30538 Problem Set 3: git Solution

Clarice Tee

2024-10-26

SOLO

Learn git branching (15 points)

Go to https://learngitbranching.js.org. This is the best visual git explainer we know of.

1. Complete all the levels of main "Introduction Sequence". Report the commands needed to complete "Git rebase" with one line per command.

```
git checkout -b bugFix
git commit
git checkout main
git commit
git checkout bugFix
git rebase main
```

2. Complete all the levels of main "Ramping up". Report the commands needed to complete "Reversing changes in git" with one line per command.

```
$ git reset HEAD~1

$ git checkout pushed

$ git revert HEAD
```

3. Complete all the levels of remote "Push & Pull – Git Remotes!". Report the commands needed to complete "Locked Main" with one line per command.

```
git reset --hard o/main

git checkout -b feature C2

git push origin feature
```

Exercises

- Basic Staging and Branching (10-15)

1. Exercise. For your pset submission, tell us only the answer to the last question (22).

```
On branch master
nothing to commit, working tree clean
```

2. Exercise. For your pset submission, tell us only the output to the last question (18).

```
$ git diff mybranch master
diff --git a/file1.txt b/file1.txt
deleted file mode 100644
index be58370..0000000
--- a/file1.txt
+++ /dev/null
@@ -1 +0,0 @@
-Clarice
diff --git a/file2.txt b/file2.txt
new file mode 100644
index 0000000..3b94f91
--- /dev/null
+++ b/file2.txt
@@ -0,0 +1 @@
+Temp
```

- Merging

1. Exercise. After completing all the steps (1 through 12), run git log –oneline –graph –all and report the output.

```
$ git log --oneline --graph --all
* 2610a7c (HEAD -> master, feature/uppercase) Changed greeting to uppercase
* 5a69c73 Add content to greeting.txt
* f034c3b Add file greeting.txt
```

2. Exercise. Report the answer to step 11.

```
c5d8a69 (HEAD -> master) Merge branch 'greeting'
|\
| * 06a310d (greeting) Update greeting with my favorite greeting
* | e425133 added readme info
|/
* 2c2819d Add content to greeting.txt
* c2795ae Add file greeting.txt
```

3. Identify the type of merge used in Q1 and Q2 of this exercise. In words, explain the difference between the two merge types, and describe scenarios where each type would be most appropriate.

I think that the Q1 merge is simpler and is best used when merging a branch that was created directly from the branch we are trying to integrate. I think that three-way merges are more widely used in larger group projects, while fast-forward merges are more common when updating code or addressing coding errors. In contrast, Q2 is most effective when we need to merge two distinct branches that have a shared last commit.

- Undo, Clean, and Ignore

1. Exercise. Report the answer to step 13.

```
$ git show
commit 9da7ef3dd19bfe5b7efb0aaadcecce2ec1791382 (HEAD -> master)
Author: git-katas trainer bot <git-katas@example.com>
Date:   Thu Oct 24 18:27:03 2024 -0500

    Revert "Add credentials to repository"

    This reverts commit ac75cc54a25f8651aa3e6f4faecddcc79fd9c662.

diff --git a/credentials.txt b/credentials.txt
deleted file mode 100644
index 8995708..0000000
--- a/credentials.txt
+++ /dev/null
@@ -1 +0,0 @@
-supersecretpassword
```

2. Exercise. Look up `git clean` since we haven't seen this before. For context, this example is about cleaning up compiled C code, but the same set of issues apply to random files

generated by knitting a document or by compiling in Python. Report the terminal output from step 7.

```
PS C:\Users\clari\git-katas\basic-cleaning\exercise> git clean -f -d
Removing README.txt~
Removing obj/
Removing src/myapp.c~
Removing src/oldfile.c~
PS C:\Users\clari\git-katas\basic-cleaning\exercise> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   src/mylib.c
```

3. Exercise. Report the answer to 15 ("What does git status say?")

```
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    file1.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .gitignore

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file1.txt
        file2.txt
        file3.txt
        foo.s
```