Setting up Your Development Environment: Git: Objectives and Outcomes

At the end of this lesson you should have set up Git on your computer. At the end of this lesson, you will be able to:

- Set up a Git repository and perform basic Git operations
- Set up and use online Git repositories

Setting up your Development Environment

Software Requirements

1. **Text editor of your choice**: Any text editor that you are already familiar with can be used for editing the project files. I will be using Visual Studio Code (https://code.visualstudio.com/) as the editor of choice in this specialization. You may also consider other editors such as Brackets (http://brackets.io/), Sublime Text (http://www.sublimetext.com/), or Atom (https://atom.io/).
2. **Browser of your choice**: You may use your preferred browser. I will be using Chrome as the browser in all the exercises. All the exercises and assignments in this course have been tested using Chrome v. 46. Please note that not all browsers may support all the HTML5 features to the same extent. You might encounter problems when using other browsers. I strongly urge you to use the latest Chrome browser for the exercises and assignments in this course so that any problems are minimized.
3. **Command line shell**: Familiarity with the command-line shell will be essential for the exercises. In Windows a cmd window or power shell with admin privileges would be needed. On a Mac or in Linux, a terminal window can be used. Please get familiar with the "sudo" command in OS X and Linux.

Note: Please remember to retain the folders and all the files that you create in the exercises. Further exercises will build upon the files that you create in the preceding exercises. DO NOT DELETE the files at the end of the exercises, unless otherwise instructed. You may wish to set up your exercise folder as a Git repository and commit the files to the repository at the end of each exercise.

Exercise (Instructions): Setting up Git

Objectives and Outcomes

In this exercise you will learn to install Git on your computer. Git is required for using all the remaining Node.js and Node based tools that we encounter in the rest of the course. At the end of this exercise, you would be able to:

- Install Git on your computer
- Ensure that Git can be used from the command-line or command-prompt on your computer
- Set up some of the basic global configuration for Git

Downloading and Installing Git

- To install Git on your computer, go to https://git-scm.com/downloads to download the Git installer for your specific computing platform.
- Then, follow the installation steps as you install Git using the installer.
- You can find more details about installing Git at https://git-scm.com/book/en/v2/Getting-Started-Installing-Git. This document lists several ways of installing Git on various platforms.
- Installing some of the GUI tools like GitHub Desktop will also install Git on your computer.
- On a Mac, setting up XCode command-line tools also will set up Git on your computer.
- You can choose any of the methods that is most convenient for you.

Some Global Configuration for Git

- Open a cmd window or terminal on your computer.
- Check to make sure that Git is installed and available on the command line, by typing the following at the command prompt:

```
git --version
```

- To configure your user name to be used by Git, type the following at the prompt:

```
git config --global user.name "Your Name"
```

- To configure your email to be used by Git, type the following at the prompt:

```
git config --global user.email <your email address>
```

- You can check your default Git global configuration, you can type the following at the prompt:

```
git config --list
```

Conclusions

At the end of this exercise you should have Git available on the command-line of your computer.

Exercise (Instructions): Basic Git Commands

Objectives and Outcomes

In this exercise you will get familiar with some basic Git commands. At the end of this exercise you will be able to:

- Set up a folder as a Git repository
- Perform basic Git operations on your Git repository

Basic Git Commands

- At a convenient location on your computer, create a folder named **git-test**.
- Open this git-test folder in your favorite editor.
- Add a file named *index.html* to this folder, and add the following HTML code to this file:

```
<!DOCTYPE html>
<html>
    <head></head>
    <body>
        <h1>This is a Header</h1>
    </body>
</html>
```

Initializing the folder as a Git repository

- Go to the git-test folder in your cmd window/terminal and type the following at the prompt to initialize the folder as a Git repository:

```
git init
```

Checking your Git repository status

- Type the following at the prompt to check your Git repository's status:

```
git status
```

Adding files to the staging area

- To add files to the staging area of your Git repository, type:

```
git add .
```

Commiting to the Git repository

- To commit the current staging area to your Git repository, type:

```
git commit -m "first commit"
```

Checking the log of Git commits

- To check the log of the commits to your Git repository, type

```
git log --oneline
```

- Now, modify the *index.html* file as follows:

```
<!DOCTYPE html>
<html>
    <head></head>
    <body>
        <h1>This is a Header</h1>
        <p>This is a paragraph</p>
    </body>
</html>
```

- Add a sub-folder named **templates** to your **git-test** folder, and then add a file named *test.html* to the templates folder. Then set the contents of this file to be the same as the *index.html* file above.
- Then check the status and add all the files to the staging area.
- Then do the second commit to your repository
- Now, modify the *index.html* file as follows:

```
<!DOCTYPE html>
<html>
    <head></head>
    <body>
        <h1>This is a Header</h1>
        <p>This is a paragraph</p>
        <p>This is a second paragraph</p>
    </body>
</html>
```

- Now add the modified index.html file to the staging area and then do a third commit.

Checking out a file from an earlier commit

- To check out the index.html from the second commit, find the number of the second commit using the git log, and then type the following at the prompt:

```
git checkout <second commit's number> index.html
```

Resetting the Git repository

- To discard the effect of the previous operation and restore index.html to its state at the end of the third commit, type:

```
git reset HEAD index.html
```

- Then type the following at the prompt:

```
git checkout -- index.html
```

- You can also use *git reset* to reset the staging area to the last commit without disturbing the working directory.

Conclusions

At the end of this exercise you should have learnt some basic Git commands. Experiment with these commands until you fully understand how to use Git.

Exercise (Instructions): Online Git Repositories

Objectives and Outcomes

In this exercise you will learn about how to set up and use an online Git repository and synchronize your local Git repository with your online repository. At the end of this exercise, you will be able to:

- Set up the online repository as a remote repository for your local Git repository
- Push your commits to the online repository
- Clone an online Git repository to your computer

Setting up an Online Git repository

- Sign up for an account either at Bitbucket (https://bitbucket.org) or GitHub (https://github.com).
- Then set up an online Git repository named **git-test**. Note the URL of your online Git repository.

Set the local Git repository to set its remote origin

- At the prompt, type the following to set up your local repository to link to your online Git repository:

```
git remote add origin <repository URL>
```

Pushing your commits to the online repository

- At the prompt, type the following to push the commits to the online repository:

```
git push -u origin master
```

Cloning an online repository

- To clone an online repository to your computer, type the following at the prompt:

```
git clone <repository URL>
```

Conclusions

In this exercise you have learnt to set up an online Git repository, synchronize your local repository with the remote repository, and clone an online repository.