

di-git-ally managing love letters

Version 1.6

April 2010

GNU Free Documentation License

Shakthi Kannan

shakthimaan@gmail.com

Warning

The characters in the presentation are purely fictional. Any reference to real life characters is merely coincidental. Viewer discretion is advised.

git config –global

```
$ git config --global user.name "Shakthi Kannan"
```

git config –global

```
$ git config --global user.name "Shakthi Kannan"  
$ git config --global user.email shakthimaan@gmail.com
```

git config –global

```
$ git config --global user.name "Shakthi Kannan"  
$ git config --global user.email shakthimaan@gmail.com
```

OR

~/.gitconfig

git config –global

```
$ git config --global user.name "Shakthi Kannan"  
$ git config --global user.email shakthimaan@gmail.com
```

OR

~/.gitconfig

[user]

```
name = Shakthi Kannan  
email = shakthimaan@gmail.com
```

git config –global

```
$ git config --global user.name "Shakthi Kannan"  
$ git config --global user.email shakthimaan@gmail.com
```

OR

~/.gitconfig

```
[user]  
    name = Shakthi Kannan  
    email = shakthimaan@gmail.com  
  
[color]  
    status = auto  
    branch = auto  
    ui = auto
```

git config –global

```
$ git config --global user.name "Shakthi Kannan"  
$ git config --global user.email shakthimaan@gmail.com
```

OR

~/.gitconfig

```
[user]  
    name = Shakthi Kannan  
    email = shakthimaan@gmail.com  
  
[color]  
    status = auto  
    branch = auto  
    ui = auto  
  
[alias]  
    gl = "log --graph --pretty=oneline --abbrev-commit"
```


git init

```
$ mkdir love-letters
```

working tree

git init

```
$ mkdir love-letters  
$ cd love-letters
```

working tree

git init

```
$ mkdir love-letters
```

```
$ cd love-letters
```

```
$ git init
```

working tree

index

repository

git init

```
$ mkdir love-letters
```

```
$ cd love-letters
```

```
$ git init
```

```
Initialized empty Git repository in /tmp/love-letters/.git/
```

working tree

index

repository

git init

```
$ mkdir love-letters
```

```
$ cd love-letters
```

```
$ git init
```

```
Initialized empty Git repository in /tmp/love-letters/.git/
```

```
$ touch to-my-dearest.txt
```

working tree

index

repository

to-my-dearest.txt

git init (2)

Edit to-my-dearest.txt.

working tree

index

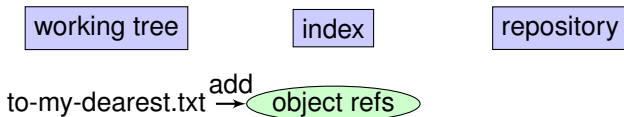
repository

to-my-dearest.txt

git init (2)

Edit to-my-dearest.txt.

```
$ git add .
```

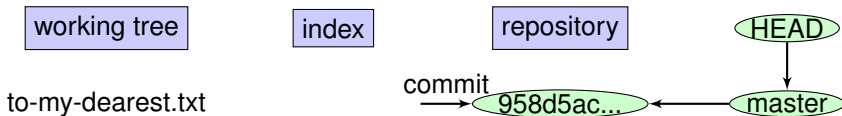


git init (2)

Edit to-my-dearest.txt.

```
$ git add .
```

```
$ git commit -m "First commit"
```



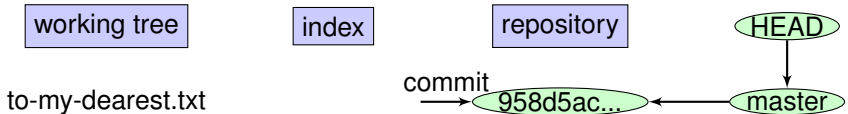
git init (2)

Edit to-my-dearest.txt.

```
$ git add .
```

```
$ git commit -m "First commit"
```

```
Created initial commit 958d5ac: First commit  
1 files changed, 1 insertions(+), 0 deletions(-)  
create mode 100644 to-my-dearest.txt
```



git init (2)

Edit to-my-dearest.txt.

```
$ git add .
```

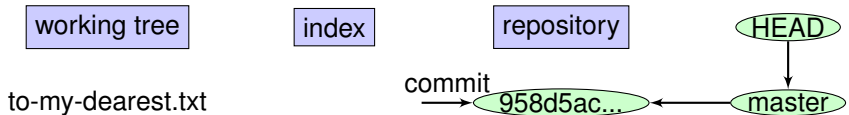
```
$ git commit -m "First commit"
```

```
Created initial commit 958d5ac: First commit
```

```
1 files changed, 1 insertions(+), 0 deletions(-)
```

```
create mode 100644 to-my-dearest.txt
```

```
$ git status
```



git init (2)

Edit to-my-dearest.txt.

```
$ git add .
```

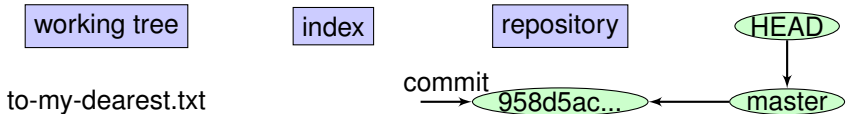
```
$ git commit -m "First commit"
```

```
Created initial commit 958d5ac: First commit  
1 files changed, 1 insertions(+), 0 deletions(-)  
create mode 100644 to-my-dearest.txt
```

```
$ git status
```

```
# On branch master
```

```
nothing to commit (working directory clean)
```



git gl

```
$ git gl
```

git gl

```
$ git gl  
* 958d5ac... First commit
```

git gl

```
$ git gl  
* 958d5ac... First commit
```

```
$ git show
```

git gl

```
$ git gl
* 958d5ac... First commit

$ git show
commit 958d5ac512dd202b65fd7214b3ebd470561dd0ac
Author: Shakthi Kannan <shakthimaan@gmail.com>
Date: Sat Apr 25 11:41:40 2009 +0530

    First commit

diff --git a/to-my-dearest.txt b/to-my-dearest.txt
new file mode 100644
index 0000000..27853ff
--- /dev/null
+++ b/to-my-dearest.txt
@@ -0,0 +1 @@
+Dear
```

git gl

```
$ git gl
* 958d5ac... First commit

$ git show
commit 958d5ac512dd202b65fd7214b3ebd470561dd0ac
Author: Shakthi Kannan <shakthimaan@gmail.com>
Date: Sat Apr 25 11:41:40 2009 +0530

    First commit

diff --git a/to-my-dearest.txt b/to-my-dearest.txt
new file mode 100644
index 0000000..27853ff
--- /dev/null
+++ b/to-my-dearest.txt
@@ -0,0 +1 @@
+Dear

$ git status
```


git gl

```
$ git gl
* 958d5ac... First commit

$ git show
commit 958d5ac512dd202b65fd7214b3ebd470561dd0ac
Author: Shakthi Kannan <shakthimaan@gmail.com>
Date: Sat Apr 25 11:41:40 2009 +0530

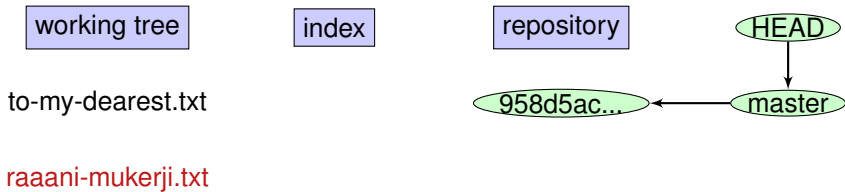
    First commit

diff --git a/to-my-dearest.txt b/to-my-dearest.txt
new file mode 100644
index 0000000..27853ff
--- /dev/null
+++ b/to-my-dearest.txt
@@ -0,0 +1 @@
+Dear

$ git status
# On branch master
nothing to commit (working directory clean)
```

git diff

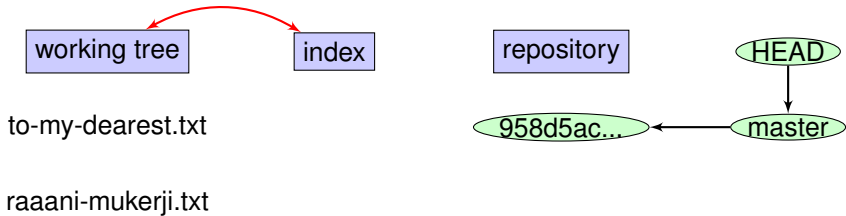
```
$ touch raaani-mukerji.txt
```



git diff

```
$ touch raaani-mukerji.txt
```

```
$ git diff
```

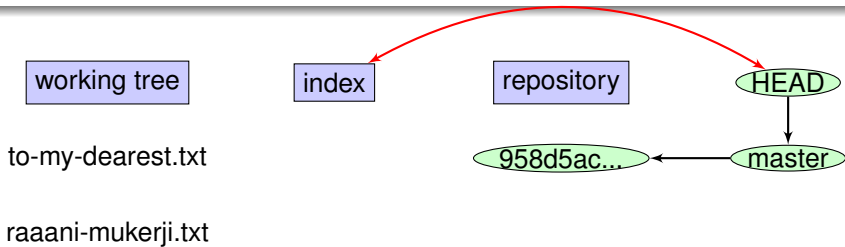


git diff

```
$ touch raaani-mukerji.txt
```

```
$ git diff
```

```
$ git diff --cached
```



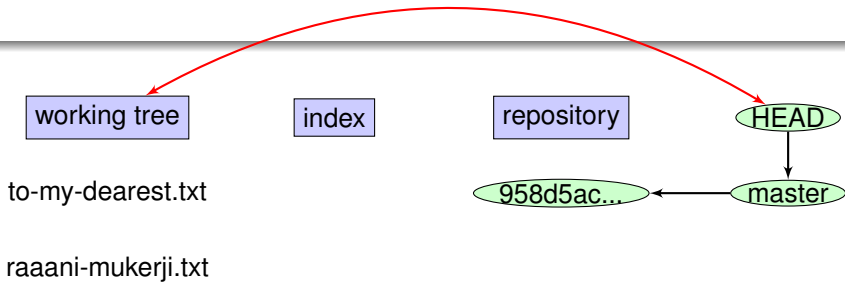
git diff

```
$ touch raaani-mukerji.txt
```

```
$ git diff
```

```
$ git diff --cached
```

```
$ git diff HEAD
```



git diff

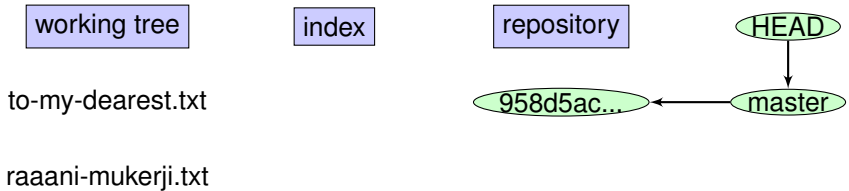
```
$ touch raaani-mukerji.txt
```

```
$ git diff
```

```
$ git diff --cached
```

```
$ git diff HEAD
```

```
$ git status
```

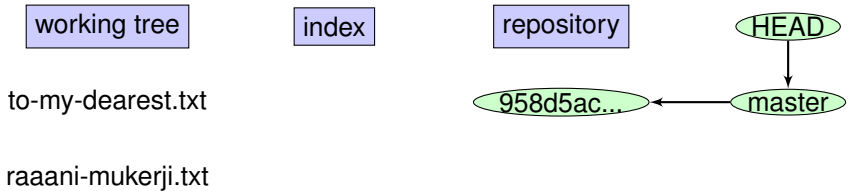


git diff

```
$ touch raaani-mukerji.txt

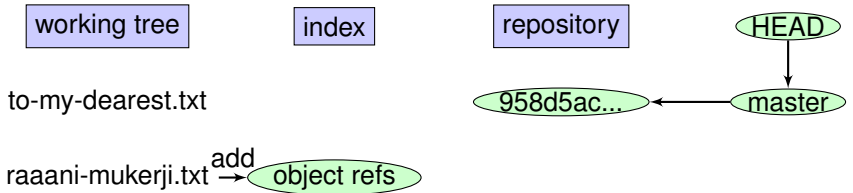
$ git diff
$ git diff --cached
$ git diff HEAD

$ git status
# On branch master
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
# raaani-mukerji.txt
nothing added to commit but untracked files present (use "git add" to track)
```



git add

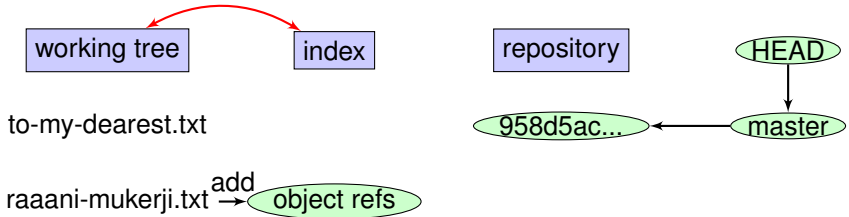
```
$ git add raaani-mukerji.txt
```



git add

```
$ git add raaani-mukerji.txt
```

```
$ git diff
```

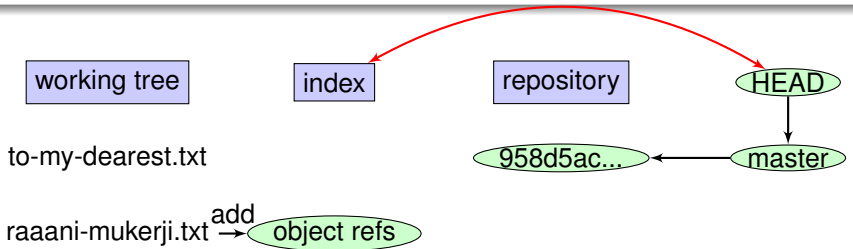


git add

```
$ git add raaani-mukerji.txt
```

```
$ git diff
```

```
$ git diff --cached
```



git add

```
$ git add raaani-mukerji.txt
```

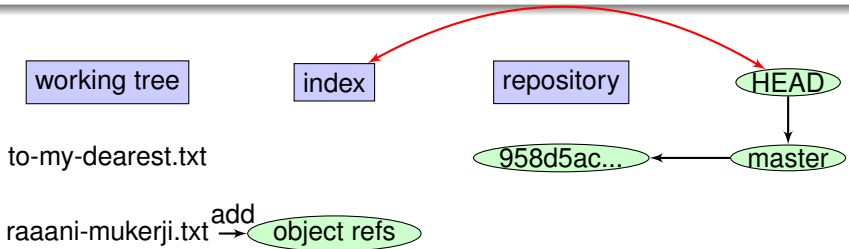
```
$ git diff
```

```
$ git diff --cached
```

```
diff --git a/raaani-mukerji.txt b/raaani-mukerji.txt
```

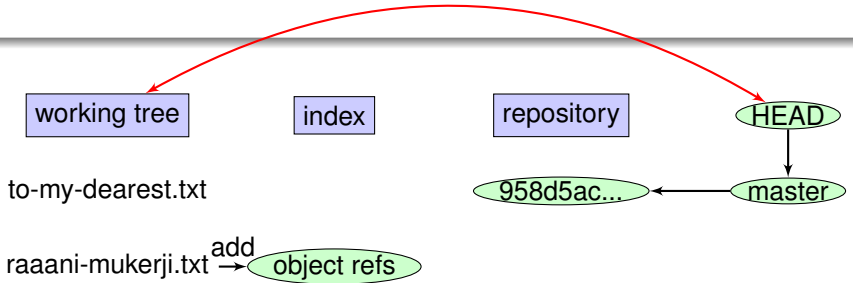
```
new file mode 100644
```

```
index 0000000..e69de29
```



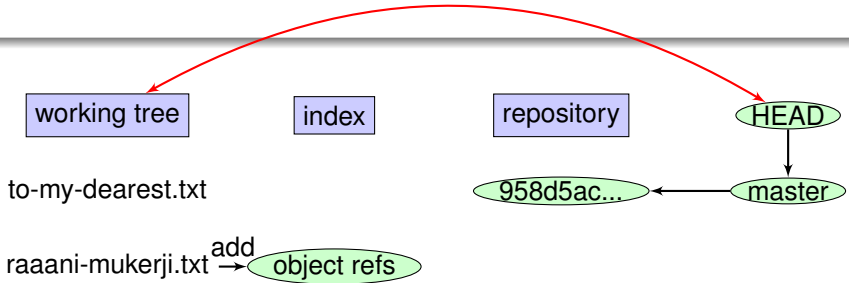
git add (2)

```
$ git diff HEAD
```



git add (2)

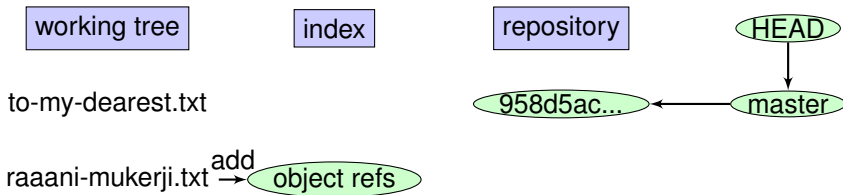
```
$ git diff HEAD
diff --git a/raaani-mukerji.txt b/raaani-mukerji.txt
new file mode 100644
index 0000000..e69de29
```



git add (2)

```
$ git diff HEAD
diff --git a/raaani-mukerji.txt b/raaani-mukerji.txt
new file mode 100644
index 0000000..e69de29
```

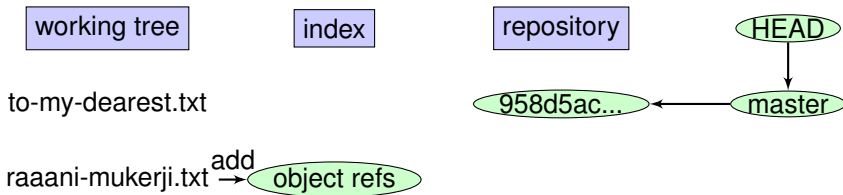
```
$ git status
```



git add (2)

```
$ git diff HEAD
diff --git a/raaani-mukerji.txt b/raaani-mukerji.txt
new file mode 100644
index 0000000..e69de29
```

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# new file:   raaani-mukerji.txt
#
```



What!

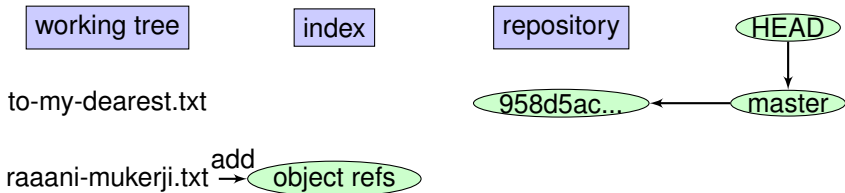
\$ *date me*

date: invalid date 'me'

How sad!

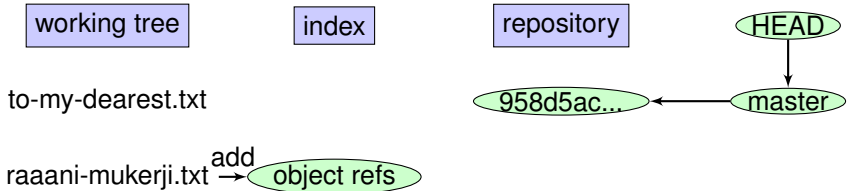
git rm -f

```
$ git rm raaani-mukerji.txt
```



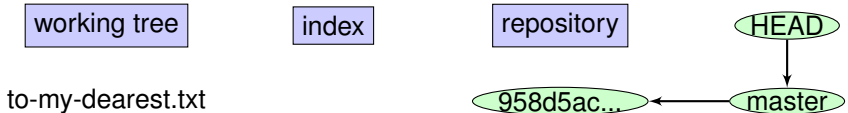
git rm -f

```
$ git rm raaani-mukerji.txt  
error: 'raaani-mukerji.txt' has changes staged in the index  
(use --cached to keep the file, or -f to force removal)
```



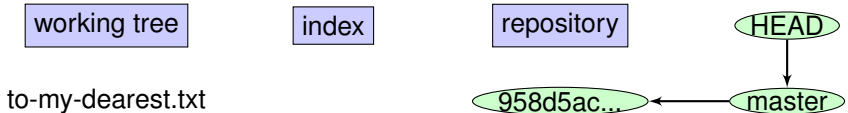
git rm -f

```
$ git rm raaani-mukerji.txt
error: 'raaani-mukerji.txt' has changes staged in the index
(use --cached to keep the file, or -f to force removal)
$ git rm -f raaani-mukerji.txt
```



git rm -f

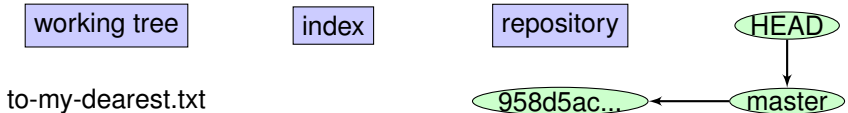
```
$ git rm raaani-mukerji.txt
error: 'raaani-mukerji.txt' has changes staged in the index
(use --cached to keep the file, or -f to force removal)
$ git rm -f raaani-mukerji.txt
rm 'raaani-mukerji.txt'
```



git rm -f

```
$ git rm raaani-mukerji.txt
error: 'raaani-mukerji.txt' has changes staged in the index
(use --cached to keep the file, or -f to force removal)
$ git rm -f raaani-mukerji.txt
rm 'raaani-mukerji.txt'
```

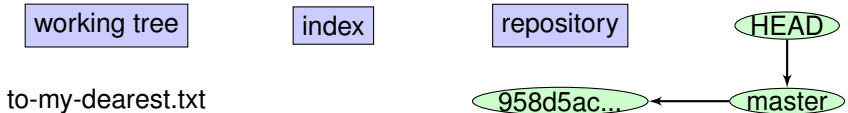
```
$ ls
```



git rm -f

```
$ git rm raaani-mukerji.txt
error: 'raaani-mukerji.txt' has changes staged in the index
(use --cached to keep the file, or -f to force removal)
$ git rm -f raaani-mukerji.txt
rm 'raaani-mukerji.txt'

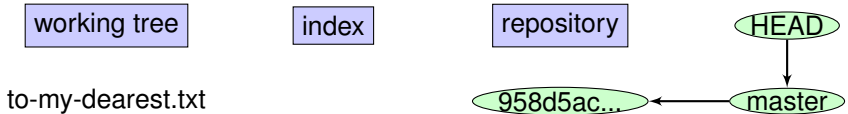
$ ls
to-my-dearest.txt
```



git rm -f

```
$ git rm raaani-mukerji.txt
error: 'raaani-mukerji.txt' has changes staged in the index
(use --cached to keep the file, or -f to force removal)
$ git rm -f raaani-mukerji.txt
rm 'raaani-mukerji.txt'

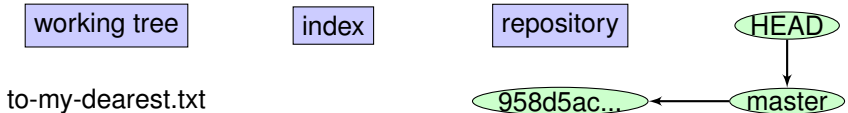
$ ls
to-my-dearest.txt
$ git status
```



git rm -f

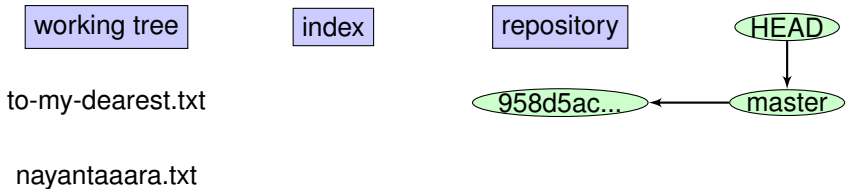
```
$ git rm raaani-mukerji.txt
error: 'raaani-mukerji.txt' has changes staged in the index
(use --cached to keep the file, or -f to force removal)
$ git rm -f raaani-mukerji.txt
rm 'raaani-mukerji.txt'

$ ls
to-my-dearest.txt
$ git status
# On branch master
nothing to commit (working directory clean)
```



git rm --cached

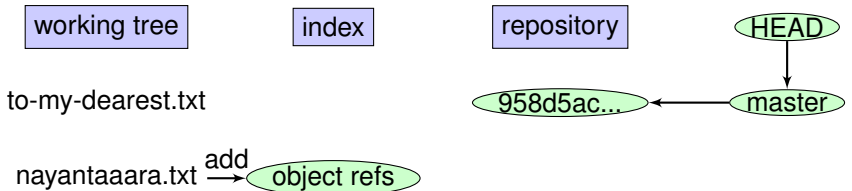
```
$ touch nayantaaara.txt
```



git rm --cached

```
$ touch nayantaaara.txt
```

```
$ git add nayantaaara.txt
```



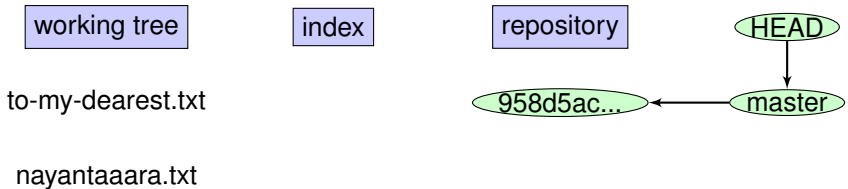
git rm --cached

```
$ touch nayantaaara.txt
```

```
$ git add nayantaaara.txt
```

Just remove from staging area (index)

```
$ git rm --cached nayantaaara.txt
```



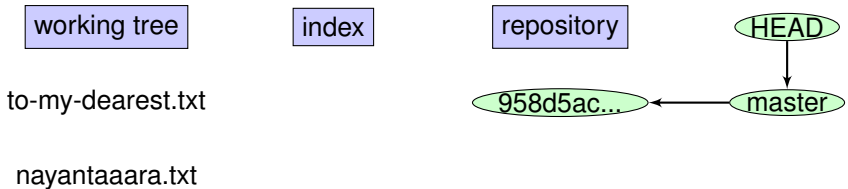
git rm --cached

```
$ touch nayantaaara.txt
```

```
$ git add nayantaaara.txt
```

Just remove from staging area (index)

```
$ git rm --cached nayantaaara.txt  
rm 'nayantaaara.txt'
```



git rm --cached

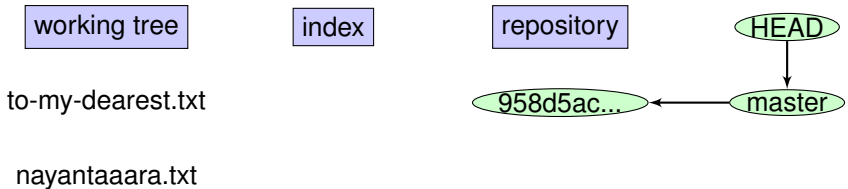
```
$ touch nayantaaara.txt
```

```
$ git add nayantaaara.txt
```

Just remove from staging area (index)

```
$ git rm --cached nayantaaara.txt  
rm 'nayantaaara.txt'
```

```
$ ls
```



git rm --cached

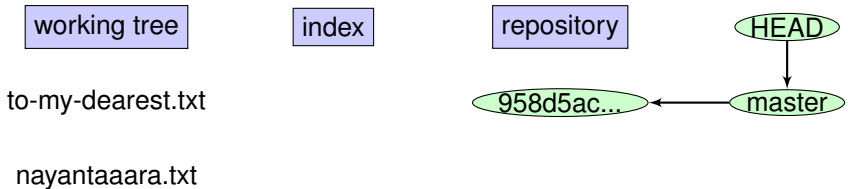
```
$ touch nayantaaara.txt
```

```
$ git add nayantaaara.txt
```

Just remove from staging area (index)

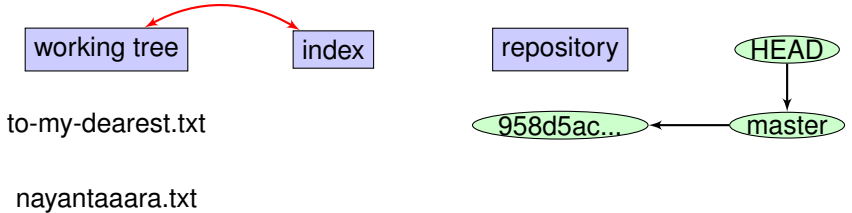
```
$ git rm --cached nayantaaara.txt  
rm 'nayantaaara.txt'
```

```
$ ls  
to-my-dearest.txt nayantaaara.txt
```



git rm –cached (2)

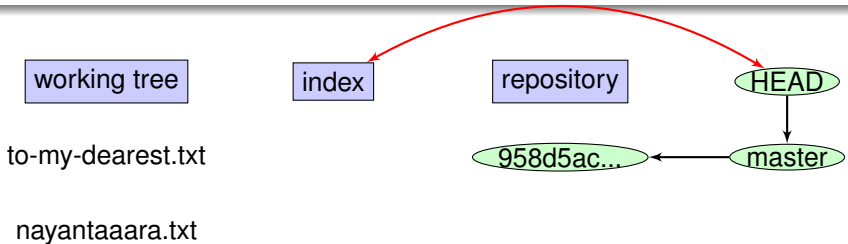
```
$ git diff
```



git rm --cached (2)

```
$ git diff
```

```
$ git diff --cached
```

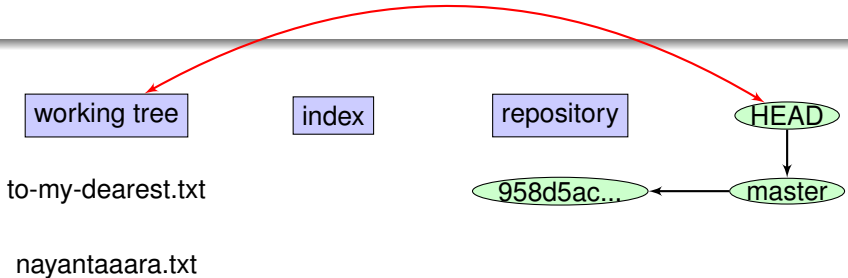


git rm --cached (2)

```
$ git diff
```

```
$ git diff --cached
```

```
$ git diff HEAD
```



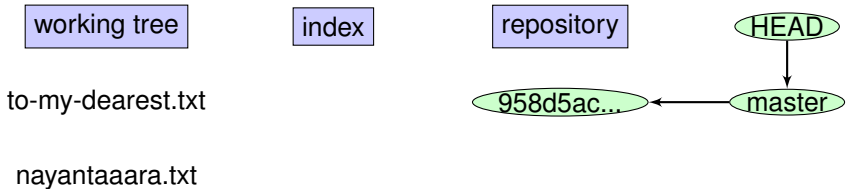
git rm --cached (2)

```
$ git diff
```

```
$ git diff --cached
```

```
$ git diff HEAD
```

```
$ git status
```



git rm --cached (2)

```
$ git diff
```

```
$ git diff --cached
```

```
$ git diff HEAD
```

```
$ git status
```

```
# On branch master
```

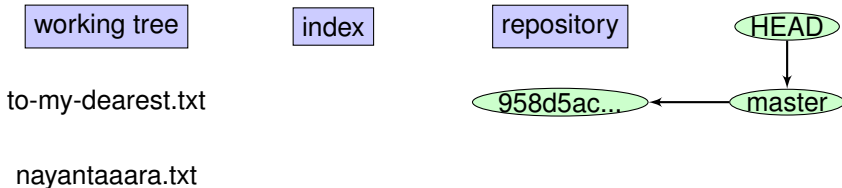
```
# Untracked files:
```

```
#   (use "git add <file>..." to include in what will be committed)
```

```
#
```

```
# nayantaaara.txt
```

```
nothing added to commit but untracked files present (use "git add" to track)
```



What!

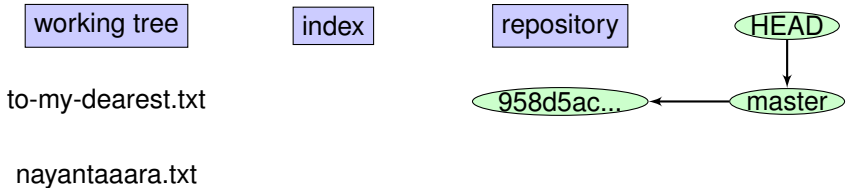
\$ *make love*

make: *** No rule to
make target 'love'. Stop

How sad!

git rm –cached (3)

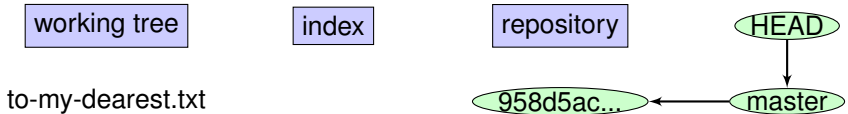
Didn't work out with nayantaaara.



git rm –cached (3)

Didn't work out with nayantaaara.

```
$ rm nayantaaara.txt
```

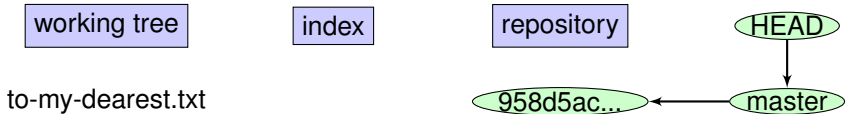


git rm –cached (3)

Didn't work out with nayantaaara.

```
$ rm nayantaaara.txt
```

```
$ git status
```



git rm –cached (3)

Didn't work out with nayantaaara.

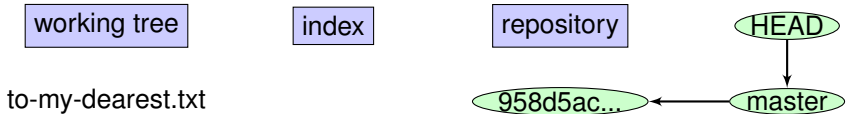
```
$ rm nayantaaara.txt
```

```
$ git status
```

```
# On branch master
```

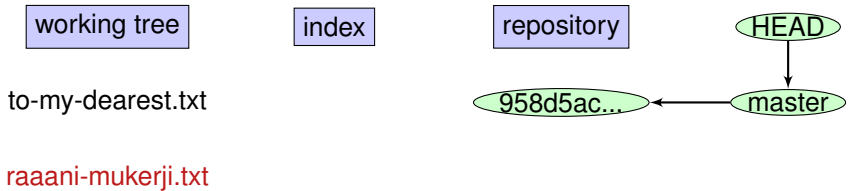
```
nothing to commit (working directory clean)
```

```
* 958d5ac... First commit
```



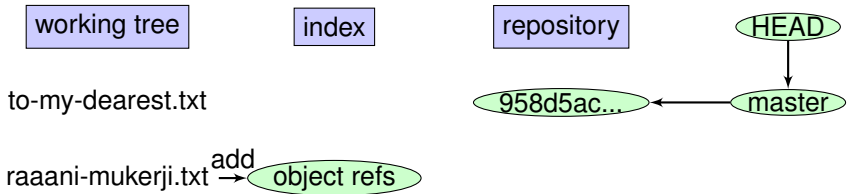
git reset HEAD file

```
$ touch raaani-mukerji.txt
```



git reset HEAD file

```
$ touch raaani-mukerji.txt  
$ git add raaani-mukerji.txt
```

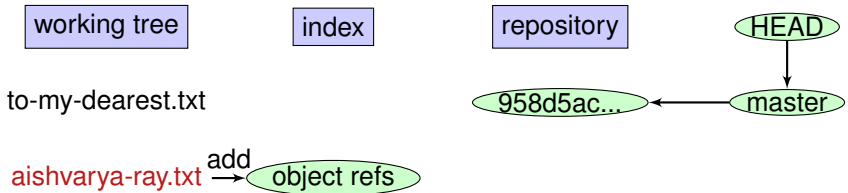


git reset HEAD file

```
$ touch raaani-mukerji.txt  
$ git add raaani-mukerji.txt
```

Already rejected. Try aishvarya-ray!

```
$ git mv raaani-mukerji.txt aishvarya-ray.txt
```

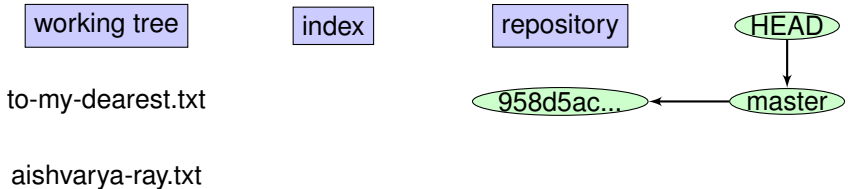


git reset HEAD file

```
$ touch raaani-mukerji.txt  
$ git add raaani-mukerji.txt
```

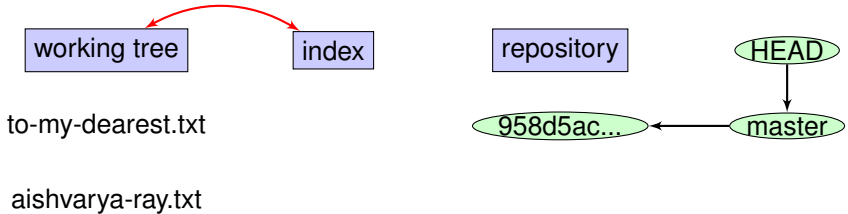
Already rejected. Try aishvarya-ray!

```
$ git mv raaani-mukerji.txt aishvarya-ray.txt  
$ git reset HEAD aishvarya-ray.txt
```



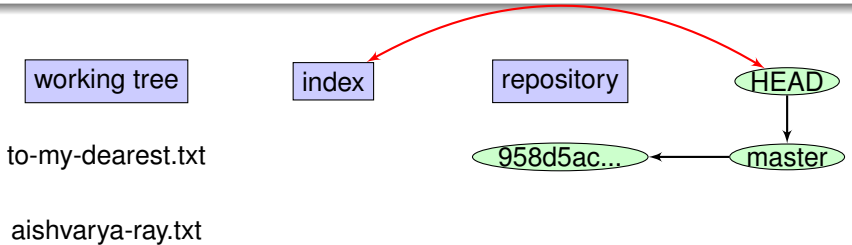
git reset HEAD file (2)

```
$ git diff
```



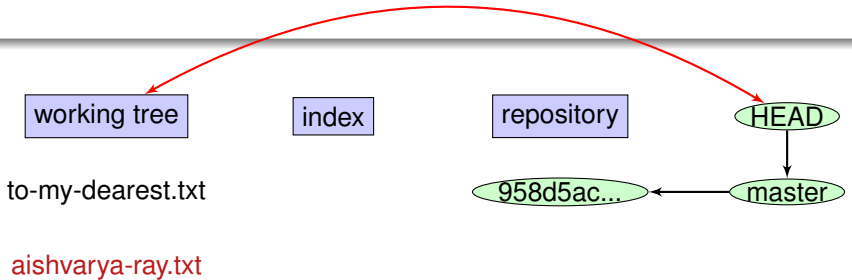
git reset HEAD file (2)

```
$ git diff  
$ git diff --cached
```



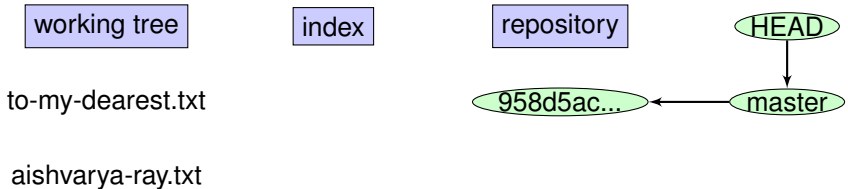
git reset HEAD file (2)

```
$ git diff  
$ git diff --cached  
$ git diff HEAD
```



git reset HEAD file (2)

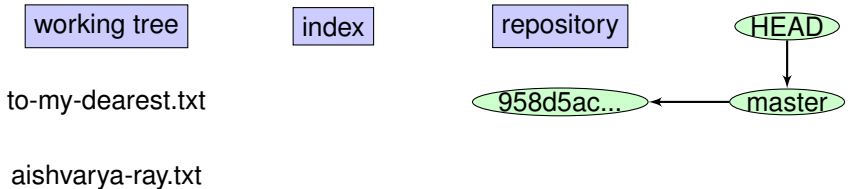
```
$ git diff  
$ git diff --cached  
$ git diff HEAD  
  
$ git status
```



git reset HEAD file (2)

```
$ git diff
$ git diff --cached
$ git diff HEAD

$ git status
# On branch master
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
# aishvarya-ray.txt
nothing added to commit but untracked files present (use "git add" to track)
```



What!

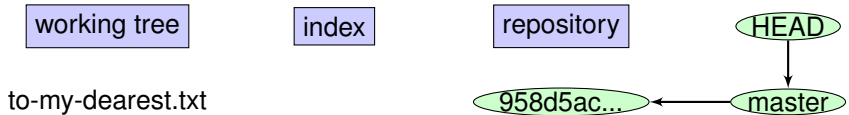
aishvarya-ray is married!

OOPS!

How sad!

git reset HEAD file (3)

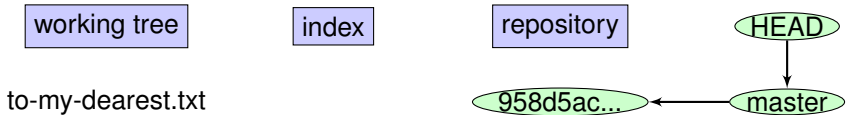
```
$ rm aishvarya-ray.txt
```



git reset HEAD file (3)

```
$ rm aishvarya-ray.txt
```

```
$ ls
```

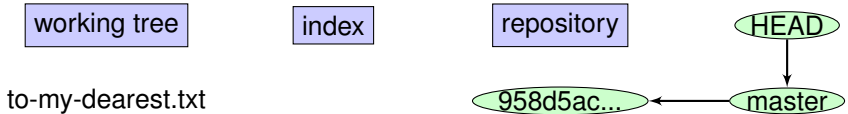


git reset HEAD file (3)

```
$ rm aishvarya-ray.txt
```

```
$ ls
```

```
to-my-dearest.txt
```

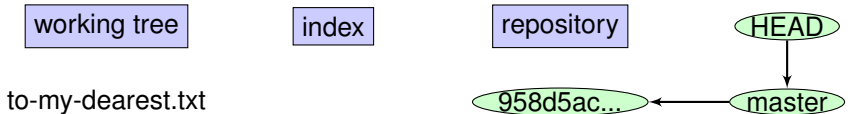


git reset HEAD file (3)

```
$ rm aishvarya-ray.txt
```

```
$ ls  
to-my-dearest.txt
```

```
$ git status
```

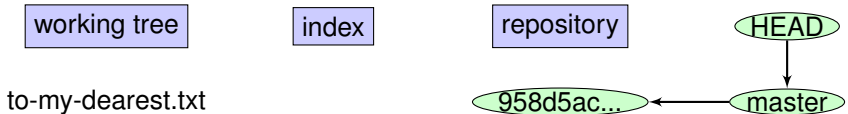


git reset HEAD file (3)

```
$ rm aishvarya-ray.txt
```

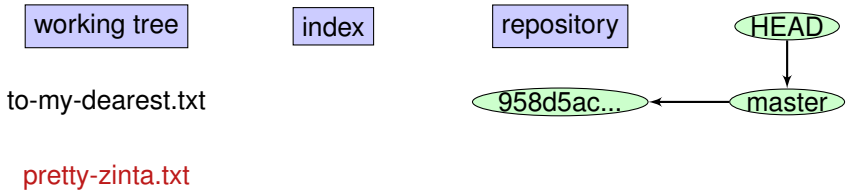
```
$ ls  
to-my-dearest.txt
```

```
$ git status  
# On branch master  
nothing to commit (working directory clean)
```



git reset –soft HEAD^

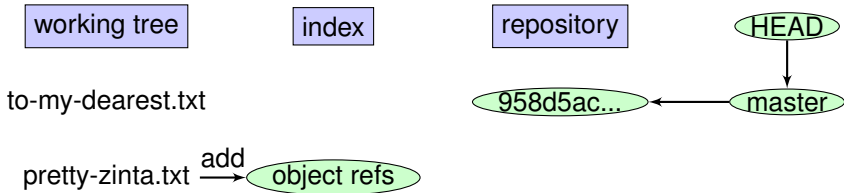
```
$ touch pretty-zinta.txt
```



git reset –soft HEAD^

```
$ touch pretty-zinta.txt
```

```
$ git add pretty-zinta.txt
```

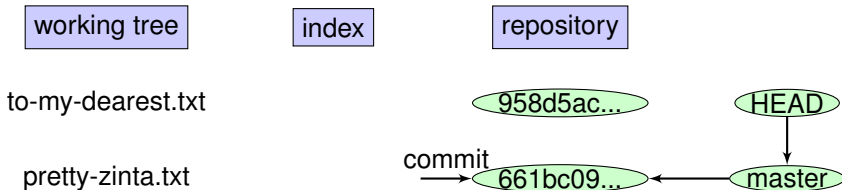


git reset –soft HEAD^

```
$ touch pretty-zinta.txt
```

```
$ git add pretty-zinta.txt
```

```
$ git commit -m "Added pretty-zinta.txt"
```



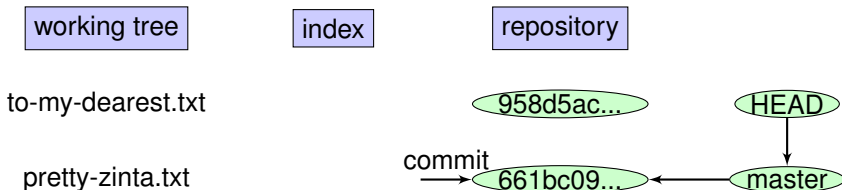
git reset –soft HEAD^

```
$ touch pretty-zinta.txt
```

```
$ git add pretty-zinta.txt
```

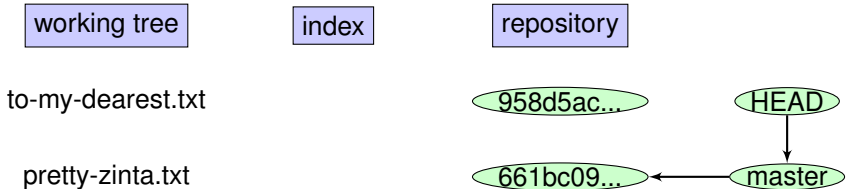
```
$ git commit -m "Added pretty-zinta.txt"
```

```
Created commit 661bc09: Added pretty-zinta.txt  
0 files changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 pretty-zinta.txt
```



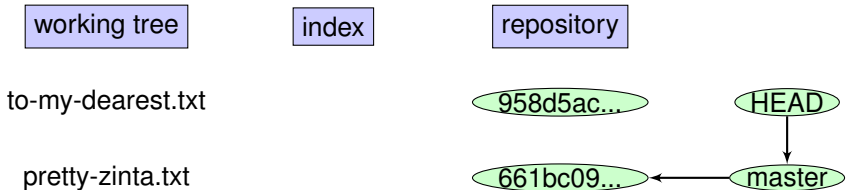
git reset --soft HEAD^ (2)

```
$ git gl
```



git reset –soft HEAD^ (2)

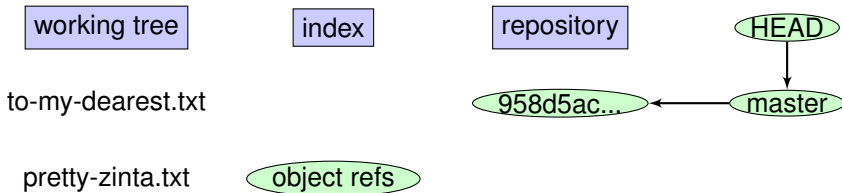
```
$ git gl
* 661bc09... Added pretty-zinta.txt
* 958d5ac... First commit
```



git reset --soft HEAD^ (2)

```
$ git gl
* 661bc09... Added pretty-zinta.txt
* 958d5ac... First commit
```

```
$ git reset --soft HEAD^
```

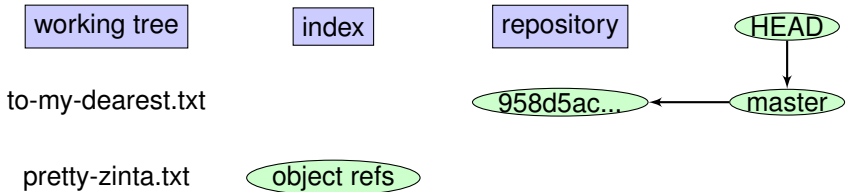


git reset --soft HEAD^ (2)

```
$ git gl
* 661bc09... Added pretty-zinta.txt
* 958d5ac... First commit

$ git reset --soft HEAD^

$ git gl
```

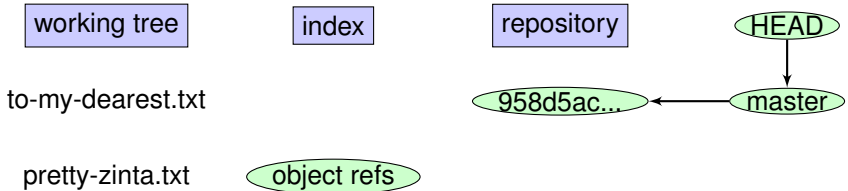


git reset --soft HEAD^ (2)

```
$ git gl
* 661bc09... Added pretty-zinta.txt
* 958d5ac... First commit

$ git reset --soft HEAD^

$ git gl
* 958d5ac... First commit
```



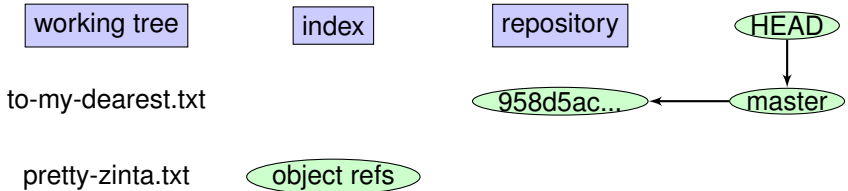
git reset --soft HEAD^ (2)

```
$ git gl
* 661bc09... Added pretty-zinta.txt
* 958d5ac... First commit

$ git reset --soft HEAD^

$ git gl
* 958d5ac... First commit

$ ls
```



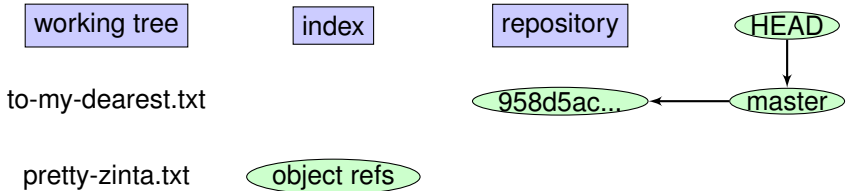
git reset --soft HEAD^ (2)

```
$ git gl
* 661bc09... Added pretty-zinta.txt
* 958d5ac... First commit

$ git reset --soft HEAD^

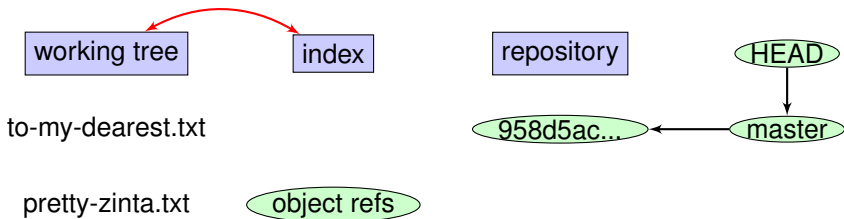
$ git gl
* 958d5ac... First commit

$ ls
pretty-zinta.txt to-my-dearest.txt
```



git reset –soft HEAD^ (3)

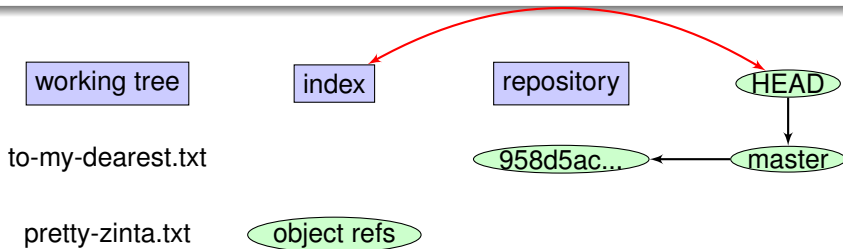
```
$ git diff
```



git reset –soft HEAD^ (3)

```
$ git diff
```

```
$ git diff --cached
```



git reset --soft HEAD^ (3)

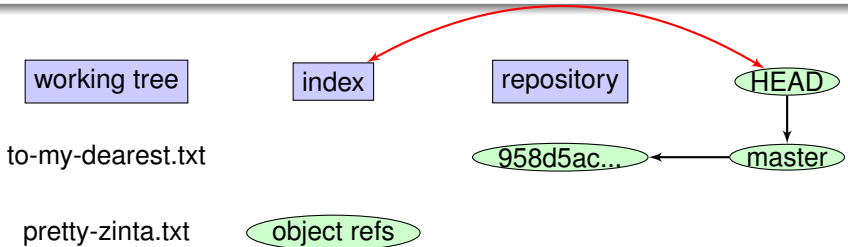
```
$ git diff
```

```
$ git diff --cached
```

```
diff --git a/pretty-zinta.txt b/pretty-zinta.txt
```

```
new file mode 100644
```

```
index 0000000..e69de29
```



git reset --soft HEAD^ (3)

```
$ git diff
```

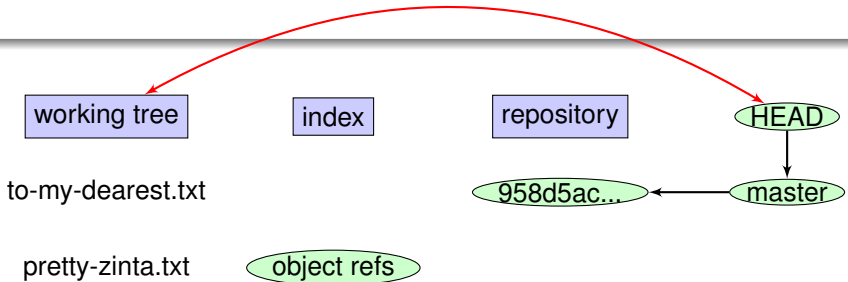
```
$ git diff --cached
```

```
diff --git a/pretty-zinta.txt b/pretty-zinta.txt
```

```
new file mode 100644
```

```
index 0000000..e69de29
```

```
$ git diff HEAD
```



git reset --soft HEAD^ (3)

```
$ git diff
```

```
$ git diff --cached
```

```
diff --git a/pretty-zinta.txt b/pretty-zinta.txt
```

```
new file mode 100644
```

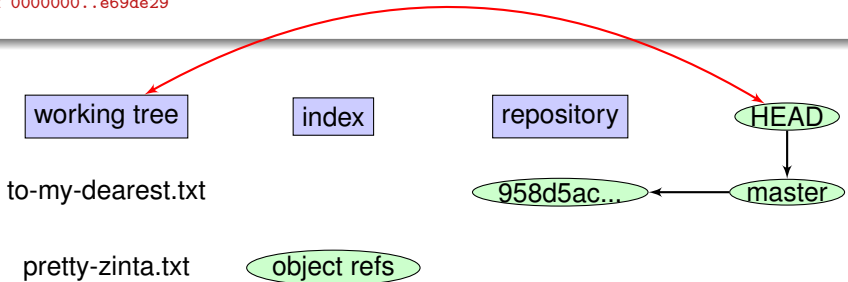
```
index 0000000..e69de29
```

```
$ git diff HEAD
```

```
diff --git a/pretty-zinta.txt b/pretty-zinta.txt
```

```
new file mode 100644
```

```
index 0000000..e69de29
```



git reset –soft HEAD^ (4)

```
$ git status
```


git reset --soft HEAD^ (4)

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# new file:   pretty-zinta.txt
#
```

git reset –soft HEAD^ (4)

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# new file:   pretty-zinta.txt
#
```

You can remove pretty-zinta.txt:

```
$ git reset HEAD pretty-zinta.txt
```

git reset –soft HEAD^ (4)

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# new file:   pretty-zinta.txt
#
```

You can remove pretty-zinta.txt:

```
$ git reset HEAD pretty-zinta.txt
$ rm pretty-zinta.txt
```

git reset –soft HEAD^ (4)

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# new file:   pretty-zinta.txt
#
```

You can remove pretty-zinta.txt:

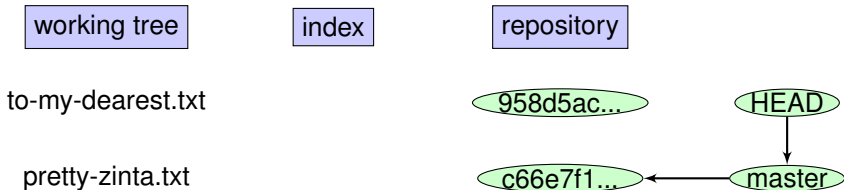
```
$ git reset HEAD pretty-zinta.txt
$ rm pretty-zinta.txt
```

OR ...

git reset –soft HEAD^ (5)

Go for it! Modify the file and commit it back:

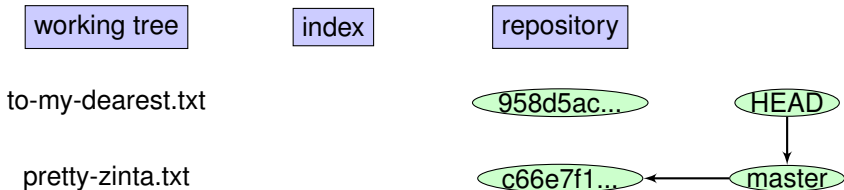
```
$ git commit -m "Re-added pretty-zinta.txt"
```



git reset –soft HEAD^ (5)

Go for it! Modify the file and commit it back:

```
$ git commit -m "Re-added pretty-zinta.txt"
Created commit c66e7f1: Re-add pretty-zinta.txt
0 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 pretty-zinta.txt
```

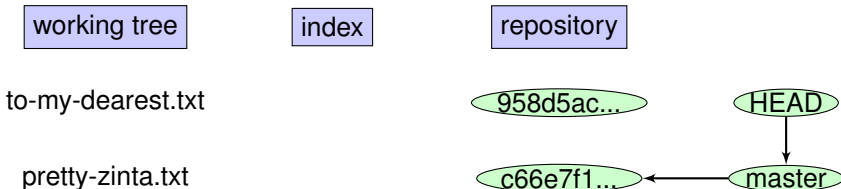


git reset –soft HEAD^ (5)

Go for it! Modify the file and commit it back:

```
$ git commit -m "Re-added pretty-zinta.txt"
Created commit c66e7f1: Re-add pretty-zinta.txt
0 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 pretty-zinta.txt
```

```
$ git gl
```

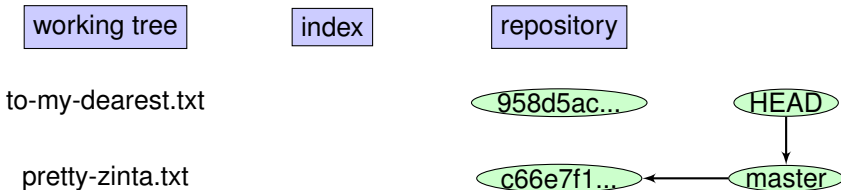


git reset –soft HEAD^ (5)

Go for it! Modify the file and commit it back:

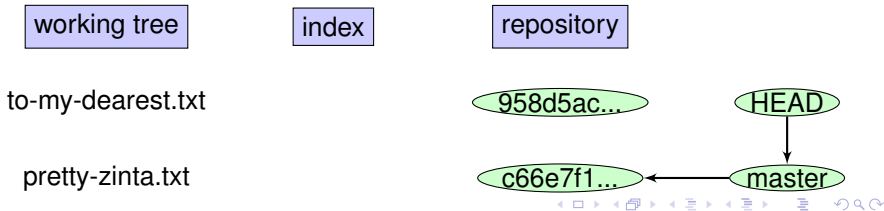
```
$ git commit -m "Re-added pretty-zinta.txt"
Created commit c66e7f1: Re-add pretty-zinta.txt
0 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 pretty-zinta.txt
```

```
$ git gl
* c66e7f1... Re-add pretty-zinta.txt
* 958d5ac... First commit
```



git commit -amend

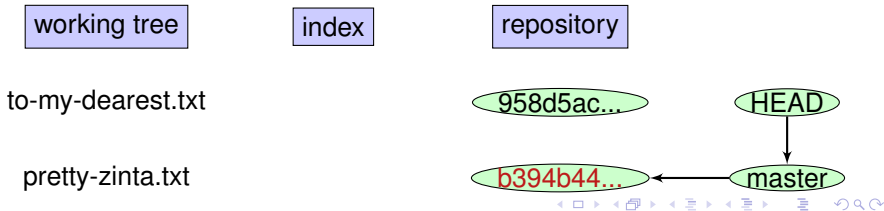
Modify a committed message:



git commit --amend

Modify a committed message:

```
$ git commit --amend
```



git commit --amend

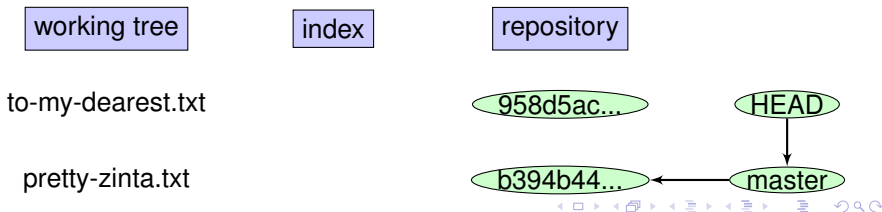
Modify a committed message:

```
$ git commit --amend
```

```
Created commit b394b44: Re-added pretty-zinta.txt to repo.
```

```
0 files changed, 0 insertions(+), 0 deletions(-)
```

```
create mode 100644 pretty-zinta.txt
```

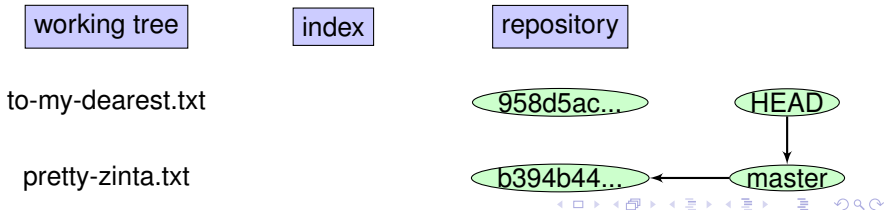


git commit --amend

Modify a committed message:

```
$ git commit --amend
Created commit b394b44: Re-added pretty-zinta.txt to repo.
0 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 pretty-zinta.txt
```

```
$ git gl
```

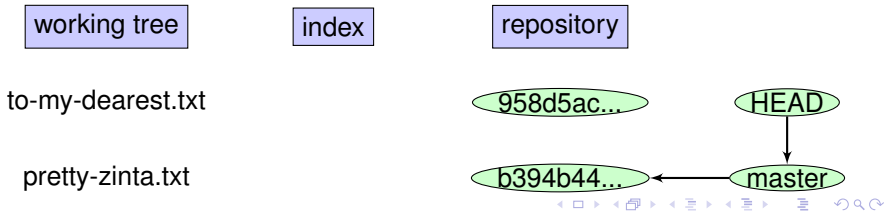


git commit --amend

Modify a committed message:

```
$ git commit --amend
Created commit b394b44: Re-added pretty-zinta.txt to repo.
0 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 pretty-zinta.txt

$ git gl
* b394b44... Re-added pretty-zinta.txt to repo.
* 958d5ac... First commit
```



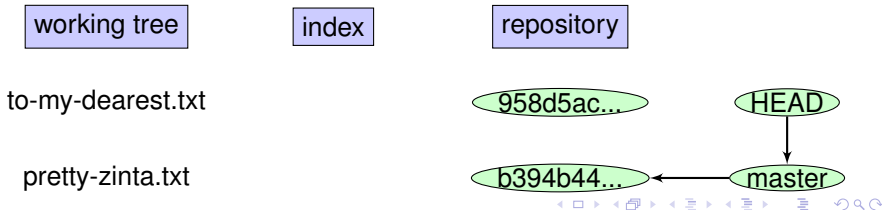
git commit --amend

Modify a committed message:

```
$ git commit --amend
Created commit b394b44: Re-added pretty-zinta.txt to repo.
0 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 pretty-zinta.txt

$ git gl
* b394b44... Re-added pretty-zinta.txt to repo.
* 958d5ac... First commit

$ git status
```



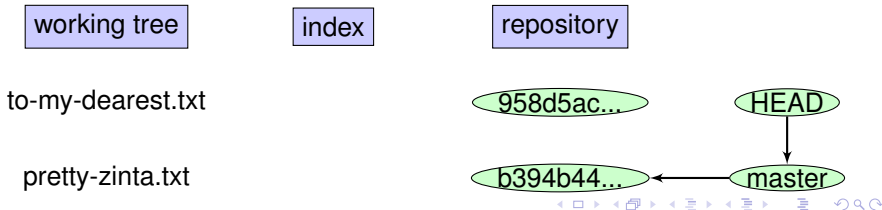
git commit --amend

Modify a committed message:

```
$ git commit --amend
Created commit b394b44: Re-added pretty-zinta.txt to repo.
0 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 pretty-zinta.txt

$ git gl
* b394b44... Re-added pretty-zinta.txt to repo.
* 958d5ac... First commit

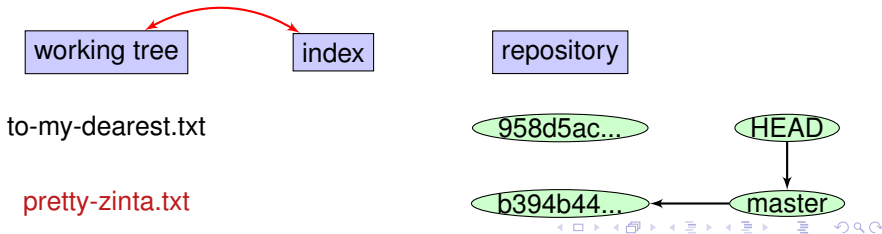
$ git status
# On branch master
nothing to commit (working directory clean)
```



git diff

Update `pretty-zinta.txt` with some changes.

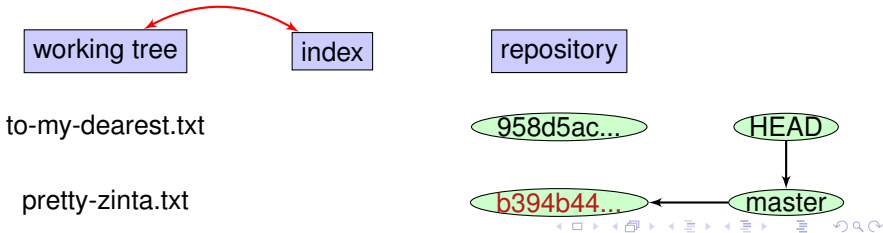
```
$ git diff
```



git diff

Update pretty-zinta.txt with some changes.

```
$ git diff
diff --git a/pretty-zinta.txt b/pretty-zinta.txt
index e69de29..10e90a6 100644
--- a/pretty-zinta.txt
+++ b/pretty-zinta.txt
@@ -0,0 +1,3 @@
+Dear Pretty:
+
+You are very pretty!
```

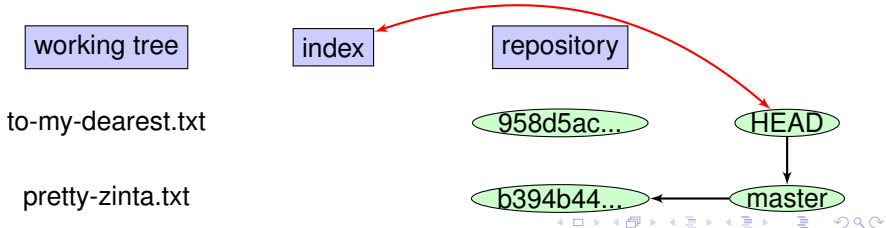


git diff

Update pretty-zinta.txt with some changes.

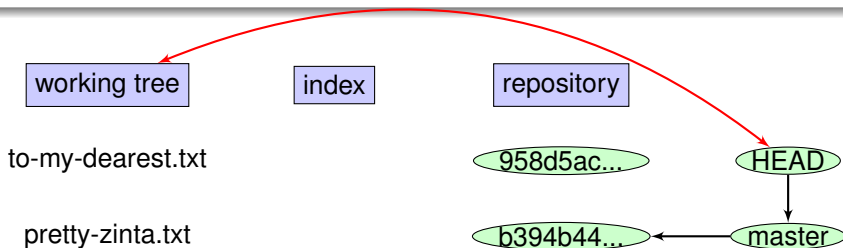
```
$ git diff
diff --git a/pretty-zinta.txt b/pretty-zinta.txt
index e69de29..10e90a6 100644
--- a/pretty-zinta.txt
+++ b/pretty-zinta.txt
@@ -0,0 +1,3 @@
+Dear Pretty:
+
+You are very pretty!
```

```
$ git diff --cached
```



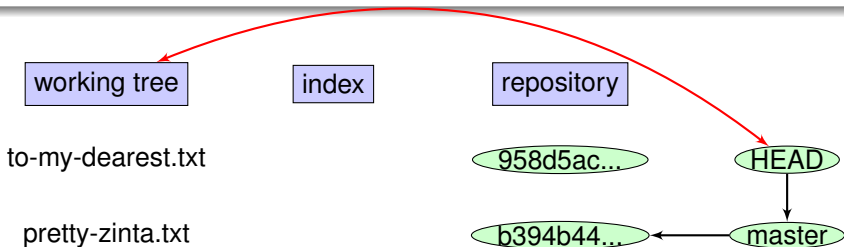
git diff (2)

```
$ git diff HEAD
```



git diff (2)

```
$ git diff HEAD
diff --git a/pretty-zinta.txt b/pretty-zinta.txt
index e69de29..10e90a6 100644
--- a/pretty-zinta.txt
+++ b/pretty-zinta.txt
@@ -0,0 +1,3 @@
+Dear Pretty:
+
+You are very pretty!
```



What!

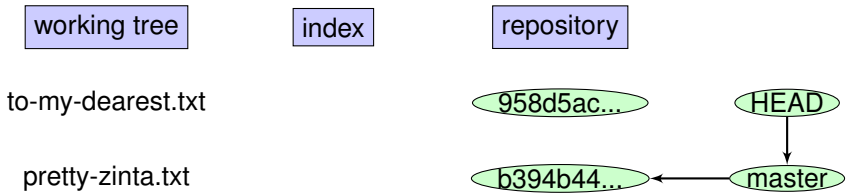
\$ *yum search pretty-zinta*

No Matches found

How sad!

git reset –hard HEAD^

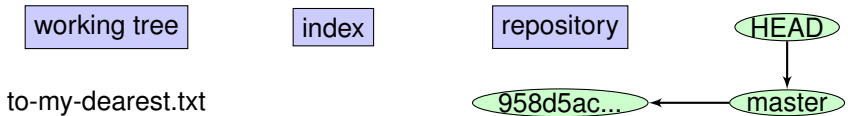
No match found. Reset!



git reset --hard HEAD^

No match found. Reset!

```
$ git reset --hard HEAD^
```

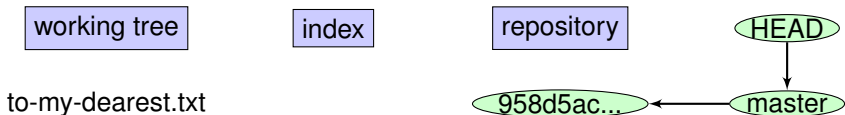


git reset --hard HEAD^

No match found. Reset!

```
$ git reset --hard HEAD^
```

HEAD is now at 958d5ac First commit

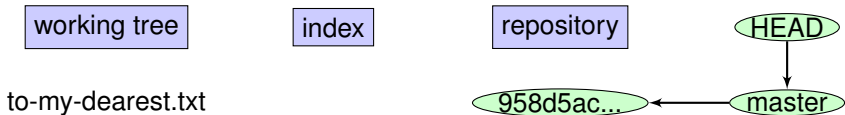


git reset --hard HEAD^

No match found. Reset!

```
$ git reset --hard HEAD^  
HEAD is now at 958d5ac First commit
```

```
$ git gl
```

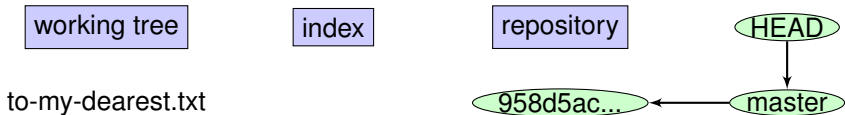


git reset --hard HEAD^

No match found. Reset!

```
$ git reset --hard HEAD^  
HEAD is now at 958d5ac First commit
```

```
$ git gl  
* 958d5ac... First commit
```



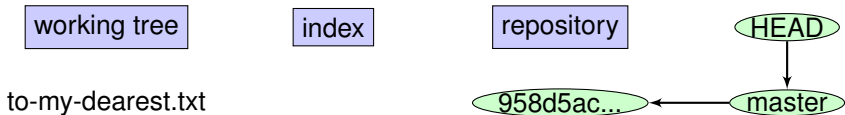
git reset --hard HEAD^

No match found. Reset!

```
$ git reset --hard HEAD^  
HEAD is now at 958d5ac First commit
```

```
$ git gl  
* 958d5ac... First commit
```

```
$ git status
```



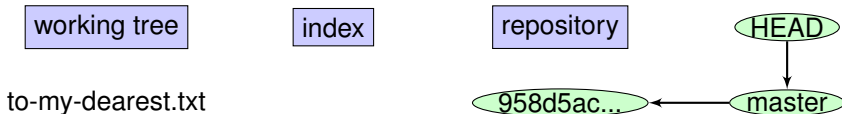
git reset --hard HEAD^

No match found. Reset!

```
$ git reset --hard HEAD^
HEAD is now at 958d5ac First commit

$ git gl
* 958d5ac... First commit

$ git status
# On branch master
nothing to commit (working directory clean)
```



git grep

Grep for a pattern 'dear':

```
$ git grep dear
```

git grep

Grep for a pattern 'dear':

```
$ git grep dear  
to-my-dearest.txt:dear
```

git grep

Grep for a pattern 'dear':

```
$ git grep dear  
to-my-dearest.txt:dear
```

Examples

```
$ git grep -n pattern
```

git grep

Grep for a pattern 'dear':

```
$ git grep dear  
to-my-dearest.txt:dear
```

Examples

```
$ git grep -n pattern
```

```
$ git grep --name-only pattern
```


git grep

Grep for a pattern 'dear':

```
$ git grep dear  
to-my-dearest.txt:dear
```

Examples

```
$ git grep -n pattern
```

```
$ git grep --name-only pattern
```

```
$ git grep pattern v1.5.0
```

git grep

Grep for a pattern 'dear':

```
$ git grep dear  
to-my-dearest.txt:dear
```

Examples

```
$ git grep -n pattern
```

```
$ git grep --name-only pattern
```

```
$ git grep pattern v1.5.0
```

```
$ git grep -e '#define' --and -e SORT_DIRENT
```

git-log, git-show

git log

```
$ git log -p
```

git show

git-log, git-show

git log

```
$ git log -p  
$ git log --stat
```

git show

git-log, git-show

git log

```
$ git log -p  
$ git log --stat  
$ git log -5 master@{yesterday}
```

git show

git-log, git-show

git log

```
$ git log -p  
$ git log --stat  
$ git log -5 master@{yesterday}  
$ git log v1.1.10.. README
```

git show

git-log, git-show

git log

```
$ git log -p
$ git log --stat
$ git log -5 master@{yesterday}
$ git log v1.1.10.. README
$ git log --since="2 weeks ago"
```

git show

git-log, git-show

git log

```
$ git log -p
$ git log --stat
$ git log -5 master@{yesterday}
$ git log v1.1.10.. README
$ git log --since="2 weeks ago"
$ git log --before="2 weeks ago"
```

git show

git-log, git-show

git log

```
$ git log -p
$ git log --stat
$ git log -5 master@{yesterday}
$ git log v1.1.10.. README
$ git log --since="2 weeks ago"
$ git log --before="2 weeks ago"
$ git log --author=shakthimaan --since="10 days ago"
```

git show

git-log, git-show

git log

```
$ git log -p
$ git log --stat
$ git log -5 master@{yesterday}
$ git log v1.1.10.. README
$ git log --since="2 weeks ago"
$ git log --before="2 weeks ago"
$ git log --author=shakthimaan --since="10 days ago"
```

git show

```
$ git show v1.0.0
```

git-log, git-show

git log

```
$ git log -p
$ git log --stat
$ git log -5 master@{yesterday}
$ git log v1.1.10.. README
$ git log --since="2 weeks ago"
$ git log --before="2 weeks ago"
$ git log --author=shakthimaan --since="10 days ago"
```

git show

```
$ git show v1.0.0
$ git show HEAD
```

git-log, git-show

git log

```
$ git log -p
$ git log --stat
$ git log -5 master@{yesterday}
$ git log v1.1.10.. README
$ git log --since="2 weeks ago"
$ git log --before="2 weeks ago"
$ git log --author=shakthimaan --since="10 days ago"
```

git show

```
$ git show v1.0.0
$ git show HEAD
$ git show HEAD^^
```

git-log, git-show

git log

```
$ git log -p
$ git log --stat
$ git log -5 master@{yesterday}
$ git log v1.1.10.. README
$ git log --since="2 weeks ago"
$ git log --before="2 weeks ago"
$ git log --author=shakthimaan --since="10 days ago"
```

git show

```
$ git show v1.0.0
$ git show HEAD
$ git show HEAD^^
$ git show master~5
```

git-log, git-show

git log

```
$ git log -p
$ git log --stat
$ git log -5 master@{yesterday}
$ git log v1.1.10.. README
$ git log --since="2 weeks ago"
$ git log --before="2 weeks ago"
$ git log --author=shakthimaan --since="10 days ago"
```

git show

```
$ git show v1.0.0
$ git show HEAD
$ git show HEAD^^
$ git show master~5
$ git show master@{May.23}
```

git stash

Edit to-my-dearest.txt. Something else to be done, stash it!

```
$ git stash
```

git stash

Edit to-my-dearest.txt. Something else to be done, stash it!

```
$ git stash
Saved working directory and index state "WIP on master: 958d5ac...
First commit"
HEAD is now at 958d5ac First commit
(To restore them type "git stash apply")
```


git stash

Edit to-my-dearest.txt. Something else to be done, stash it!

```
$ git stash
Saved working directory and index state "WIP on master: 958d5ac...
First commit"
HEAD is now at 958d5ac First commit
(To restore them type "git stash apply")
```

```
$ git stash list
```

git stash

Edit to-my-dearest.txt. Something else to be done, stash it!

```
$ git stash
Saved working directory and index state "WIP on master: 958d5ac...
First commit"
HEAD is now at 958d5ac First commit
(To restore them type "git stash apply")
```

```
$ git stash list
stash@{0}: WIP on master: 958d5ac... First commit
```

git stash (2)

Do any local changes, or commits. Reapply the stashed changes:

```
$ git stash apply
```

git stash (2)

Do any local changes, or commits. Reapply the stashed changes:

```
$ git stash apply
# On branch master
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#
# modified:   to-my-dearest.txt
#
no changes added to commit (use "git add" and/or "git commit -a")
```

git stash (2)

Do any local changes, or commits. Reapply the stashed changes:

```
$ git stash apply
# On branch master
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#
# modified:   to-my-dearest.txt
#
no changes added to commit (use "git add" and/or "git commit -a")

$ git commit -a -m "Modified to-my-dearest.txt"
```

git stash (2)

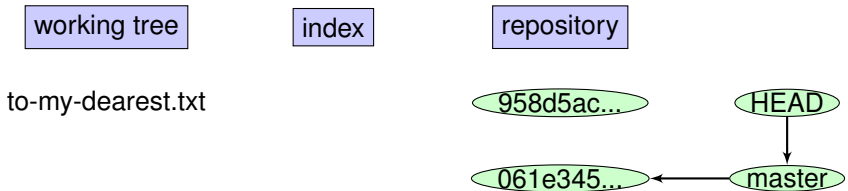
Do any local changes, or commits. Reapply the stashed changes:

```
$ git stash apply
# On branch master
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#
# modified:   to-my-dearest.txt
#
no changes added to commit (use "git add" and/or "git commit -a")

$ git commit -a -m "Modified to-my-dearest.txt"
Created commit 061e345: Modified to-my-dearest.txt
1 files changed, 3 insertions(+), 1 deletions(-)
```

git tag

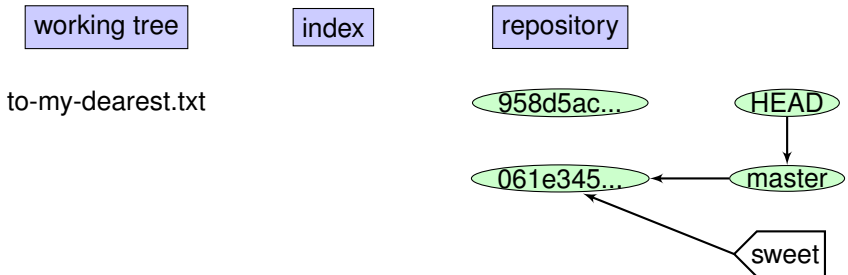
Create a tag named sweet:



git tag

Create a tag named sweet:

```
$ git tag sweet
```

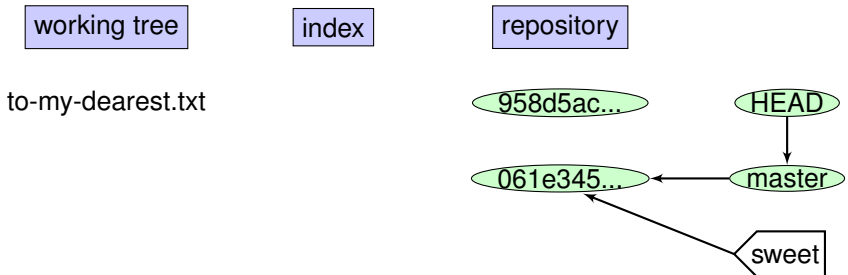


git tag

Create a tag named sweet:

```
$ git tag sweet
```

```
$ git tag
```



git tag

Create a tag named sweet:

```
$ git tag sweet
```

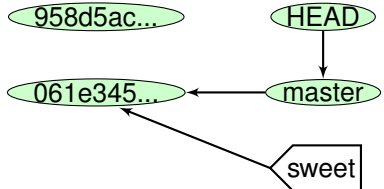
```
$ git tag  
sweet
```

working tree

index

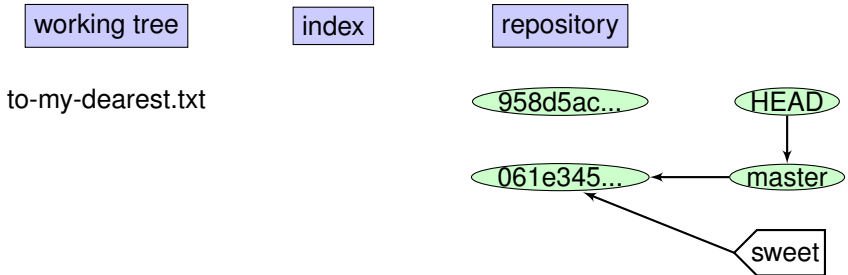
repository

to-my-dearest.txt



git tag (2)

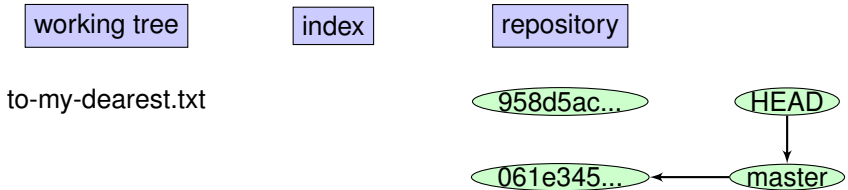
Remove a tag using -d (delete) option:



git tag (2)

Remove a tag using -d (delete) option:

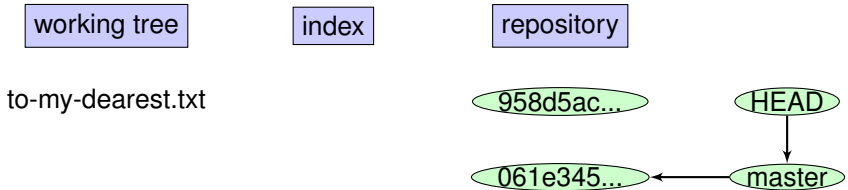
```
$ git tag -d sweet
```



git tag (2)

Remove a tag using -d (delete) option:

```
$ git tag -d sweet  
Deleted tag 'sweet'
```

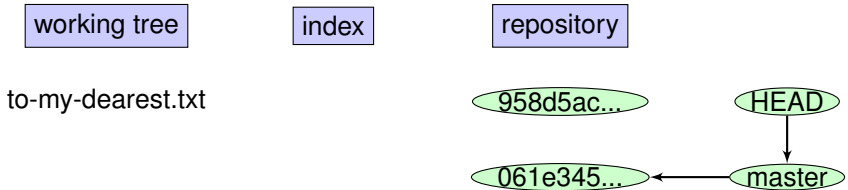


git tag (2)

Remove a tag using -d (delete) option:

```
$ git tag -d sweet  
Deleted tag 'sweet'
```

```
$ git tag
```



git tag (2)

Remove a tag using -d (delete) option:

```
$ git tag -d sweet  
Deleted tag 'sweet'
```

```
$ git tag
```

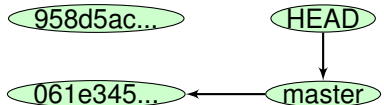
💰

working tree

index

repository

to-my-dearest.txt



git format patch

Create a patch for the last commit:

```
$ git-format-patch -1
```


git format patch

Create a patch for the last commit:

```
$ git-format-patch -1  
0001-Modified-to-my-dearest.txt.patch
```

git format patch

Create a patch for the last commit:

```
$ git-format-patch -1  
0001-Modified-to-my-dearest.txt.patch
```

Use git-send-email to send a collection of patches by email.

git format patch(2)

```
$ cat 0001-Modified-to-my-dearest.txt.patch
```

git format patch(2)

```
$ cat 0001-Modified-to-my-dearest.txt.patch
From 061e345c4079bf66f5ce5bd52539346f9edcee1c Mon Sep 17 00:00:00 2001
From: Shakthi Kannan <shakthimaan@gmail.com>
Date: Sat, 25 Apr 2009 16:31:27 +0530
Subject: [PATCH] Modified to-my-dearest.txt

---
to-my-dearest.txt |    4 +++-
1 files changed, 3 insertions(+), 1 deletions(-)

diff --git a/to-my-dearest.txt b/to-my-dearest.txt
index 27853ff..2e4b9d9 100644
--- a/to-my-dearest.txt
+++ b/to-my-dearest.txt
@@ -1,3 @@
-Dear
+Dear X,
+
+You are very sweet!
--
1.6.0.6
```

git reflog

```
$ git reflog
```

git reflog

```
$ git reflog
061e345... HEAD@{0}: commit: Modified to-my-dearest.txt
958d5ac... HEAD@{1}: HEAD~: updating HEAD
94fafbc... HEAD@{2}: commit (amend): First commit.
...
```

git reflog

```
$ git reflog
061e345... HEAD@{0}: commit: Modified to-my-dearest.txt
958d5ac... HEAD@{1}: HEAD~: updating HEAD
94fafbc... HEAD@{2}: commit (amend): First commit.
...
```

```
$ git reset --hard HEAD@{1}
```

git reflog

```
$ git reflog
061e345... HEAD@{0}: commit: Modified to-my-dearest.txt
958d5ac... HEAD@{1}: HEAD~: updating HEAD
94fafbc... HEAD@{2}: commit (amend): First commit.
...

$ git reset --hard HEAD@{1}
HEAD is now at 958d5ac First commit
```


git reflog

```
$ git reflog
061e345... HEAD@{0}: commit: Modified to-my-dearest.txt
958d5ac... HEAD@{1}: HEAD~: updating HEAD
94fafbc... HEAD@{2}: commit (amend): First commit.
...
```

```
$ git reset --hard HEAD@{1}
HEAD is now at 958d5ac First commit
```

```
$ git gl
```

git reflog

```
$ git reflog
061e345... HEAD@{0}: commit: Modified to-my-dearest.txt
958d5ac... HEAD@{1}: HEAD~: updating HEAD
94fafbc... HEAD@{2}: commit (amend): First commit.
...

$ git reset --hard HEAD@{1}
HEAD is now at 958d5ac First commit

$ git gl
* 958d5ac... First commit
```

git branch

Why have multiple files? Track content, not files!

```
$ git branch
```

git branch

Why have multiple files? Track content, not files!

```
$ git branch  
* master
```

git branch

Why have multiple files? Track content, not files!

```
$ git branch  
* master
```

```
$ git branch priyaaanka-chopra
```

git branch

Why have multiple files? Track content, not files!

```
$ git branch
```

```
* master
```

```
$ git branch priyaaanka-chopra
```

```
$ git branch
```

git branch

Why have multiple files? Track content, not files!

```
$ git branch
```

```
* master
```

```
$ git branch priyaaanka-chopra
```

```
$ git branch
```

```
* master
```

```
priyaaanka-chopra
```

git branch

Why have multiple files? Track content, not files!

```
$ git branch
```

```
* master
```

```
$ git branch priyaaanka-chopra
```

```
$ git branch
```

```
* master
```

```
  priyaaanka-chopra
```

```
$ git branch -d priyaaanka-chopra
```


git branch

Why have multiple files? Track content, not files!

```
$ git branch
```

```
* master
```

```
$ git branch priyaaanka-chopra
```

```
$ git branch
```

```
* master
```

```
  priyaaanka-chopra
```

```
$ git branch -d priyaaanka-chopra
```

```
Deleted branch priyaaanka-chopra.
```

git branch (2)

```
$ git checkout -b priyaaanka-chopra
```

git branch (2)

```
$ git checkout -b priyaaanka-chopra  
Switched to a new branch "priyaaanka-chopra"
```

git branch (2)

```
$ git checkout -b priyaaanka-chopra  
Switched to a new branch "priyaaanka-chopra"
```

```
$ git branch
```

git branch (2)

```
$ git checkout -b priyaaanka-chopra  
Switched to a new branch "priyaaanka-chopra"
```

```
$ git branch  
* priyaaanka-chopra  
  master
```

git branch (2)

```
$ git checkout -b priyaaanka-chopra  
Switched to a new branch "priyaaanka-chopra"
```

```
$ git branch  
* priyaaanka-chopra  
master
```

```
$ git checkout master
```

git branch (2)

```
$ git checkout -b priyaaanka-chopra  
Switched to a new branch "priyaaanka-chopra"
```

```
$ git branch  
* priyaaanka-chopra  
master
```

```
$ git checkout master  
Switched to branch "master"
```

git branch (2)

```
$ git checkout -b priyaaanka-chopra  
Switched to a new branch "priyaaanka-chopra"
```

```
$ git branch  
* priyaaanka-chopra  
  master
```

```
$ git checkout master  
Switched to branch "master"
```

```
$ git branch
```


git branch (2)

```
$ git checkout -b priyaaanka-chopra  
Switched to a new branch "priyaaanka-chopra"
```

```
$ git branch  
* priyaaanka-chopra  
  master
```

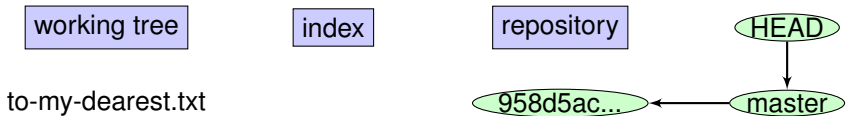
```
$ git checkout master  
Switched to branch "master"
```

```
$ git branch  
  priyaaanka-chopra  
* master
```

git merge

```
$ git checkout priyaaanka-chopra
```

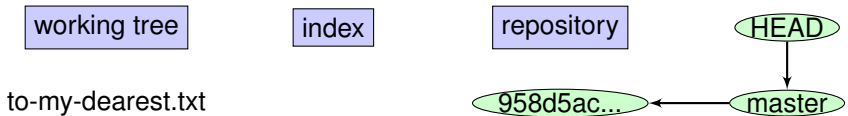
priyaaanka-chopra branch



git merge

```
$ git checkout priyaaanka-chopra  
Switched to branch "priyaaanka-chopra"
```

priyaaanka-chopra branch



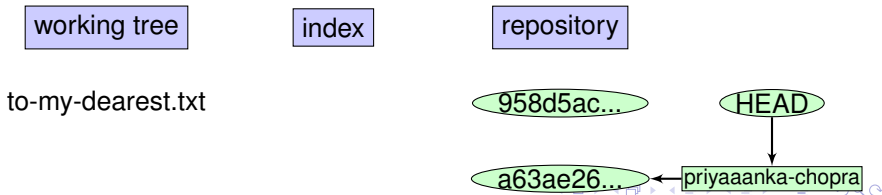
git merge

```
$ git checkout priyaaanka-chopra  
Switched to branch "priyaaanka-chopra"
```

Edit file and commit the same.

```
$ git commit -a -m "Added very sweet"
```

priyaaanka-chopra branch



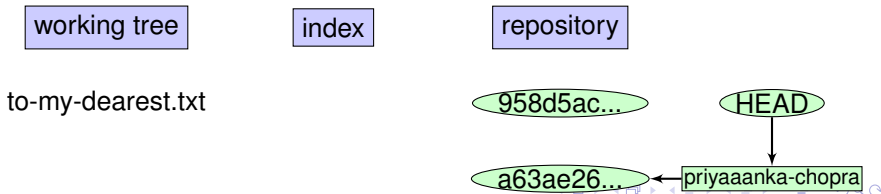
git merge

```
$ git checkout priyaaanka-chopra  
Switched to branch "priyaaanka-chopra"
```

Edit file and commit the same.

```
$ git commit -a -m "Added very sweet"  
Created commit a63ae26: Added very sweet  
1 files changed, 2 insertions(+), 0 deletions(-)
```

priyaaanka-chopra branch



git merge

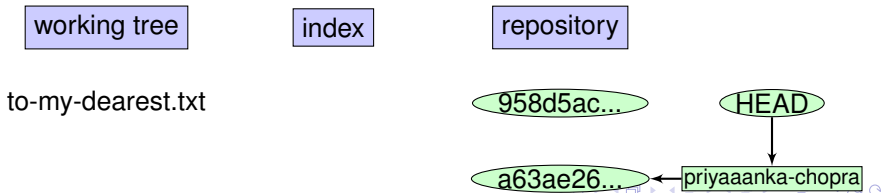
```
$ git checkout priyaaanka-chopra  
Switched to branch "priyaaanka-chopra"
```

Edit file and commit the same.

```
$ git commit -a -m "Added very sweet"  
Created commit a63ae26: Added very sweet  
1 files changed, 2 insertions(+), 0 deletions(-)
```

```
$ git gl
```

priyaaanka-chopra branch



git merge

```
$ git checkout priyaaanka-chopra  
Switched to branch "priyaaanka-chopra"
```

Edit file and commit the same.

```
$ git commit -a -m "Added very sweet"  
Created commit a63ae26: Added very sweet  
1 files changed, 2 insertions(+), 0 deletions(-)
```

```
$ git gl  
* a63ae26... Added very sweet  
* 958d5ac... First commit
```

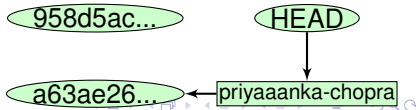
priyaaanka-chopra branch

working tree

index

repository

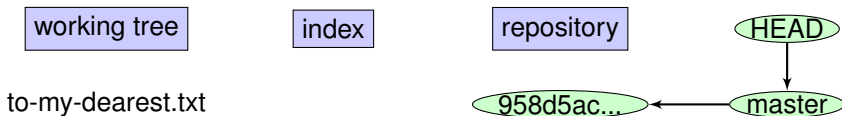
to-my-dearest.txt



git merge (2)

```
$ git checkout master
```

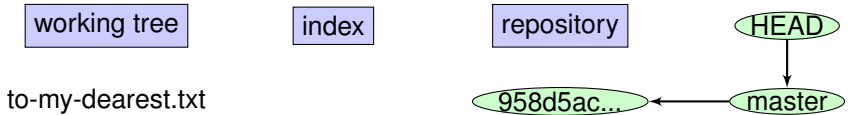
master branch



git merge (2)

```
$ git checkout master  
Switched to branch "master"
```

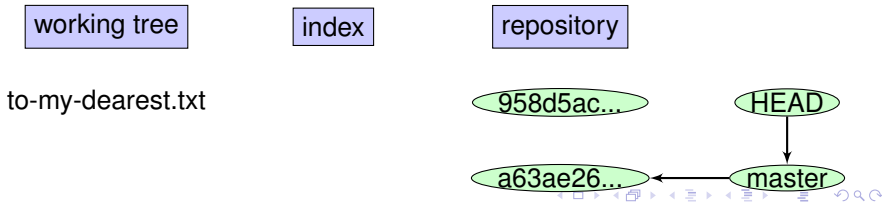
master branch



git merge (2)

```
$ git checkout master  
Switched to branch "master"  
  
$ git merge priyaaanka-chopra
```

master branch

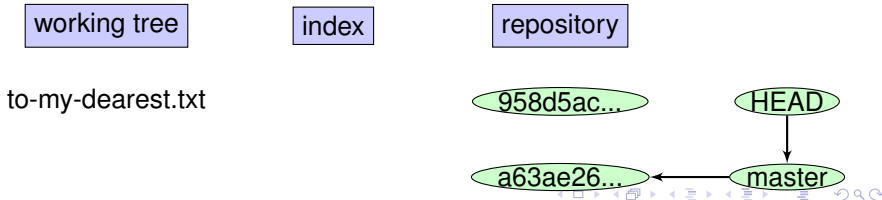


git merge (2)

```
$ git checkout master
Switched to branch "master"

$ git merge priyaaanka-chopra
Updating 958d5ac..a63ae26
Fast forward
 to-my-dearest.txt |    2 ++
 1 files changed, 2 insertions(+), 0 deletions(-)
```

master branch



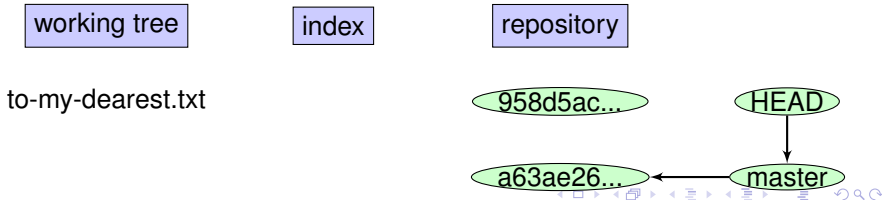
git merge (2)

```
$ git checkout master
Switched to branch "master"

$ git merge priyaaanka-chopra
Updating 958d5ac..a63ae26
Fast forward
 to-my-dearest.txt | 2 ++
1 files changed, 2 insertions(+), 0 deletions(-)

$ git gl
```

master branch



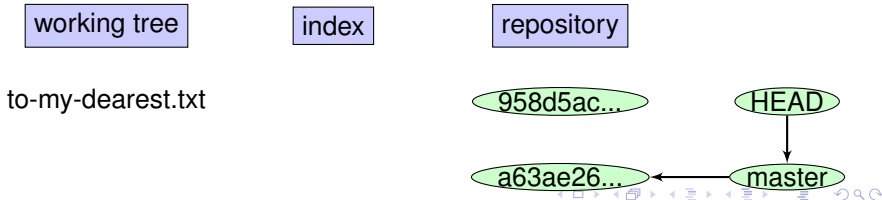
git merge (2)

```
$ git checkout master
Switched to branch "master"

$ git merge priyaaanka-chopra
Updating 958d5ac..a63ae26
Fast forward
 to-my-dearest.txt |    2 ++
 1 files changed, 2 insertions(+), 0 deletions(-)

$ git gl
* a63ae26... Added very sweet
* 958d5ac... First commit
```

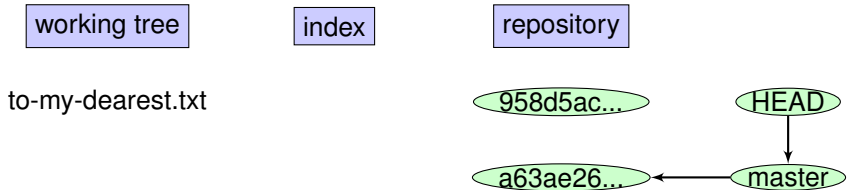
master branch



git rebase

Get changes from upstream (or remote repo), or commit manually to master branch:

master branch

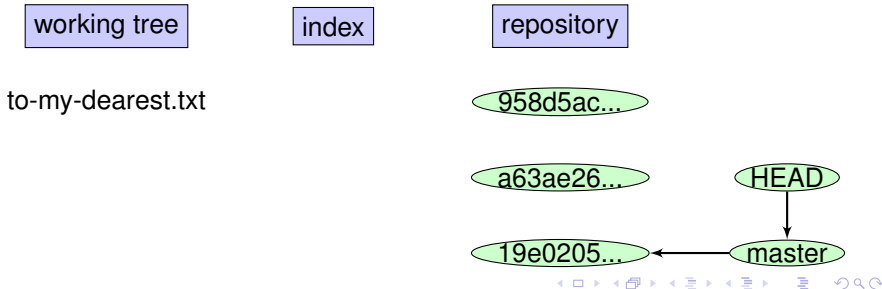


git rebase

Get changes from upstream (or remote repo), or commit manually to master branch:

```
$ git commit -a -m "Added very, very sweet"
```

master branch

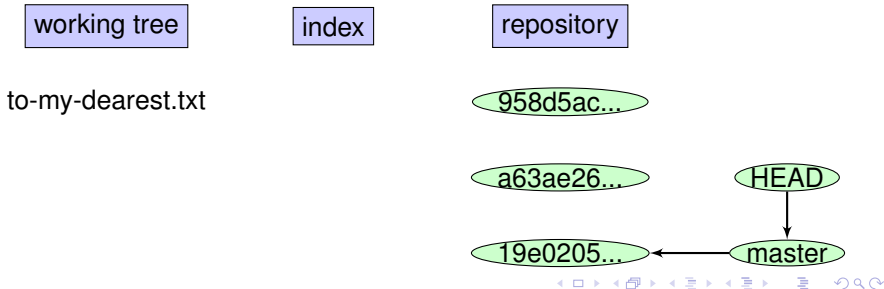


git rebase

Get changes from upstream (or remote repo), or commit manually to master branch:

```
$ git commit -a -m "Added very, very sweet"
Created commit 19e0205: Added very, very sweet
1 files changed, 1 insertions(+), 1 deletions(-)
```

master branch



git rebase (2)

```
$ git branch
```

git rebase (2)

```
$ git branch  
  priyaaanka-chopra  
* master
```

git rebase (2)

```
$ git branch  
  priyaaanka-chopra  
* master
```

```
$ git gl
```

git rebase (2)

```
$ git branch
  priyaaanka-chopra
* master

$ git gl
* 19e0205... Added very, very sweet
* a63ae26... Added very sweet
* 958d5ac... First commit
```

git rebase (2)

```
$ git branch
  priyaaanka-chopra
* master

$ git gl
* 19e0205... Added very, very sweet
* a63ae26... Added very sweet
* 958d5ac... First commit

$ git checkout priyaaanka-chopra
```

git rebase (2)

```
$ git branch
  priyaaanka-chopra
* master

$ git gl
* 19e0205... Added very, very sweet
* a63ae26... Added very sweet
* 958d5ac... First commit

$ git checkout priyaaanka-chopra
Switched to branch "priyaaanka-chopra"
```

git rebase (2)

```
$ git branch
  priyaaanka-chopra
* master

$ git gl
* 19e0205... Added very, very sweet
* a63ae26... Added very sweet
* 958d5ac... First commit

$ git checkout priyaaanka-chopra
Switched to branch "priyaaanka-chopra"

$ git branch
```

git rebase (2)

```
$ git branch
  priyaaanka-chopra
* master

$ git gl
* 19e0205... Added very, very sweet
* a63ae26... Added very sweet
* 958d5ac... First commit

$ git checkout priyaaanka-chopra
Switched to branch "priyaaanka-chopra"

$ git branch
* priyaaanka-chopra
  master
```


git rebase (2)

```
$ git branch
  priyaaanka-chopra
* master

$ git gl
* 19e0205... Added very, very sweet
* a63ae26... Added very sweet
* 958d5ac... First commit

$ git checkout priyaaanka-chopra
Switched to branch "priyaaanka-chopra"

$ git branch
* priyaaanka-chopra
  master

$ git gl
```

git rebase (2)

```
$ git branch
  priyaaanka-chopra
* master

$ git gl
* 19e0205... Added very, very sweet
* a63ae26... Added very sweet
* 958d5ac... First commit

$ git checkout priyaaanka-chopra
Switched to branch "priyaaanka-chopra"

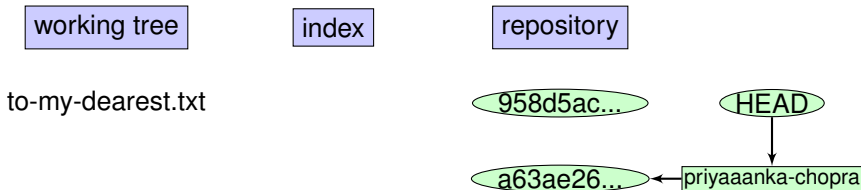
$ git branch
* priyaaanka-chopra
  master

$ git gl
* a63ae26... Added very sweet
* 958d5ac... First commit
```

git rebase (3)

Add your changes to priyaaanka-chopra branch and commit:

priyaaanka-chopra branch

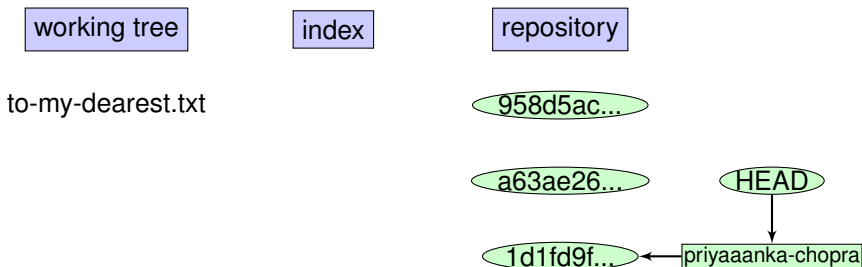


git rebase (3)

Add your changes to priyaaanka-chopra branch and commit:

```
$ git commit -a -m "Added so, so sweet"
```

priyaaanka-chopra branch

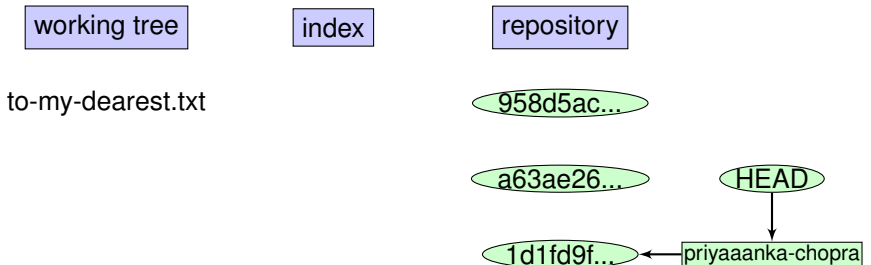


git rebase (3)

Add your changes to priyaaanka-chopra branch and commit:

```
$ git commit -a -m "Added so, so sweet"
Created commit 1d1fd9f: Added so, so sweet
1 files changed, 1 insertions(+), 1 deletions(-)
```

priyaaanka-chopra branch



git rebase(4)

Now, we have changes in master (upstream) as well as in branch. We can re-write history and update the same with rebase:

```
$ git rebase master
```

git rebase(4)

Now, we have changes in master (upstream) as well as in branch. We can re-write history and update the same with rebase:

```
$ git rebase master
```

```
First, rewinding head to replay your work on top of it...
```

```
Applying: Added so, so sweet
```

```
error: patch failed: to-my-dearest.txt:1
```

```
error: to-my-dearest.txt: patch does not apply
```

```
Using index info to reconstruct a base tree...
```

```
Falling back to patching base and 3-way merge...
```

```
Auto-merged to-my-dearest.txt
```

```
CONFLICT (content): Merge conflict in to-my-dearest.txt
```

```
Failed to merge in the changes.
```

```
Patch failed at 0001.
```

When you have resolved this problem run "git rebase --continue".

If you would prefer to skip this patch, instead run "git rebase --skip".

To restore the original branch and stop rebasing run "git rebase --abort".

git rebase(5)

This is the conflict:

Dear

```
<<<<<< HEAD:to-my-dearest.txt
You are very, very sweet!
=====
You are so, so sweet!
>>>>>> Added so, so sweet:to-my-dearest.txt
```


git rebase(5)

This is the conflict:

Dear

```
<<<<<< HEAD:to-my-dearest.txt
You are very, very sweet!
=====
You are so, so sweet!
>>>>>> Added so, so sweet:to-my-dearest.txt
```

We manually fix it:

Dear

You are very, very sweet!

git rebase(6)

```
$ git gl
```

git rebase(6)

```
$ git gl  
* 1d1fd9f... Added so, so sweet  
* a63ae26... Added very sweet  
* 958d5ac... First commit
```

git rebase(6)

```
$ git gl
* 1d1fd9f... Added so, so sweet
* a63ae26... Added very sweet
* 958d5ac... First commit
```

Since, file has been modified, we need to stage it (add to index):

```
$ git add to-my-dearest.txt
```

git rebase(6)

```
$ git gl
* 1d1fd9f... Added so, so sweet
* a63ae26... Added very sweet
* 958d5ac... First commit
```

Since, file has been modified, we need to stage it (add to index):

```
$ git add to-my-dearest.txt
```

```
$ git rebase --skip
```

git rebase(6)

```
$ git gl
* 1d1fd9f... Added so, so sweet
* a63ae26... Added very sweet
* 958d5ac... First commit
```

Since, file has been modified, we need to stage it (add to index):

```
$ git add to-my-dearest.txt
```

```
$ git rebase --skip
```

```
HEAD is now at 19e0205 Added very, very sweet
Nothing to do.
```

git rebase(6)

```
$ git gl
* 1d1fd9f... Added so, so sweet
* a63ae26... Added very sweet
* 958d5ac... First commit
```

Since, file has been modified, we need to stage it (add to index):

```
$ git add to-my-dearest.txt
```

```
$ git rebase --skip
HEAD is now at 19e0205 Added very, very sweet
Nothing to do.
```

```
$ git gl
```

git rebase(6)

```
$ git gl
* 1d1fd9f... Added so, so sweet
* a63ae26... Added very sweet
* 958d5ac... First commit
```

Since, file has been modified, we need to stage it (add to index):

```
$ git add to-my-dearest.txt
```

```
$ git rebase --skip
HEAD is now at 19e0205 Added very, very sweet
Nothing to do.
```

```
$ git gl
* 19e0205... Added very, very sweet
* a63ae26... Added very sweet
* 958d5ac... First commit
```


git rebase(6)

```
$ git gl
* 1d1fd9f... Added so, so sweet
* a63ae26... Added very sweet
* 958d5ac... First commit
```

Since, file has been modified, we need to stage it (add to index):

```
$ git add to-my-dearest.txt
```

```
$ git rebase --skip
HEAD is now at 19e0205 Added very, very sweet
Nothing to do.
```

```
$ git gl
* 19e0205... Added very, very sweet
* a63ae26... Added very sweet
* 958d5ac... First commit
```

It has overwritten previous "1d1fd9f... Added so, so sweet" commit on priyaaanka-chopra branch.

git rebase(6)

```
$ git gl
* 1d1fd9f... Added so, so sweet
* a63ae26... Added very sweet
* 958d5ac... First commit
```

Since, file has been modified, we need to stage it (add to index):

```
$ git add to-my-dearest.txt
```

```
$ git rebase --skip
HEAD is now at 19e0205 Added very, very sweet
Nothing to do.
```

```
$ git gl
* 19e0205... Added very, very sweet
* a63ae26... Added very sweet
* 958d5ac... First commit
```

It has overwritten previous "1d1fd9f... Added so, so sweet" commit on priyaaanka-chopra branch.

Always rebase on local branches.

git clone

Initial repository:

```
$ git clone git://eagain.net/gitosis.git
```

git clone

Initial repository:

```
$ git clone git://eagain.net/gitosis.git  
Initialized empty Git repository in /tmp/gitosis/.git/  
remote: Counting objects: 603, done.  
remote: Compressing objects: 100% (172/172), done.  
remote: Total 603 (delta 425), reused 597 (delta 422)  
Receiving objects: 100% (603/603), 92.87 KiB | 23 KiB/s, done.  
Resolving deltas: 100% (425/425), done.
```

git clone

Initial repository:

```
$ git clone git://eagain.net/gitosis.git
Initialized empty Git repository in /tmp/gitosis/.git/
remote: Counting objects: 603, done.
remote: Compressing objects: 100% (172/172), done.
remote: Total 603 (delta 425), reused 597 (delta 422)
Receiving objects: 100% (603/603), 92.87 KiB | 23 KiB/s, done.
Resolving deltas: 100% (425/425), done.
```

To get updates:

```
$ git fetch
```

git clone

Initial repository:

```
$ git clone git://eagain.net/gitosis.git
Initialized empty Git repository in /tmp/gitosis/.git/
remote: Counting objects: 603, done.
remote: Compressing objects: 100% (172/172), done.
remote: Total 603 (delta 425), reused 597 (delta 422)
Receiving objects: 100% (603/603), 92.87 KiB | 23 KiB/s, done.
Resolving deltas: 100% (425/425), done.
```

To get updates:

```
$ git fetch
```

To fetch and merge:

```
$ git pull
```

git clone (2)

Remote repository notations:

```
rsync://host.xz/path/to/repo.git/
```

```
http://host.xz[:port]/path/to/repo.git/  
https://host.xz[:port]/path/to/repo.git/
```

```
git://host.xz[:port]/path/to/repo.git/  
git://host.xz[:port]/~user/path/to/repo.git/
```

```
ssh://[user@]host.xz[:port]/path/to/repo.git/  
ssh://[user@]host.xz/path/to/repo.git/  
ssh://[user@]host.xz/~user/path/to/repo.git/  
ssh://[user@]host.xz/~ /path/to/repo.git
```

git remote

```
$ git clone git@gitorious.org:di-git-ally-managing-love-letters/mainline.git
```


git remote

```
$ git clone git@gitorious.org:di-git-ally-managing-love-letters/mainline.git
Initialized empty Git repository in /tmp/mainline/.git/
remote: Counting objects: 12, done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 12 (delta 4), reused 0 (delta 0)
Receiving objects: 100% (12/12), 9.41 KiB, done.
Resolving deltas: 100% (4/4), done.
```

git remote

```
$ git clone git@gitorious.org:di-git-ally-managing-love-letters/mainline.git
Initialized empty Git repository in /tmp/mainline/.git/
remote: Counting objects: 12, done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 12 (delta 4), reused 0 (delta 0)
Receiving objects: 100% (12/12), 9.41 KiB, done.
Resolving deltas: 100% (4/4), done.
```

```
$ git remote
```

git remote

```
$ git clone git@gitorious.org:di-git-ally-managing-love-letters/mainline.git
Initialized empty Git repository in /tmp/mainline/.git/
remote: Counting objects: 12, done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 12 (delta 4), reused 0 (delta 0)
Receiving objects: 100% (12/12), 9.41 KiB, done.
Resolving deltas: 100% (4/4), done.
```

```
$ git remote
origin
```

git remote

```
$ git clone git@gitorious.org:di-git-ally-managing-love-letters/mainline.git
Initialized empty Git repository in /tmp/mainline/.git/
remote: Counting objects: 12, done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 12 (delta 4), reused 0 (delta 0)
Receiving objects: 100% (12/12), 9.41 KiB, done.
Resolving deltas: 100% (4/4), done.
```

```
$ git remote
origin
```

```
$ git branch
```

git remote

```
$ git clone git@gitorious.org:di-git-ally-managing-love-letters/mainline.git
Initialized empty Git repository in /tmp/mainline/.git/
remote: Counting objects: 12, done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 12 (delta 4), reused 0 (delta 0)
Receiving objects: 100% (12/12), 9.41 KiB, done.
Resolving deltas: 100% (4/4), done.
```

```
$ git remote
origin
```

```
$ git branch
* master
```

git remote

```
$ git clone git@gitorious.org:di-git-ally-managing-love-letters/mainline.git
Initialized empty Git repository in /tmp/mainline/.git/
remote: Counting objects: 12, done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 12 (delta 4), reused 0 (delta 0)
Receiving objects: 100% (12/12), 9.41 KiB, done.
Resolving deltas: 100% (4/4), done.
```

```
$ git remote
origin
```

```
$ git branch
* master
```

```
$ git branch -r
```

git remote

```
$ git clone git@gitorious.org:di-git-ally-managing-love-letters/mainline.git
Initialized empty Git repository in /tmp/mainline/.git/
remote: Counting objects: 12, done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 12 (delta 4), reused 0 (delta 0)
Receiving objects: 100% (12/12), 9.41 KiB, done.
Resolving deltas: 100% (4/4), done.
```

```
$ git remote
origin
```

```
$ git branch
* master
```

```
$ git branch -r
origin/HEAD
origin/master
```

git remote (2)

```
$ git remote add sagarun  
git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/  
sags-di-git-ally-managing-love-letters-clone.git
```


git remote (2)

```
$ git remote add sagarun  
    git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/  
    sags-di-git-ally-managing-love-letters-clone.git
```

```
$ git remote
```

git remote (2)

```
$ git remote add sagarun  
    git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/  
    sags-di-git-ally-managing-love-letters-clone.git
```

```
$ git remote  
origin  
sagarun
```

git remote (2)

```
$ git remote add sagarun  
    git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/  
    sags-di-git-ally-managing-love-letters-clone.git
```

```
$ git remote  
origin  
sagarun
```

```
$ git fetch sagarun
```

git remote (2)

```
$ git remote add sagarun  
    git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/  
    sags-di-git-ally-managing-love-letters-clone.git  
  
$ git remote  
origin  
sagarun  
  
$ git fetch sagarun  
From git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/  
sags-di-git-ally-managing-love-letters-clone  
* [new branch]      master      -> sagarun/master
```

git remote (2)

```
$ git remote add sagarun  
    git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/  
    sags-di-git-ally-managing-love-letters-clone.git  
  
$ git remote  
origin  
sagarun  
  
$ git fetch sagarun  
From git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/  
sags-di-git-ally-managing-love-letters-clone  
* [new branch]      master      -> sagarun/master  
  
$ git branch -r
```

git remote (2)

```
$ git remote add sagarun
    git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/
    sags-di-git-ally-managing-love-letters-clone.git

$ git remote
origin
sagarun

$ git fetch sagarun
From git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/
sags-di-git-ally-managing-love-letters-clone
* [new branch]      master      -> sagarun/master

$ git branch -r
origin/HEAD
origin/master
sagarun/master
```

git remote (2)

```
$ git remote add sagarun
  git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/
  sags-di-git-ally-managing-love-letters-clone.git

$ git remote
origin
sagarun

$ git fetch sagarun
From git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/
sags-di-git-ally-managing-love-letters-clone
* [new branch]      master      -> sagarun/master

$ git branch -r
origin/HEAD
origin/master
sagarun/master

$ git diff master..sagarun/master
```

git remote (2)

```
$ git remote add sagarun
  git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/
  sags-di-git-ally-managing-love-letters-clone.git

$ git remote
origin
sagarun

$ git fetch sagarun
From git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/
sags-di-git-ally-managing-love-letters-clone
* [new branch]      master      -> sagarun/master

$ git branch -r
origin/HEAD
origin/master
sagarun/master

$ git diff master..sagarun/master
$ git checkout -b sagarun-review sagarun/master
```


git remote (2)

```
$ git remote add sagarun
  git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/
  sags-di-git-ally-managing-love-letters-clone.git

$ git remote
origin
sagarun

$ git fetch sagarun
From git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/
sags-di-git-ally-managing-love-letters-clone
* [new branch]      master      -> sagarun/master

$ git branch -r
origin/HEAD
origin/master
sagarun/master

$ git diff master..sagarun/master
$ git checkout -b sagarun-review sagarun/master
$ git reset --hard 00359690a7283d2e9db9473df88b0e9292ed0bc7
```

git remote (2)

```
$ git remote add sagarun
  git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/
  sags-di-git-ally-managing-love-letters-clone.git

$ git remote
origin
sagarun

$ git fetch sagarun
From git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/
sags-di-git-ally-managing-love-letters-clone
* [new branch]      master      -> sagarun/master

$ git branch -r
origin/HEAD
origin/master
sagarun/master

$ git diff master..sagarun/master
$ git checkout -b sagarun-review sagarun/master
$ git reset --hard 00359690a7283d2e9db9473df88b0e9292ed0bc7
```

Review changes, check if they are ok.

git remote (2)

```
$ git remote add sagarun
  git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/
  sags-di-git-ally-managing-love-letters-clone.git

$ git remote
origin
sagarun

$ git fetch sagarun
From git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/
sags-di-git-ally-managing-love-letters-clone
* [new branch]      master      -> sagarun/master

$ git branch -r
origin/HEAD
origin/master
sagarun/master

$ git diff master..sagarun/master
$ git checkout -b sagarun-review sagarun/master
$ git reset --hard 00359690a7283d2e9db9473df88b0e9292ed0bc7
```

Review changes, check if they are ok.

```
$ git checkout master
```

git remote (2)

```
$ git remote add sagarun
  git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/
  sags-di-git-ally-managing-love-letters-clone.git

$ git remote
origin
sagarun

$ git fetch sagarun
From git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/
sags-di-git-ally-managing-love-letters-clone
* [new branch]      master      -> sagarun/master

$ git branch -r
origin/HEAD
origin/master
sagarun/master

$ git diff master..sagarun/master
$ git checkout -b sagarun-review sagarun/master
$ git reset --hard 00359690a7283d2e9db9473df88b0e9292ed0bc7

Review changes, check if they are ok.
$ git checkout master
$ git merge sagarun/master
```

git remote (2)

```
$ git remote add sagarun
   git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/
   sags-di-git-ally-managing-love-letters-clone.git

$ git remote
origin
sagarun

$ git fetch sagarun
From git://gitorious.org/~sagarun/di-git-ally-managing-love-letters/
sags-di-git-ally-managing-love-letters-clone
* [new branch]      master    -> sagarun/master

$ git branch -r
origin/HEAD
origin/master
sagarun/master

$ git diff master..sagarun/master
$ git checkout -b sagarun-review sagarun/master
$ git reset --hard 00359690a7283d2e9db9473df88b0e9292ed0bc7

    Review changes, check if they are ok.

$ git checkout master
$ git merge sagarun/master
$ git push origin master
```