bhavrs@INLEN8520050892 MINGW64 ~

$ mkdir project\_dir

bhavrs@INLEN8520050892 MINGW64 ~

$ cd project\_dir

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir

$ git init

Initialized empty Git repository in C:/Users/BHAVRS/project\_dir/.git/

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ gitinit

bash: gitinit: command not found

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git init

Reinitialized existing Git repository in C:/Users/BHAVRS/project\_dir/.git/

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ touch index.html

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git status

On branch master

No commits yet

Untracked files:

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

index.html

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

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git add index.html

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: index.html

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git commit -m "committing index.html file"

[master (root-commit) c1b7f7c] committing index.html file

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 index.html

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ touch info.txt

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git status

On branch master

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: index.html

Untracked files:

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

info.txt

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

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ touch .gitignore

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git status

On branch master

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: index.html

Untracked files:

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

.gitignore

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

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git commit -m "First commit"

On branch master

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: index.html

Untracked files:

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

.gitignore

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

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git add index.html

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git commit -m "First commit"

[master 2c4f443] First commit

1 file changed, 8 insertions(+)

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git log

commit 2c4f4432ab661b384cbc23e055cbd76e7c0f158e (HEAD -> master)

Author: Bhavan <bhavanreddy1998@gmail.com>

Date: Sat Jun 18 10:24:38 2022 +0530

First commit

commit c1b7f7cfce050666b66347c27b754268d116659c

Author: Bhavan <bhavanreddy1998@gmail.com>

Date: Sat Jun 18 10:16:26 2022 +0530

committing index.html file

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git revert HEAD-1

fatal: bad revision 'HEAD-1'

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git revert~1

git: 'revert~1' is not a git command. See 'git --help'.

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git --help

usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]

[--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]

[-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]

[--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]

[--super-prefix=<path>] [--config-env=<name>=<envvar>]

<command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)

clone Clone a repository into a new directory

init Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)

add Add file contents to the index

mv Move or rename a file, a directory, or a symlink

restore Restore working tree files

rm Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)

bisect Use binary search to find the commit that introduced a bug

diff Show changes between commits, commit and working tree, etc

grep Print lines matching a pattern

log Show commit logs

show Show various types of objects

status Show the working tree status

grow, mark and tweak your common history

branch List, create, or delete branches

commit Record changes to the repository

merge Join two or more development histories together

rebase Reapply commits on top of another base tip

reset Reset current HEAD to the specified state

switch Switch branches

tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)

fetch Download objects and refs from another repository

pull Fetch from and integrate with another repository or a local branch

push Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some

concept guides. See 'git help <command>' or 'git help <concept>'

to read about a specific subcommand or concept.

See 'git help git' for an overview of the system.

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ ^C

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ ^C

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git revert HEAD --no--edit

usage: git revert [<options>] <commit-ish>...

or: git revert <subcommand>

--quit end revert or cherry-pick sequence

--continue resume revert or cherry-pick sequence

--abort cancel revert or cherry-pick sequence

--skip skip current commit and continue

--cleanup <mode> how to strip spaces and #comments from message

-n, --no-commit don't automatically commit

-e, --edit edit the commit message

-s, --signoff add a Signed-off-by trailer

-m, --mainline <parent-number>

select mainline parent

--rerere-autoupdate update the index with reused conflict resolution if possible

--strategy <strategy>

merge strategy

-X, --strategy-option <option>

option for merge strategy

-S, --gpg-sign[=<key-id>]

GPG sign commit

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git revert --skip

error: no revert in progress

fatal: revert failed

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git revert --quit

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git revert --skip

error: no revert in progress

fatal: revert failed

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git revert --continue--skip

usage: git revert [<options>] <commit-ish>...

or: git revert <subcommand>

--quit end revert or cherry-pick sequence

--continue resume revert or cherry-pick sequence

--abort cancel revert or cherry-pick sequence

--skip skip current commit and continue

--cleanup <mode> how to strip spaces and #comments from message

-n, --no-commit don't automatically commit

-e, --edit edit the commit message

-s, --signoff add a Signed-off-by trailer

-m, --mainline <parent-number>

select mainline parent

--rerere-autoupdate update the index with reused conflict resolution if possible

--strategy <strategy>

merge strategy

-X, --strategy-option <option>

option for merge strategy

-S, --gpg-sign[=<key-id>]

GPG sign commit

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ ^C

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git revert 2c4f443

error: Your local changes to the following files would be overwritten by merge:

index.html

Please commit your changes or stash them before you merge.

Aborting

fatal: revert failed

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git commit -m "second commit"

On branch master

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: index.html

Untracked files:

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

.gitignore

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

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git revert [master 2c4f443]

fatal: bad revision '[master'

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git revert [2c4f443]

fatal: bad revision '[2c4f443]'

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ ^C

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git revert c1b7f7cfce050666b66347c27b754268d116659c

CONFLICT (modify/delete): index.html deleted in (empty tree) and modified in HEA

error: could not revert c1b7f7c... committing index.html file

hint: After resolving the conflicts, mark them with

hint: "git add/rm <pathspec>", then run

hint: "git revert --continue".

hint: You can instead skip this commit with "git revert --skip".

hint: To abort and get back to the state before "git revert",

hint: run "git revert --abort".

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master|REVERTING)

$ git revert --abort

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git revert c1b7f7cfce050666b66347c27b754268d116659c

CONFLICT (modify/delete): index.html deleted in (empty tree) and modified in HEA

error: could not revert c1b7f7c... committing index.html file

hint: After resolving the conflicts, mark them with

hint: "git add/rm <pathspec>", then run

hint: "git revert --continue".

hint: You can instead skip this commit with "git revert --skip".

hint: To abort and get back to the state before "git revert",

hint: run "git revert --abort".

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master|REVERTING)

$ git log

commit 2c4f4432ab661b384cbc23e055cbd76e7c0f158e (HEAD -> master)

Author: Bhavan <bhavanreddy1998@gmail.com>

Date: Sat Jun 18 10:24:38 2022 +0530

First commit

commit c1b7f7cfce050666b66347c27b754268d116659c

Author: Bhavan <bhavanreddy1998@gmail.com>

Date: Sat Jun 18 10:16:26 2022 +0530

committing index.html file

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master|REVERTING)

$ ^C

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master|REVERTING)

$ git revert 2c4f4432ab661b384cbc23e055cbd76e7c0f158e

error: Reverting is not possible because you have unmerged files.

hint: Fix them up in the work tree, and then use 'git add/rm <file>'

hint: as appropriate to mark resolution and make a commit.

fatal: revert failed

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master|REVERTING)

$ git revert 2c4f4432ab661b384cbc23e055cbd76e7c0f158e

error: Reverting is not possible because you have unmerged files.

hint: Fix them up in the work tree, and then use 'git add/rm <file>'

hint: as appropriate to mark resolution and make a commit.

fatal: revert failed

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master|REVERTING)

$ git add .

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master|REVERTING)

$ git revert 2c4f4432ab661b384cbc23e055cbd76e7c0f158e

error: your local changes would be overwritten by revert.

hint: commit your changes or stash them to proceed.

fatal: revert failed

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master|REVERTING)

$ git commit -m "committed"

[master 7f0ea4c] committed

2 files changed, 2 insertions(+)

create mode 100644 .gitignore

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git revert 2c4f4432ab661b384cbc23e055cbd76e7c0f158e

Auto-merging index.html

CONFLICT (content): Merge conflict in index.html

error: could not revert 2c4f443... First commit

hint: After resolving the conflicts, mark them with

hint: "git add/rm <pathspec>", then run

hint: "git revert --continue".

hint: You can instead skip this commit with "git revert --skip".

hint: To abort and get back to the state before "git revert",

hint: run "git revert --abort".

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master|REVERTING)

$ git revert 2c4f4432ab661b384cbc23e055cbd76e7c0f158e

error: Reverting is not possible because you have unmerged files.

hint: Fix them up in the work tree, and then use 'git add/rm <file>'

hint: as appropriate to mark resolution and make a commit.

fatal: revert failed

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master|REVERTING)

$ git revert --abort

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git revert --abort

error: no cherry-pick or revert in progress

fatal: revert failed

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git add .

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git revert 2c4f4432ab661b384cbc23e055cbd76e7c0f158e

error: your local changes would be overwritten by revert.

hint: commit your changes or stash them to proceed.

fatal: revert failed

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git revert --skip

error: no revert in progress

fatal: revert failed

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git commit "third commit"

error: pathspec 'third commit' did not match any file(s) known to git

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git commit -m "third commit"

[master 6eba02e] third commit

1 file changed, 1 insertion(+)

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git log

commit 6eba02e160a528f162005c28a292672ea7aa3a62 (HEAD -> master)

Author: Bhavan <bhavanreddy1998@gmail.com>

Date: Sat Jun 18 11:23:11 2022 +0530

third commit

commit 7f0ea4cd83de1d2e5113d945d7a35fb933b1e8d3

Author: Bhavan <bhavanreddy1998@gmail.com>

Date: Sat Jun 18 11:16:54 2022 +0530

committed

commit 2c4f4432ab661b384cbc23e055cbd76e7c0f158e

Author: Bhavan <bhavanreddy1998@gmail.com>

Date: Sat Jun 18 10:24:38 2022 +0530

First commit

commit c1b7f7cfce050666b66347c27b754268d116659c

Author: Bhavan <bhavanreddy1998@gmail.com>

Date: Sat Jun 18 10:16:26 2022 +0530

committing index.html file

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git revert ^C

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git revert 2c4f4432ab661b384cbc23e055cbd76e7c0f158e

Auto-merging index.html

CONFLICT (content): Merge conflict in index.html

error: could not revert 2c4f443... First commit

hint: After resolving the conflicts, mark them with

hint: "git add/rm <pathspec>", then run

hint: "git revert --continue".

hint: You can instead skip this commit with "git revert --skip".

hint: To abort and get back to the state before "git revert",

hint: run "git revert --abort".

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master|REVERTING)

$ git revert 2c4f4432ab661b384cbc23e055cbd76e7c0f158e

error: Reverting is not possible because you have unmerged files.

hint: Fix them up in the work tree, and then use 'git add/rm <file>'

hint: as appropriate to mark resolution and make a commit.

fatal: revert failed

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master|REVERTING)

$ git revert -continue

error: did you mean `--continue` (with two dashes)?

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master|REVERTING)

$ git revert --continue

error: Committing is not possible because you have unmerged files.

hint: Fix them up in the work tree, and then use 'git add/rm <file>'

hint: as appropriate to mark resolution and make a commit.

fatal: Exiting because of an unresolved conflict.

U index.html

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master|REVERTING)

$ git revert 2c4f4432ab661b384cbc23e055cbd76e7c0f158e

error: Reverting is not possible because you have unmerged files.

hint: Fix them up in the work tree, and then use 'git add/rm <file>'

hint: as appropriate to mark resolution and make a commit.

fatal: revert failed

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master|REVERTING)

$ git mv rename add my-add

fatal: destination 'my-add' is not a directory

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master|REVERTING)

$ git revert --abort

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git mv rename add my-add

fatal: destination 'my-add' is not a directory

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git config --global alias.my\_add add

error: invalid key: alias.my\_add

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git config --global alias.my\_add add

error: invalid key: alias.my\_add

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git config --global alias my\_add=add

error: key does not contain a section: alias

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git config --global alias.my\_add add

error: invalid key: alias.my\_add

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ [alias] my\_add=add

bash: [alias]: command not found

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git my\_add .

git: 'my\_add' is not a git command. See 'git --help'.

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git --help

usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]

[--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]

[-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]

[--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]

[--super-prefix=<path>] [--config-env=<name>=<envvar>]

<command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)

clone Clone a repository into a new directory

init Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)

add Add file contents to the index

mv Move or rename a file, a directory, or a symlink

restore Restore working tree files

rm Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)

bisect Use binary search to find the commit that introduced a bug

diff Show changes between commits, commit and working tree, etc

grep Print lines matching a pattern

log Show commit logs

show Show various types of objects

status Show the working tree status

grow, mark and tweak your common history

branch List, create, or delete branches

commit Record changes to the repository

merge Join two or more development histories together

rebase Reapply commits on top of another base tip

reset Reset current HEAD to the specified state

switch Switch branches

tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)

fetch Download objects and refs from another repository

pull Fetch from and integrate with another repository or a local branch

push Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some

concept guides. See 'git help <command>' or 'git help <concept>'

to read about a specific subcommand or concept.

See 'git help git' for an overview of the system.

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ [alias]

bash: [alias]: command not found

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ [ alias ] my\_add = add

bash: [: missing `]'

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git config --global alias."my\_add" add

error: invalid key: alias.my\_add

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git config --global alias."my\_add" 'add'

error: invalid key: alias.my\_add

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git config --global alias.my\_add 'add'

error: invalid key: alias.my\_add

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git config --global alias.'my\_add' 'add'

error: invalid key: alias.my\_add

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ ^C

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ [alias]

bash: [alias]: command not found

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ my\_add = add

bash: my\_add: command not found

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ ~/.gitconfig

/c/Users/BHAVRS/.gitconfig: line 1: [user]: command not found

/c/Users/BHAVRS/.gitconfig: line 2: name: command not found

/c/Users/BHAVRS/.gitconfig: line 3: email: command not found

/c/Users/BHAVRS/.gitconfig: line 4: username: command not found

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ $ git config --global alias.st status

bash: $: command not found

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git config --global alias.ci "commit -v"

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git config --global alias.my\_add "commit add"

error: invalid key: alias.my\_add

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git config --global alias."my\_add" "commit add"

error: invalid key: alias.my\_add

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ vim `/gitconfig

> :q

> ^C

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ vim ~/gitconfig

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git st

git: 'st' is not a git command. See 'git --help'.

The most similar commands are

status

reset

stage

stash

svn

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ st

bash: st: command not found

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ ^C

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git config alias.a "add"

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git a .

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git commit -m "last commit got index.html"

On branch master

nothing to commit, working tree clean

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ ^C

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git config alias.my\_add "add"

error: invalid key: alias.my\_add

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git config alias.my-add "add"

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git my-add

Nothing specified, nothing added.

hint: Maybe you wanted to say 'git add .'?

hint: Turn this message off by running

hint: "git config advice.addEmptyPathspec false"

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$ git my-add .

bhavrs@INLEN8520050892 MINGW64 ~/project\_dir (master)

$

bhavrs@INLEN8520050892 MINGW64 ~

$ mkdir Assignments

bhavrs@INLEN8520050892 MINGW64 ~

$ cd Assignments

bhavrs@INLEN8520050892 MINGW64 ~/Assignments

$ git init

Initialized empty Git repository in C:/Users/BHAVRS/Assignments/.git/

bhavrs@INLEN8520050892 MINGW64 ~/Assignments (master)

$ touch README.txt

bhavrs@INLEN8520050892 MINGW64 ~/Assignments (master)

$ git add .

bhavrs@INLEN8520050892 MINGW64 ~/Assignments (master)

$ git commit -m "first commit"

[master (root-commit) 2060d7e] first commit

1 file changed, 10 insertions(+)

create mode 100644 README.txt

bhavrs@INLEN8520050892 MINGW64 ~/Assignments (master)

$ git branch html-assignments

bhavrs@INLEN8520050892 MINGW64 ~/Assignments (master)

$ git checkout html-assignments

Switched to branch 'html-assignments'

bhavrs@INLEN8520050892 MINGW64 ~/Assignments (html-assignments)

$ git branch css-assignments

bhavrs@INLEN8520050892 MINGW64 ~/Assignments (html-assignments)

$ git branch

css-assignments

\* html-assignments

master

bhavrs@INLEN8520050892 MINGW64 ~/Assignments (html-assignments)

$ checkout css-assignmenty

bash: checkout: command not found

bhavrs@INLEN8520050892 MINGW64 ~/Assignments (html-assignments)

$ checkoutbcss-assignment

bash: checkoutbcss-assignment: command not found

bhavrs@INLEN8520050892 MINGW64 ~/Assignments (html-assignments)

$ checkout css-assignments

bash: checkout: command not found

bhavrs@INLEN8520050892 MINGW64 ~/Assignments (html-assignments)

$ git checkout css-assignments

Switched to branch 'css-assignments'

bhavrs@INLEN8520050892 MINGW64 ~/Assignments (css-assignments)

$ git branch js-assignments

bhavrs@INLEN8520050892 MINGW64 ~/Assignments (css-assignments)

$ checkout js-asignments

bash: checkout: command not found

bhavrs@INLEN8520050892 MINGW64 ~/Assignments (css-assignments)

$ git checkout js-assignments

Switched to branch 'js-assignments'

bhavrs@INLEN8520050892 MINGW64 ~/Assignments (js-assignments)

$ git branch

css-assignments

html-assignments

\* js-assignments

master

bhavrs@INLEN8520050892 MINGW64 ~/Assignments (js-assignments)

$







