Git Branches: Takeaways 🖻

by Dataquest Labs, Inc. - All rights reserved © 2020

Syntax

• Create a branch

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

· Switch to branch

```
git checkout [branch name]
```

• Create and switch to branch

```
git checkout -b [branch name]
```

• Show all of the remote branches

```
git branch -r
```

• Show all of the branches availably locally

```
git branch -a
```

• Merge a branch

```
git merge
```

• Delete a branch

```
git branch -d [branch name]
```

• Update Git's list of branch names and commits

```
git fetch
```

Concepts

- Branches allow us to create several different work areas within the same repo.
- Switching branches is useful when we want to work on changes to a project that require different amounts of development time.

- Git will prevent you from switching to a different branch if there is a potential merge conflict with the branch we're switching to.
- Git uses HEAD to refer to the current branch, as well the branch with the latest commit in that branch.
- Django is a popular Web framework for Python that programmers develop using Git and GitHub.
- In order to merge branch B into branch A, switch to branch A then merge the branch.
- Pull requests will show us the differences between branches in an attractive interface, and allow other developers to add comments.
- Typical branch names:

Feature: feature/happy - bot
Fix: fix/remove - error
Chore: chore/add - analytics

Resources

- Pull Requests
- Django



Takeaways by Dataquest Labs, Inc. - All rights reserved © 2020