# September

## 17-23:

### Git and github introduction:

* [Git Tutorial (w3schools.com)](https://www.w3schools.com/git/default.asp)
* [How to Use Git and GitHub – Introduction for Beginners (freecodecamp.org)](https://www.freecodecamp.org/news/introduction-to-git-and-github/)
* [Learn Git Branching](https://learngitbranching.js.org/) (Interactive tutorial)
* [Git - Basic Branching and Merging (git-scm.com)](https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging)
* [Pull Requests in VS Code - YouTube](https://www.youtube.com/watch?v=LdSwWxVzUpo)

### Project: Create a Simple Website

Description: You will create a simple static website with a few pages and use Git and GitHub to manage the project's version control.

Tasks:

1. Setup:

* Set up Git on your local machine if you haven't already.
* Create a GitHub account if you don't have one.
* Create a new GitHub repository (let's call it "SimpleWebsite").

1. Initial Commit:

* Create a folder on your local machine for the project.
* Inside the folder, create an HTML file (e.g., index.html) and add some basic content.
* Initialize a Git repository in the project folder using the git init command.
* Add the HTML file to the repository using git add.
* Commit the changes using git commit.

1. Branching:

* Create a new branch named "feature" using git branch feature.
* Switch to the "feature" branch using git checkout feature.
* Make some changes to the HTML file in the "feature" branch, like adding a new page or modifying existing content.

1. Merging:

* Switch back to the main branch using git checkout main.
* Merge the changes from the "feature" branch into the main branch using git merge feature.
* Remote Repository:
* Link your local repository to the GitHub repository you created earlier using git remote add origin <repository URL>.
* Push your local repository to GitHub using git push -u origin main.

1. Pull Requests:

* Go to the GitHub repository.
* Create a new Pull Request (PR) to merge the "feature" branch into the main branch.
* Add a description for the PR.
* Review the changes and create the PR.

1. Collaboration (Optional):

* Invite a friend to collaborate on the project by adding them as a collaborator on the GitHub repository.
* Have your friend clone the repository, create a branch, make changes, and create a Pull Request as well.

## 24-30:

### Introduction to Dart:

* [Dart basics | Dart](https://dart.dev/language)
* [Variables | Dart](https://dart.dev/language/variables)
* [Error handling | Dart](https://dart.dev/language/error-handling)
* [Classes | Dart](https://dart.dev/language/classes)
* [Constructors | Dart](https://dart.dev/language/constructors)
* [Understanding null safety | Dart](https://dart.dev/null-safety/understanding-null-safety)

### Project: Building a Simple Task List App

**Description:** You will create a simple command-line task list application using Dart. This project will cover various Dart concepts.

1. Setup:

* Install Dart on your local machine by following the official Dart installation guide: https://dart.dev/get-dart

1. Variables:

* Create a Dart program that declares and initializes different types of variables (integers, strings, booleans, lists, etc.).
* Print the values of these variables to the console.

1. Error Handling:

* Modify your Dart program to include error handling using try-catch blocks.
* Create a situation where an error can occur, catch the error, and handle it gracefully.

1. Classes:

* Define a Dart class called Task to represent a task with properties like id, title, description, and isCompleted.
* Create instances of the Task class and manipulate their properties.

1. Constructors:

* Add a constructor to the Task class to simplify the creation of task objects.
* Create new task objects using the constructor.

1. Understanding Null Safety:

* Refactor your Dart program to enable null safety.
* Annotate variables and function parameters with appropriate null safety syntax (e.g., int?, String?, Task?).
* Ensure that you handle null values appropriately in your code.

1. Task List App:

* Build a simple task list app that allows users to add, view, update, and delete tasks.
* Implement a menu-driven interface in the command line to interact with the task list.

# October

# November

# December

# January

# February