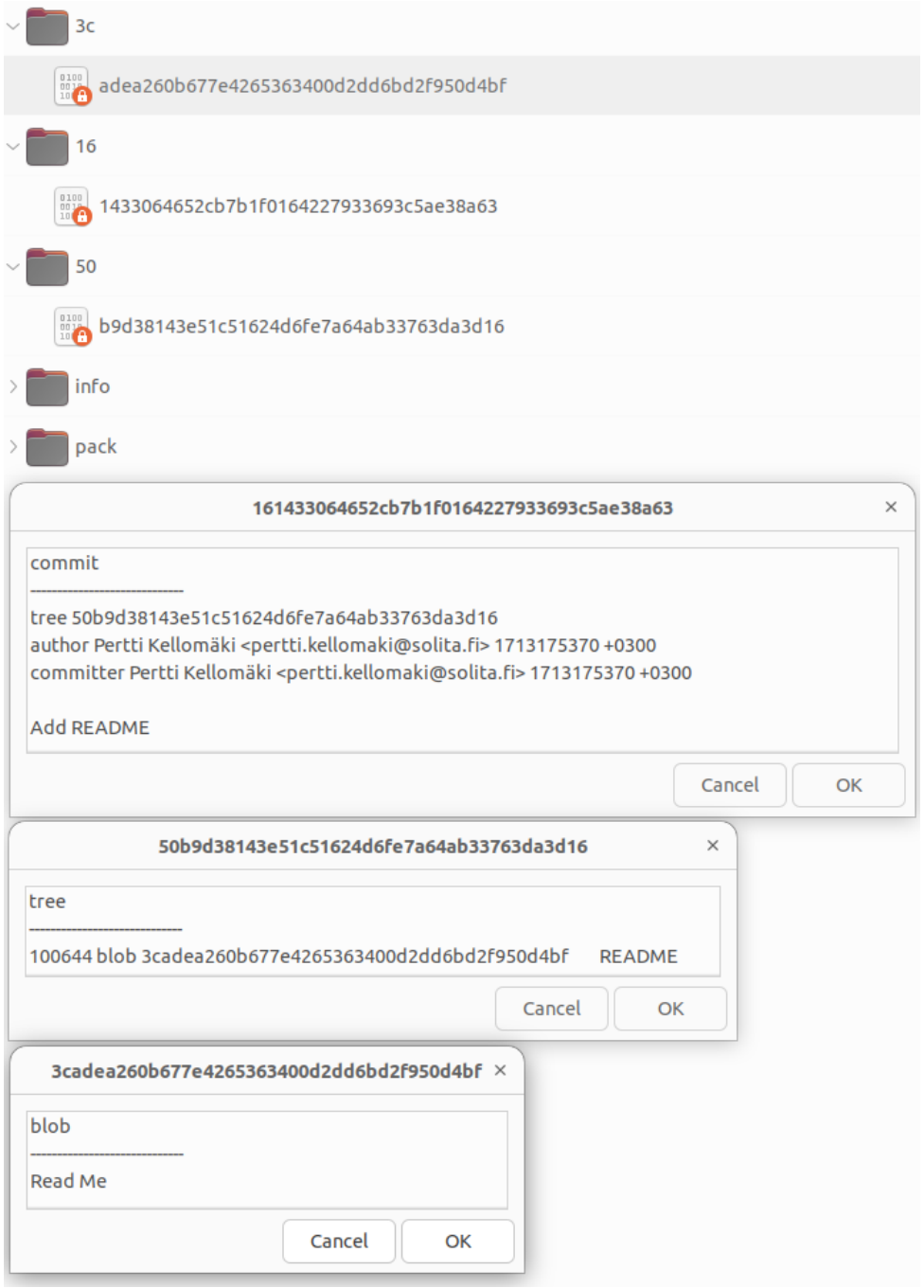


Now you understand ~50% of git

Commit

git commit –m "Add README"

=> 1614330



Now you understand ~75% of git

Subdirectory

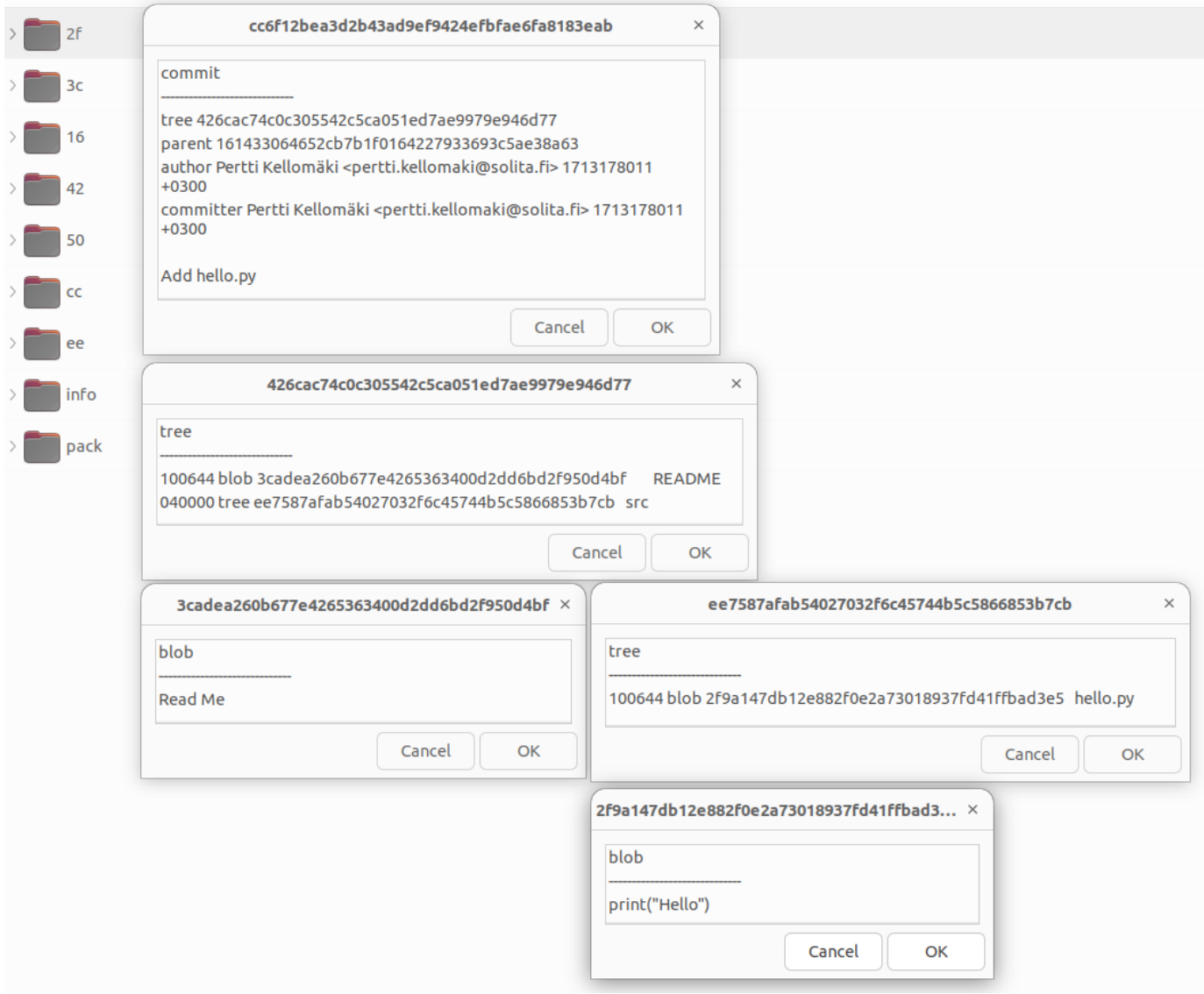
mkdir src

echo 'print("Hello")' > src/hello.py

git add src/hello.py

git commit -m "Add hello.py"

=> cc6f12b



Now you understand ~80% of git

Unchanged subdirectory

echo "Read more" >> README

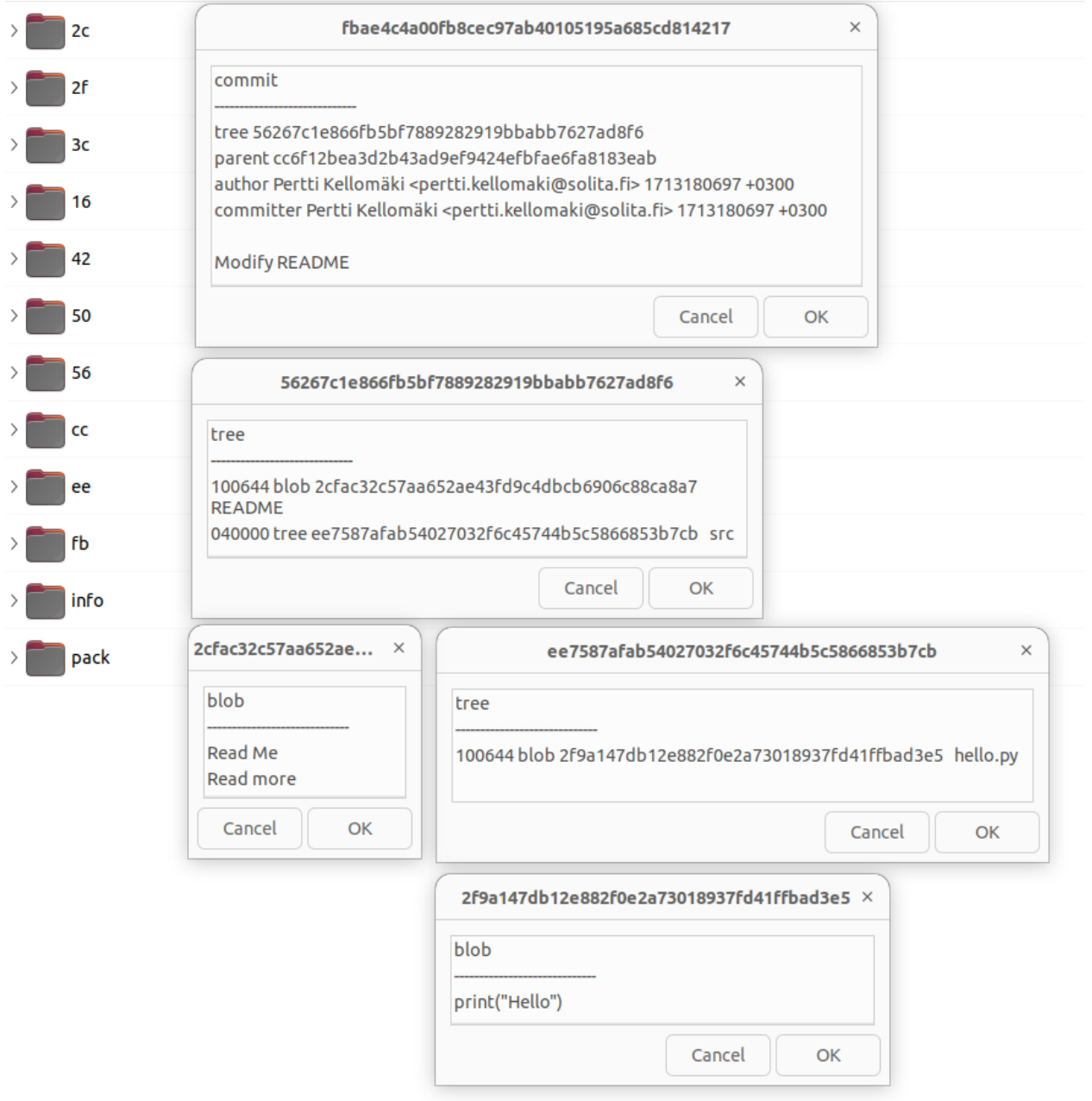
git add README

git hash-object README

=> 2cfac32c57aa652...

git commit -m "Modify README"

=> fbae4c4



Now you understand ~90% of git

Branches

git checkout -b mybranch

echo "My Branch" >> README

git add README

git commit -m "Modify README again"

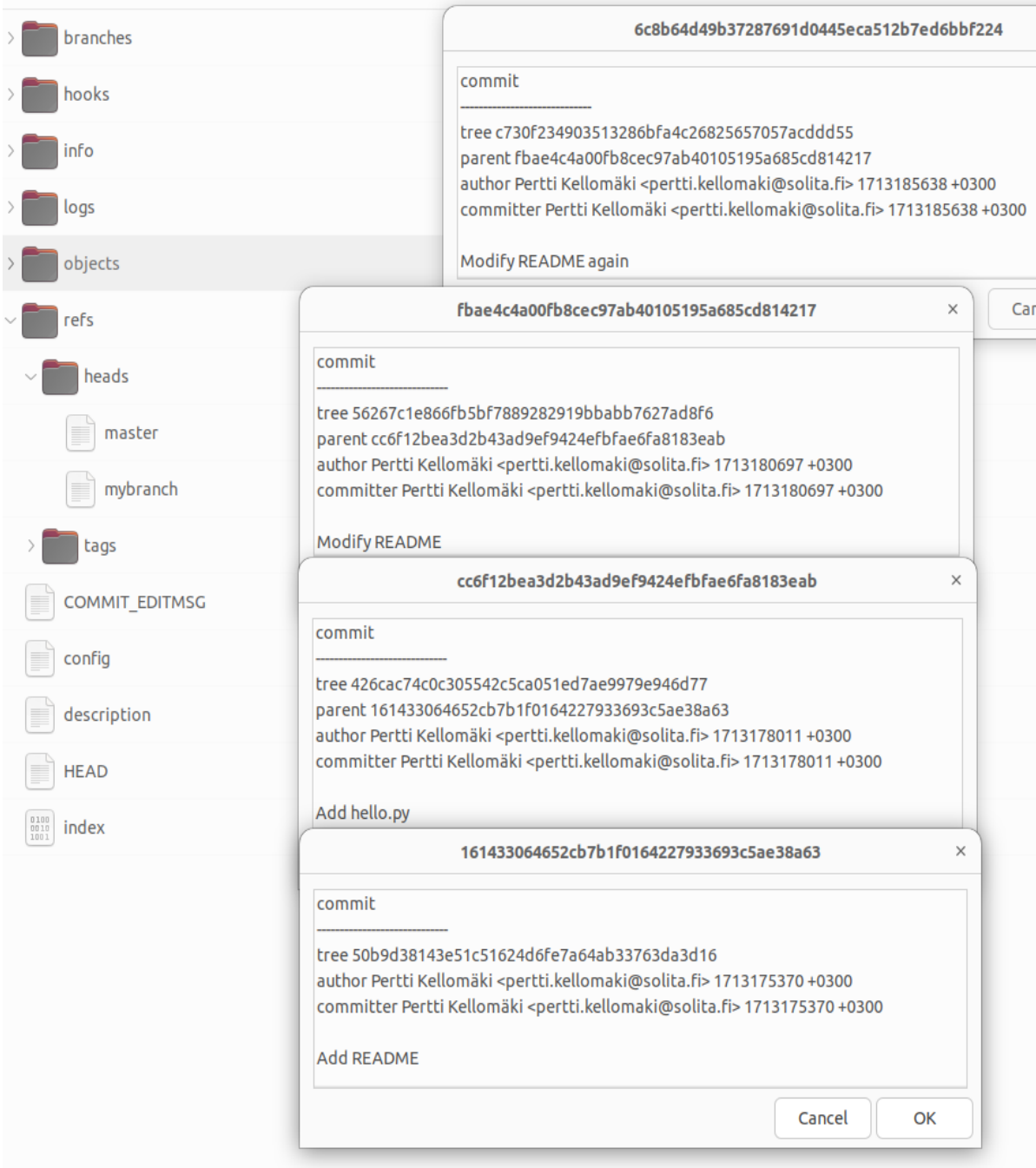
=> 6c8b64d

cat .git/refs/heads/master

=> fbae4c4a00fb8ce...

cat .git/refs/heads/mybranch

=> 6c8b64d49b372...



Now you understand ~95% of git

Git diff

- Takes two commits
- Simply compares the trees attached to the commits

```
$ git diff master..mybranch
diff --git a/README b/README
index 2cfac32..9f68d48 100644
--- a/README
+++ b/README
@@ -1,2 +1,3 @@
Read Me
Read more
+My Branch
```

Git format-patch (prequel to rebase)

- Takes two connected commits
- Creates diffs for each commit in between

```
$ git format-patch master..mybranch
0001-Modify-README-again.patch
$ cat 0001-Modify-README-again.patch
...
diff --git a/README b/README
index 2cfac32..9f68d48 100644
--- a/README
+++ b/README
@@ -1,2 +1,3 @@
 Read Me
 Read more
+My Branch
```

Git rebase

- Simple case: rebase **new-branch** to include new commits in **base-branch**
- Create diffs for each of the commits created in **new-branch** after branching
- Start with **base-branch**, and apply each diff in succession
- Update the **new-branch** ref to point to the last commit
- In reality somewhat more clever, takes into consideration where new-branch is before rebase and does 3-way merge (I think).

Git merge

- Merge branch-A into branch-B
- Each branch identifies a commit => merge trees attached to commits
- Find commit C where A and B diverge
- Perform 3-way merge for each file in tree
  - If line is the same in A and B, pick line
  - If lines are different, look in C
    - If line changed in only A or B, pick the changed line
    - Otherwise pick both and create a merge conflict

Rest of git is trivial ;-)