

001 INSTALLING GIT

002

003 1. Mac : <http://sourceforge.net/projects/git-osx-installer/>

004

005 2. Windows : <http://msysgit.github.io/>

006

007 3. Linux : apt-get install git-core OR yum install git-core

008

009 ABOUT GIT

010

011 1. Git is a version control tool that saves changes to groups of files so you can revert back if needed.

012

013 2. There are different types of version control tools

014

015 a. Local Version Control saves changes to files in a database

016

017 b. Centralized Version Control saves changes to a shared server

018

019 c. Distributed Version Control allows for easier sharing of files then LVC and also eliminates problems that could occur if access to the server is lost under a CVC system.

020

021 d. DVC clients have a complete backup of the files on their computer. If the server is lost the client just waits to regain contact and then uploads changes.

022

023 3. When you commit changes to files Git stores a reference of what the files look like at that moment. If a file isn't changed it isn't stored again.

024

025 4. Each client has a complete history of all changes stored locally. The client can also access all changes made to the files historically with a simple command. Also those files cannot be changed without Git knowing and changes are difficult to lose.

026

027 5. Files transition between 3 states with Git

028

029 a. Modified Files are files that have been recently changed

030

031 b. Staged Files have been marked to be saved

032

033 c. Committed Files are those that have been saved

034

035 6. Git saves all file changes to a directory as a compressed database.

036

```
037 a. You modify files in Working Directory
038
039 b. You notify that want to save changes in your Staging Area
040
041 c. After you Commit the file changes are saved in the Git directory
042
043 USING GIT
044
045 1. git config --global user.name "Derek Banas"
046
047 2. git config --global user.email derekbanas@verizon.net
048
049 3. git config --global core.editor "vim" # Set editor as vim
050
051 4. git config --global core.editor "edit -w" # Set editor as Text Wrangler Mac
052
053 5. git config --list # Show settings
054
055 6. git help OR git help [COMMAND] OR git help add
056
057 7. ----- Track a directory -----
058
059 a. Go to directory
060
061 b. ls -a shows all files
062
063 c. git init # Creates the .git directory
064
065 8. ----- Start tracking files -----
066
067 a. By type : git add *.java
068
069 b. By name : git add AndroidManifest.xml
070
071 9. ----- Ignore Files -----
072
073 a. Create a .gitignore file
074
075 b. https://github.com/github/gitignore
076
077 10. ----- git commit -m 'Initial project version'
078
```

```
079 a. Commits the changes and sets an abbreviated commit message
080
081 11. ----- git status -----
082
083 a. Shows the state of your files meaning if they are tracked, have been modified and the branch your on.
084
085 12. ----- Stage A Modified File -----
086
087 a. Change the file and save
088
089 b. git diff # Shows what you changed, but haven't staged
090
091 c. git add AndroidManifest.xml # Stage file
092
093 d. git diff --cached # Shows what has been staged, but not committed
094
095 13. ----- Commit The Changes -----
096
097 a. commit # Opens the editor we defined above or vi
098
099 b. In vi click [ESC] i to enter insert mode
100
101 c. Type a heading that briefly explains the changes in 50 characters or less
102
103 d. Describes the original problem that is being addressed
104
105 e. Describes the specific change being made
106
107 f. Describes the result of the change
108
109 g. Describes any future improvements
110
111 h. Post a closes bug notation Closes-Bug: #1291621
112
113 i. Hit [ESC] and type wq to save and exit
114
115 j. git commit -a -m 'Changed comment' # Skips staging and commit message
116
117 14. ----- Remove a File -----
118
119 a. rm DeleteMe.txt # If you remove a file it shows as "Changed but not updated"
120
```

```
121 b. git status # If you remove a file it shows as "Changed but not updated"
122
123 c. git rm DeleteMe.txt
124
125 d. git status # Shows that the file was deleted
126
127 e. If you have committed a file to be removed you must add the -f option
128
129 f. git rm --cached DeleteMe.txt # Keep file, but remove from staging area
130
131 g. git mv DeleteMe.txt Delete.txt # Renames a file
132
133 15. ----- Log Commit History -----
134
135 a. git log # Shows all of the previous commit messages in reverse order
136
137 b. git log --pretty=oneline # Shows commits on one line
138
139 c. git log --pretty=format:"%h : %an : %ar : %s"
140
141 I. %h - Abbreviated Hash
142
143 II. %an - Authors Name
144
145 III. %ar - Date Changed
146
147 IV. %s - First Line of Comment
148
149 d. git log -p -2 # Shows the last 2 commit changes
150
151 e. git log --stat # Prints abbreviated stats
152
153 f. git log --since=1.weeks # Show only changes in the last week
154
155 g. git log --since="2014-04-12" # Show changes since this date
156
157 h. git log --author="Derek Banas" # Changes made by author
158
159 i. git log --before="2014-04-13" # Changes made before this date
160
161 16. ----- Undoing a Commit -----
162
```

```
163 a. git commit --amend # If you want to change your previous commit
164
165 b. Normally done if you forgot to stage a file, or to change the commit message
166
167 17. ----- Unstage a File -----
168
169 a. git reset HEAD AndroidManifest.xml
```