

Nous avons commencé par ouvrir git bash, nous avons ensuite rentrer les différentes commandes pour lier notre dossier sur notre ordinateur à notre compte github en ligne

```
$ git clone https://HugosSeulin@github.com/HugosSeulin/Hugo-Seulin.git
```

```
$ cd Hugo-Seulin/
```

```
$ git status
```

```
$ls
```

```
MINGW64:/c/Users/hugos/coursprog/Hugo-Seulin

hugos@PC-Hugo MINGW64 ~
$ git status
fatal: not a git repository (or any of the parent directories): .git

hugos@PC-Hugo MINGW64 ~
$ ls
-1.14-windows.xml
'3D Objects' /
AppData/
'Application Data'@
Contacts/
Cookies@
'Creative Cloud Files' /
Documents/
Downloads/
Favorites/
IntelGraphicsProfiles/
Links/
'Local Settings'@
'Menu Démarrer'@
'Mes documents'@
Modèles@
Music/
NTUSER.DAT
NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TM.blf
NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TMContainer00000000000000000000
1.regtrans-ms
NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TMContainer00000000000000000000
2.regtrans-ms
OneDrive/
Pictures/
Recent@
'Saved Games' /
Searches/
SendTo@
Videos/
'Voisinage d'impression'@
'Voisinage réseau'@
ansel/
coursprog/
curseforge/
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini

hugos@PC-Hugo MINGW64 ~
$ git clone
fatal: You must specify a repository to clone.

usage: git clone [<options>] [--] <repo> [<dir>]

    -v, --verbose          be more verbose
    -q, --quiet            be more quiet
    --progress            force progress reporting
    --reject-shallow      don't clone shallow repository
    -n, --no-checkout     don't create a checkout
```

```

hugos@PC-Hugo MINGW64 ~
$ git clone
fatal: You must specify a repository to clone.

usage: git clone [<options>] [--] <repo> [<dir>]

-v, --verbose          be more verbose
-q, --quiet            be more quiet
--progress            force progress reporting
--reject-shallow      don't clone shallow repository
-n, --no-checkout      don't create a checkout
--bare                create a bare repository
--mirror              create a mirror repository (implies bare)
-l, --local            to clone from a local repository
--no-hardlinks        don't use local hardlinks, always copy
-s, --shared           setup as shared repository
--recurse-submodules[=<pathspec>]
                        initialize submodules in the clone
--recursive ...       alias of --recurse-submodules
-j, --jobs <n>        number of submodules cloned in parallel
--template <template-directory>
                        directory from which templates will be used
--reference <repo>    reference repository
--reference-if-able <repo>
                        reference repository
                        use --reference only while cloning
-o, --origin <name>   use <name> instead of 'origin' to track upstream
-b, --branch <branch>
                        checkout <branch> instead of the remote's HEAD
-u, --upload-pack <path>
                        path to git-upload-pack on the remote
--depth <depth>       create a shallow clone of that depth
--shallow-since <time>
                        create a shallow clone since a specific time
--shallow-exclude <revision>
                        deepen history of shallow clone, excluding rev
--single-branch        clone only one branch, HEAD or --branch
--no-tags              don't clone any tags, and make later fetches not to follow them
--shallow-submodules   any cloned submodules will be shallow
--separate-git-dir <gitdir>
                        separate git dir from working tree
-c, --config <key=value>
                        set config inside the new repository
--server-option <server-specific>
                        option to transmit
-4, --ipv4             use IPv4 addresses only
-6, --ipv6             use IPv6 addresses only
--filter <args>        object filtering
--also-filter-submodules
                        apply partial clone filters to submodules
--remote-submodules    any cloned submodules will use their remote-tracking branch
--sparse              initialize sparse-checkout file to include only files at root
--bundle-uri <uri>     a URI for downloading bundles before fetching from origin remote

```

hugos@PC-Hugo MINGW64 ~

```

hugos@PC-Hugo MINGW64 ~
$ git clone https://HugosSeulin@github.com/HugosSeulin/Hugo-Seulin.git
Cloning into 'Hugo-Seulin'...
warning: You appear to have cloned an empty repository.

hugos@PC-Hugo MINGW64 ~
$ cd Hugo-Seulin/

hugos@PC-Hugo MINGW64 ~/Hugo-Seulin (main)
$ git status
On branch main

No commits yet

nothing to commit (create/copy files and use "git add" to track)

hugos@PC-Hugo MINGW64 ~/Hugo-Seulin (main)
$ pwd
/c/Users/hugos/Hugo-Seulin

hugos@PC-Hugo MINGW64 ~/Hugo-Seulin (main)
$ cd ..

hugos@PC-Hugo MINGW64 ~
$ cd c
Contacts/          Cookies/          Creative Cloud Files/ coursprog/          curseforge/

hugos@PC-Hugo MINGW64 ~
$ cd c
Contacts/          Cookies/          Creative Cloud Files/ coursprog/          curseforge/

hugos@PC-Hugo MINGW64 ~
$ cd coursprog/
.git/              Hugo-Seulin/  README.md

hugos@PC-Hugo MINGW64 ~
$ cd coursprog/
.git/              Hugo-Seulin/  README.md

hugos@PC-Hugo MINGW64 ~
$ cd coursprog/Hugo-Seulin/

hugos@PC-Hugo MINGW64 ~/coursprog/Hugo-Seulin (main)
$ pwd
/c/Users/hugos/coursprog/Hugo-Seulin
*
hugos@PC-Hugo MINGW64 ~/coursprog/Hugo-Seulin (main)
$ *

```

Sur la capture au-dessus, nous avons dû résoudre un problème avec un de mes dossiers car il y avait eu un double clonage ce qui posait problème pour lier les deux choses.

Nous savons donc utilisés les commandes :

```

$ pwd
$ cd ..
$ cd c
$ cd c
$ cd coursprog/
$ cd coursprog/Hugo-Seulin/
$ pwd

```

```

hugos@PC-Hugo MINGW64 ~/coursprog/Hugo-Seulin (main)
$ cd Desktop/
bash: cd: Desktop/: No such file or directory

hugos@PC-Hugo MINGW64 ~/coursprog/Hugo-Seulin (main)
$ git add .AC

hugos@PC-Hugo MINGW64 ~/coursprog/Hugo-Seulin (main)
$ git add .

hugos@PC-Hugo MINGW64 ~/coursprog/Hugo-Seulin (main)
$ git commit -m
error: switch 'm' requires a value

hugos@PC-Hugo MINGW64 ~/coursprog/Hugo-Seulin (main)
$ git commit -m "ajoute fichier rendu"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'hugos@PC-Hugo.(none)')

hugos@PC-Hugo MINGW64 ~/coursprog/Hugo-Seulin (main)
$ git config --global user.email "hugo.seulin1@gmail.com"

hugos@PC-Hugo MINGW64 ~/coursprog/Hugo-Seulin (main)
$ git commit -m "ajoute fichier rendu"
[main (root-commit) 2e72959] ajoute fichier rendu
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Koro Sensei.jpg

hugos@PC-Hugo MINGW64 ~/coursprog/Hugo-Seulin (main)
$ |

```

```

$ cd Desktop/
$ git add .
$ git commit -m "ajoute fichier rendu"
$ git push

```

Une fois cela terminé, nos deux dossiers sont liés et les documents déposés sur le dossier pc sont transférés sur github.