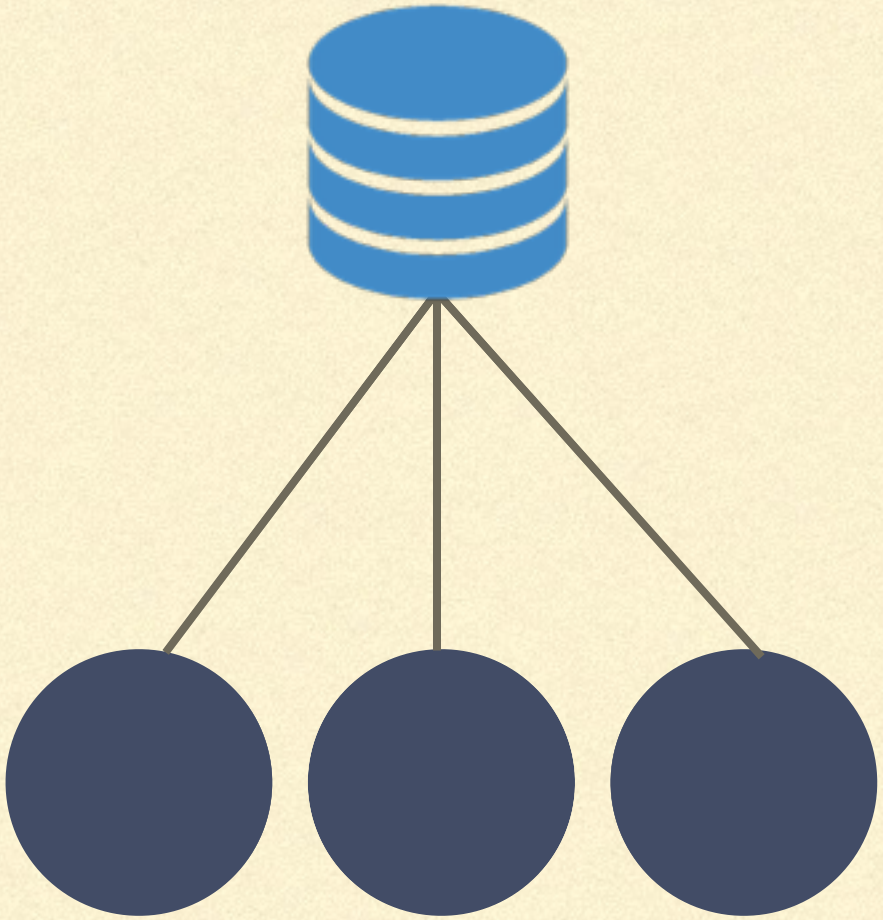
git

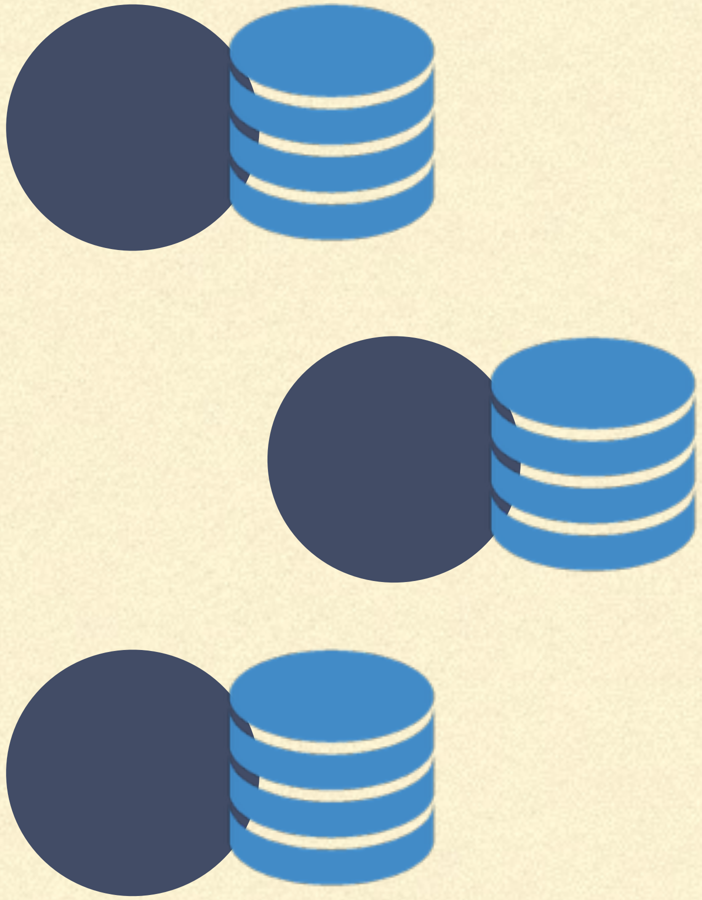
Godwin Stanislaus

DISTRIBUTED VERSION CONTROL SYSTEM

- Central Repository



- Distributed Repository



git

ORIGINS

Linus Torvalds

- Linux Kernel Project
- Meant to be distributed, fast
- Capable of handling large projects

WORKING WITH git

- Git - http://git-scm.com/
- Command line interface
- Many GUI Tools available - @ https://git-scm.com/download/gui/mac

GitHub Desktop

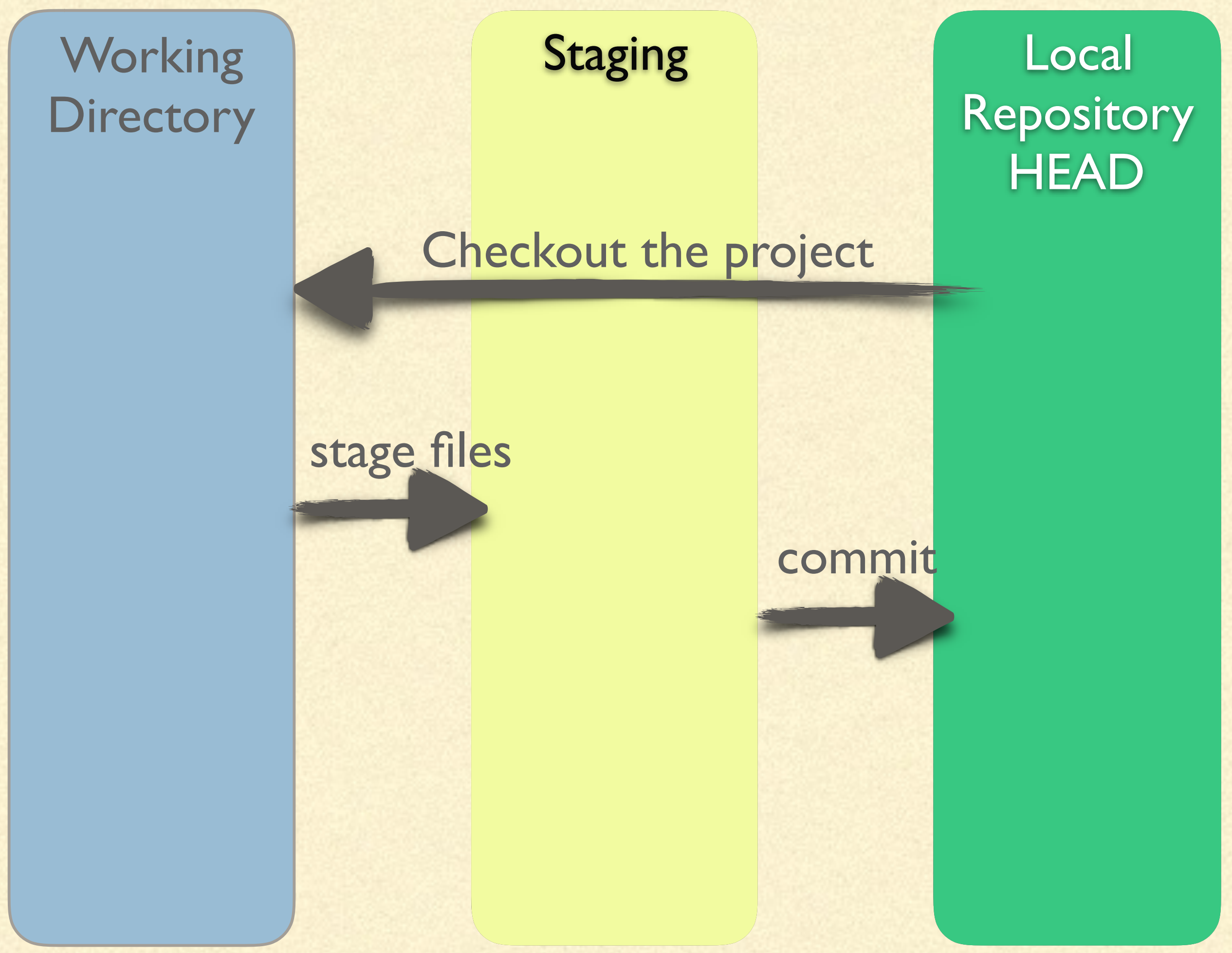
SourceTree

GitKraken

GitUp

SmartGit

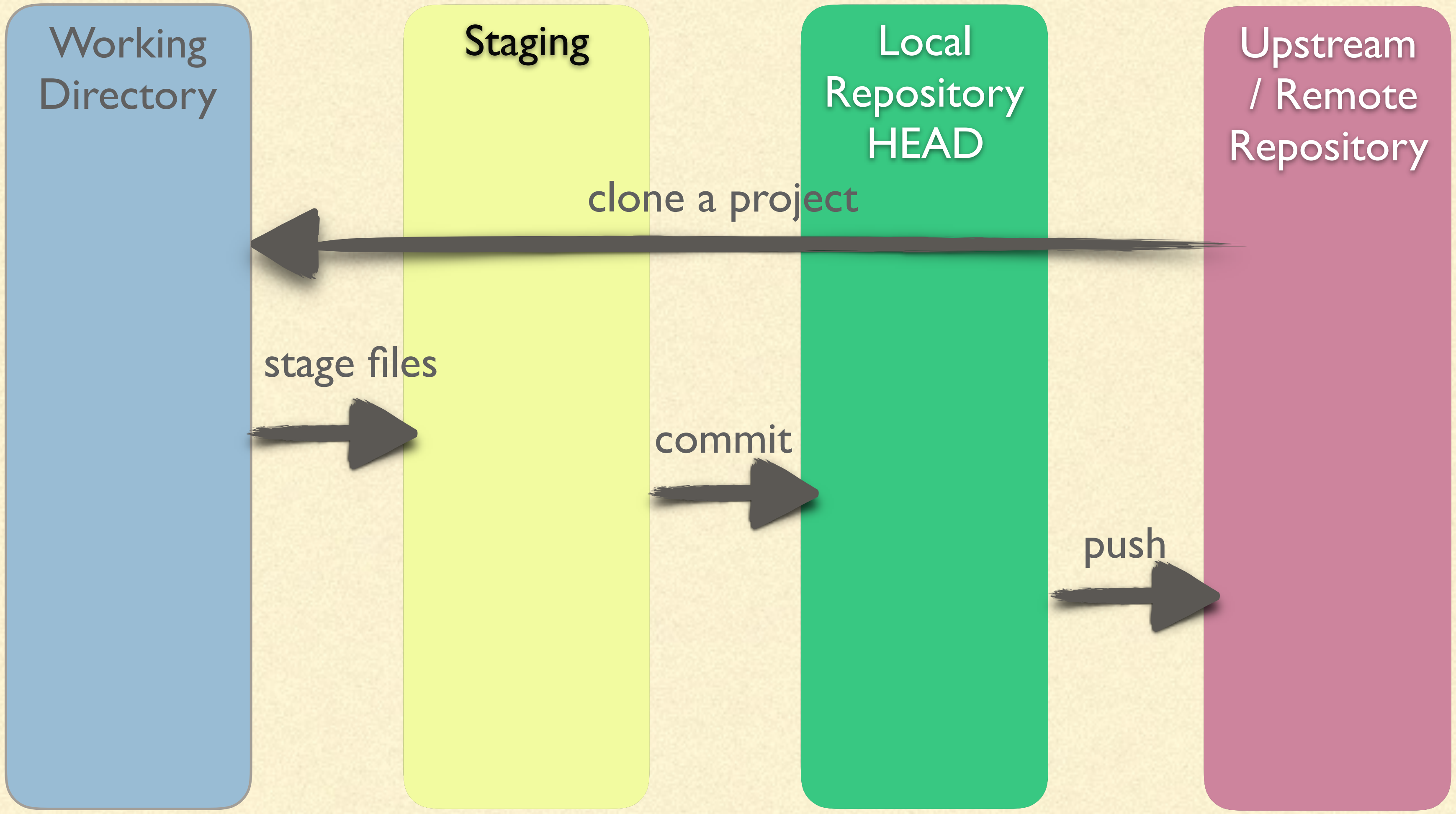
GETTING STARTED



Trees of Git

- The HEAD - last commit snapshot
- Index - Staging area
- Workding Directory - Sandbox

GETTING STARTED



1. git clone <Repository>

2. git add <list of files>

3. git commit -m “comment” files

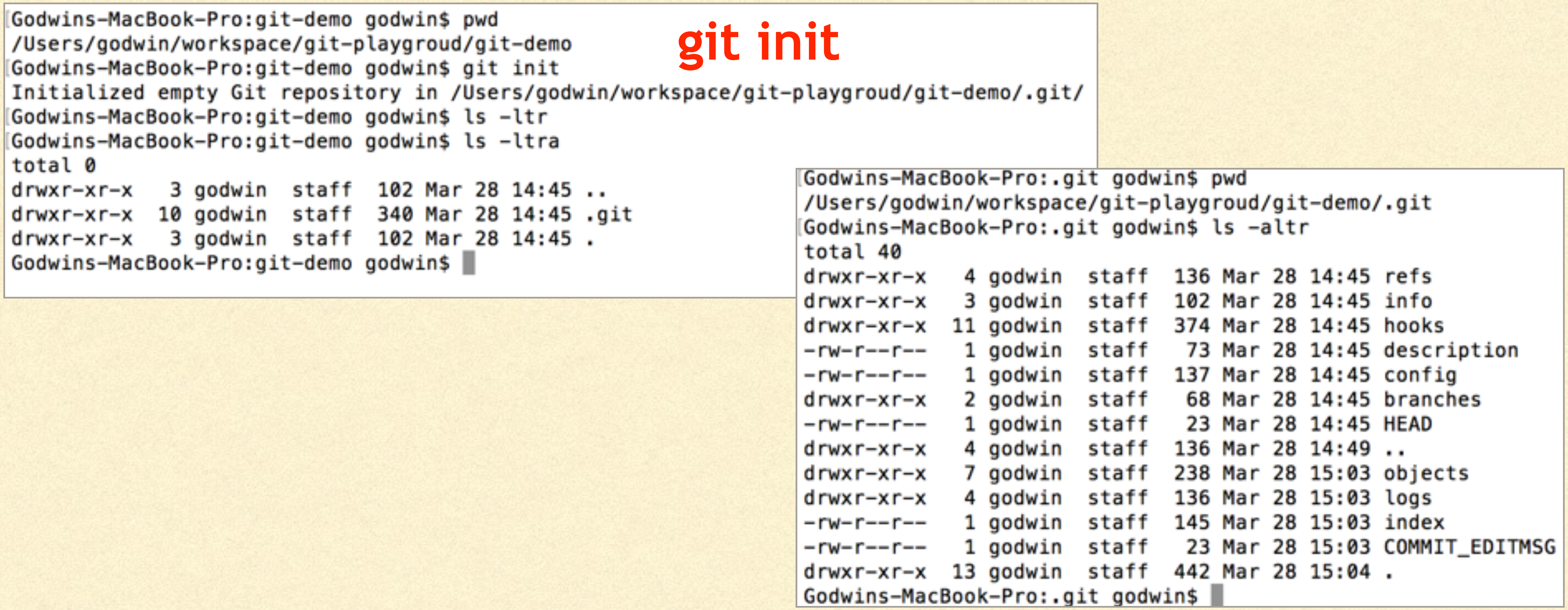
4. git push

GETTING STARTED

- A basic workflow
  - – Initialize a repository - (Possible init or clone)
  - – Edit / add files
  - – Stage the changes
  - – Review your changes
  - – Commit the changes

GETTING STARTED

- A basic workflow
  - Initialize a repository -(Possible init or clone)



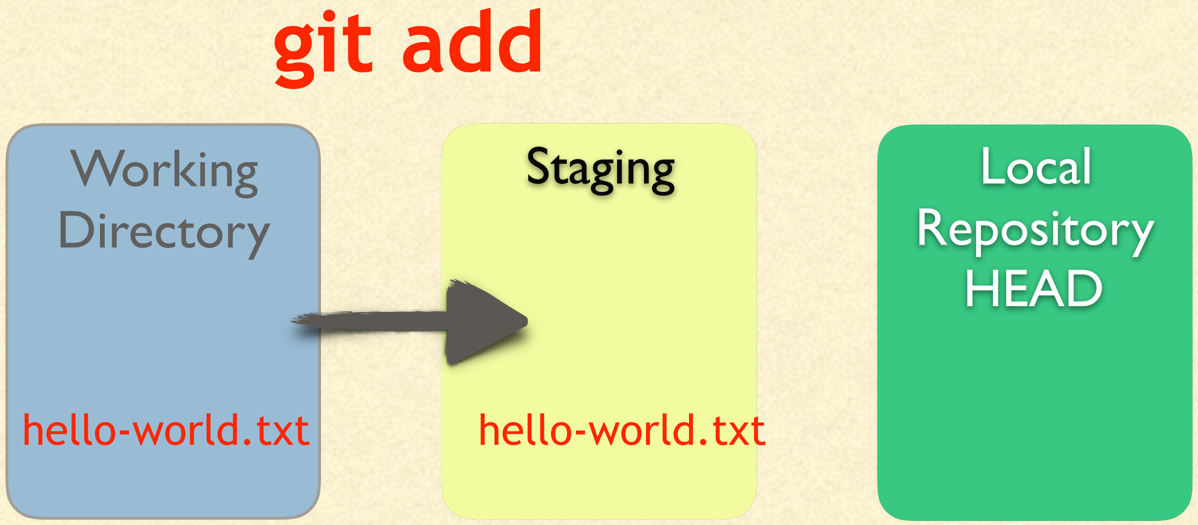
git init

```
[Godwins-MacBook-Pro:git-demo godwin$ pwd
/Users/godwin/workspace/git-playgroud/git-demo
[Godwins-MacBook-Pro:git-demo godwin$ git init
Initialized empty Git repository in /Users/godwin/workspace/git-playgroud/git-demo/.git/
[Godwins-MacBook-Pro:git-demo godwin$ ls -ltr
[Godwins-MacBook-Pro:git-demo godwin$ ls -ltra
total 0
drwxr-xr-x   3 godwin  staff  102 Mar 28 14:45 ..
drwxr-xr-x  10 godwin  staff  340 Mar 28 14:45 .git
drwxr-xr-x   3 godwin  staff  102 Mar 28 14:45 .
Godwins-MacBook-Pro:git-demo godwin$ 
```

```
[Godwins-MacBook-Pro:.git godwin$ pwd
/Users/godwin/workspace/git-playgroud/git-demo/.git
[Godwins-MacBook-Pro:.git godwin$ ls -altr
total 40
drwxr-xr-x   4 godwin  staff  136 Mar 28 14:45 refs
drwxr-xr-x   3 godwin  staff  102 Mar 28 14:45 info
drwxr-xr-x  11 godwin  staff  374 Mar 28 14:45 hooks
-rw-r--r--   1 godwin  staff   73 Mar 28 14:45 description
-rw-r--r--   1 godwin  staff  137 Mar 28 14:45 config
drwxr-xr-x   2 godwin  staff   68 Mar 28 14:45 branches
-rw-r--r--   1 godwin  staff   23 Mar 28 14:45 HEAD
drwxr-xr-x   4 godwin  staff  136 Mar 28 14:49 ..
drwxr-xr-x   7 godwin  staff  238 Mar 28 15:03 objects
drwxr-xr-x   4 godwin  staff  136 Mar 28 15:03 logs
-rw-r--r--   1 godwin  staff  145 Mar 28 15:03 index
-rw-r--r--   1 godwin  staff   23 Mar 28 15:03 COMMIT_EDITMSG
drwxr-xr-x  13 godwin  staff  442 Mar 28 15:04 .
Godwins-MacBook-Pro:.git godwin$ 
```

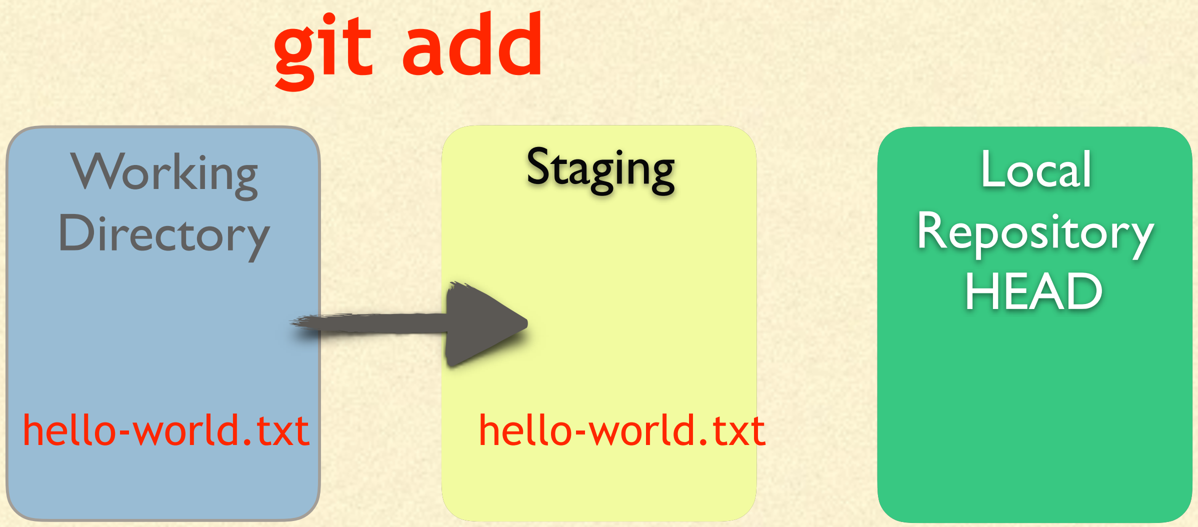
GETTING STARTED

- A basic workflow
  - Initialize a repository - (Possible init or clone)Edit files
  - Add / Edit a file
  - Stage the changes

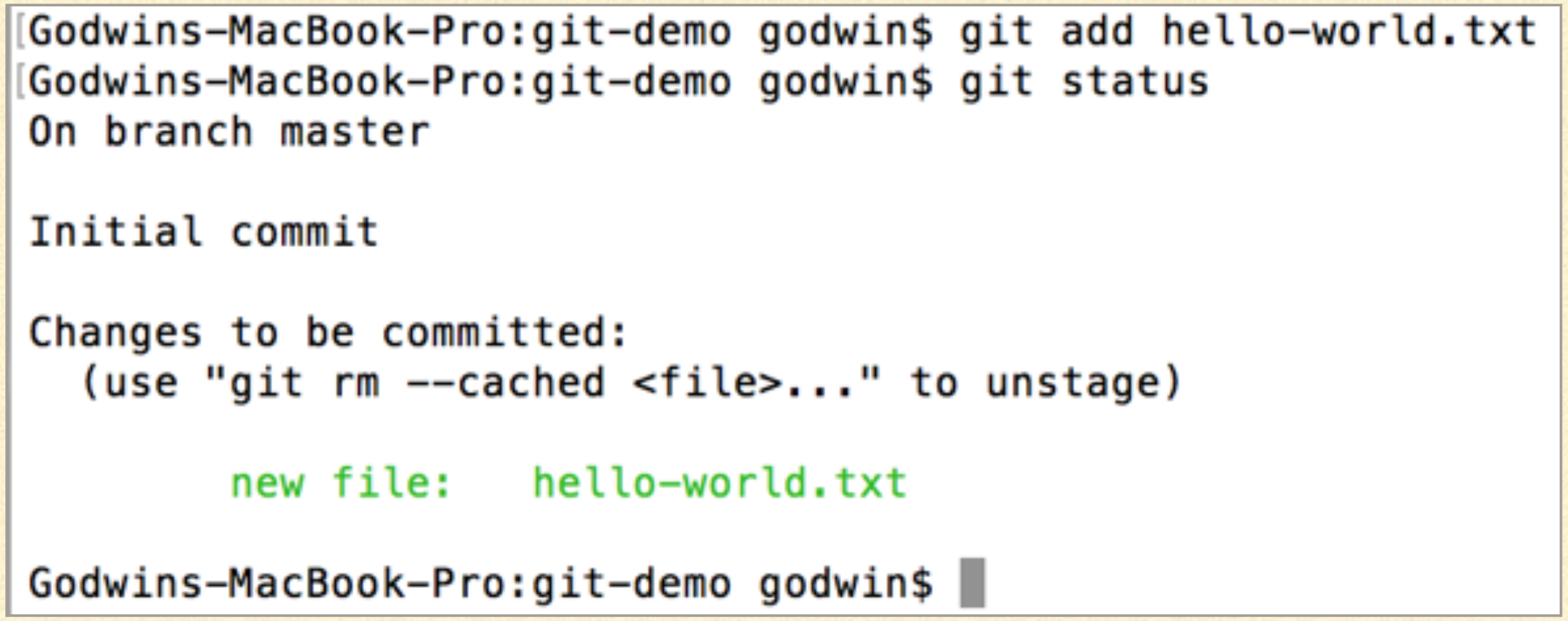


GETTING STARTED

- A basic workflow
  - Initialize a repository - (Possible init or clone)Edit files
  - Add / Edit a file
  - Stage the changes
  - Review your Changes



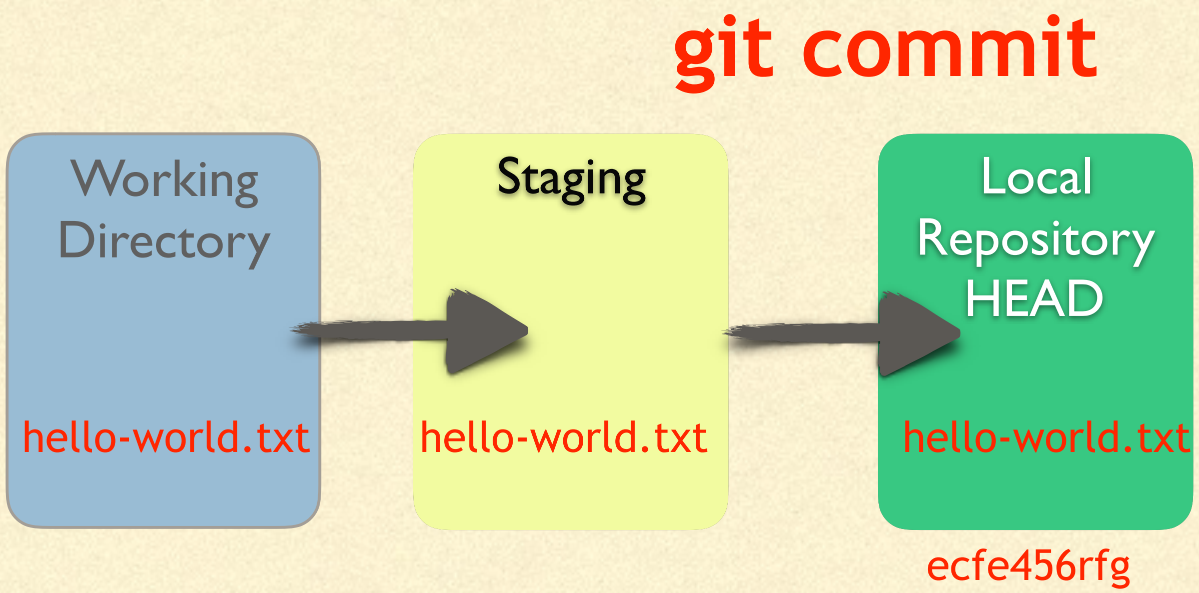
git status



```
[Godwins-MacBook-Pro:git-demo godwin$ git add hello-world.txt
[Godwins-MacBook-Pro:git-demo godwin$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   hello-world.txt

Godwins-MacBook-Pro:git-demo godwin$ 
```

GETTING STARTED

- A basic workflow
  - – Initialize a repository - (Possible init or clone)Edit files
  - – Add / Edit a file
  - – Stage the changes
  - – Review your Changes
  - – Commit the Changes



git commit

Working Directory

Staging

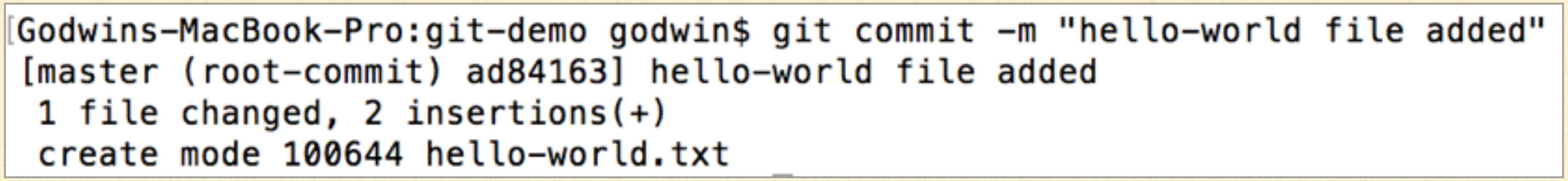
Local Repository HEAD

hello-world.txt

hello-world.txt

hello-world.txt

ecfe456rfg



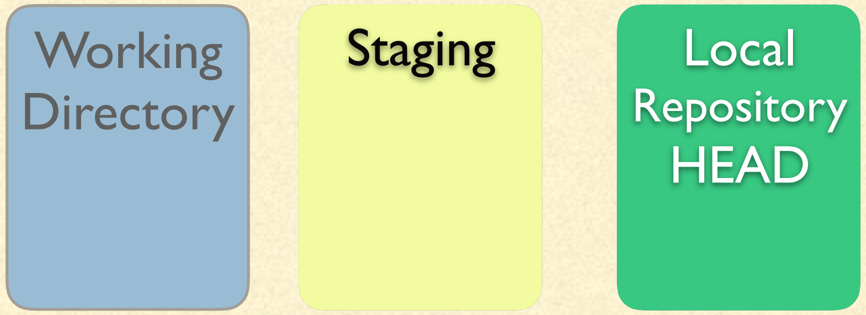
```
Godwins-MacBook-Pro:git-demo godwin$ git commit -m "hello-world file added"
[master (root-commit) ad84163] hello-world file added
 1 file changed, 2 insertions(+)
 create mode 100644 hello-world.txt
```

DIFFERENT WAYS TO ADD

- git add <list of files>   # add the list of files
- git add - -all   # add all files
- git add *.txt   # add all txt files in current directory
- git add src/*.java   # add all java files in src directory
- git add src/   # add add files in src directory
- git add “*.java”   # add all java files in the whole project

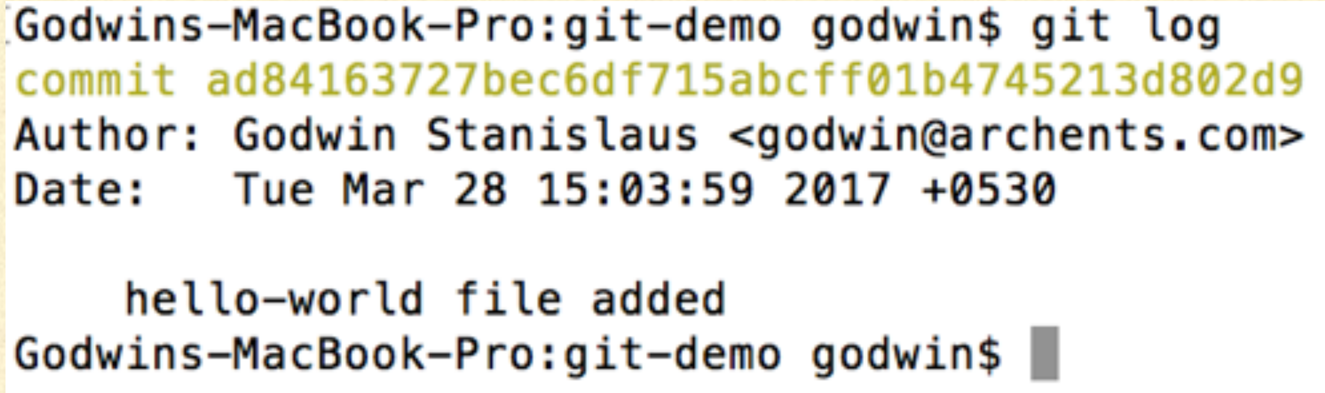
GIT DIFF

- git diff # show difference between working dir and staged
- git diff - - cached # Changes between staged and your last commit;
- git diff HEAD # Changes in the working directory since your last commit;
- git diff test # compare with the tip of "test" branch
- for more options refer "git help diff"



GIT LOG

- git log # shows timeline history
- for more options refer "git help log"



```
Godwins-MacBook-Pro:git-demo godwin$ git log
commit ad84163727bec6df715abcff01b4745213d802d9
Author: Godwin Stanislaus <godwin@archents.com>
Date:   Tue Mar 28 15:03:59 2017 +0530

    hello-world file added
Godwins-MacBook-Pro:git-demo godwin$ 
```

- git log —author=godwin will display only the commits of a certain author
- git log —pretty=oneline to see a compressed log - each commit in one line
- git log —name-status see which files have changed
- git log —help for more options

UNSTAGE & DISCARD

- git add <fileName>
- git status
- git reset HEAD <fileName>
- git status

- git add
- git status
- git checkout - - <fileName>
- git status

COMMIT CHANGES

- add and modify some files.
- git add — all   #adds all new or modified files to staging area
- git status
- git commit -m “provide your comments”
- git status

SKIP STAGING & COMMIT

- add and modify some files.
- git commit -a -m ‘provide your comments’
  - -a — add changes from all tracked files. Doesn’t add new (untracked) files.

UNDOING A COMMIT

- git reset - -soft HEAD^
  - - -soft : reset into staging
  - HEAD^: move to commit before HEAD
- now you can make changes and re-commit

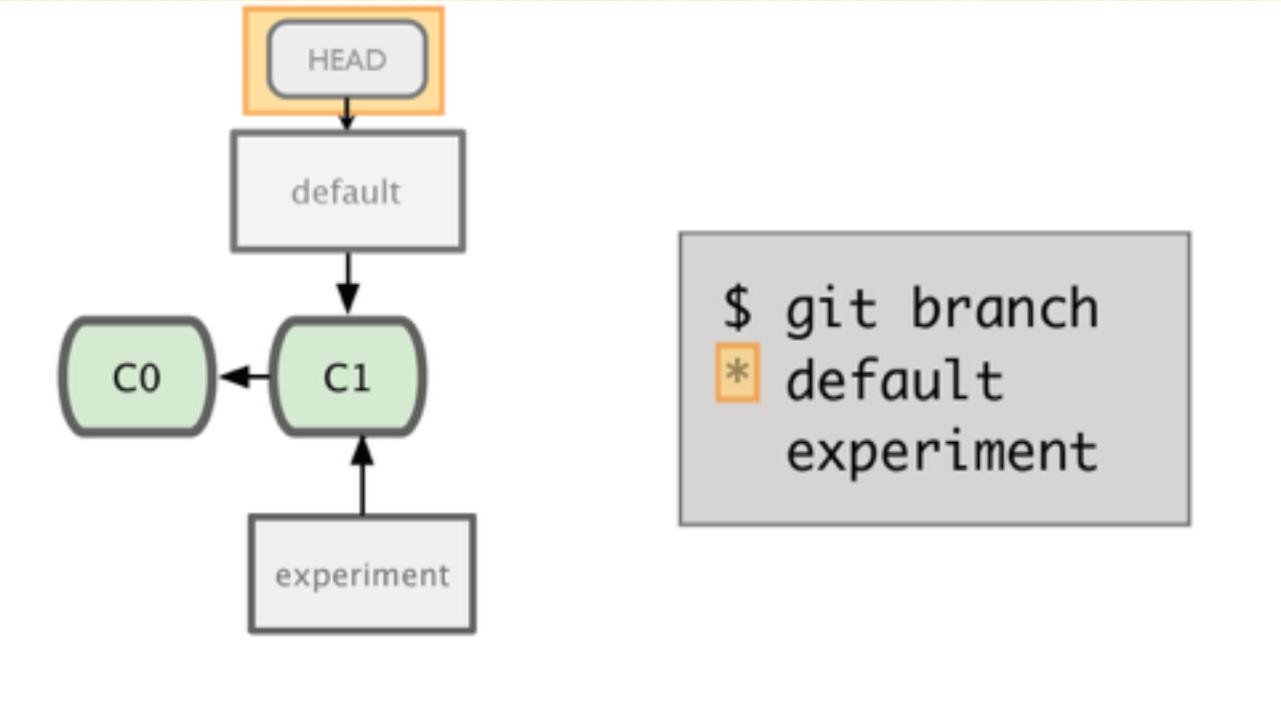
USEFUL COMMANDS

- git reset --soft HEAD^
    - undo last commit, put changes into staging
- git commit - -amend -m “new commit message”
    - change the last commit.
- git reset - -hard HEAD^
    - Undo last commit and all changes
- git reset - -hard HEAD^^
    - Undo last 2 commits and all changes

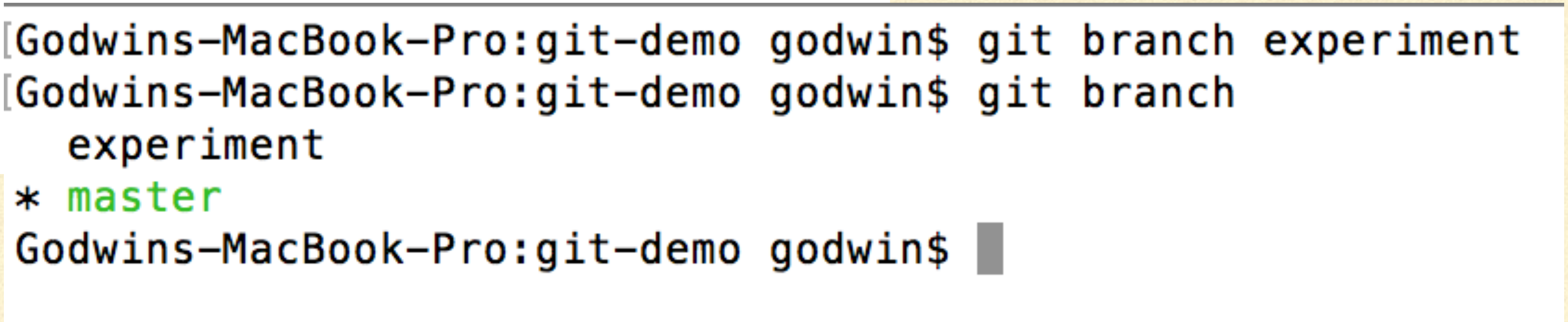
PUSHING CHANGES

- git commit — the file is committed and changes are now in the HEAD of our local working copy. To send those changes to your remote repository, execute
  - git push origin **master**
- If you have not cloned an existing repository and want to connect your repository to a remote server, you need to add it with change the last commit.
  - git remote add origin <server>
  - now you are able to push your changes to the remote server.

BRANCHING

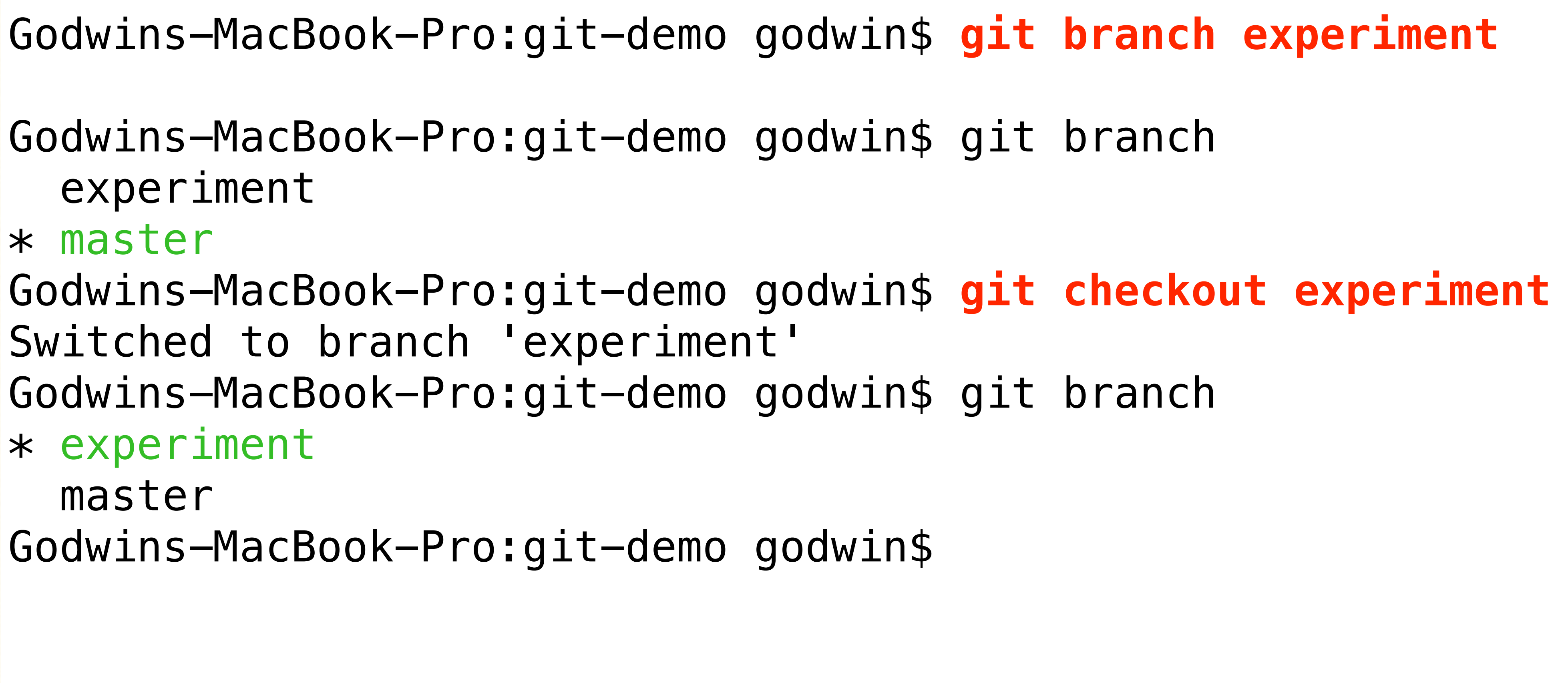


```
$ git branch
* default
  experiment
```



```
[Godwins-MacBook-Pro:git-demo godwin$ git branch experiment
[Godwins-MacBook-Pro:git-demo godwin$ git branch
  experiment
* master
Godwins-MacBook-Pro:git-demo godwin$ 
```

BRANCHING



```
Godwins-MacBook-Pro:git-demo godwin$ git branch experiment

Godwins-MacBook-Pro:git-demo godwin$ git branch
  experiment
* master
Godwins-MacBook-Pro:git-demo godwin$ git checkout experiment
Switched to branch 'experiment'
Godwins-MacBook-Pro:git-demo godwin$ git branch
* experiment
  master
Godwins-MacBook-Pro:git-demo godwin$
```

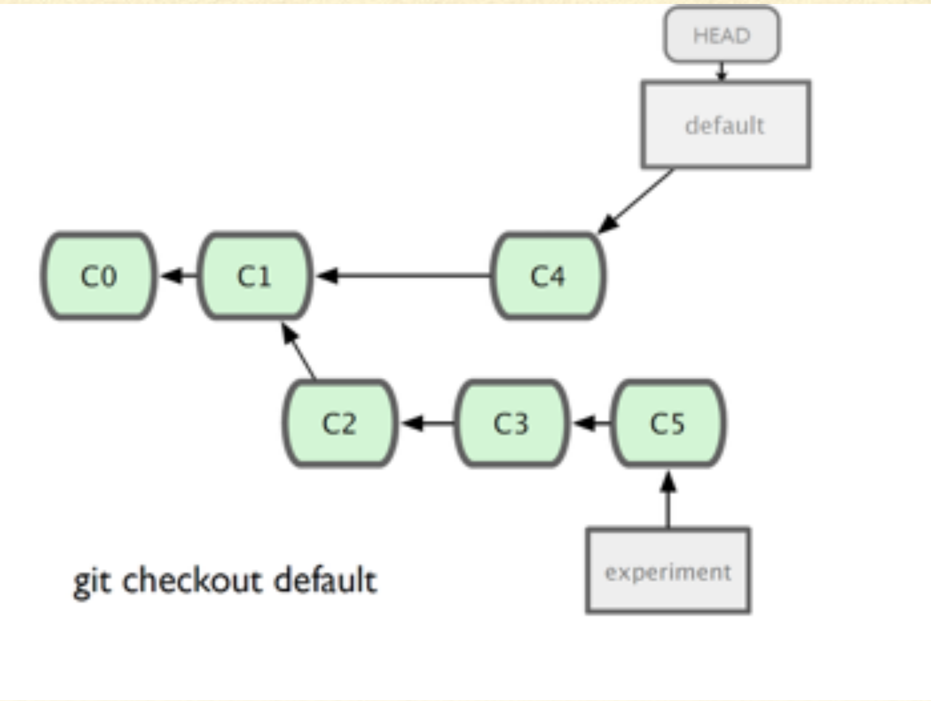
create a new branch named “feature-abc" and switch to it using
git checkout -b feature-abc

MERGING

Steps to merge two branches

– Checkout the branch you want to merge onto
– Merge the branch you want to merge

- git checkout master
- git merge experiment
- git checkout experiment



git checkout default

BRANCHING & MERGING

Why is this cool? - Non linear development

1. clone the code in production and create a brach (Feature-105) to work for the next enhancement, and you work for 30 mins.

2. someone asks you to work on a production issue, Defect-ABC.

3. You checkout ‘production’ branch and create a new branch Defect-ABC and fix it.

4. checkout ‘production’ and merge ‘Defect-ABC’ , push production

4. Checkout Feature-105 branch to continue working on the enhancement

BRANCH

Delete a branch

git branch -d feature-125

A branch is not available to others unless you push the branch to the remote server
git push origin <branch>