Lab 2 - Branching and Merging

A branch in git represents an independent line of development. You can think of branches as a way to request a brand new working directory, staging area, and project history. New commits are recorded **only** in the history for the current branch, which results as a fork in the history of the project.

Create a branch

You are just informed to create a new feature to work on the about page as a new feature. Let's create an about branch, by using the following command:

```
$ git branch about
```

To switch to a particular branch, we will use the `checkout` command:

```
$ git checkout about
```

As mentioned, each branch is considered as independent from the primary trunk or development line, so now we are at the `about` branch. If you check the status, it shows:

```
On branch about
nothing to commit, working directory clean
```

Let's create a `about.html`, with the following content.

```
<h1>This is the about page</h1>
```

Task: Add the `about.html` to staging and commit to the repository.

Switching branch

So now, let's switch it back to our `master` branch, using the `checkout` command. Before that there is a few terminologies we need to clarify:

HEAD: This is the current commit our repo is on. Most of the time HEAD points to the latest commit in your branch, but that doesn't have to be always the case. HEAD just means where is my repository currently pointing at.

master: The `master` branch is the name of the default branch that git creates for us when we first creating a repository. In most cases, **master** refers to “the main branch”.

Let’s switch to our master branch, and use the `ls` to show the folder content.

```
$ git checkout master
```

Because now we have switched to our master branch, we can only see the files that existed in `master` branch, before branching out to `about` branch. There should only be `index.html` and nothing else.

Task: Switch to `master` branch and branch it to another branch `about2`, and creat another `about.html`, assuming you are working on an A/B test or just want to try out different versions of about page. Type in different content and switch between `about` branch and `about2` branch. To visualize the difference, open the page in web browser and refresh it, and see how the modification is automatically handled by git seamlessly.

Merging branch

So now, you have been working on the `about` page for a while and you are pretty happy with the progress and after some test, you decided to merge it back to your `master` branch and continue working.

First, let’s make sure we have switched to our `master` branch.

```
$ git checkout master
```

Let’s merge with the `about` branch, using the `merge` command.

```
$ git merge about
```

Use `ls` or any other directory browsing, or browse to the folder, you will be able to see that now the `master` and `about` is now merged as one into `master`, but the `about` branch is still there, and safe for delete.