



brijpandeyji



# The Ultimate Git Command Handbook

Your Quick Reference  
Guide to Mastering  
Git



swipe

brijpandeyji



# Git Basics

Use `git help [command]` if you're stuck.

`master`

**default level branch**

`origin`

**default upstream branch**

`HEAD`

**current branch**

`HEAD^`

**parent of HEAD**

`HEAD~4`

**great-great grandparent of HEAD**

`foo..bar`

**from branch foo to branch bar**

Let us now look at the command flow of Git

swipe

brijpandeyji



# Git Command Flow

## create

init  
clone

## browse

status  
log  
blame  
show  
diff

## change

mark changes  
to be respected  
by commit;  
add

## branch

checkout  
branch

## update

pull  
fetch  
merge  
am

## revert

reset  
checkout  
revert

## commit

commit

## push

push  
format-patch

swipe

brijpandeyji



## Create

### From Existing Files

`git init`

`git add`

### From Existing Repository

`git clone ~/old ~/new`

`git clone git:// ...`

`git clone ssh:// ...`

## View

`git status`

`git diff [oldid newid]`

`git log [-p] [file|dir]`

`git blame file`

`git show id` (meta data + diff)

`git show id:file`

`git branch` (shows list, \* = current)

`git tag -l` (shows list)

swipe

brijpandeyji



## Revert

In Git, revert usually describes a new commit that undoes previous commits.

`git reset --hard` (NO UNDO) (reset to last commit)

`git revert branch`

`git commit -a --amend` (replaces prev, commit)

`git checkout id file`

## Publish

In Git, commit only respects changes that have been marked explicitly with add.

`git commit [-a]` (-a: add changed files automatically)

`git format-patch origin` (create set of diffs)

`git push remote` (push to origin or remote)

`git tag foo` (mark current version)

swipe

brijpandeyji



## Update

**git fetch** (from def. upstream)

**git fetch remote**

**git pull** (= fetch & merge)

**git am -3 patch.mbox**

**git apply patch.diff**

## Branch

**git checkout branch** (switch working dir to branch)

**git merge branch** (merge into current)

**git branch branch** (branch current)

**git checkout -b new other** (branch new from other and switch to it)

swipe

brijpandeyji



## Useful Tools

**git archive** (Create release tarball)

**git bisect** (Binary search for defects)

**git cherry-pick** (Take single commit from elsewhere)

**git fsck** (Check tree)

**git gc** (Compress metadata (performance))

**git rebase** (Forward-port local changes to remote branch)

**git remote add URL** (Register a new remote repository for  
this tree)

**git stash** (Temporarily set aside changes)

**git tag** (there is more to it)

**gitk** (Tk GUI for Git)

swipe

brijpandeyji



## Tracking Files

`git add files`

`git mv old new`

`git rm files`

`git rm - -cached files` (stop tracking but keep files in working dir)

## Conflicts

Use add to mark files as resolved

`git diff [--base]`

`git diff --ours`

`git diff --theirs`

`git log --merge`

`gitk --merg`

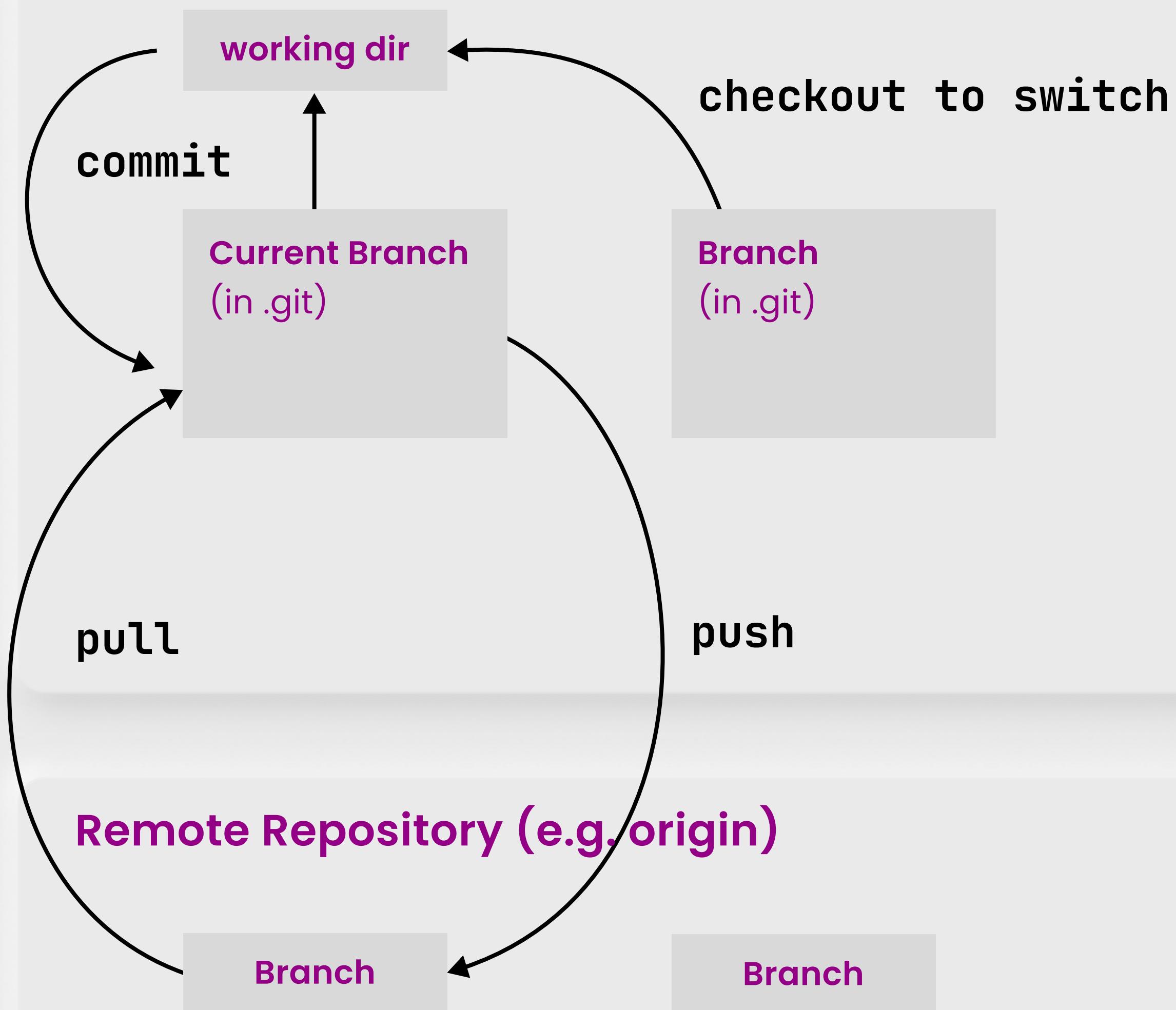
swipe

brijpandeyji



# Structure Overview

## Local Repository



swipe



brijpandeyji



**For More Interesting  
Content**



**Brij Kishore Pandey**



**Follow Me On  
LinkedIn**

<https://www.linkedin.com/in/brijpandeyji/>