# Exercises

# PreRequisites

# 1. Install git from <https://git-scm.com/downloads>

# 2. Create an account in GitHub <https://github.com>

# 3. Install Hugo <https://gohugo.io/getting-started/installing/>

# 6. Install Python <https://www.python.org/downloads/>

# 7. Install MKDocs [https://www.mkdocs.org/#installation](https://www.mkdocs.org/%23installation)

# 8. Install VS Code <https://code.visualstudio.com/download>

# Git Exercise 1

# Set up your system for git

# Create a folder in your local system

# cd to the folder

# Open Git Bash

# Execute git init

# Execute git config –-global user.email <email>

# Execute git config –-global user.name <username>

# Copy the given adoc exercise files to the folder

# Execute git add .

# Commit the files to the local repository

# Execute git commit –m “First Commit files added”

# GitHub Exercise 2

# Log into GitHub

# Create a New Repository in GitHub

# Copy the Repository URL from Quick Setup page

# cd to the folder created in Exercise 1

# Open Git Bash

# Execute git remote add origin <repository URL>

# Execute git remote –v

# Execute git add .

# Execute git commit –m “files added”

# Execute git push origin master

# Check your files in GitHub

# Git Exercise 3

# Delete your project folder created in Exercise 1 in your local system

# Let us clone the repository from GitHub

# Go to your root directory (where you save git projects) in your system

# Open Git Bash

# Execute git clone <repository URL>

# Check your folder for the files

# Git Exercise 5

# Open Git Bash in the git project

# Execute git pull

# Execute git branch first-branch

# Execute git checkout first-branch

# Execute git checkout master

# Execute git push origin <branch name>

# Supply the GitHub username and password

# Git Exercise 6

# Open Git Bash

# Execute git checkout first-branch

# Execute touch test1.adoc

# Execute git commit –m “file added to branch”

# Execute git push origin first-branch

# Check the files in GitHub

# Let us merge the branch with master

# Switch to master, git checkout master

# Execute git merge first-branch

# Check the master in GitHub

## MKDocs Exercise

## Install Python, pip, MKDocs, necessary packages

## Create a folder for named *mkdocs\_project*

## Open Power Shell or cmd in Windows, and execute the following commands:

## mkdocs new <project-name>

## Copy the given markdown files *chapter1.md*, *chapter2.md*, and *chapter3.md* to the docs folder in your newly created project

## Edit the *index.md* file in the *docs/* as you require

## Replace the *mkdocs.yml* from the zip:

## site\_name: My Asciidoctor

## nav:

## - Home: index.md

## - Chapter 1 Overview: chapter1.md

## - Chapter 2 Installation: chapter2.md

## - Chapter 3 Configuration: chapter3.md

## theme: readthedocs

## Save the *mkdocs.yml* file

## Execute the command mkdocs serve

## Check the added files and the site in your local server <http://127.0.0.1:8000>

## Build the site files using the command mkdocs build

## The html files are generated and stored in the folder */site*

## Create a new repository in GitHub

## Follow the steps in Exercise 2 and push these files to the new repository

## Then, execute the command mkdocs gh-deploy

## Check your new doc website URL given in the output

# Hugo Exercise 1

# Create a folder named hugo\_techdoc

# cd to the folder

# Type hugo version and check the hugo version

# Enter hugo new site <sitename>

# cd to <sitename>

# Following directories are created:

# 

# Execute git init

# cd to themes directory

# Execute git clone <https://github.com/thingsym/hugo-theme-techdoc.git>

# Remove the .git folder in the themes directory

# Copy the Content folder and config.toml file from the class folder named hugo and replace them respectively in your hugo project folder.

# Open cmd and execute hugo server -t hugo-theme-techdoc

# Open <http://localhost:1313> and check the doc site

# Execute the command hugo –t hugo-theme-techdoc –d docs

# A new folder docs is created with the generated files

# Create a repo in GitHub and push these files to the new repo (refer Exercise 2)

# In the GitHub Repo, go to Settings tab and select the GitHub Pages Source as master/docs folder

# Wait for the green highlighted text like this

# 

# Open the published URL and check your doc