# KataKoda Scenarios

```
        hello-world.js

nothing added to commit but untracked files present (use "git add" to track)
$
$ git add hello-world.js
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   hello-world.js

$ git commit -m "Added hello-world.js"
[master (root-commit) 9484ef9] Added hello-world.js
 1 file changed, 1 insertion(+)
 create mode 100644 hello-world.js
$ git status
On branch master
nothing to commit, working tree clean
$ vi .gitignore
$ cat .gitignore
*.tmp

$ touch file.tmp
$ ls
file.tmp  hello-world.js
$ git add file.tmp
The following paths are ignored by one of your .gitignore files:
file.tmp
Use -f if you really want to add them.
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .gitignore

nothing added to commit but untracked files present (use "git add" to track)
$ git add -A
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   .gitignore

$ git commit -m "recent update"
[master c5f007a] recent update
 1 file changed, 2 insertions(+)
 create mode 100644 .gitignore
$
```

```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   committed.js

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        untracked.js

no changes added to commit (use "git add" and/or "git commit -a")
$ git diff
ESC[1mdiff --git a/committed.js b/committed.jsESC[m
ESC[1mindex 12e7e7c..fc77969 100644ESC[m
ESC[1m--- a/committed.jsESC[m
ESC[1m+++ b/committed.jsESC[m
ESC[36m@@ -1 +1 @@ESC[m
ESC[31m-console.log("Committed File")ESC[m
ESC[32m+ESC[mESC[32mconsole.log("Demostrating changing a committed file")ESC[m
$ git add --all
$ git diff
$ git commit -m "added untracked.js and modified committed.js"
[master ef3eee5] added untracked.js and modified committed.js
 2 files changed, 2 insertions(+), 1 deletion(-)
 create mode 100644 untracked.js
$ git log
ESC[33mcommit ef3eee5d7278aad7841895ac83f169a009aa5731ESC[mESC[33m (ESC[mESC[1;36mHEAD -> ESC[mESC[1;32mmasterESC[mESC[33m)ESC[m
Author: Katacoda Scenario <scenario@katacoda.com>
Date:   Tue Aug 3 18:44:39 2021 +0000

    added untracked.js and modified committed.js

ESC[33mcommit 72037567cdfcf398be8e1fdf0e25459534a64b62ESC[m
Author: Katacoda Scenario <scenario@katacoda.com>
Date:   Tue Aug 3 18:43:33 2021 +0000

    Initial Commit
$ git show
ESC[33mcommit ef3eee5d7278aad7841895ac83f169a009aa5731ESC[mESC[33m (ESC[mESC[1;36mHEAD -> ESC[mESC[1;32mmasterESC[mESC[33m)ESC[m
Author: Katacoda Scenario <scenario@katacoda.com>
Date:   Tue Aug 3 18:44:39 2021 +0000

    added untracked.js and modified committed.js

ESC[1mdiff --git a/committed.js b/committed.jsESC[m
ESC[1mindex 12e7e7c..fc77969 100644ESC[m
ESC[1m--- a/committed.jsESC[m
ESC[1m+++ b/committed.jsESC[m
ESC[36m@@ -1 +1 @@ESC[m
ESC[31m-console.log("Committed File")ESC[m
ESC[32m+ESC[mESC[32mconsole.log("Demostrating changing a committed file")ESC[m
```

```
This is your command line, a safe place to practice & complete the scenario

$ git remote add origin /s/remote-project/1
$ git push origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 228 bytes | 228.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To /s/remote-project/1
 * [new branch]      master -> master
$ git pull origin master
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (4/4), done.
From /s/remote-project/1
 * branch            master     -> FETCH_HEAD
   0308dad..8ff20f6  master     -> origin/master
Updating 0308dad..8ff20f6
Fast-forward
 new-file.txt | 1 +
 staging.txt  | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 new-file.txt
$ git log
ESC[33mcommit 8ff20f64316abaf28360aa3534835d6cf76278d5ESC[mESC[33m (ESC[mESC[1;36mHEAD -> ESC[mESC[1;32mmasterESC[mESC[33m, ESC[m
Author: Different User <DifferentUser@JoinScrapbook.com>
Date:   Tue Aug 3 18:48:28 2021 +0000

    Fix for Bug #1234

ESC[33mcommit 0308dad1ef5ad899010bfa5990c396c45c439effESC[m
Author: Katacoda Scenario <scenario@katacoda.com>
Date:   Tue Aug 3 18:46:56 2021 +0000

    Message
...skipping...
ESC[33mcommit 8ff20f64316abaf28360aa3534835d6cf76278d5ESC[mESC[33m (ESC[mESC[1;36mHEAD -> ESC[mESC[1;32mmasterESC[mESC[33m, ESC[m
Author: Different User <DifferentUser@JoinScrapbook.com>
Date:   Tue Aug 3 18:48:28 2021 +0000

    Fix for Bug #1234

ESC[33mcommit 0308dad1ef5ad899010bfa5990c396c45c439effESC[m
Author: Katacoda Scenario <scenario@katacoda.com>
Date:   Tue Aug 3 18:46:56 2021 +0000

    Message
$ git branch -a
* ESC[32mmasterESC[m
  ESC[31mremotes/origin/masterESC[m
$ git fetch
remote: Counting objects: 2, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (2/2), done.
From /s/remote-project/1
   8ff20f6..99a9fc9  master     -> origin/master
$ git checkout remote/origin/master
error: pathspec 'remote/origin/master' did not match any file(s) known to git.
$ git checkout remotes/origin/master
Note: checking out 'remotes/origin/master'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b <new-branch-name>

HEAD is now at 99a9fc9 Fix for Bug #42
$
```

```
$ git checkout .
$ ls
committed.txt  staging.txt
$ git reset HEAD .
Unstaged changes after reset:
M       staging.txt
$ git reset --hard
HEAD is now at 118555d New File
$ git reset --hard HEAD
HEAD is now at a56d409 Commit To Revert
$ git revert HEAD...HEAD~2
[master fd8e327] Revert "Commit To Revert"
 1 file changed, 1 insertion(+), 1 deletion(-)
[master 2a93bec] Revert "New File"
 1 file changed, 1 deletion(-)
 delete mode 100644 committed.txt
$ git log --oneline
ESC[33m2a93becESC[mESC[33m (ESC[mESC[1;36mHEAD -> ESC[mESC[1;32mmasterESC[mESC[33m)ESC[m Revert "New File"
ESC[33mfd8e327ESC[m Revert "Commit To Revert"
ESC[33ma56d409ESC[m Commit To Revert
ESC[33m118555dESC[m New File
ESC[33me68eff5ESC[m Fixing Error
ESC[33mce69c14ESC[m First Commit
$
```

```
$ git merge remotes/origin/master
fatal: refusing to merge unrelated histories
$ cat staging.txt
Fixing Error, Let's Hope No-One Else Does
$
$ git diff
$ git checkout --theirs staging.txt
$ git add staging.txt
$ git commit --no-edit
On branch master
nothing to commit, working tree clean
$ git pull --no-edit origin master
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (4/4), done.
From /s/remote-project/1
 * branch            master     -> FETCH_HEAD
   03566a0..6af7539  master     -> origin/master
fatal: refusing to merge unrelated histories
$ git log --oneline
ESC[33m5d6467dESC[mESC[33m (ESC[mESC[1;36mHEAD -> ESC[mESC[1;32mmasterESC[mESC[33m)ESC[m Fix for Bug #55
ESC[33ma669934ESC[m Fixing Error
ESC[33mc29e1b6ESC[m First Commit
$ git rebase feature/10
fatal: Needed a single revision
invalid upstream 'feature/10'
$
$ git pull --rebase
There is no tracking information for the current branch.
Please specify which branch you want to rebase against.
See git-pull(1) for details.

    git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

    git branch --set-upstream-to=origin/<branch> master

$
$
```

```
This is your command line, a safe place to practice & complete the scenario

$ git checkout -b new_branch
Switched to a new branch 'new_branch'
$ git branchgit branchgit
git: 'branchgit' is not a git command. See 'git --help'.
$
$
$ git branch
  masterESC[m
* ESC[32mnew_branchESC[m
$ git branch -va
  master    ESC[m e179a86 First Commit on master
* ESC[32mnew_branchESC[m e179a86 First Commit on master
$ git checkout master
Switched to branch 'master'
$ git merge new_branch
Updating e179a86..48e8f95
Fast-forward
 new-file-6.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 new-file-6.txt
$ git push origin new_branch
Counting objects: 6, done.
Delta compression using up to 12 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 498 bytes | 498.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To /s/remote-project/1
 * [new branch]      new_branch -> new_branch
$
$ git branch -d new_branch
Deleted branch new_branch (was 48e8f95).
$
```