

```
user@LAPTOP-U4MSV6FK MINGW64 /c/oss_python/cau_oss_python_03 (main)
$ git branch -l
error: unknown switch `l'
usage: git branch [<options>] [-r | -a] [--merged] [--no-merged]
      or: git branch [<options>] [-f] [--recurse-submodules] <branch-name> [<start-point>]
      or: git branch [<options>] [-l] [<pattern>...]
      or: git branch [<options>] [-r] (-d | -D) <branch-name>...
      or: git branch [<options>] (-m | -M) [<old-branch>] <new-branch>
      or: git branch [<options>] (-c | -C) [<old-branch>] <new-branch>
      or: git branch [<options>] [-r | -a] [--points-at]
      or: git branch [<options>] [-r | -a] [--format]

Generic options
  -v, --verbose      show hash and subject, give twice for upstream branch
  -q, --quiet        suppress informational messages
  -t, --track[=(direct|inherit)]
                    set branch tracking configuration
  -u, --set-upstream-to <upstream>
                    change the upstream info
  --unset-upstream   unset the upstream info
  --color[=<when>]   use colored output
  -r, --remotes      act on remote-tracking branches
  --contains <commit>
                    print only branches that contain the commit
  --no-contains <commit>
                    print only branches that don't contain the commit
  --abbrev[=<n>]     use <n> digits to display object names

Specific git-branch actions:
  -a, --all          list both remote-tracking and local branches
  -d, --delete       delete fully merged branch
  -D                delete branch (even if not merged)
  -m, --move         move/rename a branch and its reflog
  -M                move/rename a branch, even if target exists
  -c, --copy         copy a branch and its reflog
  -C                copy a branch, even if target exists
  -l, --list         list branch names
  --show-current     show current branch name
  --create-reflog    create the branch's reflog
  --edit-description edit the description for the branch
  -f, --force        force creation, move/rename, deletion
  --merged <commit> print only branches that are merged
  --no-merged <commit>
                    print only branches that are not merged
  --column[=<style>] list branches in columns
  --sort <key>       field name to sort on
  --points-at <object>
                    print only branches of the object
  -i, --ignore-case  sorting and filtering are case insensitive
  --recurse-submodules
                    recurse through submodules
  --format <format> format to use for the output
```

```
user@LAPTOP-U4MSV6FK MINGW64 /c/oss_python/cau_oss_python_03 (main)
$ git branch hw5
```

```
user@LAPTOP-U4MSV6FK MINGW64 /c/oss_python/cau_oss_python_03 (main)
$ git branch -l
error: unknown switch `l'
usage: git branch [<options>] [-r | -a] [--merged] [--no-merged]
      or: git branch [<options>] [-f] [--recurse-submodules] <branch-name> [<start-
point>]
      or: git branch [<options>] [-l] [<pattern>...]
      or: git branch [<options>] [-r] (-d | -D) <branch-name>...
      or: git branch [<options>] (-m | -M) [<old-branch>] <new-branch>
      or: git branch [<options>] (-c | -C) [<old-branch>] <new-branch>
      or: git branch [<options>] [-r | -a] [--points-at]
      or: git branch [<options>] [-r | -a] [--format]
```

Generic options

-v, --verbose	show hash and subject, give twice for upstream branch
-q, --quiet	suppress informational messages
-t, --track[=(direct inherit)]	set branch tracking configuration
-u, --set-upstream-to <upstream>	change the upstream info
--unset-upstream	unset the upstream info
--color[=<when>]	use colored output
-r, --remotes	act on remote-tracking branches
--contains <commit>	print only branches that contain the commit
--no-contains <commit>	
	print only branches that don't contain the commit
--abbrev[=<n>]	use <n> digits to display object names

Specific git-branch actions:

-a, --all	list both remote-tracking and local branches
-d, --delete	delete fully merged branch
-D	delete branch (even if not merged)
-m, --move	move/rename a branch and its reflog
-M	move/rename a branch, even if target exists
-c, --copy	copy a branch and its reflog
-C	copy a branch, even if target exists
-l, --list	list branch names
--show-current	show current branch name
--create-reflog	create the branch's reflog
--edit-description	edit the description for the branch
-f, --force	force creation, move/rename, deletion
--merged <commit>	print only branches that are merged
--no-merged <commit>	print only branches that are not merged
--column[=<style>]	list branches in columns
--sort <key>	field name to sort on
--points-at <object>	print only branches of the object
-i, --ignore-case	sorting and filtering are case insensitive
--recurse-submodules	recurse through submodules
--format <format>	format to use for the output

```
user@LAPTOP-U4MSV6FK MINGW64 /c/oss_python/cau_oss_python_03 (main)
$ git switch hw5
Switched to branch 'hw5'
```

```
user@LAPTOP-U4MSV6FK MINGW64 /c/oss_python/cau_oss_python_03 (hw5)
$ git status
On branch hw5
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md
        modified:   figure.py
        modified:   figure_main.py

no changes added to commit (use "git add" and/or "git commit -a")

user@LAPTOP-U4MSV6FK MINGW64 /c/oss_python/cau_oss_python_03 (hw5)
$ git add figure.py figure_main.py

user@LAPTOP-U4MSV6FK MINGW64 /c/oss_python/cau_oss_python_03 (hw5)
$ git commit -m 'implement hw5'
[hw5 2eb55e5] implement hw5
 2 files changed, 72 insertions(+), 44 deletions(-)
```

```
user@LAPTOP-U4MSV6FK MINGW64 /c/oss_python/cau_oss_python_03 (hw5)
$ git switch main
Switched to branch 'main'
M       README.md
Your branch is up to date with 'origin/main'.

user@LAPTOP-U4MSV6FK MINGW64 /c/oss_python/cau_oss_python_03 (main)
$ git merge hw5
Updating bc93bc5..2eb55e5
Fast-forward
 figure.py      | 78 ++++++++++++++++++++++++++++++++++++++-----
 figure_main.py | 38 ++++++-----
 2 files changed, 72 insertions(+), 44 deletions(-)
```