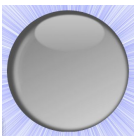


# Version Control, git and Github

First Contributing to SasView workshop

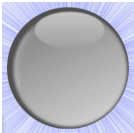
Paul Butler  
Virtually May 20, 2021



# What to expect

No knowledge of any of these concepts is assumed here.

These few slides are intended to provide a quick orientation to how the three concepts are related (and how they are different) and some basics about each. They are intended to be followed immediately by a tour of github using the SasView organization github



# Version Control

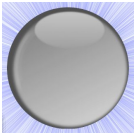
**git**

**Github**

A concept for collaborative software development to keep track of changes and conflicts.

One of many “protocols” of how to implement Version Control. Also name of software implementing the protocol

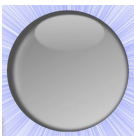
A web based Software development environment which provides a number of services including an implementation of git for version control



# VERSION CONTROL

- ALL HISTORY is maintained .. nothing is deleted
- You can go backward and forward in time to see who made what change when
- You can revert to a previous time
- Tools to do more sophisticated surgery for the ninjas -- will not cover

Basically a big database that tracks and records every difference (along with time and who made the change) - not saving every changed file.

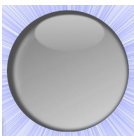


# BASIC GIT CONCEPTS/COMMANDS

Git is basically a protocol for how to do version control  
But also basically an implementation of that protocol

## KEY CONCEPTS

- At its core it is meant to be de-centralized. So everybody has a copy of the FULL repository (repo). In principle “all are created equal”
- For simplicity SasView uses Github as the master repo on github
- NOTHING changes when you change and save a file (need some git commands)
- Git changing things locally is a commit to the repo ... BUT only your cop. Nobody knows what you have done yet - you need to synch the new state of your repo with “origin” on github



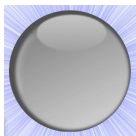
# BASIC GIT CONCEPTS/COMMANDS

## BASIC CONCEPTS/COMMANDS:

- Repo (git init; git clone https://www.github.com/xxxx)
- Branches (git branch mybranch; git checkout mybranch; git switch mybranch)
  - Create and checkout branch in one command (git checkout -b mybranch)
- **add and commit, (git add file1 + git commit OR git commit -am “message”)**
- push upstream, (git push; git push origin -u mybranch remote branch)
- pull from remote, (git fetch, git pull)
- Merge, (git merge otherbranch)
- History (git log; git blame ... or git praise :-))
- Status (git status)

Handy ref:

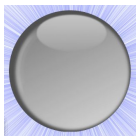
<https://www.digitalocean.com/community/cheatsheets/how-to-use-git-a-reference-guide>



# GitHub

A service provided at *github.com* that includes:

- Version control repository (using git)
- Ticket/Issue tracker system
- Continuous Integration tools (but travis CI going away)
- Website
- Wiki
- Release management
- And more



# NOW A QUICK TOUR

- The different repositories within the SasView github account
- In sasview repo
  - Last commit and log (git log)
  - Commits and diffs view and blame
  - Branches (main and master) and making PR
  - PRs and how to work them
  - Insights -> Network: the bird's eye view
  - Issues - where discussion happens (labels and milestones and linking to PR/closing)