

```
1 Create a folder
2 e:\practice\GIT
3 open cmd command prompt (run the cmd.exe and traverse to the said folder)
4 type in
5 e:\practice\GIT>git --version
6 If the command returns the GIT version then GIT is installed successfully !!
7
8 Add username for GIT
9 e:\practice\GIT>git config --global user.name ameya
10 Add email for GIT
11 e:\practice\GIT>git config --global user.email amsalways@gmail.com
12 To see the above configs
13 e:\practice\GIT>git config user.name
14 e:\practice\GIT>git config user.email
15
16 To set the default text editor for git
17 e:\practice\GIT>git config --global core.editor "atom --wait"
18 -----Install
19 AtomIDE-----
20 Double click the executable installer
21 ---install and access integrated terminal.
22 In the Atom IDE:
23 Open file>settings
24 Click "+" (install)
25 Search for a terminal package called "platformio-ide-terminal" Click "install". (Needs
    Python to be installed so if not installed first install python)
26 Press Ctrl+` to toggle the terminal
27 -----Create
28 Repo-----
29 create the folders git-one and git-two within e:\practice\GIT (We will create repos
    in both the folders)
30 e:\practice\GIT>md git-one
31 e:\practice\GIT>md git-two
32
33 In Atom file-->add project folder traverse to e:\practice\GIT and add it as project
    folder
34 Traverse to git-one folder and open cmd prompt
35 (can be done by rt click folder in atom copy full path and paste in cmd prompt as
    parameter to cd )
36
37 To create a repo
38 E:\practice\GIT\git-one>git init
39 (Initialized empty Git repository in E:/practice/GIT/git-one/.git/)
40
41 create a new file index.html in git-one folder by rt click git-one in atom and select
    new file
```

42 Add some <html></html> in the file
43
44 create a file index.html in git-two
45 then from cmd prompt e:\practice\git\git-two>run git init
46 (Here we are demonstrating that a repo neednot be empty folder but existing folder
with pre-existing files can also be a repo)
47
48 -----Adding to staging
area-----
49
50 --- To check which files are in staging area etc
51
52 E:\practice\GIT\git-one>git status
53
54 This will show the files that are in modified area/ staged area etc.
55
56 --- To stage the index.html
57 E:\practice\GIT\git-one>git add index.html
58
59 --- To unstage the staged file index.html
60 E:\practice\GIT\git-one>git rm --cached index.html
61
62 This will move the index.html from staged to modified area
63
64 --- To stage multiple files from a folder
65
66 --- create a new file styles.css
67
68 body{
69 background:red;
70 }
71
72 Check using e:\practice\GIT\git-one>git status
73 Both the files will be in modified area
74
75 --- To add both the files (index.html and styles.css) to staging area
76 e:\practice\GIT\git-one>git add .
77
78 Check with e:\practice\GIT\git-one>git status
79 Both the files will be seen in staged area
80
81 ----- Committing the staging area files

82
83 --- Check the files from staging area that are to be committed
84 e:\practice\GIT\git-one>git status
85 Both the files index.html and styles.css are green color that means they are in
staging area and can be now committed.

```
86
87 --- Commit the staging area files to committed
88 E:\practice\GIT\git-one>git commit -m "index.html,styles.html
   commit_01-10/1/21-9:27am"
89 Below is the o/p
90 [master (root-commit) a345cbb] index.html,styles.html commit-10/1/21-9:27am
91  2 files changed, 9 insertions(+)
92  create mode 100644 index.html
93  create mode 100644 styles.css
94 This means we have made a commit the id of the commit is a345cbb
95 The commit is made on the master branch
96 2 files had been changed and there are 13 insertions, that is there are 13 lines of
   codes in the files we committed
97 This commit is added to commit history.
98 Lets make a couple of more commits and then view the history.
99
100 --- Modify index.html and add <title>My second commit</title> and save the file
101 --- Check the status and observe the o/p
102 E:\practice\GIT\git-one>git status
103 On branch master
104 Changes not staged for commit:
105   (use "git add <file>..." to update what will be committed)
106   (use "git checkout -- <file>..." to discard changes in working directory)
107
108       modified:   index.html
109
110 no changes added to commit (use "git add" and/or "git commit -a")
111
112 --- Now add the file to staging area and observe the status (git add . followed by
   git status shown below)
113 E:\practice\GIT\git-one>git add .
114
115 E:\practice\GIT\git-one>git status
116 On branch master
117 Changes to be committed:
118   (use "git reset HEAD <file>..." to unstage)
119
120       modified:   index.html
121
122 --- Commit the staging files
123 E:\practice\GIT\git-one>git commit -m "Added title to index.html"
124 [master 6ad7e1a] Added title to index.html
125  1 file changed, 1 insertion(+), 1 deletion(-)
126
127 --- Modify the styles.css as follows
128 body{
129     background:red;
130     font-size:2em;
```

```
131 }
132
133 --- Add the file to staging and make sure the status of files shows it in staging area
134 (git add . and git status)
135 E:\practice\GIT\git-one>git add .
136
137 E:\practice\GIT\git-one>git status
138 On branch master
139 Changes to be committed:
140   (use "git reset HEAD <file>..." to unstage)
141
142       modified:   styles.css
143
144 --- Commit the staging files
145 E:\practice\GIT\git-one>git commit -m "Changed body font-size in styles.css"
146 [master 6f3b742] Changed body font-size in styles.css
147   1 file changed, 1 insertion(+)
148
149 Now we have 3 commits done i.e. we have 3 different snapshots of commits in our
    commit history.
150
151 --- View the commit history
152 E:\practice\GIT\git-one>git log
153 commit 6f3b742b0a935ada2a98e11d6198747c55737beb (HEAD -> master)
154 Author: ameya <amsalways@gmail.com>
155 Date:   Mon Dec 10 11:03:41 2018 +0530
156
157     Changed body font-size in styles.css
158
159 commit 6ad7e1a053e2443e8088147618d385bcc7c5ed28
160 Author: ameya <amsalways@gmail.com>
161 Date:   Mon Dec 10 10:57:37 2018 +0530
162
163     Added title to index.html
164
165 commit a22b2f5027a3c5938c97b65b7c2d00a141513268
166 Author: ameya <amsalways@gmail.com>
167 Date:   Mon Dec 10 10:22:38 2018 +0530
168
169     Added index and styles files
170
171 Observe the o/p You may see that the id which is very long, the first 7 characters
    are same as the ids we saw when we actually committed the files. Those were the
    short version of the ids. These are the full versions of the ids.
172
173 --- View the condensed (online) version of commit history
174 E:\practice\GIT\git-one>git log --oneline
175 6f3b742 (HEAD -> master) Changed body font-size in styles.css
```

```
176 6ad7e1a Added title to index.html
177 a22b2f5 Added index and styles files
178
179 --- Few more commits (necessary to show next topic)
180 --- Add the new file feature.js
181 console.log('hey there');
182
183 --- Add to staging area
184 e:\practice\GIT\git-one>git add .
185 --- Commit
186 e:\practice\GIT\git-one>git commit -m "Added new feature"
187
188 --- Add the new <h1>Added new Heading<h1> to index.html
189 --- Add to staging area
190 e:\practice\GIT\git-one>git add .
191 --- Commit
192 e:\practice\GIT\git-one>git commit -m "Added new heading"
193
194 ----- Undoing the commits
195 -----
196
196 --- Now let us say I want to see the state of the code in earlier commit lets say
added title to index file
197
198 E:\practice\GIT\git-one>git checkout 6ad7e1a
199 Note: checking out '6ad7e1a'.
200
201 You are in 'detached HEAD' state. You can look around, make experimental
202 changes and commit them, and you can discard any commits you make in this
203 state without impacting any branches by performing another checkout.
204
205 If you want to create a new branch to retain commits you create, you may
206 do so (now or later) by using -b with the checkout command again. Example:
207
208     git checkout -b <new-branch-name>
209
210 HEAD is now at 6ad7e1a Added title to index.html
211
212 Observe the files they will have been altered as per the commit
213
214 --- To go back to previous state before we checkedout
215 E:\practice\GIT\git-one>git checkout master (Dont worry too much about branch
name master we are going to see it later)
216 Previous HEAD position was 6ad7e1a Added title to index.html
217 Switched to branch 'master'
218 Observe the files again they will have been restored.
219
220 When we checkedout to a particular commit we had detached ourself from master
```

branch and this command re-attaches us to master branch

```
221
222 --- Undo using revert (remember revert undoes one particular commit)
223 --- Add two more commits
224 add new file feature.js
225 git add .
226 git commit -m "Added new Feature"
227
228 modify index.html and add <h1>This is new heading</h1>
229 git add .
230 git commit -m "Added new heading to index.html"
231
232 --- Check the log
233 E:\practice\GIT\git-one>git log --oneline
234 005ee46 (HEAD -> master) Added new heading
235 18f3e6a Added new feature
236 6f3b742 Changed body font-size in styles.css
237 6ad7e1a Added title to index.html
238 a22b2f5 Added index and styles files
239
240 --- undo using revert
241 E:\practice\GIT\git-one>git revert 18f3e6a
242
243 --- Now check logs
244 E:\practice\GIT\git-one (master -> origin)
245 ? git log --oneline
246 d08f909 (HEAD -> master) Revert "Added new feature"
247 2b4a958 Added new Heading to index
248 8a592fd Added new feature
249 f856672 Changed body fonst-size in styles.css
250 d1517ec Added title to index.html
251 4804ecc Added index and styles files
252
253 This shows exactly what has happened. Though we dont see the feature.js file
    actually it is there and just that commit has been reverted by adding a new revert
    commit.
254
255 --- undo using reset (Here we reset to Added title to index.html)
256 E:\practice\GIT\git-one (master -> origin)
257 ? git log --oneline
258 d08f909 (HEAD -> master) Revert "Added new feature"
259 2b4a958 Added new Heading to index
260 8a592fd Added new feature
261 f856672 Changed body fonst-size in styles.css
262 d1517ec Added title to index.html
263 4804ecc Added index and styles files
264
265 E:\practice\GIT\git-one (master -> origin)
```

```
266 ? git reset d1517ec
267 Unstaged changes after reset:
268 M      index.html
269 M      styles.css
270
271 E:\practice\GIT\git-one (master -> origin)
272 ? git log --oneline
273 d1517ec (HEAD -> master) Added title to index.html
274 4804ecc Added index and styles files
275
276 Although you see the logs have been reset to the Added title to index.html we still
    see the contents in the files which we had done in later commits.
277 This is sort of soft reset
278
279 --- If you want to reset permanently i.e. hard reset check below command
280 E:\practice\GIT\git-one (master -> origin)
281 ? git reset d1517ec --hard
282 HEAD is now at d1517ec Added title to index.html
283
284 E:\practice\GIT\git-one (master -> origin)
285 ? git log --oneline
286 d1517ec (HEAD -> master) Added title to index.html
287 4804ecc Added index and styles files
288
289 --- Second time
290 E:\practice\GIT\git-one (master -> origin)
291 ? git log --oneline
292 d1517ec (HEAD -> master) Added title to index.html
293 4804ecc Added index and styles files
294
295 E:\practice\GIT\git-one (master -> origin)
296 ? git reset 4804ecc --hard
297 HEAD is now at 4804ecc Added index and styles files
298
299 This will remove the title we added in reset commit.
300
301 However Now the commits have been permanently reset and there is no way to get
    the contents back.
302
```