

# GIT Mini XTR Exercises

## PreRequisites

### Git Installation

- Unix : <http://git-scm.com/download/linux>
- Windows : <http://git-scm.com/download/win>
- Mac : <https://mac.github.com/>

### GitK Installation

- Unix : **sudo apt-get install git-gui gitk**
- Windows : **should come with git installation**
- Mac : **should come with git installation**

### Installation Test

- command Prompt : git (*Command should be recognized*)
- command Prompt : gitk (*Command should be recognized*)

## Session 1 - Exercise1

### Step1 - Git Configuration

- git config --global user.name <name>
- git config --global user.email <email>
- git config --global color.ui auto
  
- git config <parameter>
- git config --list

### Step 2 - Registration

- Register yourself on [www.github.com](http://www.github.com)

### Step 3 - SSH Keys Generation

- **Linux :**
- **Windows:** <https://help.github.com/articles/generating-ssh-keys#platform-windows>
- **Mac:** <https://help.github.com/articles/generating-ssh-keys#platform-mac>

## **Session 2 - Exercise 1**

- 1) Create a repository on your github account.
- 2) Create a local folder **session1**
- 3) Shell/Command Prompt : Go to **session1** folder and create this project  
mvn archetype:generate -DgroupId=com.xebia.xtr -DartifactId=git-demo  
-DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
- 4) Shell/Command Prompt : Go to **session1** folder
- 5) Shell/Command Prompt : git init
- 6) Verify that .git folder is created
- 7) Shell/Command Prompt : Go to git-demo folder git add -a
- 8) Shell/Command Prompt : git commit -m "<Some Commit Message>"
- 9) Shell/Command Prompt : git remote add origin [git@github.com](mailto:git@github.com):<<Git Repo>>
- 10) Shell/Command Prompt : git push origin master
- 11) Verify that files reached on github account via browser

### **Additional**

- 1) Add a new file **file.txt** in the git-demo project
- 2) Make some changes to the file
- 3) Commit that file
- 4) Do git status and see the output
- 5) Modify the same file again
- 6) Do git status and observe the status

## **Session 2 - Exercise 2**

- 1) Continue using the repository create in the previous exercise.
- 2) Create file **exercise2.txt** , add, commit and push it to the remote. Note the SHA code
- 3) Update the file **exercise2.txt** add, commit and push it to the remote.
- 4) Shell/Command Prompt : git revert SHA Code
- 5) Shell/Command Prompt : git push origin master
- 6) Observe the commits on the git hub repo and contents of **exercise2.txt** (They should be same as first time commit).

### **Additional**

- 1) Edit , add , commit and push **exercise2.txt** 3 - 4 times more.
- 2) Now try to bring the **exercise2.txt** contents to the original contents.

## **Session 2 - Exercise 3**

- 1) Clone the project from this url [git@github.com:ScalaTribelIndia/session2.git](https://github.com/ScalaTribelIndia/session2.git) in folder **/home/session2**
- 2) Edit the file Name.txt and enter the line "**Committed by <<Your Name>>**"
- 3) Commit and push this change to the repository.

## **Session 3 - Exercise 1**

- 1) Clone the project from this url [git@github.com:ScalaTribelIndia/session3.git](https://github.com/ScalaTribelIndia/session3.git) in folder **/home/session3**
- 2) Create a branch from master with **<yourname>**.
- 4) Create one more branch **<yourname>-1** from the **<yourname>** branch.
- 5) Add 2 new files(file2.txt,file3.txt) and commits in the **<yourname>-1** branch , don't PUSH.
- 3) Move to **<yourname>** branch and Create a file **file1.txt** and commit it in this branch.
- 6) Visualize using GitK (yourname and yourname-1 should diverge)
- 7) Now bring all the commits of **<yourname>-1** branch to **<yourname>** branch without a merge object.
- 8) Push the **yourname** branch.
- 9) Delete the local branch **<yourname> - 1**

### **Additional**

- 11) Try Git Squash Command by making more than one commit on **<yourname>-1** branch

## **Session 3 - Exercise 2**

- 1) Clone the project from this url [git@github.com:ScalaTribelIndia/session4.git](https://github.com/ScalaTribelIndia/session4.git) in folder **/home/session4**
- 2) Create a branch from master with **<yourname>**.
- 3) Create another **<yourname>-1** branch from **<yourname>** branch.
- 4) Go to yourname branch and edit the **"file.txt"** (Don't add or commit it).
- 5) Now go to **<yourname>-1** branch (If any errors try to resolve them)
- 6) Edit the same file **"file.txt"** and commit the file.
- 7) Edit file **"file2.txt"** and commit the file (Don't Push).
- 8) Edit file **"file2.txt"** again and add a new line commit the file (Don't Push).
- 9) Go back to branch **<yourname>** branch.
- 10) Now cherry pick the first edit done on file2.txt.
- 11) Commit the **<yourname>** branch.
- 12) Now merge the **<yourname>-1** branch onto **<yourname>** branch.
- 13) See what happens ?