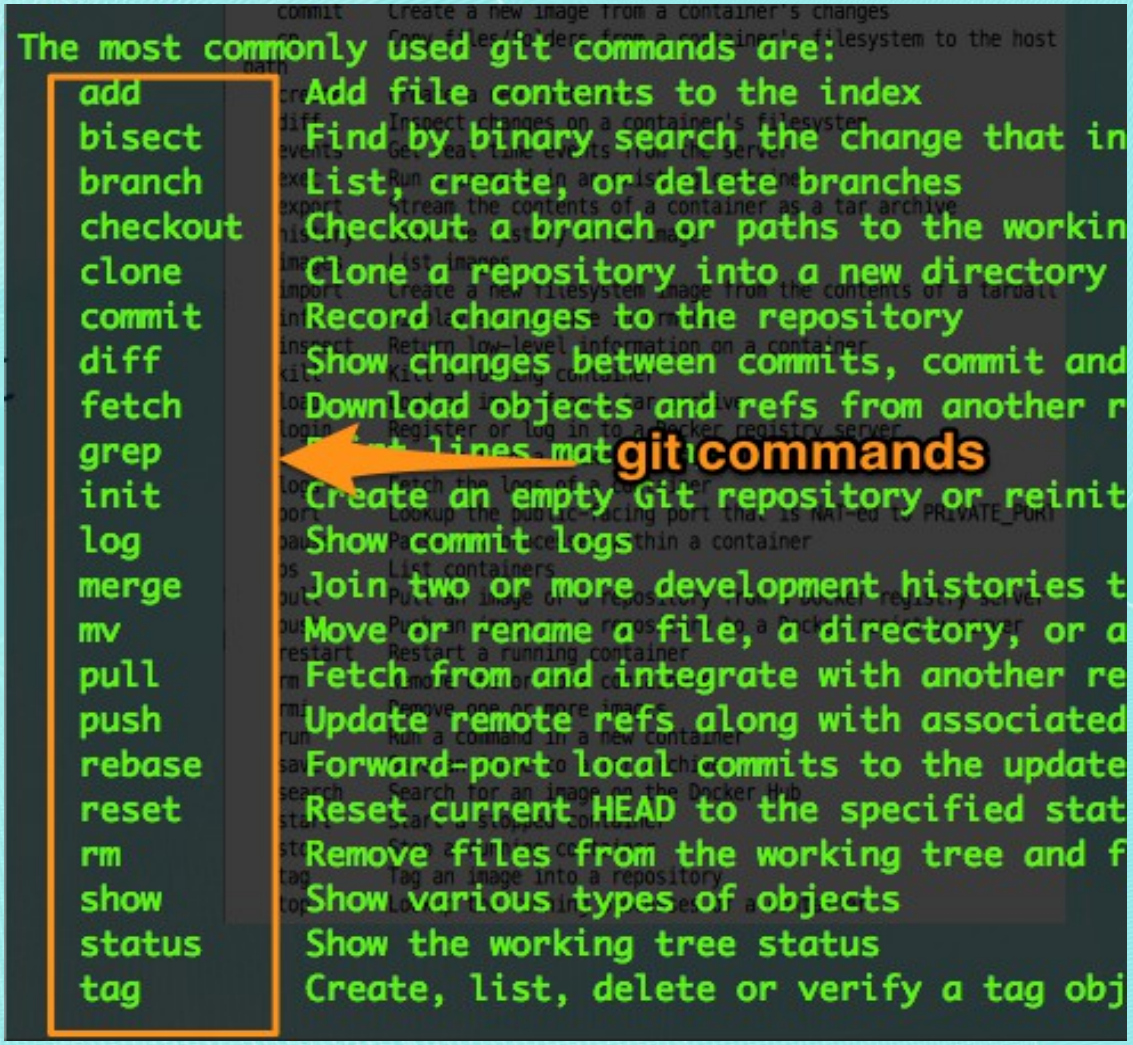
Getting Started with GIT

Objectives

- Configure GIT
- Find help, basic operations
- Create a Repository
- Clone existing Repository

$ git


The most commonly used git commands are:
add Add file contents to the index
bisect Find by binary search the change that in
branch List, create, or delete branches
checkout Checkout a branch or paths to the workin
clone Clone a repository into a new directory
commit Record changes to the repository
diff Show changes between commits, commit and
fetch Download objects and refs from another r
grep
init Create an empty Git repository or reinit
log Show commit logs
merge Join two or more development histories t
mv Move or rename a file, a directory, or a
pull Fetch from and integrate with another re
push Update remote refs along with associated
rebase Forward-port local commits to the update
reset Reset current HEAD to the specified stat
rm Remove files from the working tree and f
show Show various types of objects
status Show the working tree status
tag Create, list, delete or verify a tag obj
git commands

Finding Help

Syntax:

git <command> --help

example

```
$ git commit -help
$ git commit -h
```

Configuring GIT

```
$ git config -h
$ git config -l
```

The very first thing you would do is set global options for your git installation.

This includes letting git know who you are

Setting Global Configs

```
$ git config --global user.name “First Last”
$ git config --global user.email “abc@xyz.com”
$ git config --global color.ui auto
$ git config --global core.editor vim
```

Write to global config

Instead of repo specific

Setting up the learning paths

```
$ cd learn
$ mkdir git
$ cd git
```

Creating a repository

Lets create a empty git repository from scratch

```
$ git init --bare myapp.git
```

--bare

- No working tree
- Can not edit, add files etc.
- No .git dir, everything under the main dir
- Servers as a central repository
- Used to share code with others

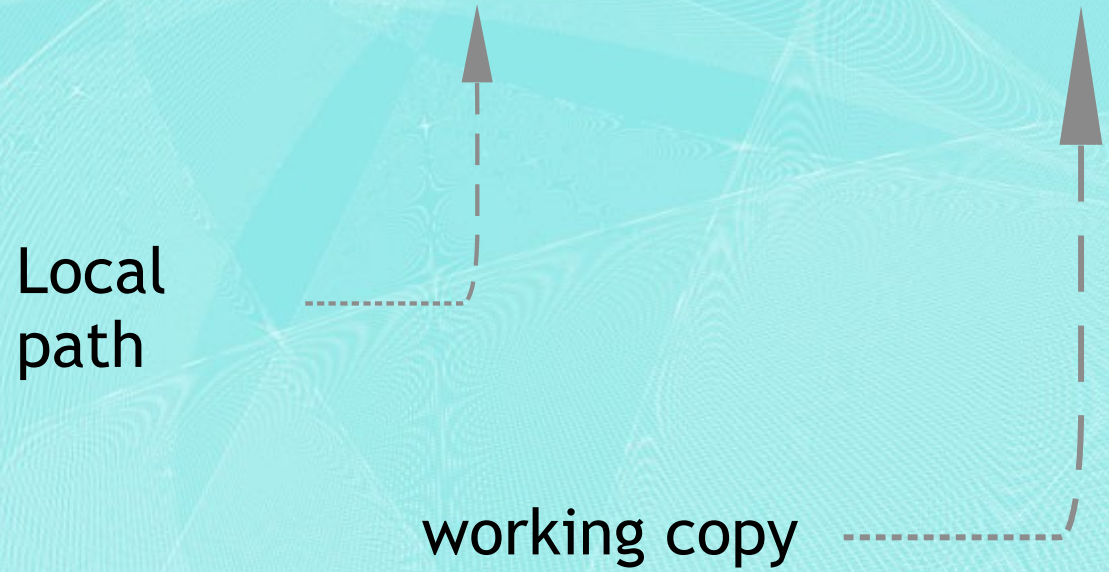
Importing Existing Code to GIT

If you have existing code which you would like to version control, import it to git without using --bare option

```
$ git init  myapp
```

Creating a working copy

```
$ cd learn/git
$ git clone --local myapp.git myapp
```



Examine Working Copy

```
$ cd myapp
$ ls -a .git/
```

```
Gouravs-MacBook-Pro:myapp gouravshah$ ls -a .git/
.               HEAD            config          hooks           objects
..              branches        description     info            refs
```