GANATAN

Getting started with git

Complete Guide

Update on February 27, 2025

Git is a must-have version control system for developers.

It allows:

- to **follow** the progress of a project,
- to **collaborate** effectively as a team,
- to **manage** different versions of the code.

Our goal

- **Optimize** our development workflow,
- **Ensure** accurate tracking of changes,
- Make branch management **easier** .

Documentation

Site officiel

https://git-scm.com/

Documentation

https://git-scm.com/book/fr/v2

Installation under Ubuntu

```
# Installation
sudo apt-get --yes install git

# Uninstall
sudo apt-get --yes purge git
```

Configuration

```
# See configuration
git config --list

# Configure the default account
git config --global user.email "user_email"
git config --global user.name "user_name"

# Configure the use of notepad
git config --global core.editor "notepad"
```

First deposit

Creating the first repo

```
# Creating the README file
echo "# essai" >> README.md

# Initializing the repository
git init

# Add the README file to the index
git add README.md

# Adds the latest modified files
git add *

# Commit
git commit -m "first commit"

# Remote without login
git remote add origin https://github.com/user-name/example.git

# Remote with identifier
git remote add origin https://user-name:user-password@github.com/user-name/example.git

# Push
git push -u origin master

# Pull
git pull
```

Orders

```
# Initialize the project in git repository
git init

git config

# Configure the default account
git config --global user.email "user_email"
git config --global user.name "user_name"

# Undo the last commit
git commit --amend

# Check project status
git status
```

```
# Send files to the stage
git add *

# Check the differences
git diff

# send a commit
git commit -m "feature01"

# make an add and send a commit
git commit -a -m "feature01"

# List commits with an editor
git log

# List commits on one line
git log --oneline

# List detailed commits on one line on a file
git log --oneline -p index.html

git push

# sets to the state at the time of commit id
git checkout id

# revert to current state # git checkout master
# fetch a file at a time and make a commit

# undo a commit
git checkout id index.html
git commit -a -m "nouveau commit"

# Cancel internship
git revert id


git reset HEAD fichier
git merge
```

```
# Create a prototype branch
git branch prototype

# List of branches
git branch

# Selecting a branch
git checkout prototype

# Selecting the master branch
git checkout master
```