Lessons Learnt from Assignment 1

- Start working on the Assignment from the very beginning even if the deadline is far away.

- If you do not know the answer to your problem Google it.
  - Familiarise with sources such as Stackoverflow

Lecture 4

- How to Generate a random number
- Introduction to GIT

Generating a Random Number

```
void srand(unsigned int seed)
```

It seeds the random number generator used by the function **rand**

seed – is an integer value to be used as seed by the pseudo-random number generator algorithm

If you give this function the same seed as input, it will return the same sequence of random numbers the same time it is invoked.

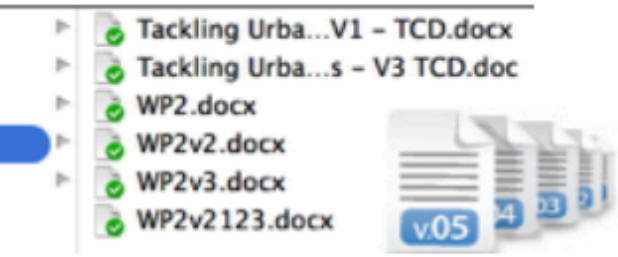
Generating a Random Number

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
   int i, n;
   time_t t;
   n = 5;

   /* Intializes random number generator */
   srand((unsigned) time(&t));

   /* Print 5 random numbers from 0 to 50 */
   for( i = 0 ; i < n ; i++ )
      printf("%d\n", rand() % 50);

   return(0);
}
```

Introduction to Git

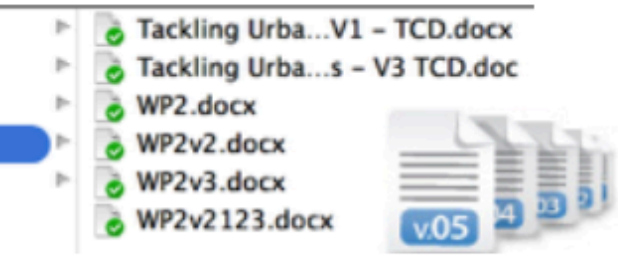
Source: https://git-scm.com/book/en/v2

A Brief History of versioning Control Systems



since the dawn of time

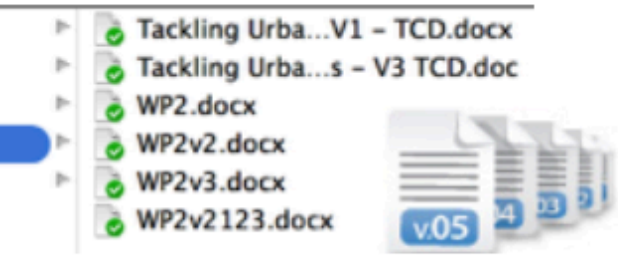
A Brief History of versioning Control Systems



since the dawn of time

Keeping multiple versions
of a file in a single folder

A Brief History of versioning Control Systems

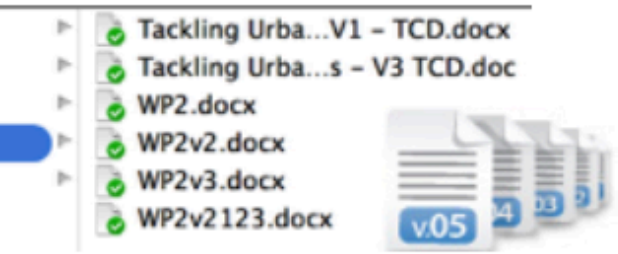


since the dawn of time

Keeping multiple versions
of a file in a single folder

- Inconsistent versioning between partners
- Cumbersome revisions
- Manual merging
- Distribution overhead (via mail)
- Disk cluttering

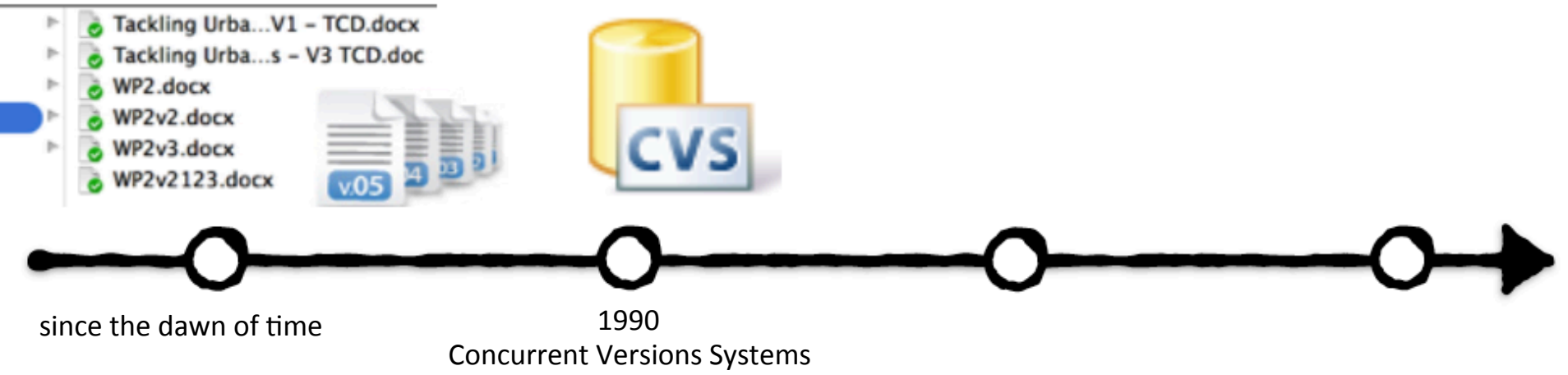
A Brief History of versioning Control Systems



since the dawn of time

1990
Concurrent Versions Systems

A Brief History of versioning Control Systems



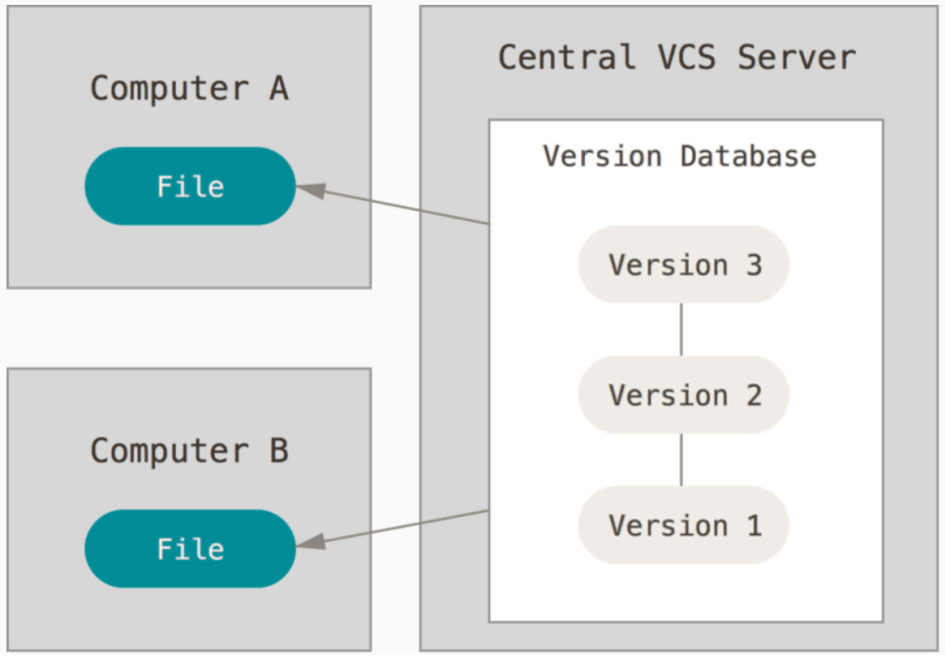
since the dawn of time

1990
Concurrent Versions Systems

Automatic versioning:

- Keep track of versions and changes in data files
- Files centrally stored in a server

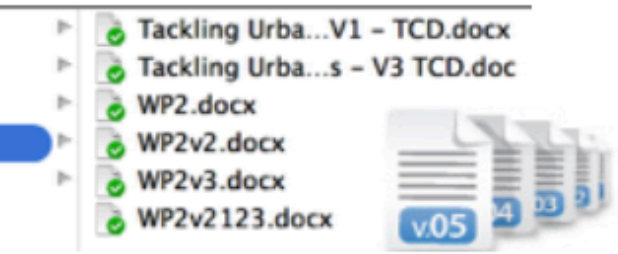
Centralised Version Control Systems



A single server contains all versions of the files

Single point of failure

A Brief History of versioning Control Systems



since the dawn of time

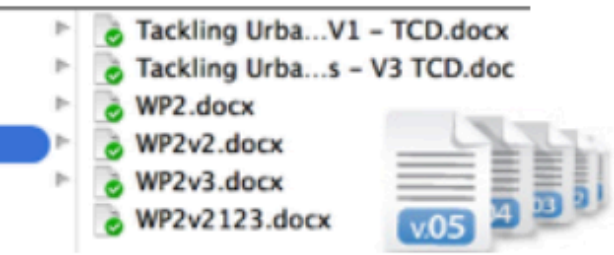
1990
Concurrent Versions Systems

Automatic versioning:

- Keep track of versions and changes in data files
- Files centrally stored in a server

- Versioning is too coarse
- Interrupted commit would cause repository inconsistency and corruption
- Centralized revision control

A Brief History of versioning Control Systems

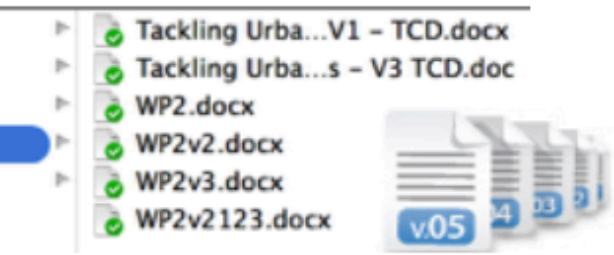


since the dawn of time

1990
Concurrent Versions Systems

2000
Subversion

A Brief History of versioning Control Systems



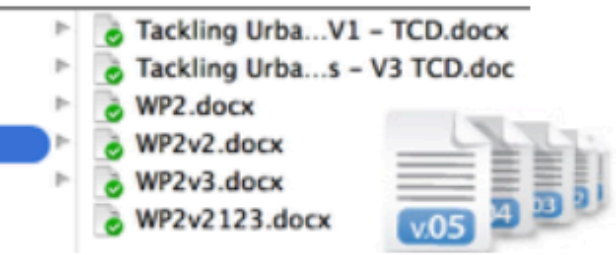
since the dawn of time

1990
Concurrent Versions Systems

2000
Subversion

- Finer grained version system
- Atomic commits

A Brief History of versioning Control Systems



since the dawn of time

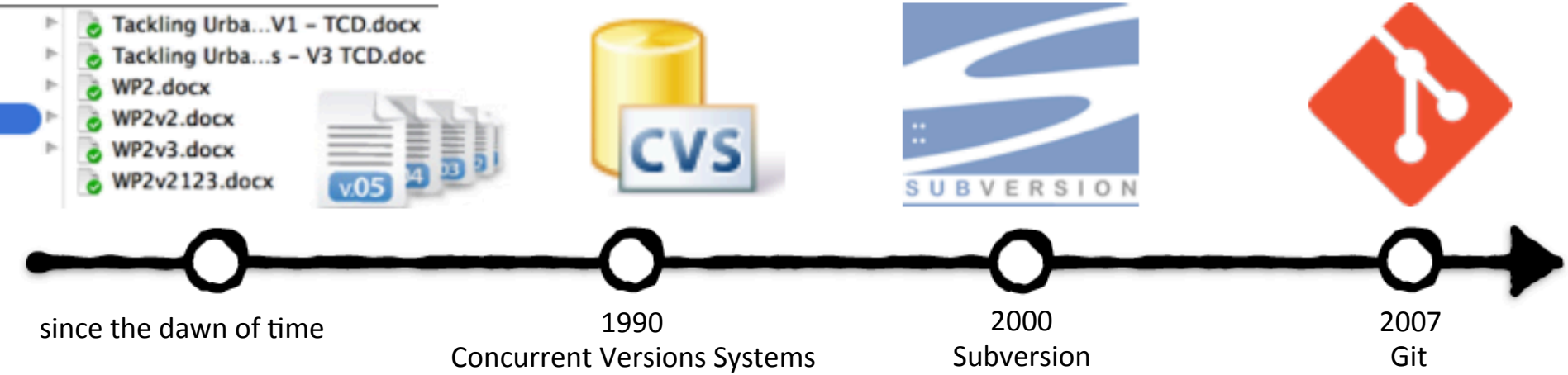
1990
Concurrent Versions Systems

2000
Subversion

- Finer grained version system
- Atomic commits

- Centralized revision control
- Modifications are not time tagged

A Brief History of versioning Control Systems



since the dawn of time

1990
Concurrent Versions Systems

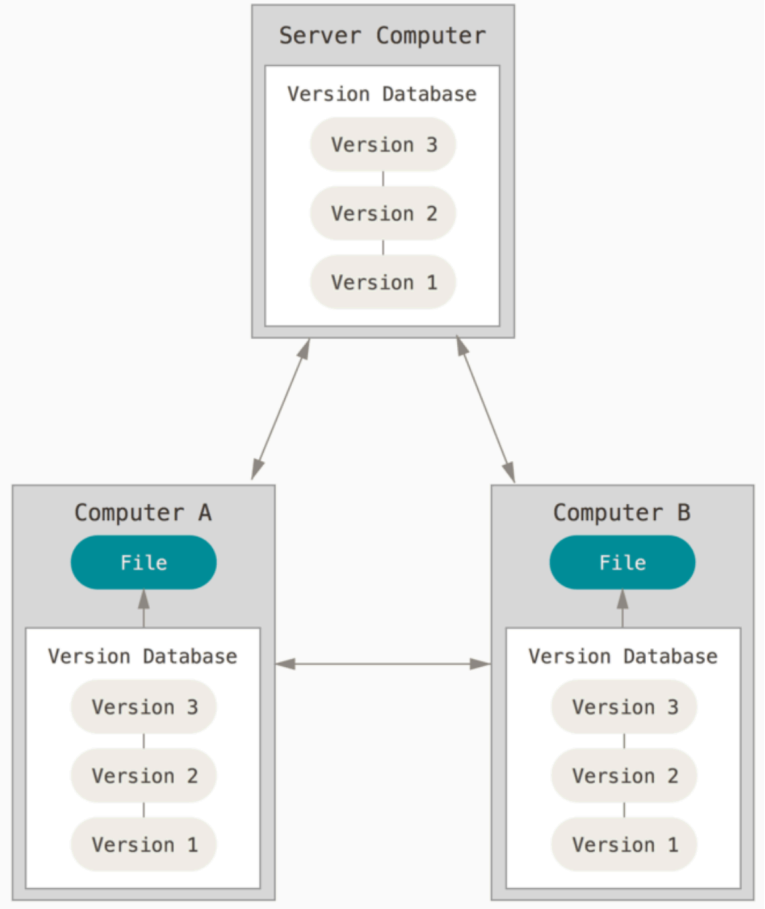
2000
Subversion

2007
Git

- Distributed version control system
- Handle large projects

Being new is awesome and everyone loves it!

Distributed Version Control Systems



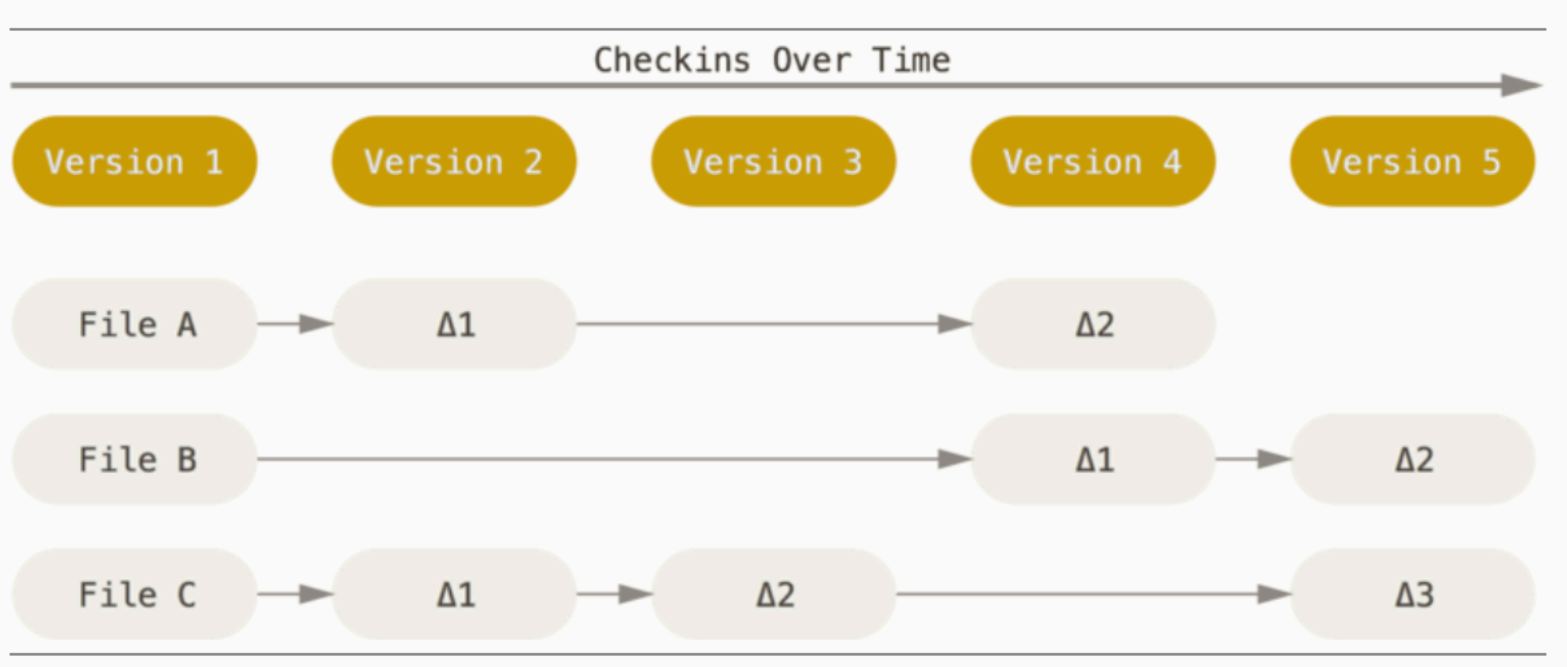
Clients fully mirror the repository including all project versions.

Most operations only need local files and resources to operate

Most operations seem to be executed almost instantaneously.

How Data Are Stored

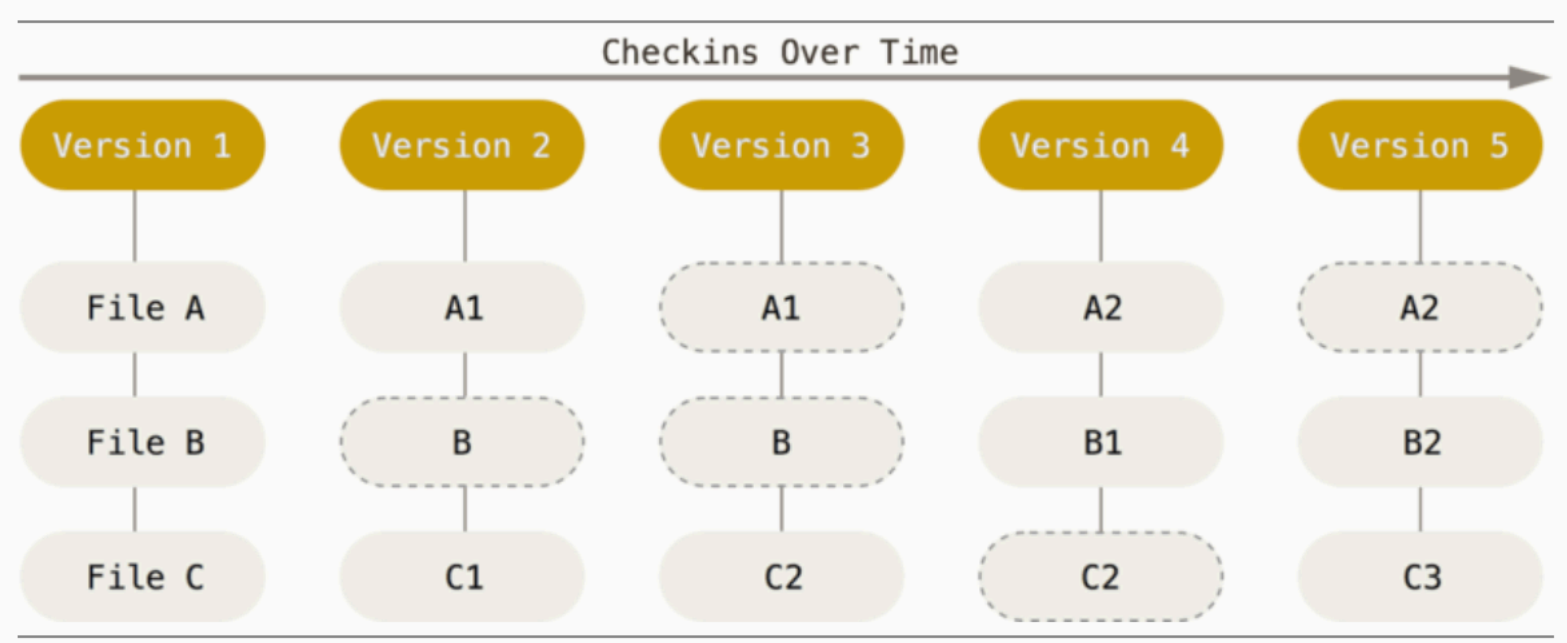
CVS and SVN



Data are stored as changes to a base version of the same file

How Data Are Stored

Git



Data are stored as snapshots of the project over time.

Most Important Thing to Remember About Git

A File can be in 3 main states:

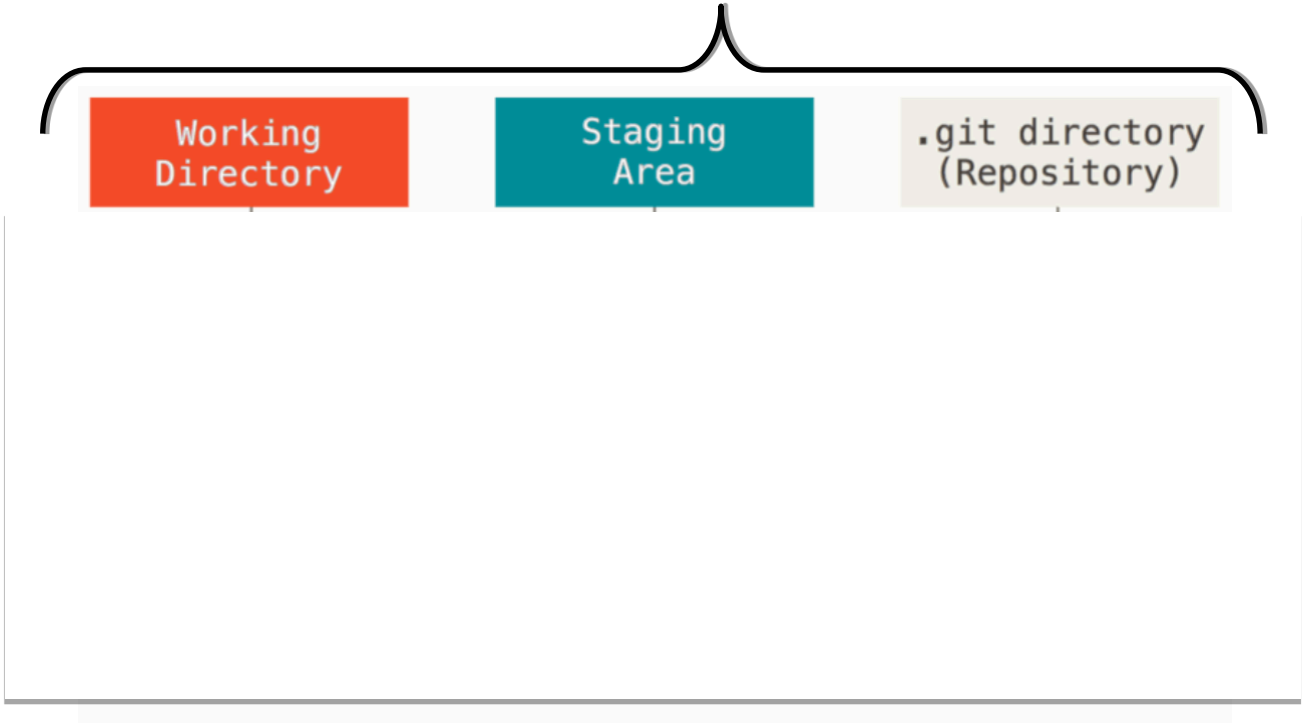
- **Committed:** the data is safely stored in your local database.
- **Modified:** the file has been changed but not yet committed.
- **Staged:** the file is marked as modified and will be included in the next commit snapshot.

Most Important Thing to Remember About Git

A File can be in 3 main states:

- **Committed:** the data is safely stored in your local database.
- **Modified:** the file has been changed but not yet committed.
- **Staged:** the file is marked as modified and will be included in the next commit snapshot.

Main Sections of a Git project
(created locally on your machine)



Most Important Thing to Remember About Git

A File can be in 3 main states:

- **Committed:** the data is safely stored in your local database.
- **Modified:** the file has been changed but not yet committed.
- **Staged:** the file is marked as modified and will be included in the next commit snapshot.

Main Sections of a Git project
(created locally on your machine)

- Metadata and object database for the project.
- This is what is copied when a repository is cloned from another computer.

Most Important Thing to Remember About Git

A File can be in 3 main states:

- **Committed:** the data is safely stored in your local database.
- **Modified:** the file has been changed but not yet committed.
- **Staged:** the file is marked as modified and will be included in the next commit snapshot.

Main Sections of a Git project
(created locally on your machine)

- Single checkout of one version of the project.
- These files are pulled out of the compressed database in the .git directory and placed on disk for you to use or modify.

Most Important Thing to Remember About Git

A File can be in 3 main states:

- **Committed:** the data is safely stored in your local database.
- **Modified:** the file has been changed but not yet committed.
- **Staged:** the file is marked as modified and will be included in the next commit snapshot.

Main Sections of a Git project
(created locally on your machine)

- It is a file, generally contained in your .git directory, that stores information about what will go into your next commit.

Git Workflow

A File can be in 3 main states:

- **Committed:** the data is safely stored in your local database.
- **Modified:** the file has been changed but not yet committed.
- **Staged:** the file is marked as modified and will be included in the next commit snapshot.

Main Sections of a Git project
(created locally on your machine)