

# Version control system: git

Madhu Belur

Control & Computing group,  
Electrical Engineering Dept, IIT Bombay

FOSSEE project, IIT Bombay

This talk (updated/corrected) at:

<http://www.ee.iitb.ac.in/%7Ebelur/talks/>

Dec 2017: Induction Training Programme

- **Version control**
- **Dropbox/Git/Google-Drive (GDrive)**
- **Advantages/disadvantages of above 3 examples**
- **Git commands**
- **Spoken tutorials**

- **Why version control?**
- hard-disk crash! (actually very **rare**, these days)
- **delete** file by mistake
- **overwrite** by mistake
- add/edit from the **wrong** past version
- difficulty remembering which is the **latest**
- final.doc, final1.doc, finalest.doc, most-final.doc, final8thDec.doc, final8thDec.doc (but last-edited on 13thDec) final8thDecWithAdditions.doc

- Why version control?
- **hard-disk crash!** (actually very **rare**, these days)
- **delete** file by mistake
- **overwrite** by mistake
- add/edit from the **wrong** past version
- difficulty remembering which is the **latest**
- final.doc, final1.doc, finalest.doc, most-final.doc, final8thDec.doc, final8thDec.doc (but last-edited on 13thDec) final8thDecWithAdditions.doc

- Why version control?
- hard-disk crash! (actually very **rare**, these days)
- **delete** file by mistake
- **overwrite** by mistake
- add/edit from the **wrong** past version
- difficulty remembering which is the **latest**
- final.doc, final1.doc, finalest.doc, most-final.doc, final8thDec.doc, final8thDec.doc (but last-edited on 13thDec) final8thDecWithAdditions.doc

- Why version control?
- hard-disk crash! (actually very **rare**, these days)
- **delete** file by mistake
- **overwrite** by mistake
- add/edit from the **wrong** past version
- difficulty remembering which is the **latest**
- final.doc, final1.doc, finalest.doc, most-final.doc, final8thDec.doc, final8thDec.doc (but last-edited on 13thDec) final8thDecWithAdditions.doc

- Why version control?
- hard-disk crash! (actually very **rare**, these days)
- **delete** file by mistake
- **overwrite** by mistake
- add/edit from the **wrong** past version
- difficulty remembering which is the **latest**
- final.doc, final1.doc, finalest.doc, most-final.doc, final8thDec.doc, final8thDec.doc (but last-edited on 13thDec) final8thDecWithAdditions.doc

- Why version control?
- hard-disk crash! (actually very **rare**, these days)
- **delete** file by mistake
- **overwrite** by mistake
- add/edit from the **wrong** past version
- difficulty remembering which is the **latest**
- final.doc, final1.doc, finalest.doc, most-final.doc, final8thDec.doc, final8thDec.doc (but last-edited on 13thDec) final8thDecWithAdditions.doc



- Why version control?
- hard-disk crash! (actually very **rare**, these days)
- **delete** file by mistake
- **overwrite** by mistake
- add/edit from the **wrong** past version
- difficulty remembering which is the **latest**
- final.doc, final1.doc, finalest.doc, most-final.doc,  
final8thDec.doc,  
final8thDec.doc (but last-edited on 13thDec)  
final8thDecWithAdditions.doc

- Why version control?
- hard-disk crash! (actually very **rare**, these days)
- **delete** file by mistake
- **overwrite** by mistake
- add/edit from the **wrong** past version
- difficulty remembering which is the **latest**
- final.doc, final1.doc, finalest.doc, most-final.doc, final8thDec.doc,  
final8thDec.doc (but last-edited on 13thDec)  
final8thDecWithAdditions.doc

- Why version control?
- hard-disk crash! (actually very **rare**, these days)
- **delete** file by mistake
- **overwrite** by mistake
- add/edit from the **wrong** past version
- difficulty remembering which is the **latest**
- final.doc, final1.doc, finalest.doc, most-final.doc, final8thDec.doc, final8thDec.doc (but last-edited on 13thDec)  
final8thDecWithAdditions.doc

- Why version control?
- hard-disk crash! (actually very **rare**, these days)
- **delete** file by mistake
- **overwrite** by mistake
- add/edit from the **wrong** past version
- difficulty remembering which is the **latest**
- final.doc, final1.doc, finalest.doc, most-final.doc, final8thDec.doc, final8thDec.doc (but last-edited on 13thDec) final8thDecWithAdditions.doc

- Why version control?
- hard-disk crash! (actually very **rare**, these days)
- **delete** file by mistake
- **overwrite** by mistake
- add/edit from the **wrong** past version
- difficulty remembering which is the **latest**
- final.doc, final1.doc, finalest.doc, most-final.doc, final8thDec.doc, final8thDec.doc (but last-edited on 13thDec) final8thDecWithAdditions.doc

- Collaboration/sharing
- Need for **systematic** file-naming (if multiple versions)
- Need to **revert** back to a particular version (line-number, etc.)
- Call past versions with short and long descriptions
- extract differences about what changes were done/resolved

- Collaboration/sharing
- Need for **systematic** file-naming (if multiple versions)
- Need to **revert** back to a particular version (line-number, etc.)
- Call past versions with short and long descriptions
- extract differences about what changes were done/resolved

- Collaboration/sharing
- Need for **systematic** file-naming (if multiple versions)
- Need to **revert** back to a particular version (line-number, etc.)
- Call past versions with short and long descriptions
- extract differences about what changes were done/resolved



- Collaboration/sharing
- Need for **systematic** file-naming (if multiple versions)
- Need to **revert** back to a particular version (line-number, etc.)
- Call past versions with short and long descriptions
- extract differences about what changes were done/resolved

- Collaboration/sharing
- Need for **systematic** file-naming (if multiple versions)
- Need to **revert** back to a particular version (line-number, etc.)
- Call past versions with short and long descriptions
- extract differences about what changes were done/resolved

- **Edit locally (using any plain text editor: Notepad/Emacs/Vim/Gedit/Pico/Nano).** Your favorite IDE: for Python, Latex, C, C++)
- Available for GNU-Linux/Windows/Mac
- Easy to install
- Very useful for both local and online version control
- Github: just an online repository: uses git commands
- Bitbucket: also an online repository: uses git commands

Also, SVN and Mercurial ( $\equiv$  hg) are good version control systems.

- **Edit locally (using any plain text editor: Notepad/Emacs/Vim/Gedit/Pico/Nano). Your favorite IDE: for Python, Latex, C, C++)**
- **Available for GNU-Linux/Windows/Mac**
- Easy to install
- Very useful for both local and online version control
- Github: just an online repository: uses git commands
- Bitbucket: also an online repository: uses git commands

Also, SVN and Mercurial ( $\equiv$  hg) are good version control systems.

- **Edit locally (using any plain text editor: Notepad/Emacs/Vim/Gedit/Pico/Nano). Your favorite IDE: for Python, Latex, C, C++)**
- **Available for GNU-Linux/Windows/Mac**
- **Easy to install**
- Very useful for both local and online version control
- Github: just an online repository: uses git commands
- Bitbucket: also an online repository: uses git commands

Also, SVN and Mercurial ( $\equiv$  hg) are good version control systems.

- **Edit locally (using any plain text editor: Notepad/Emacs/Vim/Gedit/Pico/Nano). Your favorite IDE: for Python, Latex, C, C++)**
- **Available for GNU-Linux/Windows/Mac**
- **Easy to install**
- **Very useful for both local and online version control**
- Github: just an online repository: uses git commands
- Bitbucket: also an online repository: uses git commands

Also, SVN and Mercurial ( $\equiv$  hg) are good version control systems.

- Edit locally (using any plain text editor: Notepad/Emacs/Vim/Gedit/Pico/Nano). Your favorite IDE: for Python, Latex, C, C++)
- Available for GNU-Linux/Windows/Mac
- Easy to install
- Very useful for both local and online version control
- Github: just an online repository: uses git commands
- Bitbucket: also an online repository: uses git commands

Also, SVN and Mercurial ( $\equiv$  hg) are good version control systems.

- Edit locally (using any plain text editor: Notepad/Emacs/Vim/Gedit/Pico/Nano). Your favorite IDE: for Python, Latex, C, C++)
- Available for GNU-Linux/Windows/Mac
- Easy to install
- Very useful for both local and online version control
- Github: just an online repository: uses git commands
- Bitbucket: also an online repository: uses git commands

Also, SVN and Mercurial ( $\equiv$  hg) are good version control systems.



- Edit locally (using any plain text editor: Notepad/Emacs/Vim/Gedit/Pico/Nano). Your favorite IDE: for Python, Latex, C, C++)
- Available for GNU-Linux/Windows/Mac
- Easy to install
- Very useful for both local and online version control
- Github: just an online repository: uses git commands
- Bitbucket: also an online repository: uses git commands

Also, SVN and Mercurial ( $\equiv$  hg) are good version control systems.

- **Can edit locally**
- Need not be plain text
- Can retrieve from internet
- Auto-syncing (when internet is on)
- Shows conflicts if somebody else updated before we 'pushed' changes

- **Can edit locally**
- **Need not be plain text**
- Can retrieve from internet
- Auto-syncing (when internet is on)
- Shows conflicts if somebody else updated before we 'pushed' changes

- **Can edit locally**
- **Need not be plain text**
- **Can retrieve from internet**
- Auto-syncing (when internet is on)
- Shows conflicts if somebody else updated before we 'pushed' changes

- Can edit locally
- Need not be plain text
- Can retrieve from internet
- Auto-syncing (when internet is on)
- Shows conflicts if somebody else updated before we 'pushed' changes

- Can edit locally
- Need not be plain text
- Can retrieve from internet
- Auto-syncing (when internet is on)
- Shows conflicts if somebody else updated before we 'pushed' changes

- **Need internet to edit/access**
- **Need Gmail account**
- Not good for confidential information
- Can edit directly on the drive (doc/spreadsheet/ppt)
- Can see the collaborator's cursor
- Automatically no 'conflicts'
- Each small edit is a separate version: **too** many versions

- **Need internet to edit/access**
- **Need Gmail account**
- **Not good for confidential information**
- Can edit directly on the drive (doc/spreadsheet/ppt)
- Can see the collaborator's cursor
- Automatically no 'conflicts'
- Each small edit is a separate version: **too** many versions



- Need internet to edit/access
- Need Gmail account
- Not good for confidential information
- Can edit directly on the drive (doc/spreadsheet/ppt)
- Can see the collaborator's cursor
- Automatically no 'conflicts'
- Each small edit is a separate version: **too** many versions

- Need internet to edit/access
- Need Gmail account
- Not good for confidential information
- Can edit directly on the drive (doc/spreadsheet/ppt)
- Can see the collaborator's cursor
- Automatically no 'conflicts'
- Each small edit is a separate version: **too** many versions

- Need internet to edit/access
- Need Gmail account
- Not good for confidential information
- Can edit directly on the drive (doc/spreadsheet/ppt)
- Can see the collaborator's cursor
- Automatically no 'conflicts'
- Each small edit is a separate version: **too** many versions

- Need internet to edit/access
- Need Gmail account
- Not good for confidential information
- Can edit directly on the drive (doc/spreadsheet/ppt)
- Can see the collaborator's cursor
- Automatically no 'conflicts'
- Each small edit is a separate version: **too** many versions

- **Git:** for simple to very complex coding: advanced features for this
- **Dropbox:** for any type of file: automatic syncing
- **GDrive:** no local editing. Only directly on Internet

- **Git:** for simple to very complex coding: advanced features for this
- **Dropbox:** for any type of file: automatic syncing
- **GDrive:** no local editing. Only directly on Internet

- **Git:** for simple to very complex coding: advanced features for this
- **Dropbox:** for any type of file: automatic syncing
- **GDrive:** no local editing. Only directly on Internet

**Favorite editor:** say  
**emacs/vim/gedit/notepad/wordpad/nano**

- Git: any editor
- Dropbox: any editor
- GDrive: (at least default): forced to use very-basic editor



**Favorite editor:** say  
**emacs/vim/gedit/notepad/wordpad/nano**

- Git: any editor
- Dropbox: any editor
- GDrive: (at least default): forced to use very-basic editor

**Favorite editor:** say  
emacs/vim/gedit/notepad/wordpad/nano

- **Git:** any editor
- **Dropbox:** any editor
- **GDrive:** (at least default): forced to use very-basic editor

**Favorite editor:** say  
emacs/vim/gedit/notepad/wordpad/nano

- **Git:** any editor
- **Dropbox:** any editor
- **GDrive:** (at least default): forced to use very-basic editor

**Favorite editor: say  
emacs/vim/gedit/notepad/wordpad/nano**

- **Git: any editor**
- **Dropbox: any editor**
- **GDrive: (at least default): forced to use very-basic editor**

- **Git:** shows pin-pointed differences, and shortcuts for resolving
- **Dropbox:** reasonably (I believe)
- **GDrive:** shows with colour about changes

- **Git:** shows pin-pointed differences, and shortcuts for resolving
- **Dropbox:** reasonably (I believe)
- **GDrive:** shows with colour about changes

- **Git:** shows pin-pointed differences, and shortcuts for resolving
- **Dropbox:** reasonably (I believe)
- **GDrive:** shows with colour about changes

- **Git:** shows pin-pointed differences, and shortcuts for resolving
- **Dropbox:** reasonably (I believe)
- **GDrive:** shows with colour about changes



- **Git: manual syncing: recommended for complex code (with shortcuts)**  
Ensure no bugs before 'committing' changes or letting others see
- Others (Dropbox/GDrive) automatic: and can be irritating or advantageous
- Each local small change keeps syncing: bad for latex and collaborators, etc.
- GDrive: **nothing** local anyway

- **Git: manual syncing: recommended for complex code (with shortcuts)**  
Ensure no bugs before ‘committing’ changes or letting others see
- Others (Dropbox/GDrive) automatic: and can be irritating or advantageous
- Each local small change keeps syncing: bad for latex and collaborators, etc.
- GDrive: **nothing** local anyway

- **Git: manual syncing: recommended for complex code (with shortcuts)**  
Ensure no bugs before ‘committing’ changes or letting others see
- **Others (Dropbox/GDrive) automatic: and can be irritating or advantageous**
- Each local small change keeps syncing: bad for latex and collaborators, etc.
- GDrive: **nothing** local anyway

- **Git: manual syncing: recommended for complex code (with shortcuts)**  
Ensure no bugs before ‘committing’ changes or letting others see
- **Others (Dropbox/GDrive) automatic: and can be irritating or advantageous**
- **Each local small change keeps syncing: bad for latex and collaborators, etc.**
- **GDrive: nothing** local anyway

- **Git: manual syncing: recommended for complex code (with shortcuts)**  
Ensure no bugs before ‘committing’ changes or letting others see
- **Others (Dropbox/GDrive) automatic: and can be irritating or advantageous**
- **Each local small change keeps syncing: bad for latex and collaborators, etc.**
- **GDrive: nothing local anyway**

- **Git: manual syncing: recommended for complex code (with shortcuts)**  
Ensure no bugs before ‘committing’ changes or letting others see
- **Others (Dropbox/GDrive) automatic: and can be irritating or advantageous**
- **Each local small change keeps syncing: bad for latex and collaborators, etc.**
- **GDrive: nothing local anyway**

- **GDrive: only** online repository
- Local repo possible for Dropbox and Git
- Online repository: relevant for sharing or multiple PCs for oneself

- GDrive: **only** online repository
- Local repo possible for Dropbox and Git
- Online repository: relevant for sharing or multiple PCs for oneself



- GDrive: **only** online repository
- Local repo possible for Dropbox and Git
- Online repository: relevant for sharing or multiple PCs for oneself

- GDrive: **only** online repository
- Local repo possible for Dropbox and Git
- Online repository: relevant for sharing or multiple PCs for oneself

## Relevant for code: when packages provide support for multiple versions

- Git: targets such purposes: internationally used by individuals and companies
- Dropbox/GDrive: need to just make copies. No 'merging'

**Relevant for code: when packages provide support for multiple versions**

- **Git: targets such purposes: internationally used by individuals and companies**
- **Dropbox/GDrive: need to just make copies. No 'merging'**

Relevant for code: when packages provide support for multiple versions

- **Git:** targets such purposes: internationally used by individuals and companies
- **Dropbox/GDrive:** need to just make copies. No ‘merging’

Relevant for code: when packages provide support for multiple versions

- **Git:** targets such purposes: internationally used by individuals and companies
- **Dropbox/GDrive:** need to just make copies. No ‘merging’

- **Github and bitbucket: some free/paid private repositories**
- Public repositories (meaning anybody can VIEW, but edit-access for few only.)
- Dropbox/GDrive: can share to specific email-addresses/accounts (any number)
- GDrive allows: view/edit: but also **'commenting'** (intermediate level access)

- Github and bitbucket: some free/paid private repositories
- Public repositories (meaning anybody can VIEW, but edit-access for few only.)
- Dropbox/GDrive: can share to specific email-addresses/accounts (any number)
- GDrive allows: view/edit: but also 'commenting' (intermediate level access)



- Github and bitbucket: some free/paid private repositories
- Public repositories (meaning anybody can VIEW, but edit-access for few only.)
- Dropbox/GDrive: can share to specific email-addresses/accounts (any number)
- GDrive allows: view/edit: but also 'commenting' (intermediate level access)

- Github and bitbucket: some free/paid private repositories
- Public repositories (meaning anybody can VIEW, but edit-access for few only.)
- Dropbox/GDrive: can share to specific email-addresses/accounts (any number)
- GDrive allows: view/edit: but also 'commenting' (intermediate level access)

- Github and bitbucket: some free/paid private repositories
- Public repositories (meaning anybody can VIEW, but edit-access for few only.)
- Dropbox/GDrive: can share to specific email-addresses/accounts (any number)
- GDrive allows: view/edit: but also **'commenting'** (intermediate level access)

## Why Git?

- **Very relevant for code**
- Internationally the norm: only **'laymen'** shy away from git/github
- Advanced features: **fork** (edit independently and perhaps merge)
- Ask owner for **'pull request'** to merge our fork back
- For alpha/beta versions of new packages
- Github shows full code history: allows automatic **testing** (nose-tests, test driven development)

## Why Git?

- Very relevant for code
- Internationally the norm: only **'laymen'** shy away from git/github
- Advanced features: **fork** (edit independently and perhaps merge)
- Ask owner for **'pull request'** to merge our fork back
- For alpha/beta versions of new packages
- Github shows full code history: allows automatic **testing** (nose-tests, test driven development)

## Why Git?

- Very relevant for code
- Internationally the norm: only **'laymen'** shy away from git/github
- Advanced features: **fork** (edit independently and perhaps merge)
- Ask owner for **'pull request'** to merge our fork back
- For alpha/beta versions of new packages
- Github shows full code history: allows automatic **testing** (nose-tests, test driven development)

## Why Git?

- Very relevant for code
- Internationally the norm: only **'laymen'** shy away from git/github
- Advanced features: **fork** (edit independently and perhaps merge)
- Ask owner for **'pull request'** to merge our fork back
- For alpha/beta versions of new packages
- Github shows full code history: allows automatic **testing** (nose-tests, test driven development)

## Why Git?

- Very relevant for code
- Internationally the norm: only **'laymen'** shy away from git/github
- Advanced features: **fork** (edit independently and perhaps merge)
- Ask owner for **'pull request'** to merge our fork back
- For alpha/beta versions of new packages
- Github shows full code history: allows automatic **testing** (nose-tests, test driven development)



## Why Git?

- Very relevant for code
- Internationally the norm: only **'laymen'** shy away from git/github
- Advanced features: **fork** (edit independently and perhaps merge)
- Ask owner for **'pull request'** to merge our fork back
- For alpha/beta versions of new packages
- Github shows full code history: allows automatic **testing** (nose-tests, test driven development)

## Why Git?

- Very relevant for code
- Internationally the norm: only **'laymen'** shy away from git/github
- Advanced features: **fork** (edit independently and perhaps merge)
- Ask owner for **'pull request'** to merge our fork back
- For alpha/beta versions of new packages
- Github shows full code history: allows automatic **testing** (nose-tests, test driven development)

## Why Git?

- Very relevant for code
- Internationally the norm: only **'laymen'** shy away from git/github
- Advanced features: **fork** (edit independently and perhaps merge)
- Ask owner for **'pull request'** to merge our fork back
- For alpha/beta versions of new packages
- Github shows full code history: allows automatic **testing** (nose-tests, test driven development)

- Google for git MS-windows setup file
- Open MS command prompt and make a suitable directory for testing purposes
- md and cd are MD-DOS commands for making a directory and changing into that directory

- `sudo apt-get install git`

- `git config --global user.name "Madhu Belur"`
- `git config --global user.email name@example.com`
- **(without ‘--global’ flag: only for that directory)**
- `git init`
- `git add filename`

- `git config --global user.name "Madhu Belur"`
- `git config --global user.email name@example.com`
- **(without ‘--global’ flag: only for that directory)**
- `git init`
- `git add filename`

- `git config --global user.name "Madhu Belur"`
- `git config --global user.email name@example.com`
- **(without ‘--global’ flag: only for that directory)**
- `git init`
- `git add filename`



- **git commit**
- **git diff**
- `git revert #` to the version you want
- In bash, `#` means everything after `#` is ‘comment’ (on that line: ignore)

Many spoken-tutorials available at  
<http://spoken-tutorial.org>

- `git commit`
- `git diff`
- `git revert #` to the version you want
- In bash, `#` means everything after `#` is ‘comment’ (on that line: ignore)

Many spoken-tutorials available at  
<http://spoken-tutorial.org>

- `git commit`
- `git diff`
- `git revert #` to the version you want
- In bash, `#` means everything after `#` is ‘comment’ (on that line: ignore)

Many spoken-tutorials available at  
<http://spoken-tutorial.org>

- `git commit`
- `git diff`
- `git revert #` to the version you want
- In bash, `#` means everything after `#` is ‘comment’ (on that line: ignore)

Many spoken-tutorials available at  
<http://spoken-tutorial.org>

## How to change editor

- `git config --global core.editor notepad #` (for MS Windows)
- `git config --global core.editor nano #` (for linux, etc)
- `git config --global core.editor emacs #` (for linux, etc)

<https://help.github.com/articles/associating-text-editors-with-git>

<https://help.github.com/articles/associating-text-editors-with-git/#platform-windows>

## *How to delete a file*

- First delete the file the regular way
- Tell git you are no longer interested in the file by  
`git rm filename`
- `git commit #` tell as comment that you deleted the file (intentionally)

## *How to retrieve deleted file*

- If file was never committed to git, then git cannot help:-)
- If file was committed at least once, and then we retrieve the deleted (and committed) file by:  
`git checkout HEAD^ filename`
- This will retrieve the version just before deleting
- Need to commit again (to tell git about the latest version: post un-deleting the file)
- Can retrieve previously committed versions also (using revert)
- <https://stackoverflow.com/questions/953481/find-and-restore-a-deleted-file-in-a-git-repository>

- Need to understand small but complex difference between
  - staging area
  - checkout
  - HEAD
  - MASTER Branch
- HEAD means current (latest committed)
- HEAD 1 means just previously committed one (HEAD minus 1)

git log file or git log  
gives log (with your descriptions)



# *How to see differences*

- `git diff filename`  
shows difference between just-previously committed one and latest

- **git clone # optional (for taking previous repo)**
- git add
- change, commit, 'push' # push also optional
- Or request for pulling our changes
- pull-request: the owner pulls our changes, so we request him/her to pull our changes

git from browser allows interactive displays: shows conflicts with syntax highlighting

- **git clone # optional (for taking previous repo)**
- **git add**
- change, commit, 'push' # push also optional
- Or request for pulling our changes
- pull-request: the owner pulls our changes, so we request him/her to pull our changes

git from browser allows interactive displays: shows conflicts with syntax highlighting

- **git clone # optional (for taking previous repo)**
- **git add**
- **change, commit, 'push' # push also optional**
- Or request for pulling our changes
- pull-request: the owner pulls our changes, so we request him/her to pull our changes

git from browser allows interactive displays: shows conflicts with syntax highlighting

- git clone # optional (for taking previous repo)
- git add
- change, commit, 'push' # push also optional
- Or request for pulling our changes
- pull-request: the owner pulls our changes, so we request him/her to pull our changes

git from browser allows interactive displays: shows conflicts with syntax highlighting

- `git clone` # optional (for taking previous repo)
- `git add`
- change, commit, 'push' # push also optional
- Or request for pulling our changes
- **pull-request: the owner pulls our changes, so we request him/her to pull our changes**

*git from browser allows interactive displays: shows conflicts with syntax highlighting*

- `git clone` # optional (for taking previous repo)
- `git add`
- change, commit, 'push' # push also optional
- Or request for pulling our changes
- **pull-request: the owner pulls our changes, so we request him/her to pull our changes**

*git from browser allows interactive displays: shows conflicts with syntax highlighting*

- `git clone` # optional (for taking previous repo)
- `git add`
- change, commit, 'push' # push also optional
- Or request for pulling our changes
- **pull-request: the owner pulls our changes, so we request him/her to pull our changes**

*git from browser allows interactive displays: shows conflicts with syntax highlighting*



- `git clone` # optional (for taking previous repo)
- `git add`
- change, commit, 'push' # push also optional
- Or request for pulling our changes
- pull-request: the owner pulls our changes, so we request him/her to pull our changes

`git` from browser allows interactive displays: shows conflicts with syntax highlighting

- **Learn to keep past versions ‘systematically’**
- final.doc, final1.doc, finalest.doc, most-final.doc, final8thDec.doc, final8thDec.doc (but last-edited on 13thDec) final8thDecWithAdditions.doc
- Relevant for retrieving: recommended for local/individual usage also

- Learn to keep past versions ‘systematically’
- **final.doc, final1.doc, finalest.doc, most-final.doc,**  
final8thDec.doc,  
final8thDec.doc (but last-edited on 13thDec)  
final8thDecWithAdditions.doc
- Relevant for retrieving: recommended for  
local/individual usage also

- Learn to keep past versions ‘systematically’
- **final.doc, final1.doc, finalest.doc, most-final.doc, final8thDec.doc,**  
final8thDec.doc (but last-edited on 13thDec)  
final8thDecWithAdditions.doc
- Relevant for retrieving: recommended for local/individual usage also

- Learn to keep past versions ‘systematically’
- `final.doc`, `final1.doc`, `finalest.doc`, `most-final.doc`,  
`final8thDec.doc`,  
`final8thDec.doc` (but last-edited on 13thDec)  
`final8thDecWithAdditions.doc`
- Relevant for retrieving: recommended for  
local/individual usage also

- Learn to keep past versions ‘systematically’
- final.doc, final1.doc, finalest.doc, most-final.doc, final8thDec.doc, final8thDec.doc (but last-edited on 13thDec) final8thDecWithAdditions.doc
- Relevant for retrieving: recommended for local/individual usage also

- Learn to keep past versions ‘systematically’
- final.doc, final1.doc, finalest.doc, most-final.doc, final8thDec.doc, final8thDec.doc (but last-edited on 13thDec) final8thDecWithAdditions.doc
- Relevant for retrieving: recommended for local/individual usage also

- Learn to keep past versions ‘systematically’
- final.doc, final1.doc, finalest.doc, most-final.doc, final8thDec.doc, final8thDec.doc (but last-edited on 13thDec) final8thDecWithAdditions.doc
- Relevant for retrieving: recommended for local/individual usage also



- Tell git explicitly to maintain only **'source'** (and not **'generated'** files)
- Extra past source-files saved: extra disk-usage: overhead
- Negligible overhead (of saving past versions)
- Can quickly search through comments of past versions
- Internet bandwidth: always will be inadequate: video calls, HD-movies
- Safer/wiser to be independent of Internet

Latest version of this ppt at:

<http://www.ee.iitb.ac.in/%7Ebelur/talks/>

Thank you

- Tell git explicitly to maintain only **'source'** (and not **'generated'** files)
- **Extra past source-files saved: extra disk-usage: overhead**
- Negligible overhead (of saving past versions)
- Can quickly search through comments of past versions
- Internet bandwidth: always will be inadequate: video calls, HD-movies
- Safer/wiser to be independent of Internet

Latest version of this ppt at:

<http://www.ee.iitb.ac.in/%7Ebelur/talks/>

Thank you

- Tell git explicitly to maintain only **'source'** (and not 'generated' files)
- Extra past source-files saved: extra disk-usage: overhead
- Negligible overhead (of saving past versions)
- Can quickly search through comments of past versions
- Internet bandwidth: always will be inadequate: video calls, HD-movies
- Safer/wiser to be independent of Internet

Latest version of this ppt at:

<http://www.ee.iitb.ac.in/%7Ebelur/talks/>

Thank you

- Tell git explicitly to maintain only **'source'** (and not 'generated' files)
- Extra past source-files saved: extra disk-usage: overhead
- Negligible overhead (of saving past versions)
- Can quickly search through comments of past versions
- Internet bandwidth: always will be inadequate: video calls, HD-movies
- Safer/wiser to be independent of Internet

Latest version of this ppt at:

<http://www.ee.iitb.ac.in/%7Ebelur/talks/>

Thank you

- Tell git explicitly to maintain only **'source'** (and not 'generated' files)
- Extra past source-files saved: extra disk-usage: overhead
- Negligible overhead (of saving past versions)
- Can quickly search through comments of past versions
- Internet bandwidth: always will be inadequate: video calls, HD-movies
- Safer/wiser to be independent of Internet

Latest version of this ppt at:

<http://www.ee.iitb.ac.in/%7Ebelur/talks/>

Thank you

- Tell git explicitly to maintain only **‘source’** (and not **‘generated’** files)
- Extra past source-files saved: extra disk-usage: overhead
- Negligible overhead (of saving past versions)
- Can quickly search through comments of past versions
- Internet bandwidth: always will be inadequate: video calls, HD-movies
- Safer/wiser to be independent of Internet

Latest version of this ppt at:

<http://www.ee.iitb.ac.in/%7Ebelur/talks/>

Thank you

- Tell git explicitly to maintain only **‘source’** (and not **‘generated’** files)
- Extra past source-files saved: extra disk-usage: overhead
- Negligible overhead (of saving past versions)
- Can quickly search through comments of past versions
- Internet bandwidth: always will be inadequate: video calls, HD-movies
- Safer/wiser to be independent of Internet

Latest version of this ppt at:

<http://www.ee.iitb.ac.in/%7Ebelur/talks/>

Thank you

- Tell git explicitly to maintain only **‘source’** (and not **‘generated’** files)
- Extra past source-files saved: extra disk-usage: overhead
- Negligible overhead (of saving past versions)
- Can quickly search through comments of past versions
- Internet bandwidth: always will be inadequate: video calls, HD-movies
- Safer/wiser to be independent of Internet

Latest version of this ppt at:

<http://www.ee.iitb.ac.in/%7Ebelur/talks/>

Thank you



- Tell git explicitly to maintain only **‘source’** (and not **‘generated’** files)
- Extra past source-files saved: extra disk-usage: overhead
- Negligible overhead (of saving past versions)
- Can quickly search through comments of past versions
- Internet bandwidth: always will be inadequate: video calls, HD-movies
- Safer/wiser to be independent of Internet

Latest version of this ppt at:

<http://www.ee.iitb.ac.in/%7Ebelur/talks/>

Thank you