

## TP1 : Git

### I. Débuter avec git

Git est un logiciel open source et peut être obtenu ici : <http://git-scm.com/downloads>

Git permet de gérer les versions et évolutions de code source (et par extension de tout fichier texte). Pour ce faire, pour chaque modification significative du code (ajout de fonctionnalités, correction de bugs, etc.), on pourra faire un commit, c'est-à-dire sauvegarder une version du code.

Git stocke les versions successives d'un dossier. Une fois « surveillé » par Git, un dossier est un repository ou « repo ». En français on parlera de « dépôt ».

Git permet également de travailler à plusieurs. Afin de savoir qui fait quoi, git permet de stocker des informations à apposer à chaque commit de l'utilisateur.

- `git config user.name "<nom>"` : tous les commit effectués sur ce dépôt sont crédités à l'utilisateur appelé <nom> ;
- `git config --global user.name "<nom>"` : tous les commit effectués à partir de ce compte utilisateur (quel que soit le dépôt) sont crédités à l'utilisateur appelé <nom> ;
- `git config --global user.email "<email>"` : l'adresse e-mail de l'utilisateur est <email>.

### II. Créer un dossier un dépôt Git

Pour créer un dépôt git vous pouvez :

- prendre n'importe quel dossier et l'initialiser en tapant `git init` si vous êtes dans le dossier ou `git init <dossier>` ;
- cloner un dépôt git existant avec `git clone <dépôt> <directory>` (si <directory> n'est pas spécifié, c'est dans le répertoire courant que le dépôt sera cloné.)

### III. Effectuer un "commit"

Le flux de travail de base de git consiste en le fait d'effectuer des modifications puis de les grouper en ensemble cohérent avant de les stocker dans un état du code. On appelle l'action de créer une version du code un commit.

- La (sous-)commande **add** permet d'ajouter le ou les fichiers ainsi sélectionnés au prochain commit:
- `git add <fichier>` : ajoute <fichier> au prochain commit ;
- `git add <dossier>` : ajoute tout le contenu de <dossier> au prochain commit ;
- `git add -p` : ajoute tous les changements au prochain commit ;
- `git add --all <fichier>` : ajoute tous les changements à <fichier> (y compris la suppression) pour le prochain commit ;
- `git reset <fichier>` : retirer les modifications à <fichier> de la sélection pour le prochain commit.

- **git status** : permet d'avoir la liste des fichiers modifiés et leur situation par rapport au prochain commit ;
- **git log** : permet d'avoir la liste des commits pour la branche courante ;

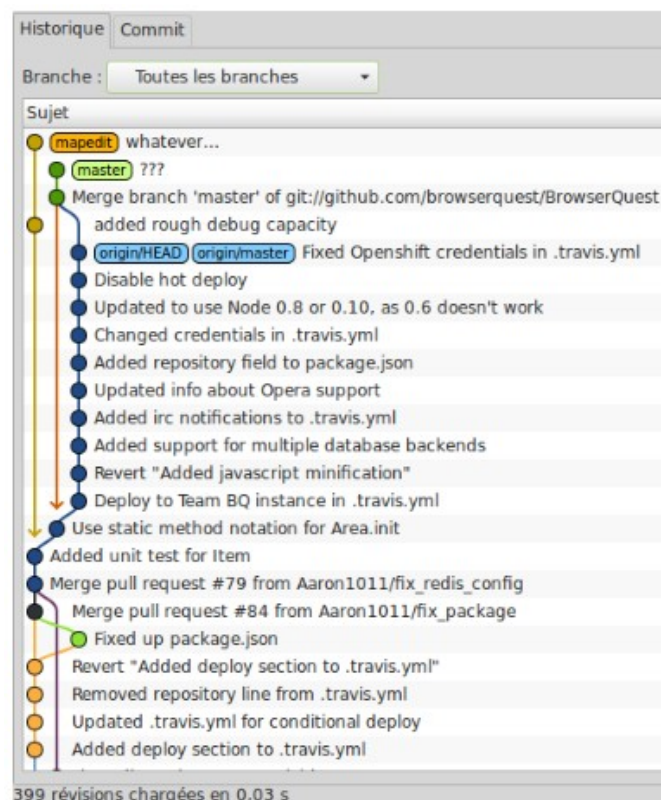
La (sous-)commande **commit** permet de créer une nouvelle version du code 2 :

- **git commit** : ouvre un éditeur de texte pour spécifier les informations du commit, qui est effectué à la fermeture de l'éditeur de texte ;
- **git commit -m "<message>"** : effectue le commit avec comme informations <message> ;
- **git commit -a** : ajoute au commit suivant toutes les modifications effectuées sur des fichiers préalablement inclus dans un commit (les créations de fichiers ne sont pas ajoutées au commit) ;
- **git commit --amend** : met à jour le dernier commit avec les changements sélectionnés.

Attention : cette commande est à utiliser avec parcimonie, surtout quand vous avez plusieurs dépôts pour le même projet. `git commit --amend` « réécrit l'historique », et ne doit être utilisé que si le commit précédent n'a pas été publié ou utilisé par ailleurs.

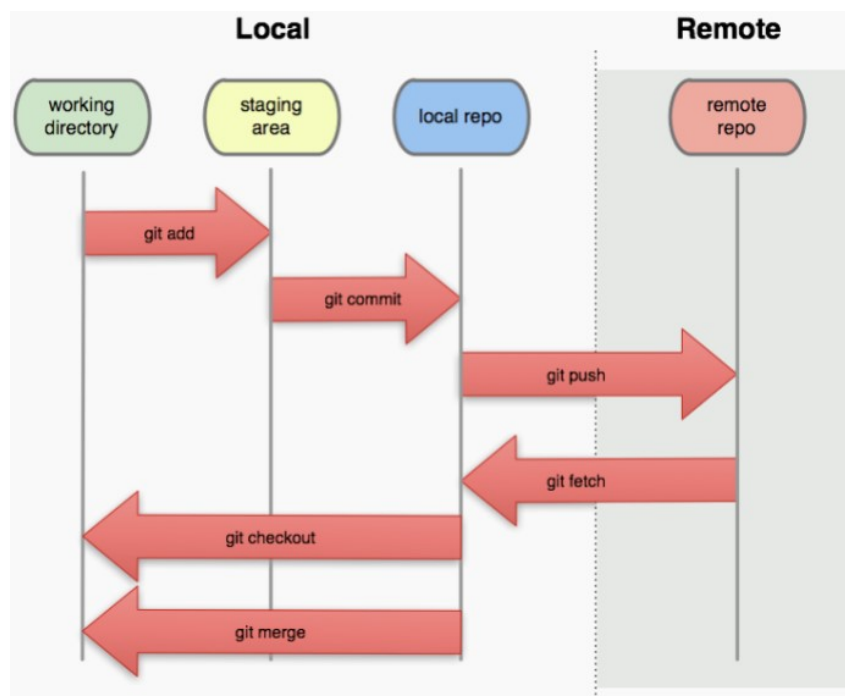
## IV. Gérer des branches

Git permet également d'expérimenter, de faire des tests sans affecter le code stable. Pour ce faire, on va avoir recours à ce que l'on appelle des « branches », qui permettront de scinder le projet en différente version. Dans la figure suivante, 4 branches sont représentées simultanément (il est à noté qu'elles sont nommées).



- **git branch** : lister toutes les branches du dépôt ;
- **git branch <nomBranche>** : créer la branche <nomBranche> (pas d'espace dans les noms de branche) ;
- **git checkout <nomBranche>** : passer sur la branche <nomBranche> (elle doit exister) ;
- **git merge <nomDUneBrancheNonManipulée>** : Fusionne la branche courante avec le contenu de la branche <nomDUneBrancheNonManipulée> ;

**NB** : Il n'est pas possible de changer de branche tant que toutes les modifications n'ont pas été soit annulées soit « committées ». Pour traiter ce genre de problématique la commande **git stash** existe, mais dépasse le cadre de cette formation. Il est bon de savoir qu'elle existe, malgré tout. De plus, fusionner des branches, peut créer des conflits, dans ce cas, il peut être bien pratique d'avoir un outil de fusion (merge tool) correctement configuré .



### # Installation de l'outil GIT, TP effectué sur Ubuntu 20.04 LTS#

```
karou@BEN:~$ sudo apt install git
```

```
karou@BEN:~$ git --help
```

```
karou@BEN MINGW64 /d/0 Maitre technologue/22-Devops/TPs
```

```
$ git --help
```

```
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
  [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
  [-p | --paginate] [-P | --no-pager] [--no-replace-objects] [--bare]
  [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
  [--config-env=<name>=<envvar>] <command> [<args>]
```

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)

clone	Clone a repository into a new directory
init	Create an empty Git repository or reinitialize an existing one
work on the	current change (see also: git help everyday)
add	Add file contents to the index
mv	Move or rename a file, a directory, or a symlink
restore	Restore working tree files
rm	Remove files from the working tree and from the index
examine the	history and state (see also: git help revisions)
bisect	Use binary search to find the commit that introduced a bug
diff	Show changes between commits, commit and working tree, etc
grep	Print lines matching a pattern
log	Show commit logs
show	Show various types of objects
status	Show the working tree status
grow, mark	and tweak your common history
branch	List, create, or delete branches
commit	Record changes to the repository
merge	Join two or more development histories together
rebase	Reapply commits on top of another base tip
reset	Reset current HEAD to the specified state
switch	Switch branches
tag	Create, list, delete or verify a tag object signed with GPG
collaborate	(see also: git help workflows)
fetch	Download objects and refs from another repository
pull	Fetch from and integrate with another repository or a local branch
push	Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some concept guides. See 'git help <command>' or 'git help <concept>' to read about a specific subcommand or concept.  
See 'git help git' for an overview of the system.

### # Configuration des paramètres global sur GIT #

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs
$ git config --global user.name "Karim Ben Romdhane"

karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs
$ git config --global user.email "karoumbr@gmail.com"
```

### # Working Directory : Zone de travail Local #

#### # Création d'un projet GIT #

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs
$ mkdir gitapp && cd gitapp
```

#### # creation de quelques fichiers #

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp
$ touch file1 file2

karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp
$ echo "first line" >> file1

karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp
$ cat file1
first line
```

#### # Initiation d'un projet git vide #

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp
```

```
$ ls -la
```

```
total 1
drwxr-xr-x 1 karou 197121 0 Oct 1 16:17 ./
drwxr-xr-x 1 karou 197121 0 Oct 1 16:16 ../
-rw-r--r-- 1 karou 197121 11 Oct 1 16:17 file1
-rw-r--r-- 1 karou 197121 0 Oct 1 16:17 file2
```

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp
```

```
$ git init
```

```
Initialized empty Git repository in D:/0 Maitre technologie/22-Devops/TPs/gitapp/.git/
```

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp (master)
```

```
$ ls -la
```

```
total 5
drwxr-xr-x 1 karou 197121 0 Oct 1 16:23 ./
drwxr-xr-x 1 karou 197121 0 Oct 1 16:16 ../
drwxr-xr-x 1 karou 197121 0 Oct 1 16:23 .git/
-rw-r--r-- 1 karou 197121 11 Oct 1 16:17 file1
-rw-r--r-- 1 karou 197121 0 Oct 1 16:17 file2
```

**#création d'un répertoire caché .git. Quel est son rôle ? #**

**# Indexation des fichiers dans gitapp (staging area)#**

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp (master)
```

```
$ git add file1
```

```
warning: in the working copy of 'file1', LF will be replaced by CRLF the next time
Git touches it
```

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp (master)
```

```
$ git status
```

```
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   file1
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
file2
```

**#### Annuler l'indexation si l'on désire ####**

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp (master)
```

```
$ git rm --cached file1
```

```
rm 'file1'
```

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp (master)
```

```
$ git status
```

```
On branch master
```

```
No commits yet
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
file1
```

```
file2
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp (master)
```

```
$ git add . #le point indique all#
```

```
warning: in the working copy of 'file1', LF will be replaced by CRLF the next time  
git touches it
```

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp (master)
```

```
$ git status
```

```
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   file1
```

```
new file:   file2
```

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp (master)
```

```
$ touch file3
```

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp (master)
```

```
$ git status
```

```
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   file1
```

```
new file:   file2
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
file3
```

### ##### faire un snapshot ou instantané (commit) en indiquant un message #####

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp (master)
```

```
$ git commit -m "file1 et file2 added"
```

```
[master (root-commit) 097b744] file1 et file2 added
```

```
2 files changed, 1 insertion(+)
```

```
create mode 100644 file1
```

```
create mode 100644 file2
```

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp (master)
```

```
$ git log
```

```
commit 097b744404316c2ca9effd996d7ea3da3dc83419 (HEAD -> master)
```

```
Author: Karim Ben Romdhane <karoumbr@gmail.com>
```

```
Date: Sun Oct 1 16:36:25 2023 +0100
```

```
file1 et file2 added
```

#Ici on remarque un seul commit représenter par un hashcode qui commence par  
28bba39 #

### # Modification de file1 #####

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp (master)
```

```
$ echo "second line" >> file1
```

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp (master)
```

```
$ cat file1
```

```
first line
```

```
second line
```

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp (master)
```

```
$ git status
```

```
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   file1

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file3

no changes added to commit (use "git add" and/or "git commit -a")
```

### # Revenir à l'état précédent #####

```
karou@BEN MINGW64 /d/0 Maitre technologue/22-Devops/TPs/gitapp (master)
$ git restore file1
```

```
karou@BEN MINGW64 /d/0 Maitre technologue/22-Devops/TPs/gitapp (master)
$ git status
```

```
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file3

nothing added to commit but untracked files present (use "git add" to track)
```

```
karou@BEN MINGW64 /d/0 Maitre technologue/22-Devops/TPs/gitapp (master)
$ cat file1
first line
```

```
karou@BEN MINGW64 /d/0 Maitre technologue/22-Devops/TPs/gitapp (master)
$ git log
```

```
commit 097b744404316c2ca9effd996d7ea3da3dc83419 (HEAD -> master)
Author: Karim Ben Romdhane <karoumbr@gmail.com>
Date:   Sun Oct 1 16:36:25 2023 +0100

    file1 et file2 added
```

```
karou@BEN MINGW64 /d/0 Maitre technologue/22-Devops/TPs/gitapp (master)
$ git add file3
```

```
karou@BEN MINGW64 /d/0 Maitre technologue/22-Devops/TPs/gitapp (master)
$ git commit -m "file3 added"
```

```
[master cd94ba9] file3 added
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file3
```

### #Ici on remarque deux commit avec deux hashcodes différents #

```
karou@BEN MINGW64 /d/0 Maitre technologue/22-Devops/TPs/gitapp (master)
$ git log
```

```
commit cd94ba971e4d4069e4179d581699d480b9a91e70 (HEAD -> master)
Author: Karim Ben Romdhane <karoumbr@gmail.com>
Date:   Sun Oct 1 16:55:54 2023 +0100

    file3 added #second commit#

commit 097b744404316c2ca9effd996d7ea3da3dc83419
Author: Karim Ben Romdhane <karoumbr@gmail.com>
Date:   Sun Oct 1 16:36:25 2023 +0100

    file1 et file2 added #first commit#
```

```
karou@BEN MINGW64 /d/0 Maitre technologue/22-Devops/TPs/gitapp (master)
$ touch file4 file5
```

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp (master)
$ git add .
```

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp (master)
$ git commit -m "rest of files added"
```

```
[master d258c2a] rest of files added
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file4
create mode 100644 file5
```

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp (master)
$ git log
```

```
commit d258c2aefef3afea21bbb0266fd171a44da664c4 (HEAD -> master)
Author: Karim Ben Romdhane <karoumbr@gmail.com>
Date: Sun Oct 1 16:59:28 2023 +0100

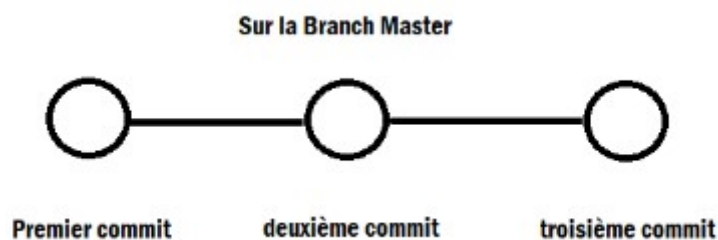
    rest of files added #third commit#

commit cd94ba971e4d4069e4179d581699d480b9a91e70
Author: Karim Ben Romdhane <karoumbr@gmail.com>
Date: Sun Oct 1 16:55:54 2023 +0100

    file3 added #second commit#

commit 097b744404316c2ca9effd996d7ea3da3dc83419
Author: Karim Ben Romdhane <karoumbr@gmail.com>
Date: Sun Oct 1 16:36:25 2023 +0100

    file1 et file2 added #first commit#
```



**# voici une représentation graphique de nos commits sur une seule branche qui est la branche Master #**

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp (master)
$ git status
```

```
On branch master
nothing to commit, working tree clean
```

**####Création de gitApp sur un dépôt distant (ex:Github)###**


**# Prérequis : Allez sur Github et créer un compte #**

```
$ git branch
* master #je m'assure que je suis sur la branche Master#
```

**# voici quelques captures pour vous aider à créer un repository sur Github#**

<https://github.com/login>





Sign in to GitHub

Username or email address

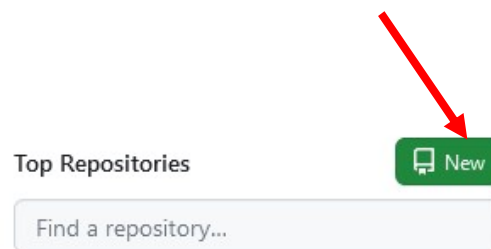
Password [Forgot password?](#)

[Sign in](#)

[Sign in with a passkey](#)  
New to GitHub? [Create an account](#)

[Terms](#) [Privacy](#) [Docs](#) [Contact GitHub Support](#)

Figure 1: Se connecter avec le compte





Click sur « New » afin de créer un nouveau dépôt distant(Remote Repository)

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).



Owner \* Repository name \*

 karoumbr / gitApp 

✔ gitApp is available.

Great repository names are short and memorable. Need inspiration? How about [reimagined-disco](#) ?

Description (optional)

- ☒  **Public**  
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**  
You choose who can see and commit to this repository.

Initialize this repository with:

-  ☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: **None** ▼

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: **None** ▼

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account. 

Create repository

Création d'un repository : gitapp

## Repository default branch




Choose the default branch for your new personal repositories. You might want to change the default name due to different workflows, or because your integrations still require "master" as the default branch name. You can always change the default branch name on individual repositories. [Learn more about default branches.](#)

main



## Repository default branch

Choose the default branch for your new personal repositories. You might want to change the default name due to different workflows, or because your integrations still require "master" as the default branch name. You can always change the default branch name on individual repositories. [Learn more about default branches.](#)

 master   Update




New repositories you create will use 'master' as their default branch name.

Figure 2: Changement de Branche par défaut de main à Master

## Create a new repository



A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \*  Repository name \*  
 karoumbr / gitApp  
 gitApp is available.

Great repository names are short and memorable. Need inspiration? How about [redesigned-couscous](#) ?

Description (optional)

- ☒  **Public**  
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**  
You choose who can see and commit to this repository.

Initialize this repository with:

- ☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

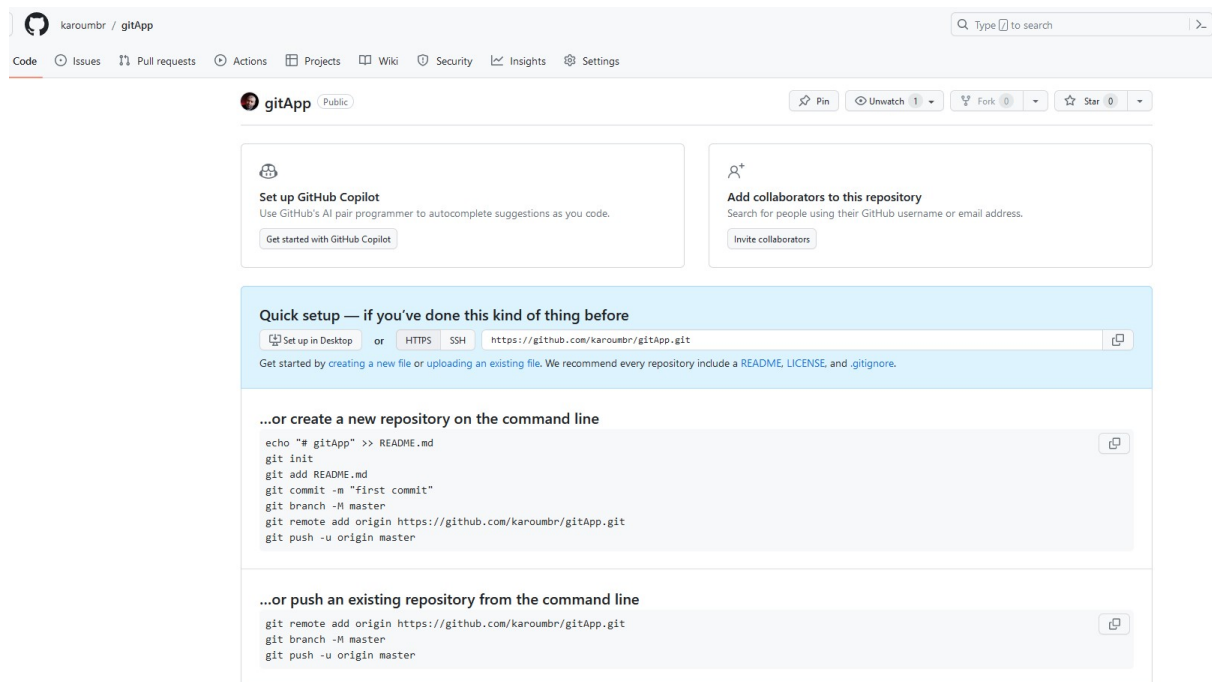
Choose a license

License: None

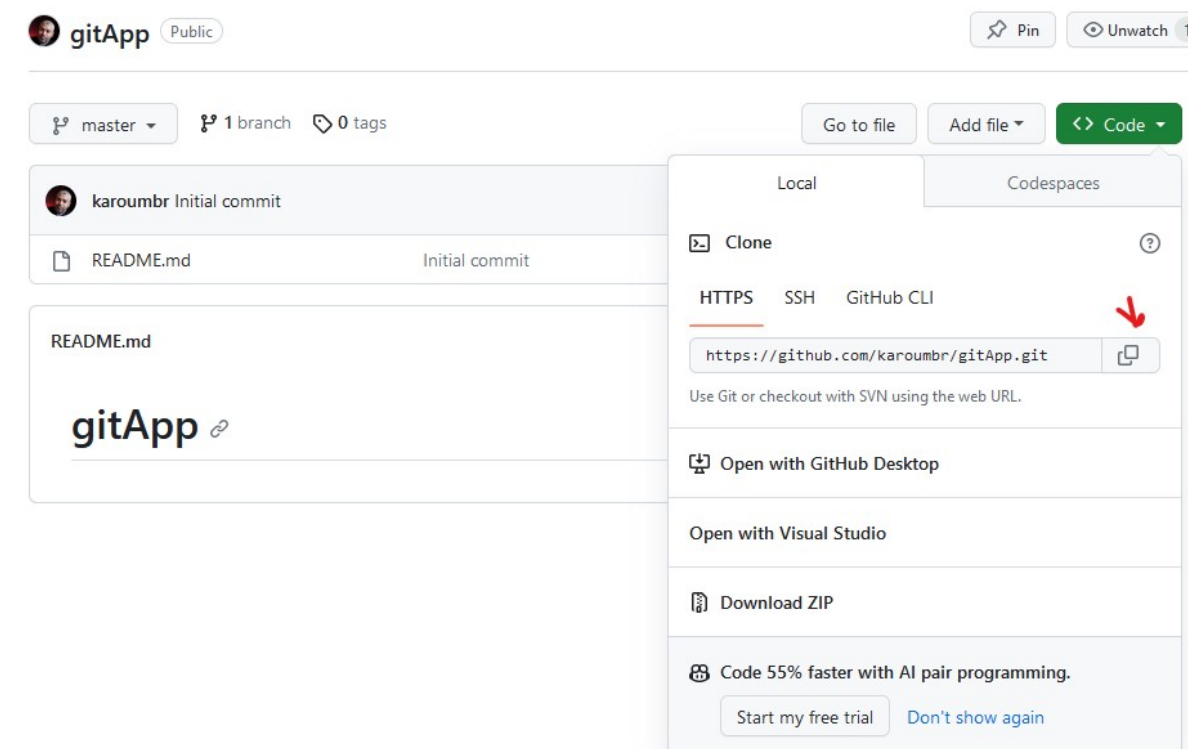
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

 You are creating a public repository in your personal account.

Create repository



Vérifier la bonne branche et récupérer l'URL



### #On va utiliser l'URL pour faire le push vers Github (mettre le dernier commit) ###

```
karou@BEN MINGW64 /d/0 Maitre technologue/22-Devops/TPs/gitapp (master)
$ git remote add origin https://github.com/karoumbr/gitApp.git
```

```
karou@BEN MINGW64 /d/0 Maitre technologue/22-Devops/TPs/gitapp (master)
$ git push -u origin master
```

```
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 628 bytes | 628.00 KiB/s, done.
Total 8 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/karoumbr/gitApp.git
* [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

gitApp Public

Pin Unwatch 1 Fork 0 Star 0

master 1 branch 0 tags

Go to file Add file <> Code

About No description, website, or topics provided.

Activity 0 stars 1 watching 0 forks

Releases No releases published Create a new release

Packages No packages published Publish your first package

karoumbr rest of files added d258c2a 23 minutes ago 3 commits

File	Commit Message	Time Ago
file1	file1 et file2 added	46 minutes ago
file2	file1 et file2 added	46 minutes ago
file3	file3 added	27 minutes ago
file4	rest of files added	23 minutes ago
file5	rest of files added	23 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README

Figure 3: Visualisation de mes fichiers sur gitApp dans Github

Pour consulter la liste des commits :

gitApp Public

Pin Unwatch 1

master 1 branch 0 tags

Go to file Add file <> Code

About No description, website, or topics provided.

Activity 0 stars 1 watching 0 forks

Releases No releases published Create a new release

Packages No packages published Publish your first package

karoumbr rest of files added d258c2a 25 minutes ago 3 commits

File	Commit Message	Time Ago
file1	file1 et file2 added	48 minutes ago
file2	file1 et file2 added	48 minutes ago
file3	file3 added	29 minutes ago
file4	rest of files added	25 minutes ago
file5	rest of files added	25 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README

## Commits

master

Commits on Oct 1, 2023

rest of files added karoumbr committed 25 minutes ago	d258c2a
file3 added karoumbr committed 29 minutes ago	cd94ba9
file1 et file2 added karoumbr committed 49 minutes ago	097b744

### #Utiliser le repository gitapp (cloner) avec un autre utilisateur afin de travailler en équipe par exemple #

Cloner le dépôt distant gitapp sur un autre pc (ou dans un autre répertoire pour le test):

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/gitapp (master)
$ cd ..
```

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs
$ mkdir myGitClone && cd myGitClone
```

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/myGitClone
$ git clone https://github.com/karoumbr/gitApp.git
```

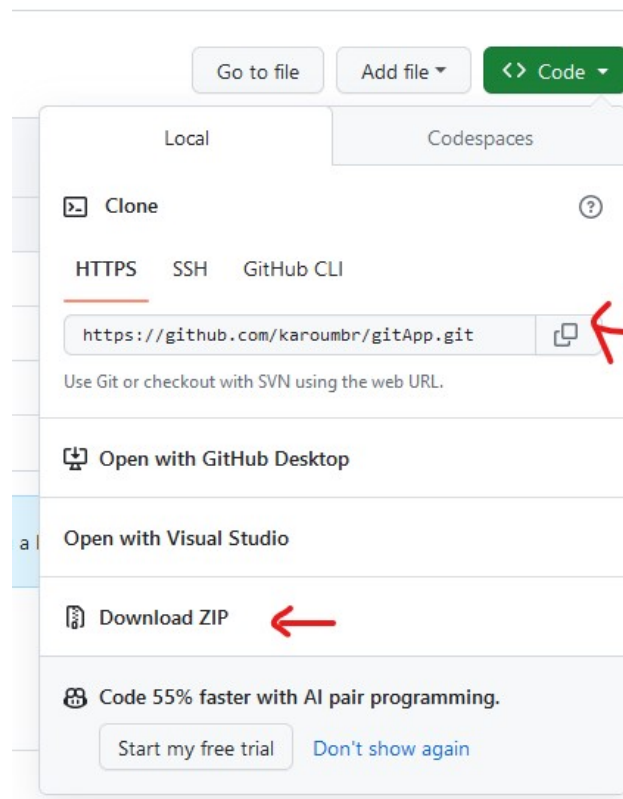
```
Cloning into 'gitApp'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 8 (delta 2), reused 8 (delta 2), pack-reused 0
Receiving objects: 100% (8/8), done.
Resolving deltas: 100% (2/2), done.
```

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/myGitClone
$ ls -la
```

```
total 0
drwxr-xr-x 1 karou 197121 0 Oct 1 17:44 ./
drwxr-xr-x 1 karou 197121 0 Oct 1 17:44 ../
drwxr-xr-x 1 karou 197121 0 Oct 1 17:44 gitApp/
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/myGitClone
$ cd gitApp/
```

```
karou@BEN MINGW64 /d/0 Maitre technologie/22-Devops/TPs/myGitClone/gitApp (master)
$ ls -la
```

```
total 5
drwxr-xr-x 1 karou 197121 0 Oct 1 17:44 ./
drwxr-xr-x 1 karou 197121 0 Oct 1 17:44 ../
drwxr-xr-x 1 karou 197121 0 Oct 1 17:44 .git/
-rw-r--r-- 1 karou 197121 12 Oct 1 17:44 file1
-rw-r--r-- 1 karou 197121 0 Oct 1 17:44 file2
-rw-r--r-- 1 karou 197121 0 Oct 1 17:44 file3
-rw-r--r-- 1 karou 197121 0 Oct 1 17:44 file4
-rw-r--r-- 1 karou 197121 0 Oct 1 17:44 file5
```



Ou télécharger le projet en cliquant sur « Download ZIP ».