**Monday 23rd of January.**

**GitHub**

- Deep Dive into GitHub with charlie.
- Working with Git Interns .
- Git Premier Kartoza Handbook.

**Purpose**

How to user git to implement new features or changes into a project without breaking the existing code base. Focus on topics like Branches , Monitoring Changes , Committing , Conflict resolution's , History , Pull Requests , Merge Request and CI/CD

**Markdown**

Markdown Cheat sheet

**Purpose**

The markdown cheat sheet provides a quick overview of all the markdown syntax elements.

Markdown cheatsheet

**Activity**

- Create GitHub account
- Forked The Kartoza Handbook GitHub repository
- Uploaded a screenshot of my GitHub account

**Refactoring and Project Response**

Worked on my CRS to be more user readable name in my simple Africa Map Project

**Tuesday 24th January.**

- Open Street Map (OSM)
- Watch this introduction to OSM video on YouTube Open  Street Map

**Purpose**

Open Street Map is a free, editable map of the whole world that is being built by volunteers largely from scratch and released with an open-content license. The Open Street Map license allows free access to their map image and all or their underlying map data . This is a good way to contribute to open source as a beginner

**Activity**

- Create a open source account
- Go through the documentation
- Contribute to OSM

**Wednesday 25th January.**

**Activities**

- Continue with QGIS module
- Read Kartoza handbook
- Learning  how to contribute to OSM
- Continue contributing to OSM

**Thursday 26th January**

**Activities**

- Continue with the Kartoza Handbook
- QGIS Module Analysis and Data Acquisition
- Contribute to OSM

**Summary**
**QGIS.**

**QGIS Definitions and Terminologies**

- **QGIS** (Quantum Geographic Information System) is a free, open-source software that allows users to create, edit, visualize, analyze, and publish geospatial information

- **Symbology** is the use of symbols to represent the features and attributes of a map layer. For example, in a layer of cities, black circles might symbolize the cities. The size of the circles might be varied to symbolize each city's population

- **Layers** allows you to create new layers in different formats. It provides tools for creating Geo Package, Shape-file, SpatiaLite, GPX format and Temporary Scratch layers (aka memory layers).

- **Labels** serve as a communication medium between the users creating the map and the end user. They provide user information regarding it's purpose.

- **Raster dataset** is composed of rows (running across) and columns (running down) of pixels (also know as cells). Each pixel represents a geographical region, and the value in that pixel represents some characteristic of that region.

- **Vector data** is represented as a collection of simple geometric objects such as points, lines, polygons, arcs, circles, etc. For example, a city may be represented by a point, a road may be represented by a collection of lines, and a state may be represented as a polygon

**GitHub Definitions and Terminologies**

- **Branch** a version of the repository that diverges from the main working project. Branches can be a new version of a repository, experimental changes or personal **forks or a repository for users to alter and test changes**
- **Clone i**s a copy of a repository or the action of copying repository. When cloning a repository into another branch , the new branch becomes a remotely tracing branch that can talk upstream to its origin branch via push , pulls and fetches
- **Fetch** by performing a Git fetch, you are downloading and copying that branch's files to your workstation . Multiple branches can be fetched at once and you can rename the branches when running the command to suit your needs.
- **Fork** creates a copy of a repository

- **Head i**s a reference variable used to denote the most current commit of the repository which you are working. When you add a new commit , Head will then become that ne commit
- **Index** the working or staging area of Git files that have been changed, added and deleted will be staged within the index until you are ready to commit the files . To see what is set in your Git index , run “git status” within your repository. The green **files** are staged and ready to commit, whereas the red files have not yet been added to staging for the next commit
- **Master** is the primary branch of all repositories , all committed and accepted changes should be on the master branch your can work directly from the master branch, or create other branches
- **Merge** taking the changes from one branch and adding them into another branch, these commits are usually first requested via pull request before merged by a project maintainer.
- **Origin** the conventional name for the primary version fo a repository. Git also uses origin as a system alias for pushing and fetching data to and from the primary branch. For example, “git push origin master” , when run on a remote , will push the changes to the master branch of the primary repository database.
- **Pull Request** if someone has changed code on a septate branch of a project and wants it to be reviewed to add to the master branch, that person can put in  pull request. Pull request ask the repository maintainers to review the commits made and the, if acceptable, merge the changes upstream. A pul happens when adding the changes to the master branch
- **Push** updates a remote branch with the commits made to the current branch. You are literally pushing your changes onto the remote
- **Remote** a copy of the original branch. When you clone a branch, that new branch is a remote, or clone. Remotes can talk to the origin branch, as well as other branches easier
- **Repository “Repo” i**n many ways, you can think of a Git repository as a directory that stores all the files ,folders and content needed for your