Code Collaboration and Submission

Git, GitHub, and Devpost

Andrew Kerr
me@andrewjkerr.com

whoami

whoami

- Fifth year Software Engineering @ UF

whoami

- Fifth year Software Engineering @ UF
- Officer of the UF Student InfoSec Team

whoami

- Fifth year Software Engineering @ UF
- Officer of the UF Student InfoSec Team
- Full stack web developer

QUOTTLY

whoami

- Fifth year Software Engineering @ UF
- Officer of the UF Student InfoSec Team
- Full stack web developer
- Currently web eng lead at Quottly

whoami

- Fifth year Software Engineering @ UF
- Officer of the UF Student InfoSec Team
- Full stack web developer
- Currently web eng lead at Quottly
- Former security intern at Tumblr

whoami

- Fifth year Software Engineering @ UF
- Officer of the UF Student InfoSec Team
- Full stack web developer
- Currently web eng lead at Quottly
- Former security intern at Tumblr
- Former intern at BlockScore

whoami

Andrew Kerr

@andrew

Professional nerd.

- Fifth year Software Engineering @ UF
- Officer of the UF Student InfoSec Team
- Full stack web developer
- Currently web eng lead at Quottly
- Former security intern at Tumblr
- Former intern at BlockScore
- Reach me @andrew on Slack

Git

At it's core, git is version control

What is version control?

Version control tracks changes in your code

```
diff --git a/presentation.md b/presentation.md
index 08d6441..c24a825 100644
--- a/presentation.md
+++ b/presentation.md
@@ -102,4 +102,6 @@ slidenumbers: true

 ## so what is version control?

+---
+
+## version control tracks changes in your code
(END)
```

Why is this useful?

Collaboration

How does this work?

How does this work?

1. Download code from remote server

How does this work?

1. Download code from remote server
2. Make your changes

How does this work?

1. Download code from remote server
2. Make your changes
3. Upload code to remote server

How does this work?

1. Download code from remote server
2. Make your changes
3. Upload code to remote server
4. Version control figures out what has changed and applies those changes to the codebase

Cool, so let's talk about git

Git Basics

Open up a terminal!

1. Make a new directory

2. `git init`

3. `git status`

4. Make a file

5. `git status`

6. `git add .`

6. `git add .`

- "staged files" are files that are ready to be committed to git
- "unstaged files" are files that have changes that have not been prepared to be committed

can also use `git add [filename]` for individual files

7. `git status`

8. `git commit -m 'Summary of changes'`

9. `git log`

Any questions?

Git Branches

Git Branches

- A "copy" of code that developers can work on independently of the main code base

Git Branches

- A "copy" of code that developers can work on independently of the main code base
- The main code base is normally "master"
- Once the developer is done, can "merge" it back into master

```
git branch [branchname]
```

Once we have a new branch, we follow the git flow

We have new code on a branch, how do we merge back?

```
git checkout master
```

```
git merge [branchname]
```

Using branches allows for multiple developers to work on the same code base

Any questions?

Git Remotes

Git Remotes

- A remote server for your git repository!

Git Remotes

- A remote server for your git repository!
- This is where GitHub comes in...

GitHub

Powerful collaboration, code review, and code management for open source and private projects.

Helps put a GUI on top of git!

Sign up for GitHub at github.com

Once signed up, create a repository

Get the remote `ssh` url

Back to the command line...

```
git remote add origin [url]
```

```
git remote -v
```

So, how do we get code on our remote?

```
git push
```

Awesome! Now we've got code on GitHub

Once a teammate pushes changes, make sure to use `git pull` to download the new changes

Any questions?

GitHub + Devpost

Projects Hackathons Teams ALPHA

Post a new project

Please respect our community guidelines.

Import from GitHub Save time by importing your project name, tagline, and README from GitHub.

Preparing your GitHub repo for submission

1. Create a README.md file

Example: https://github.com/andrewjkerr/hackathon-mentor-request-slackbot

Mentor Request for Slack

A slackbot to help hackathons manage mentorship requests during their hackathon. Written during KnightHacks from Jan 15, 2016 - Jan 16, 2016.

Usage

There are three kinds of users: attendees, mentors, and admins.

Attendees

Want help with something? Use this Slack bot to help! There are a few commands that you should be aware of:

`.mentor`

The base command that the bot looks for. It also brings up the help menu 😄

2. Set the description

"A slackbot to help manage mentorship requests for hackathons (aka SwampHacks)."

Anddd that's it! Pretty easy, eh?

To review

To review

1. `git pull`

To review

1. `git pull`
2. `git checkout -b branch`

To review

1. `git pull`
2. `git checkout -b branch`
3. Make changes

To review

1. `git pull`
2. `git checkout -b branch`
3. Make changes
4. `git add .`

To review

1. `git pull`
2. `git checkout -b branch`
3. Make changes
4. `git add .`
5. `git commit -m 'commit summary'`

To review

1. `git pull`
2. `git checkout -b branch`
3. Make changes
4. `git add .`
5. `git commit -m 'commit summary'`
6. `git checkout master`

To review (continued...)

1. `git pull`

To review (continued...)

1. `git pull`
2. `git merge branch`

To review (continued...)

1. `git pull`
2. `git merge branch`
3. `git push`

To review (continued...)

1. `git pull`
2. `git merge branch`
3. `git push`
4. Do it all again!

Git and GitHub can help teams work more efficiently on code

(And it doesn't hurt that GitHub works well with Devpost)

Any questions?

Slides available at talk.andrewjkerr.com