Last login: Sun Apr 1 10:52:36 on ttys002

10:53 ~ cd flask\_tutorial

-bash: cd: flask\_tutorial: No such file or directory

10:53 ~ cd cjn-python

10:53 cjn-python cd flask\_tutorial

10:54 flask\_tutorial [master !] git status

On branch master

Your branch is up-to-date with 'origin/master'.

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

modified: app.db

10:54 flask\_tutorial [master !] git branch checkout -b "downgrade"

error: unknown switch `b'

usage: git branch [<options>] [-r | -a] [--merged | --no-merged]

or: git branch [<options>] [-l] [-f] <branch-name> [<start-point>]

or: git branch [<options>] [-r] (-d | -D) <branch-name>...

or: git branch [<options>] (-m | -M) [<old-branch>] <new-branch>

Generic options

-v, --verbose show hash and subject, give twice for upstream branch

-q, --quiet suppress informational messages

-t, --track set up tracking mode (see git-pull(1))

--set-upstream change upstream info

-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

--abbrev[=<n>] use <n> digits to display SHA-1s

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

--list list branch names

-l, --create-reflog create the branch's reflog

--edit-description edit the description for the branch

-f, --force force creation, move/rename, deletion

--no-merged <commit> print only not merged branches

--merged <commit> print only merged branches

--column[=<style>] list branches in columns

10:54 flask\_tutorial [master !] git branch checkout -b downgrade

error: unknown switch `b'

usage: git branch [<options>] [-r | -a] [--merged | --no-merged]

or: git branch [<options>] [-l] [-f] <branch-name> [<start-point>]

or: git branch [<options>] [-r] (-d | -D) <branch-name>...

or: git branch [<options>] (-m | -M) [<old-branch>] <new-branch>

Generic options

-v, --verbose show hash and subject, give twice for upstream branch

-q, --quiet suppress informational messages

-t, --track set up tracking mode (see git-pull(1))

--set-upstream change upstream info

-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

--abbrev[=<n>] use <n> digits to display SHA-1s

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

--list list branch names

-l, --create-reflog create the branch's reflog

--edit-description edit the description for the branch

-f, --force force creation, move/rename, deletion

--no-merged <commit> print only not merged branches

--merged <commit> print only merged branches

--column[=<style>] list branches in columns

10:55 flask\_tutorial [master !] git checkout -b

error: switch `b' requires a value

usage: git checkout [<options>] <branch>

or: git checkout [<options>] [<branch>] -- <file>...

-q, --quiet suppress progress reporting

-b <branch> create and checkout a new branch

-B <branch> create/reset and checkout a branch

-l create reflog for new branch

--detach detach the HEAD at named commit

-t, --track set upstream info for new branch

--orphan <new-branch>

new unparented branch

-2, --ours checkout our version for unmerged files

-3, --theirs checkout their version for unmerged files

-f, --force force checkout (throw away local modifications)

-m, --merge perform a 3-way merge with the new branch

--overwrite-ignore update ignored files (default)

--conflict <style> conflict style (merge or diff3)

-p, --patch select hunks interactively

--ignore-skip-worktree-bits

do not limit pathspecs to sparse entries only

--ignore-other-worktrees

do not check if another worktree is holding the given ref

10:55 flask\_tutorial [master !] git checkout -b downgrade

M app.db

Switched to a new branch 'downgrade'

10:55 flask\_tutorial [downgrade !] git add .

10:55 flask\_tutorial [downgrade !] git commit -m "downgrade"

[downgrade 8c3ddc6] downgrade

Committer: Clark Ngo <clarkngo@Clarks-MacBook-Pro.local>

Your name and email address were configured automatically based

on your username and hostname. Please check that they are accurate.

You can suppress this message by setting them explicitly. Run the

following command and follow the instructions in your editor to edit

your configuration file:

git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

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

10:56 flask\_tutorial [downgrade] git push

warning: push.default is unset; its implicit value has changed in

Git 2.0 from 'matching' to 'simple'. To squelch this message

and maintain the traditional behavior, use:

git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

git config --global push.default simple

When push.default is set to 'matching', git will push local branches

to the remote branches that already exist with the same name.

Since Git 2.0, Git defaults to the more conservative 'simple'

behavior, which only pushes the current branch to the corresponding

remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.

(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode

'current' instead of 'simple' if you sometimes use older versions of Git)

fatal: The current branch downgrade has no upstream branch.

To push the current branch and set the remote as upstream, use

git push --set-upstream origin downgrade

10:56 flask\_tutorial [downgrade] git push origin master

Everything up-to-date

10:57 flask\_tutorial [downgrade] git push origin downgrade

Counting objects: 3, done.

Delta compression using up to 8 threads.

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 365 bytes | 0 bytes/s, done.

Total 3 (delta 2), reused 0 (delta 0)

remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

To https://github.com/clarkngo/flask\_tutorial.git

\* [new branch] downgrade -> downgrade

10:57 flask\_tutorial [downgrade] git checkout master

Switched to branch 'master'

Your branch is up-to-date with 'origin/master'.

11:09 flask\_tutorial [master] git checkout downgrade

Switched to branch 'downgrade'

11:09 flask\_tutorial [downgrade] git checkout master

Switched to branch 'master'

Your branch is up-to-date with 'origin/master'.

11:09 flask\_tutorial [master] git checkout downgrade

Switched to branch 'downgrade'

11:10 flask\_tutorial [downgrade] touch test.py

11:10 flask\_tutorial [downgrade ?] git checkout master

Switched to branch 'master'

Your branch is up-to-date with 'origin/master'.

11:11 flask\_tutorial [master ?] subl .

11:11 flask\_tutorial [master ?] git pull

Already up-to-date.

11:11 flask\_tutorial [master ?] git checkout 9a0439a485

Note: checking out '9a0439a485'.

You are in 'detached HEAD' state. You can look around, make experimental

changes and commit them, and you can discard any commits you make in this

state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may

do so (now or later) by using -b with the checkout command again. Example:

git checkout -b <new-branch-name>

HEAD is now at 9a0439a... post\_model

11:15 flask\_tutorial [(HEAD detached at 9a0439a) ?] git checkout john

error: pathspec 'john' did not match any file(s) known to git.

11:23 flask\_tutorial [(HEAD detached at 9a0439a) ?] git fetch

remote: Counting objects: 3, done.

remote: Compressing objects: 100% (1/1), done.

Unpacking objects: 100% (3/3), done.

remote: Total 3 (delta 2), reused 3 (delta 2), pack-reused 0

From https://github.com/clarkngo/flask\_tutorial

\* [new branch] john -> origin/john

11:23 flask\_tutorial [(HEAD detached at 9a0439a) ?] git checkout john

Previous HEAD position was 9a0439a... post\_model

Branch john set up to track remote branch john from origin.

Switched to a new branch 'john'

11:23 flask\_tutorial [john ?] git status

On branch john

Your branch is up-to-date with 'origin/john'.

Untracked files:

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

test.py

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

11:23 flask\_tutorial [john ?] git stash

No local changes to save

11:24 flask\_tutorial [john ?] git status

On branch john

Your branch is up-to-date with 'origin/john'.

Untracked files:

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

test.py

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

11:24 flask\_tutorial [john ?] git add .

11:24 flask\_tutorial [john +] git status

On branch john

Your branch is up-to-date with 'origin/john'.

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: test.py

11:24 flask\_tutorial [john +] git reset HEAD test.py

11:24 flask\_tutorial [john ?] git status

On branch john

Your branch is up-to-date with 'origin/john'.

Untracked files:

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

test.py

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

11:25 flask\_tutorial [john ?] git add .

11:25 flask\_tutorial [john +] git stash

Saved working directory and index state WIP on john: fcf980a deleted routes

HEAD is now at fcf980a deleted routes

11:25 flask\_tutorial [john] git status

On branch john

Your branch is up-to-date with 'origin/john'.

nothing to commit, working directory clean

11:25 flask\_tutorial [john] git checkout downgrade

Switched to branch 'downgrade'

11:26 flask\_tutorial [downgrade] git checkout master

Switched to branch 'master'

Your branch is up-to-date with 'origin/master'.

11:26 flask\_tutorial [master] git checkout john

Switched to branch 'john'

Your branch is up-to-date with 'origin/john'.

11:26 flask\_tutorial [john] git reset HEAD

11:27 flask\_tutorial [john] git stash apply

On branch john

Your branch is up-to-date with 'origin/john'.

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: test.py

11:27 flask\_tutorial [john +] git stash

Saved working directory and index state WIP on john: fcf980a deleted routes

HEAD is now at fcf980a deleted routes

11:28 flask\_tutorial [john] git stash list

stash@{0}: WIP on john: fcf980a deleted routes

stash@{1}: WIP on john: fcf980a deleted routes

11:28 flask\_tutorial [john] git stash show

test.py | 1 +

1 file changed, 1 insertion(+)

11:28 flask\_tutorial [john] git stash apply

On branch john

Your branch is up-to-date with 'origin/john'.

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: test.py

11:29 flask\_tutorial [john +]