1. Why use Git?

To provide “training wheels” for complex coding concepts

To test proper syntax in your code

To secure a codebase against hackers

To keep track of changes made to a project over time

1. What does “git init” do

Add all files to the staging area

Initialize a new Git project

Commit your files to the repository

Delete a Git project

1. Where in Git do you typically create, edit, delete, and organize project files?

The .gitignore directory

The initialize folder

The staging area

The working directory

1. What’s wrong with the code below?

**git commit -m Add new scene to screenplay**

The commit message lacks quotation marks

The commit message should be in all caps

The “-m” option goes before the word “commit”

The “-m” option is not necessary here

1. In Git, a commit

Records only code with correct syntax

Saves every file in your working directory

Temporarily stores changes from the staging area in the repository

Permanently stores changes from the staging area in the repository

1. What is the purpose of Git’s staging area?

To list notes about your project

To list files you don’t want Git to track

To stage file changes for a commit

To show a list of your project’s commits

1. In the code below, what will you replace “filename” with?

git add filename

None of these

The name of the Git repository

The file you wish to remove from the working directory

The file you wish to add to the staging area

1. The output below is typical of which command?

**commit bda95786432d142bbff996ad32045fa4f32ec619**

**Author: codecademy <ccuser@codecademy.com>**

**Date: on Nov 16 13:13:33 2015 -0500**

**First commit**

git log

git diff

git add filename

git status

1. The command “git status” shows

Untracked files and file changes staged for commit

File changes staged for commit

Untracked files only

A Git project’s commit history