SDEV 255 Handbook Welcome

Description of Book

Welcome to SDEV 255 - Web Application Development. Throughout this course you will use this handbook to cover topics not available in your text, as well as to setup certain labs/environments for your assignments. Feel free to come back to this book as needed for supplementary help!

Part 1 - Collaboration and Setup for Front End Development

Introduction

In Part 1 of this handbook, you will be introduced to using Git, Github, and an IDE(s) of your choosing. The topics will include:

- Understanding and using version control
- Understanding the need for version control
- Installing and configuring Git
- Understand distributed version control models such as Github
- Become familiar with popular Git terms and definitions
- Setup your code environment using Replit
- Setup your local code environment using VS Code
- Utilizing all the above to connect and manage your code projects.

Collaboration and Code Management - Git, Github, and Version Control

Introduction to Version Control

Throughout this course, you will be utilizing version control software to help manage your projects. But, what is version control?

Version control is tracking software changes, and the ability to analyze or revert to previous changes at a later time. Consider version control as as sort of "save point" system. You pick a point in time that you wish to "freeze" the state of your code, and then if you wish, you can revert back to that state at any point. To look at what version control is deeper, read the following document from the Git website below:

 Git – About Version Control

---

Also, take a look at the explanation of Version Control for the Atlassian website located below. The video on this website is a great introduction to the topic:

 What is version control | Atlassian Git Tutorial

---

There also many terms/definitions associated with version control software (specifically Git). Here are some things you need to know:

Basic Setup

- **Repository (repo):** The database storing the files.
- **Server:** The computer storing the repo.
- **Client:** The computer connecting to the repo.
- **Working Set/Working Copy:** Your local directory of files, where you make changes.
- **Trunk/Main:** The primary location for code in the repo. Think of code as a family tree — the trunk is the main line.

Basic Actions

- **Add**: Put a file into the repo for the first time, i.e. begin tracking it with Version
- Control.
- **Revision**: What version a file is on (v1, v2, v3, etc.).
- **Head**: The latest revision in the repo.
- **Check out**: Download a file from the repo.
- **Check in**: Upload a file to the repository (if it has changed). The file gets a new revision number, and people can “check out” the latest one.
- **Checkin Message**: A short message describing what was changed.
- **Changelog/History**: A list of changes made to a file since it was created.
- **Update/Sync**: Synchronize your files with the latest from the repository. This lets you grab the latest revisions of all files.
- **Revert**: Throw away your local changes and reload the latest version from the repository.

Git

So how do we use version control? Well, we need to use version control software. The most popular and defacto standard is Git. Git utilizes LOCAL version control to maintain software. You can also use the distributed model, but often we utilize another service or software in order to do this. For more information on what Git is, look at the following website:

 Git - Documentation

---

We will be utilizing Git later on in this course, so for now, just become familiar with what Git it, and the service provides.

Github

If Git is the software that version controls the software, then Github would be the "Google Drive" of the software development world. Github is used to share code, collaborate, and has a variety of team and project management tools built in. Watch the following video to get a better understanding of Github:

What is GitHub? | What is GitHub and How To Use It? | GitHub Tutorial for Beginners | Simplilearn

---

Your Turn

Now we are going to begin to setup our software for our class. The first thing you are going to do is install the Git software on your machine. So perform the following steps

- Go to https://git-scm.com/downloads
- Click on the version of Git you need for your computer
- Install the software with all the defaults. You should not have to do anything else but click next (in Windows) to install the software.

Next you will need to create a Github account.

- Go to https://github.com/
- Click the Sign Up button in the upper right hand corner
- Fill out the form using your Ivy Tech email address
- An email will be sent to confirm your account. Make sure to click on the link within the email to activate the account.
- DON'T FORGET YOUR LOGIN INFORMATION

After completing the above steps, you are ready to move on to the next section in this tutorial.

IDEs, Text Editors, and Code Environments

Now we will be working on setting up your code-environment / IDE. We will be utilizing Replit in this course for both our code editor and hosting service. Utilizing VS Code (or Visual Studio Code) is somewhat optional. I would, however, highly recommend that you follow the tutorial in setting up BOTH editors. It will make the future back-end setup much easier.

Replit Setup

Replit Setup

During this part of the tutorial, you will begin to setup you environment utilizing Replit. Replit is a cloud-based code environment that allows you to code from anywhere. Below is a video tutorial on how to setup Replit, but if you prefer a quick text-based summary, do the following:

- Go to https://replit.com/
- Click on the Sign-in button
- Click on the Google "G" symbol, and login using your Ivy Tech email and password After logging in, click on the + button in the upper right-hand corner.
- In the drop-down menu, click on the HTML, CSS, and JS environment
- Name you code environment
- Click Create repl

Video

Replit Setup SDEV - 255

Visual Studio Code (VS Code) Setup

VS Code

Next, we will setup VS Code to be our local code editor. This can be installed on any machine that has a Linux, Windows, or Mac OS. Make sure that you installed Git from the first section of this tutorial before you begin setting up. Once again, there is a video tutorial on how to setup VS Code, as well as some additional features of the software. In summary you should:

- Go to https://code.visualstudio.com/ and download the VS Code software for your operating system
- Create a folder to work in (perhaps on your
- Desktop) Install the following extensions:
    - Live Server
    - Live Share
    - Prettier

Video

VS Code Setup – SDEV 255

Bringing It All Together

Bring It Home!

For the final part of this tutorial, we will setup our Replit and VS Code environments to be able to version control our projects, as well as share between the two code editors. Watch and follow along the video below in order to correctly setup your environment for the first part of this course.

Video

 Bringing It All Together – SDEV 255
