codecademy

Basic Git Workflow

Initializing a Git Repository

The `git init` command creates or initializes a new Git project, or *repository*. It creates a **.git** folder with all the tools and data necessary to maintain versions. This command only needs to be used once per project to complete the initial setup. For instance, the code block sets up the **home** folder as a new git repository.

```
$ cd /home
$ git init
```

Checking the Status of a Git Repository

The `git status` command is used within a Git repository to its current status including the current commit, any modified files, and any new files not being tracked by Git. The output of `git status` can vary widely, and it often includes helpful messages to direct the user to manage their repository. For example, `git status` will show the user the files they would commit by running `git commit` and the files they could commit by running `git add` before running `git commit` .

Git

Git is a command line software that keeps track of changes made to a project over time. Git works by recording the changes made to a project, storing those changes, then allowing a programmer to reference them as needed.
All Git commands follow the pattern `git <action>` and, in order to use Git for a project, a project must first be initialized using the `git init` command in the project's root directory.

Git Project Workflow

A Git project has three parts:

- A Working Directory: where files are created, edited, deleted, and organized
- A Staging Area: where changes that are made to the working directory are listed
- A Repository: where Git permanently stores changes as different versions of the project

The Git workflow consists of editing files in the working directory, adding files to the staging area, and saving changes to a Git repository.

Adding Changes to the Staging Area

The `git add filename` command is used to add the `filename` file to the staging area. After your changes have been staged, you can use the `git commit` command to permanently store your changes.

```
$ git add scene-1.txt
$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   scene-1.txt
```

Displaying Differences with Git Diff

The `git diff filename` command will display the differences between the working directory and the staging area in one specific file. Use `git diff filename` before adding new content to ensure that you are making the changes you expect.

```
$ git diff hello.txt
diff --git a/hello.txt b/hello.txt
index 557db03..980a0d5 100644
--- a/hello.txt
+++ b/hello.txt
@@ -1 +1 @@
-Hello World
+Hello World!
```

Committing Your Code

The `git commit -m "log message here"` command creates a new commit containing:

- The current contents of the staging area
- A log message describing the changes to the repository

A commit is the last step in our Git workflow. A commit permanently stores changes from the staging area inside the repository. This command is almost always used in conjunction with the `git add` command as `git add` is used to add files to the staging area.

```
$ git commit -m "Added About section to
README"
[master 9d63f80] Added About section to
README
1 file changed, 10 insertions(+),
1 deletion(-)
```

Showing Git Commit Logs

In Git, the `git log` command shows all of the commit logs for a project. The following is displayed for each commit:

- A 40-character code, called a SHA, that uniquely identifies the commit.
- The commit author
- The date and time of the commit
- The commit message

This command is particularly useful when you need to refer back to an old version of your project. The unique SHA code allows you to identify a point in your program's history that you would like to revert to.

```
$ git log
commit
9d63f80111447544c303e9f1776fa08593a87310
Author: codecademy
<exampleuser@codecademy.com>
Date:   Wed Jan 13 18:55:53 2021 +0000

    Added updates to the file

commit
3ba6efbeece6ed530d85de5e313e52123fdf8cb4
Author: codecademy
<exampleuser@codecademy.com>
Date:   Wed Jan 6 10:11:13 2021 -0400

    Completed first line of dialogue
```