git commit -a

A shortcut to stage any changes to tracked files and commit them in one step

Question

If we're making a small change and want to skip the staging step, which two flags do we need to add to the git commit command? Check all that apply.

- ☑ *-m*

  ✓ **Correct**

  Right on! The -m flag allows us to directly add the commit message to the command.

- ☐ *-t*

- ☐ *-l*

- ☑ *-a*

  ✓ **Correct**

  Awesome! The -a flag lets us add and commit in the same step.

Skip   **Continue**

Git uses the HEAD alias to represent the currently checked-out snapshot of your project.

Question

If we want to see a specific commit, which command would we use along with the commit ID?

○ git log --stat

◉ git show

○ git log -p

○ git commit -am

✓ Correct

Question

If we need to delete a file from our repository, we'll need to run a command to delete the file and then stage and commit the change. Which command would we use to delete the file?

◉ *git rm*

○ *git mv*

○ *git diff*

○ *git del*

✓ Correct

Question

What is the purpose of the git checkout command?

○ It finalizes staged changes.

◉ It reverts changes to modified files before they are staged.

○ It skips staging and directly commits.

○ It displays the current status of the commit.

✓ Correct

Question

What does the **_git commit --amend_** do?

○ Add an error log to the commit.

○ Remove files from the staging area.

○ Change the commit message.

◉ Overwrite the previous commit.

✓ Correct

Avoid amending commits that have already been made public.

Question

Which of the following is true about the *git revert* command?

○ It undoes a commit as though it never happened.

◉ It creates a new commit with inverse changes.

○ The output of *git revert* is not the same as a regular commit.

○ It does not include the ID of the commit that was reverted.

✓ Correct

Question

Which of the following is NOT true about the SHA1 hash numbers that Git uses to identify commits?

○ They provide the consistency that is critical for distributed systems such as Git.

○ They are created using the commit message, date, author, and the snapshot taken of the working tree.

◉ Git requires the entire hash ID to identify a commit.

○ They are composed of 40 characters.

✓ Correct

Branch

A pointer to a particular commit

The default branch that git creates for you when a new repository is initialized is called **master.**

Question

What is the purpose of organizing repositories into branches?

- ⦿ To enable changes to be worked on without disrupting the most current working state.
- ○ To make it easier to undo commits.
- ○ To enable changes to the repository to permanently replace previous commits.
- ○ To give users a place to keep notes.

✓ **Correct**

We can use the git branch command to list, create, delete, and manipulate branches.

Question

How does *git checkout* switch branches?

○ By creating a new commit on a new branch.

◉ By updating the working tree to match the selected branch.

○ By moving the HEAD to the previous commit.

○ By amending the commit with the provided ID.

✓ **Correct**

Merging

The term that Git uses for combining branched data and history together
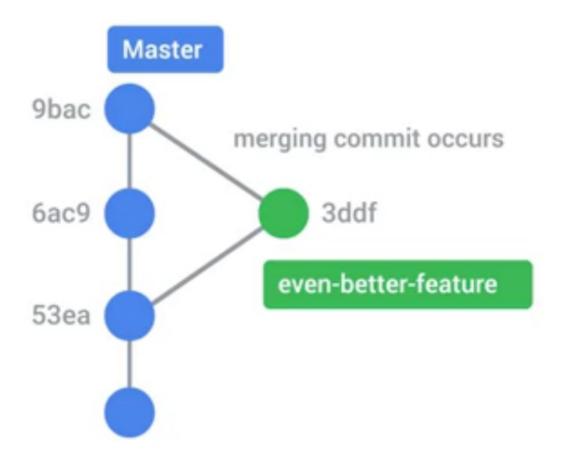
Question

What happens when we merge two branches?

○ The HEAD points at the master branch.

◉ Both branches are pointed at the same commit.

○ One of the former branches disappears.

○ Two independent snapshots will now share the same name.

✓ **Correct**

Git uses two different algorithms to perform a merge: fast-forward and three-way merge.

Git Fast forward Merge:

Git Three way Merge:

Question

What's the advantage of Git throwing a merge conflict error in cases of overlap?

- ⦿ It prevents loss of work if two lines overlap.
- ○ It helps us understand which changes to keep.
- ○ It warns us of all potential problems.
- ○ It tells us whether the commit is a merge

✓ **Correct**