Git: Basic Usage

Now What?

So you have a git repository set up and you now want to start making forward progress on your project.

These commands should provide basic knowledge on how to manage changes to your repository.

Cloning

Use the `git clone` command to make a copy of an existing repository and store it on your local machine.

Example usage:

```
git clone [user]@git.pioneering.csun.edu:/srv/git/[directory]/[name].git
```

Status

Check your changes, and check them often with the `git status` command.

This will show you changes that have been made since your last commit (more on this later).

Branches

Think of branches as an isolated development environment within your project’s repository. If you want to experiment with a feature, but don’t want it potentially breaking your main development branch, you can test it on another branch.

Example usage:

```
git branch [new branch name]
```

Adding Files

After you have made your changes to your file(s), you will want to add them to the staging area to be committed.

Example usage:

```
git add [filename]
OR
git add * // adds all changes; use with caution
```

Commits

After adding your files, you will want to commit them (saving a snapshot of the data) before pushing your latest changes to the current branch.

Example usage:

```
git commit -m “Fixed buffer overflow issue”
```

Pushing

After you are done with your commits and want to update your, say, master branch's data, you will want to do a push.

Example usage:

$$\texttt{git push -u } \overset{\text{remote name}}{\texttt{origin}} \ \overset{\text{branch name}}{\texttt{master}}$$

Fetching

So what happens if you are on another machine that does not have your up-to-date data? Fetch it!

Example usage:

```
git fetch [remote name] // fetches all branches
OR
git fetch [remote name] [branch name]
```

Merging

This allows you to take two completely independent lines of development branches and combines them into one.

Example usage:

branch you are merging

branch you are merging into

```
git merge new-feature master
```

Pulling

You can also use the command: `git pull` to automatically fetch and merge data. In general though, use `fetch`, compare changes, *then* `merge`.

(Doing a straight out `pull` will not allow you to examine any code changes before making the decision to merge them)