# Branching with Git

| | |
|---|---|
| **Time** | ~ 2 hours |
| **Learning Goals** | • Understanding project workflow with git.<br>• Understanding the working tree and branches in git.<br>• Understand what is add, commit, fork, clone, and checkout in git.<br>• Learning to work as a team. |

Remember Tommy and Jane where they eventually realised they needed a platform for code collaboration. Indeed, programming is more often than not, a team sport. Much like in any workplace environment, it's important to be able to track workflow within a team, experiment with changes locally without affecting a central system and revisit the history of changes made. Git is an extremely powerful source control system and understanding how it works is critical to working effectively in a team.

In this challenge, you will learn how to start a git project. You will also learn about the typical workflow after you start a project (which you have been doing already). You will also be learning about the working tree and its branches, and why it's good practice to work on a feature branch instead of the master branch. Let's get started!

## Objectives

### Create a GitHub Repository (approx. 20 - 30 minutes)

Based on the guide in the slides, create a local repository on your computer and a remote repository (use your Github account).

### Understanding Branches and Learn how to do Code Collaboration (approx. 30 - 40 minutes)

You MUST get a partner to work with you on this challenge otherwise you won't be able to learn how to do code collaboration.

Go through the slides and learn how to create and merge a branch. When it comes to the part on code collaboration, do the following:

1. Choose one of you to be the initiater, let's call him/her coder A.
2. In Github, coder A invites coder B to collaborate on a project (go to "Settings" in your repo and look for "Collaborators").
3. When coder B accepts the request, git clone the repo (don't fork here!).
4. Don't forget that coder A also has to 'git clone' the repo onto his/her local machine.

Both of you can then work on the exercises as given in the slides.

### Consolidate Understanding (approx. 30 minutes - 1 hour)

# Branching with Git

Now let's get to some note writing based on the questions below. Please do not skip this exercise. Write your notes in the file `my_git_project.rb` so that you know what you know and don't know. Besides, you can always refer back to these notes later on!

In your own words, answer the following questions:

1. What is git?
2. What is the git project workflow?
3. How do you initialize a new git project?
4. What is the difference between making a `fork` and making a `clone`?
5. What is the difference between `git add` and `git commit`?
6. What is the difference between `git push -u origin master` and `git push`?
7. What is the difference between a local repository and a remote repository?
8. What does `origin` refer to?
9. Why is important to do branching?
10. How do we create a new branch?
11. How do we merge back to the master branch?
12. What is a pull request?
13. How do we resolve conflicts when they arise?

To learn more about what git has to offer, bookmark this git cheatsheet.