

```
1 ----- Creating Branches
2 -----
3 --- check the git log
4 E:\practice\GIT\git-one (master -> origin)
5 ? git log --oneline
6 4804ecc (HEAD -> master) Added index and styles files
7
8 There is only one commit.
9
10 --- update index.html add <title>This is for Branches topic</title>
11
12 --- add the file to staging and then commit (Observe the color of master->origin
    after changing the index.html and later after committing)
13 E:\practice\GIT\git-one (master -> origin)
14 ? git add .
15
16 E:\practice\GIT\git-one (master -> origin)
17 ? git commit -m "added index title"
18 [master d9ea7b1] added index title
19 1 file changed, 1 insertion(+), 1 deletion(-)
20
21 --- Check the logs
22 E:\practice\GIT\git-one (master -> origin)
23 ? git log --oneline
24 d9ea7b1 (HEAD -> master) added index title
25 4804ecc Added index and styles files
26
27 --- Create a new branch feature-1
28 E:\practice\GIT\git-one (master -> origin)
29 ? git branch feature-1
30
31 --- View all the branches
32 E:\practice\GIT\git-one (master -> origin)
33 ? git branch -a
34 feature-1
35 * master
36
37 The asterisk prior to master denotes that you are currently on that branch.
38
39 --- Switch to the feature-1 branch from current branch
40 E:\practice\GIT\git-one (master -> origin)
41 ? git checkout feature-1
42 Switched to branch 'feature-1'
43
44 --- View all the branches
45 E:\practice\GIT\git-one (feature-1 -> origin)
46 ? git branch -a
```

```
47 * feature-1
48   master
49
50 --- Add new file feature-1.js as follows
51 console.log('Feature 1');
52
53 (Note that now we are on the feature-1 branch not on master as we already
   switched to this branch)
54
55 --- Check the status
56 E:\practice\GIT\git-one (feature-1 -> origin)
57 ? git status
58 On branch feature-1
59 Untracked files:
60   (use "git add <file>..." to include in what will be committed)
61
62       feature-1.js
63
64 nothing added to commit but untracked files present (use "git add" to track)
65
66 --- Stage and Commit the code to this (feature-1) branch
67 E:\practice\GIT\git-one (feature-1 -> origin)
68 ? git add .
69
70 E:\practice\GIT\git-one (feature-1 -> origin)
71 ? git commit -m "Added New Feature File"
72 [feature-1 69ceafc] Added New Feature File
73 1 file changed, 1 insertion(+)
74 create mode 100644 feature-1.js
75
76 --- To confirm that we are indeed working in an isolated environment go back to
   master branch from fetature-1
77 E:\practice\GIT\git-one (feature-1 -> origin)
78 ? git checkout master
79 Switched to branch 'master'
80
81 Observe in Atom editor You no longer have the javascript file that was created
   however on the feature-1 branch.
82
83 --- Again go to feature-1 branch and observe the Atom editor
84 E:\practice\GIT\git-one (master -> origin)
85 ? git checkout feature-1
86 Switched to branch 'feature-1'
87
88 --- Show the differences in the files in various branches
89 --- Switch to master
90 E:\practice\GIT\work\git-one (master -> origin)
91 ? git checkout master
```

```
92 --- run git diff as below
93 git diff master..feature-1
94 diff --git a/feature-1.js b/feature-1.js
95 new file mode 100644
96 index 0000000..f945880
97 --- /dev/null
98 +++ b/feature-1.js
99 @@ -0,0 +1 @@
100 +console.log('Added new Feature-1')
101
102 --- run the git diff command
103 E:\practice\GIT\work\git-one (master -> origin)
104 ? git diff feature-1
105 diff --git a/feature-1.js b/feature-1.js
106 deleted file mode 100644
107 index 6a6ae5c..0000000
108 --- a/feature-1.js
109 +++ /dev/null
110 @@ -1 +0,0 @@
111 -console.log('Added new feature-1');
112
113
114 --- Delete the branch feature-1 (Let us say that I dont need this branch)
115 --- First goto master branch
116 E:\practice\GIT\git-one (feature-1 -> origin)
117 ? git checkout master
118 Switched to branch 'master'
119
120 --- Delete the feature-1 branch
121 E:\practice\GIT\git-one (master -> origin)git checkout master
122
123 ? git branch -D feature-1
124 Deleted branch feature-1 (was 69ceafc).
125
126 As the feature-1 branch is not yet merged (which we will see later) we are using -D
(in Caps)flag else we could use -d (in lowercase)flag
127
128 --- view all branches
129 E:\practice\GIT\git-one (master -> origin)
130 ? git branch -a
131 * master
132
133 --- create and checkout to a branch (feature-a) using one command
134 E:\practice\GIT\git-one (master -> origin)
135 ? git checkout -b feature-a
136 Switched to a new branch 'feature-a'
137
138 E:\practice\GIT\git-one (feature-a -> origin)
```

```
139 ?
140
141 --- add new file feature-a.js as follows
142 console.log('Feature a');
143
144 --- stage and commit the code
145 E:\practice\GIT\git-one (feature-a -> origin)
146 ? git add .
147
148 E:\practice\GIT\git-one (feature-a -> origin)
149 ? git commit -m "Added feature-a File"
150 [feature-a 8b4b466] Added feature-a File
151 1 file changed, 1 insertion(+)
152 create mode 100644 feature-a.js
153
154 --- create and checkout to a branch (feature-b) using one command
155 --- first checkout to master and then create the other branch
156 E:\practice\GIT\git-one (feature-a -> origin)
157 ? git checkout master
158 Switched to branch 'master'
159
160 E:\practice\GIT\git-one (master -> origin)
161 ? git checkout -b feature-b
162 git cSwitched to a new branch 'feature-b'
163
164 --- Add new feature-b.js as follows
165 console.log('Feature b');
166
167 --- Stage and commit
168 E:\practice\GIT\git-one (master -> origin)
169 ? git add .
170
171 E:\practice\GIT\git-one (master -> origin)
172 ? git commit -m "Added feature-b File"
173 [master 3b8fb7c] Added feature-b File
174 1 file changed, 1 insertion(+)
175 create mode 100644 feature-b.js
176
177 --- Now one by one checkout to each of the branch and observe the Atom editor
178 E:\practice\GIT\git-one (feature-b -> origin)
179 ? git checkout master
180 Switched to branch 'master'
181
182 E:\practice\GIT\git-one (master -> origin)
183 ? git checkout feature-a
184 git Switched to branch 'feature-a'
185
186 E:\practice\GIT\git-one (feature-a -> origin)
```

```
187 ? git checkout feature-b
188 Switched to branch 'feature-b'
189
190 ----- Merge the branches -----
191
192 --- Switch to the branch that you want to merge into (In this case master)
193 E:\practice\GIT\git-one (feature-b -> origin)
194 ? git checkout master
195 Switched to branch 'master'
196
197 --- Now merge feature-a with the master
198 E:\practice\GIT\git-one (master -> origin)
199 ? git merge feature-a
200 Updating d9ea7b1..8b4b466
201 Fast-forward -->becoz master branch was already unaltered at the time we merged
202 feature-a.js | 1 +
203 1 file changed, 1 insertion(+)
204 create mode 100644 feature-a.js
205
206 Observe the Atom editor feature-a.js will be now there in master branch
207
208 --- Now merge feature-b with the master
209 E:\practice\GIT\git-one (master -> origin)
210 ? git merge feature-b
211 Merge made by the 'recursive' strategy. -->becoz master branch ahs been now
    modified as a result of previous merge
212 feature-b.js | 1 +
213 1 file changed, 1 insertion(+)git branch -a
214 create mode 100644 feature-b.js
215
216 Observe the Atom editor feature-b.js will also be there now in master branch
217
218 --- Handling conflicts
219 --- create and checkout a new branch feature-c
220 E:\practice\GIT\git-one (master -> origin)
221 ? git checkout -b feature-c
222 Switched to a new branch 'feature-c'
223
224 --- Switch back to master
225 E:\practice\GIT\git-one (feature-c -> origin)
226 ? git checkout master
227 Switched to branch 'master'
228
229 Let us say a team member decide to add margin to styles.css He does it so and
    stages and commits it
230
231 --- Add following to styles.css
232 body{
```

```
233     background:red;
234     margin: 20px;
235 }
236
237 --- stage and commit on master branch
238 E:\practice\GIT\git-one (master -> origin)
239 ? git add .
240
241 E:\practice\GIT\git-one (master -> origin)
242 ? git commit -m "Added margin to selector to styles.css"
243 [master 7e51798] Added margin to selector to styles.css
244 1 file changed, 1 insertion(+)
245
246 In meanwhile I am say working in feature-c and am unaware of the additions made to
    styles.css on master branch
247 and I add a padding to styles.css in feature-c
248 --- Switch to feature-c
249 E:\practice\GIT\git-one (master -> origin)
250 ? git checkout feature-c
251 Switched to branch 'feature-c'
252
253 --- add following to styles.css
254 body{
255     background:red;
256     padding: 30px;
257 }
258
259 --- stage and commit the code on feature-c branch
260 E:\practice\GIT\git-one (feature-c -> origin)
261 ? git add .
262
263 E:\practice\GIT\git-one (feature-c -> origin)
264 ? git commit -m "Added padding to styles.css"
265 [feature-c 6a14f36] Added padding to styles.css
266 1 file changed, 1 insertion(+)
267
268 Now I want to merge this back to the master so...
269
270 --- switch to master
271 E:\practice\GIT\git-one (feature-c -> origin)
272 ? git checkout master
273 Switched to branch 'master'
274
275 --- merge feature-c with the master branch
276 E:\practice\GIT\git-one (master -> origin)
277 ? git merge feature-c
278 Auto-merging styles.css
279 CONFLICT (content): Merge conflict in styles.css
```

```
280 Automatic merge failed; fix conflicts and then commit the result.
281
282 OOOHHH !!! We have a conflict here... becoz in both the branches (master and
    feature-c) the
283 developers have been working on same body selector from the same file styles.css
    and so you get a
284 message as on the screen that tells you there is a conflict and in which file the
    conflict occurs and also
285 quite some more lines in styles.css (watch them in Atom editor)
286 Here git cannot resolve the ambiguity of simultaneous modification of same files in a
    way.
287 So how we resolve this
288 --- first Bring the styles.css in the form you want by removing the "git added code"
289
290 i.e.remove the things u dont want
291 body{
292     background:red;
293 <<<<<< HEAD
294     margin: 20px;
295 =====
296     padding: 30px;
297 >>>>>> feature-c
298 }
299
300 that will make the file as below
301 git merge body{
302     background:red;
303     margin: 20px;
304     padding: 30px;
305 }
306
307 --- stage the code
308 E:\practice\GIT\git-one (master -> origin)
309 ? git add .
310
311 --- commit to master branch (this time it is a merge commit so no need to provide
    any message)
312 E:\practice\GIT\git-one (master -> origin)
313 ? git commit
314 hint: Waiting for your editor to close the file...
315 [master 991256c] Merge branch 'feature-c'
316
317 You might see some message file opened in Atom editor just informative message that
    can be closed.
318
319 --- check the git log and go thru it
320 E:\practice\GIT\git-one (master -> origin)
321 ? git log --oneline
```

```
322 991256c (HEAD -> master) Merge branch 'feature-c'
323 6a14f36 (feature-c) Added padding to styles.css
324 7e51798 Added margin to selector to styles.css
325 3ec5518 Merge branch 'feature-b'
326 8ded181 (feature-b) Added feature-b File
327 48eb19f mistake
328 3b8fb7c Added feature-b File
329 8b4b466 (feature-a) Added feature-a File
330 d9ea7b1 added index title
331 4804ecc Added index and styles files
332
333 You may come with maybe more complex conflicts...but then it is all about
    communicating with the concerned team member and deciding what the code shud be
    in the end.
334
335 ----- git rebase
    -----
336 goto git-two repo local using cd
337
338 --- stage and commit on master
339 E:\practice\GIT\git-two (master -> origin)
340 ? git add .
341
342 E:\practice\GIT\git-two (master -> origin)
343 ? git commit -m "First Commit"
344 [master (root-commit) c07bd2a] First Commit
345 1 file changed, 10 insertions(+)
346 create mode 100644 index.html
347
348 --- check log
349 E:\practice\GIT\git-two (master -> origin)
350 ? git log --oneline
351 c07bd2a (HEAD -> master) First Commit
352
353 --- Modify the index.html add <title>This is for rebase demo</title>
354 ---stage and commit to master
355 E:\practice\GIT\git-two (master -> origin)
356 ? git add .
357
358 E:\practice\GIT\git-two (master -> origin)
359 ? git commit -m "Added title to index html"
360 [master 078b6b1] Added title to index html
361 1 file changed, 1 insertion(+), 1 deletion(-)
362
363 --- check log again
364 E:\practice\GIT\git-two (master -> origin)
365 ? git log --oneline
366 078b6b1 (HEAD -> master) Added title to index html
```



```
367 c07bd2a First Commit
368
369 --- create and checkout to feature branch
370 E:\practice\GIT\git-two (master -> origin)
371 ? git checkout -b feature
372 Switched to a new branch 'feature'
373
374 --- create a new feature.js file
375 --- stage and commit to feature branch
376 E:\practice\GIT\git-two (feature -> origin)
377 ? git add .
378
379 E:\practice\GIT\git-two (feature -> origin)
380 ? git commit -m "added feature.js"
381 [feature 5bf0fef] added feature.js
382 1 file changed, 0 insertions(+), 0 deletions(-)
383 create mode 100644 feature.js
384
385 --- edit feature.js and add some code
386 console.log('rebase demo');
387
388 ---stage and commit to feature branch
389 E:\practice\GIT\git-two (feature -> origin)
390 ? git add .
391
392 E:\practice\GIT\git-two (feature -> origin)
393 ? git commit -m "edited the feature.js"
394 [feature 5492073] edited the feature.js
395 1 file changed, 1 insertion(+)
396
397 --- check log on feature branch
398 E:\practice\GIT\git-two (feature -> origin)
399 ? git log --oneline
400 5492073 (HEAD -> feature) edited the feature.js
401 5bf0fef added feature.js
402 078b6b1 (master) Added title to index html
403 c07bd2a First Commit
404
405 --- Check log for master as well as feature branches Observe the head for both
406 E:\practice\GIT\git-two (feature -> origin)
407 ? git checkout master
408 Switched to branch 'master'
409
410 E:\practice\GIT\git-two (master -> origin)
411 ? git log --oneline
412 078b6b1 (HEAD -> master) Added title to index html
413 c07bd2a First Commit
414
```

```
415 E:\practice\GIT\git-two (master -> origin)
416 ? git checkout feature
417 Switched to branch 'feature'
418
419 E:\practice\GIT\git-two (feature -> origin)
420 ? git log --oneline
421 5492073 (HEAD -> feature) edited the feature.js
422 5bf0fef added feature.js
423 078b6b1 (master) Added title to index.html
424 c07bd2a First Commit
425
426 --- switch to master
427 E:\practice\GIT\git-two (feature -> origin)
428 ? git checkout master
429 Switched to branch 'master'
430
431 --- edit index.html and add <h1>Some heading</h1>
432 --- stage and commit
433 E:\practice\GIT\git-two (master -> origin)
434 ? git add .
435
436 E:\practice\GIT\git-two (master -> origin)
437 ? git commit -m "Another heading to index.html"
438 [master 755a9b8] Another heading to index.html
439 1 file changed, 1 insertion(+), 1 deletion(-)
440
441 --- check the log
442 --- switch to feature branch and check log
443 E:\practice\GIT\git-two (feature -> origin)
444 ? git log --oneline
445 5492073 (HEAD -> feature) edited the feature.js
446 5bf0fef added feature.js
447 078b6b1 Added title to index.html
448 c07bd2a First Commit
449
450 --- rebase feature branch onto master
451 E:\practice\GIT\git-two (feature -> origin)
452 ? git rebase master
453 First, rewinding head to replay your work on top of it...
454 Applying: added feature.js
455 Applying: edited the feature.js
456
457 --- check log for feature branch
458 E:\practice\GIT\git-two (feature -> origin)
459 ? git log --oneline
460 4a58673 (HEAD -> feature) edited the feature.js
461 923fcc3 added feature.js
462 755a9b8 (master) Another heading to index.html
```

```
463 078b6b1 Added title to index.html
464 c07bd2a First Commit
465
466 Observe the difference
467 You will see the commits from master branch done afterwards are also seen
468
469 --- switch to master
470 --- check log
471 E:\practice\GIT\git-two (feature -> origin)
472 ? git checkout master
473 Switched to branch 'master'
474
475 E:\practice\GIT\git-two (master -> origin)
476 ? git log --oneline
477 755a9b8 (HEAD -> master) Another heading to index.html
478 078b6b1 Added title to index.html
479 c07bd2a First Commit
480
481 but the above has not yet been applied to master branch
482
483 --- Now merge the feature with master
484 E:\practice\GIT\git-two (master -> origin)
485 ? git merge feature
486 Updating 755a9b8..4a58673
487 Fast-forward
488  feature.js | 1 +
489  1 file changed, 1 insertion(+)
490  create mode 100644 feature.js
491
492 --- Check the log
493 E:\practice\GIT\git-two (master -> origin)
494 ? git log --oneline
495 4a58673 (HEAD -> master, feature) edited the feature.js
496 923fcc3 added feature.js
497 755a9b8 Another heading to index.html
498 078b6b1 Added title to index.html
499 c07bd2a First Commit
500
501 Also check in Atom the new files and contents will be visible
502
503 ----- git cherry-pick
504 -----
505
506 As discussed in the slides let us say that I want to apply changes from one commit
507 from feature branch
508 to a new branch say cherrypick-feature
509
510 --- goto git-two and switch to feature branch
```

```
509 E:\practice\GIT\git-two (master -> origin)
510 ? git checkout feature
511 Switched to branch 'feature'
512
513 --- modify the fetaure.js
514 console.log('Cherrypicking');
515
516 --- stage and commit the changes
517 E:\practice\GIT\git-two (feature -> origin)
518 ? git add .
519
520 E:\practice\GIT\git-two (feature -> origin)
521 ? git commit -m "Added the feature for cherrypicking"
522 [feature f6660a0] Added the feature for cherrypicking
523 1 file changed, 2 insertions(+), 1 deletion(-)
524
525 --- check the log for commits
526 E:\practice\GIT\git-two (feature -> origin)
527 ? git log --oneline
528 f6660a0 (HEAD -> feature) Added the feature for cherrypicking
529 4a58673 (master) edited the feature.js
530 923fcc3 added feature.js
531 755a9b8 Another heading to index.html
532 078b6b1 Added title to index html
533 c07bd2a First Commit
534
535 --- check the staus so as there is nothing to commit and work tree is clean
536 E:\practice\GIT\git-two (feature -> origin)
537 ? git status
538 On branch feature
539 nothing to commit, working tree clean
540
541 ---switch back to master
542 E:\practice\GIT\git-two (feature -> origin)
543 ? git checkout master
544 Switched to branch 'master'
545
546 Now base the new branch on the master It will not have the latest commit as the
    feature branch is not yet merged with master
547
548 --- check log on master
549 E:\practice\GIT\git-two (master -> origin)
550 ? git log --oneline
551 4a58673 (HEAD -> master) edited the feature.js
552 923fcc3 added feature.js
553 755a9b8 Another heading to index.html
554 078b6b1 Added title to index html
555 c07bd2a First Commit
```

```
556
557
558 --- create and switch to new branch cherrypick-feature (this shud be in sync with
      master)
559 E:\practice\GIT\git-two (feature -> origin)
560 ? git checkout -b cherrypick-feature master
561 Switched to a new branch 'cherrypick-feature'
562
563 See that this new branch will have all the files as per the last commit in master branch
564 But as discussed I want that this new branch must be in state as when the Another
      heading to index.html commit was done in feature branch.
565
566 --- check logs
567 E:\practice\GIT\git-two (cherrypick-feature -> origin)
568 ? git log --oneline
569 4a58673 (HEAD -> cherrypick-feature, master) edited the feature.js
570 923fcc3 added feature.js
571 755a9b8 Another heading to index.html
572 078b6b1 Added title to index html
573 c07bd2a First Commit
574
575 Also see this in atom editor
576
577 --- switch to feature branch and check log
578 E:\practice\GIT\git-two (cherrypick-feature -> origin)
579 ? git checkout feature
580 Switched to branch 'feature'
581
582 E:\practice\GIT\git-two (feature -> origin)
583 ? git log --oneline
584 f6660a0 (HEAD -> feature) Added the feature for cherrypicking
585 4a58673 (master, cherrypick-feature) edited the feature.js
586 923fcc3 added feature.js
587 755a9b8 Another heading to index.html
588 078b6b1 Added title to index html
589 c07bd2a First Commit
590
591 --- Now switch back to cherrypick-feature branch
592 E:\practice\GIT\git-two (feature -> origin)
593 ? git checkout cherrypick-feature
594 Switched to branch 'cherrypick-feature'
595
596 --- cherry-pick the commit "Added the feature for cherrypicking" using the sha id
      from the log for that commit on the cherrypick-feature branch
597 E:\practice\GIT\git-two (cherrypick-feature -> origin)
598 ? git cherry-pick f6660a0
599 [cherrypick-feature 80840a8] Added the feature for cherrypicking
600 Date: Mon Dec 17 11:06:59 2018 +0530
```

```
601 1 file changed, 2 insertions(+), 1 deletion(-)
602
603 --- see the change in atom as well as check log
604 E:\practice\GIT\git-two (cherrypick-feature -> origin)
605 ? git log --oneline
606 80840a8 (HEAD -> cherrypick-feature) Added the feature for cherrypicking
607 4a58673 (master) edited the feature.js
608 923fcc3 added feature.js
609 755a9b8 Another heading to index.html
610 078b6b1 Added title to index.html
611 c07bd2a First Commit
612
613 ----- Using Github/bitbucket -----
614
615 --- Signup to github/bitbucket
616 --- Sign in to github/bitbucket
617
618 --- Push existing local repo to github/bitbucket
619 Sign in to github/bitbucket
620 From rt top corner dropdown select your profile option
621 select repositories tab
622 click on new button
623 Repository name = git-one (just for being in sync with local one as its name is git-one)
624 Choose the public/private option
625 we select public
626 and also if you want readme file
627 gitignore file and a license file
628 Don't Forget to give branch name as master (By default it may be main)
629 click on create repository button
630 on the page that comes copy the url from the textbox
631 https://github.com/amsalways/git-one.git
632
633 --- to push to remote first ensure all the code is committed in local repo
634 E:\practice\GIT\git-one (master -> origin)
635 ? git status
636 On branch master
637 nothing to commit, working tree clean
638
639 --- push the code to the remote repo (A pop-up asking your github/bitbucket
credentials might be there just provide them)
640 E:\practice\GIT\git-one (master -> origin)
641 ? git push https://github.com/amsalways/git-one.git master
642 Enumerating objects: 28, done.
643 Counting objects: 100% (28/28), done.
644 Delta compression using up to 4 threads
645 Compressing objects: 100% (22/22), done.
646 Writing objects: 100% (28/28), 2.52 KiB | 645.00 KiB/s, done.
647 Total 28 (delta 8), reused 0 (delta 0)
```

```
648 remote: Resolving deltas: 100% (8/8), done.
649 remote:
650 remote: Create a pull request for 'master' on GitHub by visiting:
651 remote:      https://github.com/amsalways/git-one/pull/new/master
652 remote:
653 To https://github.com/amsalways/git-one.git
654 * [new branch]      master -> master
655
656 --- Check the remote repo by clicking on the name of the repo on top of the page u
must see all the files we pushed from local repo
657 --- U can check the commits by clicking the commits tab
658 --- U can see the branches by clicking the branches tab
659
660
661
662 --- Modify the local repo and again push to remote
663
664
665 --- modify index.html and add <h1>some msg</h1> within the body tag
666
667 --- stage and commit (to local repo)
668 E:\practice\GIT\git-one (master -> origin)
669 ? git add .
670
671 E:\practice\GIT\git-one (master -> origin)
672 ? git commit -m "Added Heading h1 to index.html"
673 [master 6dddf7] Added Heading h1 to index.html
674 1 file changed, 1 insertion(+), 1 deletion(-)
675
676 --- create alias to remote repo (Normally default alias in github is origin)
677 E:\practice\GIT\git-one (master -> origin)
678 ? git remote add git-one-remote https://github.com/amsalways/git-one.git
679
680 Here the alias name is git-one-remote and this will be mapped to the remote url so
u may not give the remote url everytime
681
682 --- push to remote using alias
683 E:\practice\GIT\git-one (master -> origin)
684 ? git remote add git-one-remote https://github.com/amsalways/git-one.git
685
686 E:\practice\GIT\git-one (master -> origin)
687 ? git push git-one-remote master
688 Enumerating objects: 5, done.
689 Counting objects: 100% (5/5), done.
690 Delta compression using up to 4 threads
691 Compressing objects: 100% (3/3), done.
692 Writing objects: 100% (3/3), 335 bytes | 335.00 KiB/s, done.
693 Total 3 (delta 2), reused 0 (delta 0)
```

```
694 remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
695 To https://github.com/amsalways/git-one.git
696   991256c..6dddfd7 master -> master
697 Goto remote repo and check the index.html
698 also see the commits value has changed
699
700
701 --- push to remote repo where no local repo exists
702 --- create a new remote repo with name git-one-nolocalrepo
703 --- check the initialize with readme file checkbox
704 --- click on create repository
705 --- Now as there is no existing local repo u will have to clone this repo to ur machine
706 so click on clone or download button and copy the url
707
708 --- clone it to local (first come out of existing local repo)
709 E:\practice\GIT\git-one (master -> origin)
710 ? cd ..
711
712 E:\practice\GIT
713 ? git clone https://github.com/amsalways/git-one-nolocalrepo.git
714 Cloning into 'git-one-nolocalrepo'...
715 remote: Enumerating objects: 3, done.
716 remote: Counting objects: 100% (3/3), done.
717 remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
718 Unpacking objects: 100% (3/3), done.
719
720 --- Change to this new repo
721 E:\practice\GIT
722 ? cd git-one-nolocalrepo\
723
724 --- In Atom editor Add project folder git-one-nolocalrepo
725
726 --- create a new file index.html
727 populate the title with some value
728
729 --- stage and commit to git-one-nolocalrepo local
730 E:\practice\GIT\git-one-nolocalrepo (master -> origin) ? git add
731 .
732 E:\practice\GIT\git-one-nolocalrepo (master -> origin)
733 ? git commit -m "Added index file"
734 [master 128d137] Added index file
735   1 file changed, 10 insertions(+)
736   create mode 100644 index.html
737
738 --- Now push to remote repo git-one-nolocalrepo
739 (While pushing as we have already cloned this local repo from the remote, git has
provided us with default alias "origin" we will use that)
```



```
740
741 --- check the alias provided by git
742 E:\practice\GIT\git-one-nolocalrepo (master -> origin)
743 ? git remote -v
744 origin https://github.com/amsalways/git-one-nolocalrepo.git (fetch)
745 origin https://github.com/amsalways/git-one-nolocalrepo.git (push)
746
747 --- push to remote repo
748 E:\practice\GIT\git-one-nolocalrepo (master -> origin)
749 ? git push origin master
750 Enumerating objects: 4, done.
751 Counting objects: 100% (4/4), done.
752 Delta compression using up to 4 threads
753 Compressing objects: 100% (3/3), done.
754 Writing objects: 100% (3/3), 395 bytes | 395.00 KiB/s, done.
755 Total 3 (delta 0), reused 0 (delta 0)
756 To https://github.com/amsalways/git-one-nolocalrepo.git
757    eb4645e..128d137  master -> master
758
759 --- Check in the remote repo the files will be pushed
760
761 -----Commit Java App to Git local
and push to remote thru eclipse-----
762
763 Step 1
764 Create GitHub account and SignIn
765 Step 2
766 Start a Project = Create a repository
767 Step 3
768 Start Eclipse
769 Step 4
770 Goto Perspective - Git Repositories and click on Add Git Repo
771 Step 5
772 Create a project in Eclipse
773 Step 6
774 Do a right click on Project - Team - Share - Add to git repo
775 Step 7
776 Commit and Push the project to the repo --- Provide the remote url,
username/password while pushing
777 Step 8
778 Commit and Push every change to the repo
779
```