

Programming tips. Git (1).

Git or CTRL+S evolution

[Rémi Ete](#)

DESY

November 12, 2018



Today: Git in a nutshell



What is Git?

In a nutshell

- Version control system
- Developed by Linus Tavarid (2005)
- (Not-really) competitors:
 - SVN
 - Mercurial
 - CVS
- What is it about?



What is Git?

In a nutshell

- Version control system
- Developed by Linus Tovarld (2005)
- (Not-really) competitors:
 - SVN
 - Mercurial
 - CVS
- What is it about?
 - "I want to save the history of these files"



What is Git?

In a nutshell

- Version control system
- Developed by Linus Tavarld (2005)
- (Not-really) competitors:
 - SVN
 - Mercurial
 - CVS
- What is it about?
 - "I want to save the history of these files"
 - "I want to make versions of my code"



What is Git?

In a nutshell

- Version control system
- Developed by Linus Tavarld (2005)
- (Not-really) competitors:
 - SVN
 - Mercurial
 - CVS
- What is it about?
 - "I want to save the history of these files"
 - "I want to make versions of my code"
 - "My boss wants me to compare my current results with the one of 2 years ago"



What is Git?

In a nutshell

- Version control system
- Developed by Linus Tavarld (2005)
- (Not-really) competitors:
 - SVN
 - Mercurial
 - CVS
- What is it about?
 - "I want to save the history of these files"
 - "I want to make versions of my code"
 - "My boss wants me to compare my current results with the one of 2 years ago"
 - "I want to try something new in my code but I don't want to break its current state"



What is Git?

In a nutshell

- Version control system
- Developed by Linus Tovarld (2005)
- (Not-really) competitors:
 - SVN
 - Mercurial
 - CVS
- What is it about?
 - "I want to save the history of these files"
 - "I want to make versions of my code"
 - "My boss wants me to compare my current results with the one of 2 years ago"
 - "I want to try something new in my code but I don't want to break its current state"
 - "I want to develop many ideas at the same time and don't want to mess up the current state"



What is Git?

In a nutshell

- Version control system
- Developed by Linus Tovarld (2005)
- (Not-really) competitors:
 - SVN
 - Mercurial
 - CVS
- What is it about?
 - "I want to save the history of these files"
 - "I want to make versions of my code"
 - "My boss wants me to compare my current results with the one of 2 years ago"
 - "I want to try something new in my code but I don't want to break its current state"
 - "I want to develop many ideas at the same time and don't want to mess up the current state"
 - "I want to work on the same piece of code with other persons at the same time"



What is Git?

In a nutshell

- Version control system
- Developed by Linus Tovarld (2005)
- (Not-really) competitors:
 - SVN
 - Mercurial
 - CVS
- What is it about?
 - "I want to save the history of these files"
 - "I want to make versions of my code"
 - "My boss wants me to compare my current results with the one of 2 years ago"
 - "I want to try something new in my code but I don't want to break its current state"
 - "I want to develop many ideas at the same time and don't want to mess up the current state"
 - "I want to work on the same piece of code with other persons at the same time"
- Typical use cases:
 - Code: ILCSof, your ROOT macros / Marlin processors / shell or job scripts
 - Documentation: doxygen, wikis
 - PhD thesis
 - Papers: articles, proceedings, abstracts



Git in practice



Git is not Github

- Version control system
- A C library with bindings (python, java, ...)
- Client/server command line tools (git, git-gui)
- **For you:**
 - Client command-line on your computer

Git



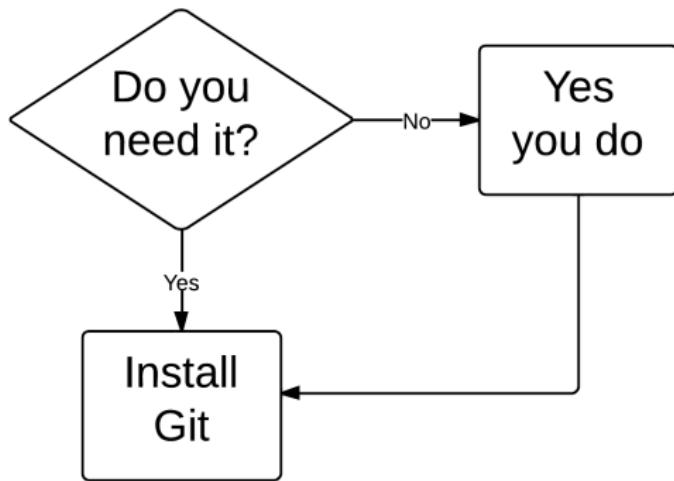
-
- A domain hosting Git servers: commercial, Microsoft
 - Host Git repositories (public, private)
 - Web client interface: github.com
 - Continus integration tools: TravisCI, Zenodo
 - Alternatives: Gitlab, Bitbucket, SourceForge
 - **For you:**
 - Host your remote repository

Github



Do you need git ?

Version Control Flowchart



Some Git definitions



Some Git definitions

- **Working directory:** local directory where you have content that you want to manage with git



Some Git definitions

- **Working directory:** local directory where you have content that you want to manage with git
- **Commit:** full snapshot of your working directory contents. Uniquely identified by a 40 character SHA1 hash. Refer to the exact state of your project and can be used for copying or restoring your project.



Some Git definitions

- **Working directory:** local directory where you have content that you want to manage with git
- **Commit:** full snapshot of your working directory contents. Uniquely identified by a 40 character SHA1 hash. Refer to the exact state of your project and can be used for copying or restoring your project.
- **Index:** local staging area in your working directory for your changes before being committed.



Some Git definitions

- **Working directory:** local directory where you have content that you want to manage with git
- **Commit:** full snapshot of your working directory contents. Uniquely identified by a 40 character SHA1 hash. Refer to the exact state of your project and can be used for copying or restoring your project.
- **Index:** local staging area in your working directory for your changes before being committed.
- **Branch:** a series of commits (history). A branch has a name (default is *master*).



Some Git definitions

- **Working directory:** local directory where you have content that you want to manage with git
- **Commit:** full snapshot of your working directory contents. Uniquely identified by a 40 character SHA1 hash. Refer to the exact state of your project and can be used for copying or restoring your project.
- **Index:** local staging area in your working directory for your changes before being committed.
- **Branch:** a series of commits (history). A branch has a name (default is *master*).
- **HEAD:** a commit pointer stored by Git. Points on the current revision/version of your files. Should always point on the latest commit of the current branch, else you are in a state called *DETACHED HEAD*.



Some Git definitions

- **Working directory:** local directory where you have content that you want to manage with git
- **Commit:** full snapshot of your working directory contents. Uniquely identified by a 40 character SHA1 hash. Refer to the exact state of your project and can be used for copying or restoring your project.
- **Index:** local staging area in your working directory for your changes before being committed.
- **Branch:** a series of commits (history). A branch has a name (default is *master*).
- **HEAD:** a commit pointer stored by Git. Points on the current revision/version of your files. Should always point on the latest commit of the current branch, else you are in a state called *DETACHED HEAD*.
- **Remote repository:** remote location where a copy of your working directory can be stored (i.e Github). On your computer, a remote location associate a name (i.e *origin*) to a repository URL (e.g <https://github.com/username/reponame.git>).

