30538 Problem Set 3: git Solution

Peter Ganong, Maggie Shi, and Dema Therese Maria

2024-10-21

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

- Set Up

```
$ git clone https://github.com/eficode-academy/git-katas.git
$ cd basic-staging
$ source setup.sh
```

- 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).

```
diff --git a/file.txt b/file2.txt
similarity index 100%
rename from file.txt
rename to file2.txt
```

- Merging

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

```
* 96e5a9f (HEAD -> master, feature/uppercase) test_commit_message
* 89cd95d Add content to greeting.txt
* fa150e7 Add file greeting.txt
```

2. Exercise. Report the answer to step 11.

```
*   beb175a (HEAD -> master) Merge branch 'greeting'
|\
| * 8fdb0be (greeting) test_commit_message
* | c7bc1b5 Adding README
|/
* 7de71f9 Add content to greeting.txt
* 09f3ecc 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.

Ans1 Merging is of Fast-Forward Merge. Branch pointer moved forward without creating a new commit. Used when the branch history is linear and no additional commits have been made on the base branch since branching.

Ans2 Merging is of 3-Way Merge. Git reconciles divergent changes from both branches and creates a new commit of combined changes. Used when both branches have made new commits since they diverged from a common ancestor.

- Undo, Clean, and Ignore

1. Exercise. Report the answer to step 13.

```
commit 716b7d3e81c69cfae0989ceff55735f40e0c2d8a (HEAD -> master)
Author: deetherese <theresemaria98@gmail.com>
Date:   Tue Oct 15 23:28:49 2024 -0500

    Revert "Add credentials to repository"

    This reverts commit 7240514d3c6f74e2b1e2312923dcb6e67dd09001.

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.

```
Removing README.txt~
Removing obj/
Removing src/myapp.c~
Removing src/oldfile.c~
```

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

```
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .gitignore
        new file:   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)
        file3.txt
        foo.s
```

Partnered

Expected final ps3_pair.qmd

```
import pandas as pd
import numpy as np
import altair as alt

np.random.seed(42)
data = pd.DataFrame(np.random.rand(100), columns=['Random Numbers'])

def preview_data(df):
    head_output = df.head()
    describe_output = df.describe()
    histogram_plot = alt.Chart(df).mark_bar().encode(
        alt.X('Random Numbers:Q', bin=True),
        alt.Y('count():Q')
    )

    # Return a list with the three outputs
    return [head_output, describe_output, histogram_plot]
```

```
preview_data(data)
```

```
[   Random Numbers
 0        0.374540
 1        0.950714
 2        0.731994
 3        0.598658
 4        0.156019,
        Random Numbers
 count      100.000000
 mean         0.470181
 std          0.297489
 min          0.005522
 25%          0.193201
 50%          0.464142
 75%          0.730203
 max          0.986887,
 alt.Chart(...)]
```