Git Basic Workflow

Stat 133 with Gaston Sanchez

Git is a Version Control System (VCS)

Repository Initialization

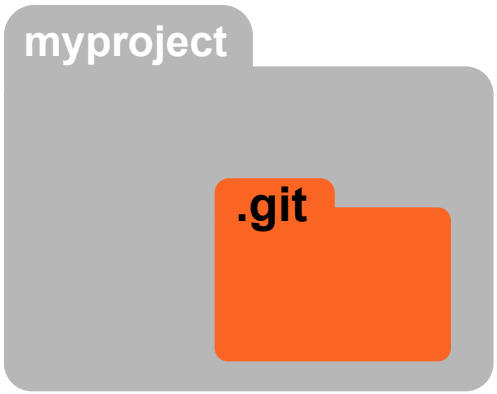
Git Initialization (of a repository)

Let's assume you are starting to work on a project (in its directory).

Open a terminal and `cd` to the project's directory.

Run `git init` to tell Git to start tracking changes.

Repository



Repository:
Database (hooked to a project) where Git stores all the versions and metadata of the project.

a project's repository is the `.git` directory

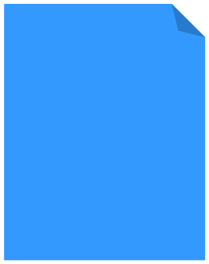
Initialization message

```
Initialized empty Git repository in
/Users/gaston/Documents/myproject/.git/
```

this is where Git will be storing
information about its tracking

Basic Workflow

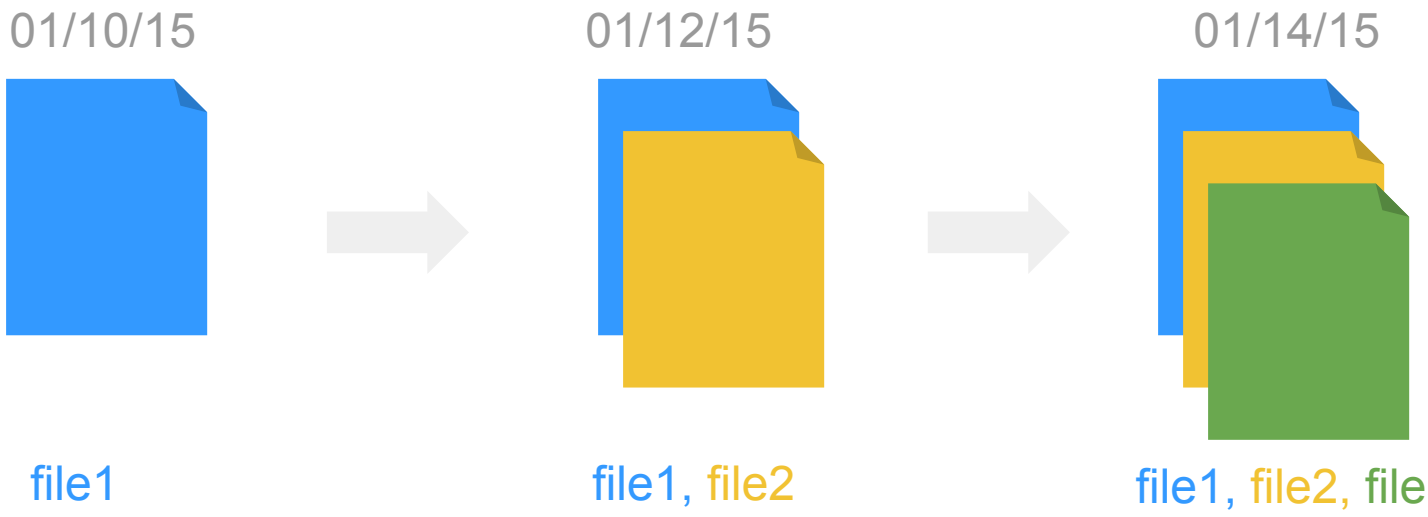
01/10/15



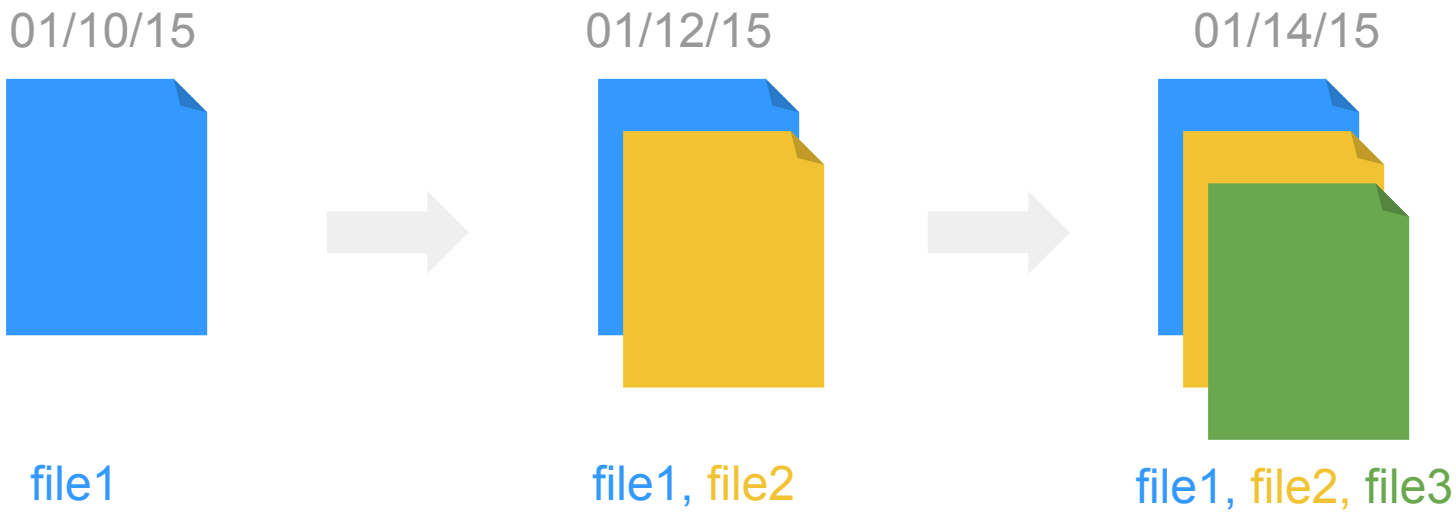
file1

```
git add file1
git status
git commit -m 'add file1'
```


```
git add file2
git status
git commit -m 'add file2'
```




```
git add file3
git status
git commit -m 'add file3'
```




```
git status
git log
```

Basic Workflow

Three basic commands

```
git add file1
```

```
git status
```

```
git commit -m "add file1"
```

Use **git status** as often as possible

Commit Messages

Commit

- to give in trust or charge; consign.
- to entrust, especially for safekeeping; commend: *(to commit one's soul to God)*
- to do; perform; perpetrate *(to commit murder; to commit an error)*

DICTIONARY DEFINITIONS

Commit

- to give in trust or charge; consign.
- to entrust, especially for safekeeping; commend: *(to commit one's soul to God)*
- to do; perform; perpetrate *(to commit murder; n error)*

VCS MEANING

- Snapshot of a complete project's state in a given point in time

Short Commit Messages

```
git commit -m 'first commit'
```

```
git commit -m 'add index file'
```

Long Commit Messages

```
git commit -m "something short

A longer description of what the
commit is all about (what it does)"
```

Suggestions

Bullet points are usually asterisks or hyphens

develop shorthand notation style:

- "`[py,R]`" (modifying python, R files)
- "`bugfix: …`" (all bug fixes commits)
- "`#35890`" (tracking numbers)

Writing Commit Messages

Bad example:

```
'fix typo'
```

Better:

```
'add missing > in index.html'
```

Writing Commit Messages

Bad:

```
'Updated cleaning data function'
```

Better:

```
'Remove missing values in cleaning
data function'
```

Writing Commit Messages

Bad:

```
'Changed correlation function. We sould discuss this next meeting'
```

Better:

```
'Changed correlation function'
```

Inspecting a Repository

Main git commands to inspect a repo

`git status` & `git log`

About git status

The `git status` command displays the state of the working directory and the staging area. It lets you see:

- which changes have been staged
- which changes haven't been staged
- which files aren't being tracked by Git

About git log

The `git log` command displays information regarding the committed project history.

- Lists the project history
- Filters the project history
- Searches for specific changes

Using git log

```
# lists history of commits
git log

# lists 5 last commits
git log -n 5

# short format listing
git log --oneline
```

Using git log

```
# searches for a particular author
git log --author="gaston"

# searches for a specifc pattern
git log --grep="readme"

# commits from a specific date
git log --since="2018-01-01"
```

Retrieving old versions (go back in time)

You can inspect the contents of a repo at a given point in time.

```
git checkout SHA#
```

```
git checkout 132cb71
git checkout 132cb71 -- file2.txt
```