To find out what version of Linux (distro) you are running, enter the following [cat command](https://www.cyberciti.biz/faq/linux-unix-appleosx-bsd-cat-command-examples/) at the shell prompt:

$ cat /etc/\*-release

++++++++++++++++++++++++++++++++++++++++

**Git install on linux**

sudo su - (to switch to root user)

Git

Yum install git

Git --version

mkdir git-demo

cd git-demo

mkdir project

Cd project

Git init

++++++++++++++++++++++++++++++++++++

**Working directory /Staging Area/Local Repository**

17 git status

18 clear

19 ls

20 git status

21 git add index1.html

22 git status

**git config --global user.name "your name”**

**git config --global user.email "your emaild"**

23 git commit -m "first Commit"

24 clear

27 git commit -m "first Commit"

29 clear

30 ls

31 vim index2.html

32 ls

33 git status

34 git ls-files

35 git add index2.html

36 git status

37 git commit -m "second file"

38 git status

39 ls

40 git ls-files

41 clear

42 ls

43 vim index1.html

44 git status

45 git commit -a -m "edit first file"

46 git status

47 git log

48 clear

49 git log --oneline

50 git show 28fbc14

51 clear

52 ls

53 vim file1.txt

54 cat file1.txt

55 git status

56 git add file1.txt

57 git commit -m "text file"

58 ls

59 git status

60 clear

61 vim file1.txt

62 git status

63 git diff file1.txt

64 git add file1.txt

65 git status

66 git diff --staged file1.txt

67 vim file1.txt

68 git diff file1.txt

**To unstage:**

**git restore --staged index1.html**

69 git diff --staged file1.txt

70 git status

71 git commit -a -m "txt file edited"

72 git status

73 history

74 ls

75 git ls-files

76 history

77 cat file1.txt

78 ll -al

79 cat file1.txt

80 git log --oneline

81 git show 1a65f8b

82 clear

83 ls

++++++++++++++++++++++++++++++  
**DELETING FILES**

84 git rm file1.txt

85 ls

86 git ls-files

87 git status

88 git commit -m "del txt file"

89 git status

90 ls

91 git ls-files

92 clear

93 ls

94 git rm --cached index2.html

95 ls

96 git ls-files

97 git status

98 git commit -m "del html from local repo"

99 git status

100 clear

101 git status

102 ls

+++++++++++++++++++++++++++++++  
**Git ignore**

103 vim .gitignore

104 git status

105 git add .gitignore

106 git commit -m "added ignore file"

107 git status

108 ll -al

109 git ls-files

110 clear

111 ls

112 git log --oneline

+++++++++++++++++++++++++++++++++  
**Git revert**

113 git revert 7aae155

114 git log --oneline

115 ll -al

116 git status

117 git log --oneline

118 git revert 2cdee85

119 git log --oneline

120 ll -al

121 git status

122 cat .gitignore

123 vim .gitignore

124 clear

125 git log --oneline

\_+++++++++++++++++++++++++++++

**RESET**

126 git log

127 clear

128 git log --oneline

129 git reset --hard HEAD~1

130 git log --oneline

131 git reset --hard HEAD~2

132 git log --oneline

133 git status

134 clear

135 git log --oneline

136 git reset --hard 1d2a29f

137 git log --oneline

138 git show 1d2a29f

139 clear

140 git log --oneline

+++++++++++++++++++++++++++++++

**BRANCHING/MERGING**

141 git status

142 ls

143 git branch

144 git branch b1 master

145 git branch

146 git status

147 git log --online master

148 git log --oneline master

149 ls

150 clear

151 git checkout b1

152 git branch

153 git status

154 ls

155 git log --oneline b1

156 ls

157 vim index3.html

158 git status

159 git add index3.html

160 git commit -m "on b1"

161 ls

162 git log --oneline

163 clear

164 ls

165 git branch

166 git checkout master

167 ls

168 git merge b1 master

169 ls

170 git log --oneline

171 git diff b1 master

172 clear

173 ls

174 vim index4.html

175 git status

176 git add index4.html

177 git commit -m "on master"

178 ls

179 git log --oneline

180 clear

181 git checkout b1

182 ls

183 vim index4.html

184 git add index4.html

185 git commit -m "on b1"

186 ls

187 git status

188 git checkout master

189 ls

190 clear

191 git diff master b1

192 clear

193 git merge b1 master

194 ls

195 vim index4.html

196 git add index4.html

197 git commit -m "resolved the conflicts"

198 git status

199 git merge b1 master

200 cat index4.html

201 ll -al

202 git checkout b1

203 cat index4.html

204 clear

205 ls

206 git branch

207 git checkout master

208 ls

209 clear

210 ls

++++++++++++++++++++++++++++++  
**STASH**

211 git status

212 ls

213 vim index1.html

214 vim index2.html

215 git status

**git checkout**

216 git stash

217 git status

218 clear

219 git stash list

220 git show stash@{0}

221 git status

222 clear

223 git stash list

224 git stash pop stash@{0}

225 git status

226 clear

227 git status

228 git stash

229 git status

230 git stash list

231 git show stash@{0}

232 clear

233 git stash apply stash@{0}

234 git status

235 git stash list

236 clear

237 git status

238 git stash -p

239 clear

240 git status

241 git stash list

242 git stash drop stash@{0}

243 git stash list

244 git stash clear

245 git stash list

246 history

+++++++++++++++++++++++++++++++++++

**REBASE** 247 git status

248 clear

249 git branch

250 git status

251 git commit -a -m "last"

252 git status

253 clear

254 git log --oneline

255 git reset --hard 1d2a29f

256 git log --oneline

257 clear

258 git log --oneline

259 git branch b2 master

260 git branch

261 git log --oneline b2

262 git status

263 ls

264 vim index3.html

265 ls

266 git add index3.html

267 git commit -m "on master"

268 clear

269 git log --oneline master

270 git checkout b2

271 ls

272 vim index4.html

273 git add index4.html

274 git commit -m "file on b2"

275 git log --oneline b2

276 git branch

277 git rebase master

278 git log --oneline b2

279 git log index1.html

280 clear

281 git log --help

++++++++++++++++++++++++++++++++

288 cd /git-demo/

289 ls

290 cd project/

291 ls

292 git status

293 clear

294 ls

295 git status

296 git branch

297 ll -al

298 clear

+++++++++++++++++++++++++++++++++

**Remote repos**

299 git remote add origin url

300 git remote -v

301 git push origin master

302 git push origin b1

303 clear

304 git branch

305 git branch -d b2

306 git checkout master

307 git branch -d b2

308 git branch -D b2

309 git branch

310 git branch -d b1

311 git branch -D b1

312 git branch

313 clear

314 git push origin --delete b1

316 cd ..

317 ls

318 mkdir remote-project

319 cd remote-project/

320 ls

321 ll -al

322 clear

323 ls

++++++++++++++++++++++++++++++++++++

**Cloning remote repos**

324 git clone url

325 ll -al

326 cd repo2/

327 ll -al

328 git remote -v

329 clear

330 vim index1.html

331 git status

332 git status index1.html

333 git add index1.html

334 git commit -m "first chang"

335 git status

336 clear

337 ls

338 git push origin master

339 ls

340 git pull origin

341 ls

342 clear

343 git fetch origin

344 ls

345 git status

346 git pull origin

++++++++++++++++++++++++++++++++++++++++

**Git pull Vs git pull -r (no conflict scenario)**

1)create a file and a commit on remote

2)create some other file on local and commit

3)git push origin master

4)git pull origin master

5)git push origin master

You will see an additional commit on master which is merge commit

Now replicate the same with git pull -r

1)create a file and a commit on remote

2)create some other file on local and commit

3)git push origin master

4)git pull origin master -r

5)git push origin master

You will not see additional merge commit added

++++++++++++++++++++++++++++++++++++++++

**Git pull Vs git pull -r ( conflict scenario)**

1)create a file and a commit on remote

2)create the same file on local and commit

3)git push origin master

4)git pull origin master

5)resolve the conflict, add ,commit

6)git push origin master

You will see an additional commit on master which is merge commit

Now replicate the same with git pull -r

1)create a file and a commit on remote

2)create the same file on local and commit

3)git push origin master

4)git pull origin master -r

5)vim file

6) git add file

7)git status

8)git rebase --continue

9)git push origin master