Git Tagging

Simple Tagging Example / Lightweight Tags

Simple example with *lightweight* tags and some basic tag commands.

Command Listing

```
pwd
cd projects/starter-web/
git status
git log --oneline --decorate --graph --all
git tag myTag
git log --oneline --decorate --graph --all
git tag --list
git show myTag
git tag --list
git tag --delete myTag
git tag --list
git log --oneline --decorate --graph --all
```

Annotated Tags

Annotated tags are tags with more information, like comments, associated with them.

Command Listing

```
pwd
git status
clear
git tag -a v-1.0
git tag --list
git log --oneline --decorate --graph --all
git show v-1.0
```

Comparing Tags

Tags mark important milestones in a project, so they are great way to compare what has happened between those milestones.

Command Listing

```
pwd
git status
git tag --list
ls
mate index.html
git commit -a
git log --oneline --decorate --graph --all
git tag -a v-1.1
mate simple.html
git commit -am "Updating for tag 1.1"
git commit --amend
git tag v-1.2 -m "Release 1.2"
git tag --list
git log --oneline --decorate --graph --all
git diff v-1.0 v-1.2
git difftool v-1.0 v-1.2
```

Tagging a Specific Comment

If you forget to tag a commit in the past, you can simply pass in the commit id while creating the tag.

Command Listing

```
pwd
git status
git log --oneline --decorate --graph --all
git tag -a v-0.9-beta 96ef75b
git log --oneline --decorate --graph --all
git tag -a v-0.8-alpha ab0d621
git log --oneline --decorate --graph --all
```

Updating a Tag

Sometimes mistakes happen and you'll need to update a tag.

Command Listing

```
pwd
git status
git log --oneline --decorate --graph --all
git tag -a v-0.8-alpha -f bd35d46
git log --oneline --decorate --graph --all
```

Remote Tagging (GitHub)

Working with remote tags on GitHub.

Command Listing

```
pwd
git status
git tag --list
git log --oneline --decorate --graph --all
git push origin v-0.9-beta
git push origin v-1.1
clear
git push origin master --tags
git push origin :v-0.8-alpha
```