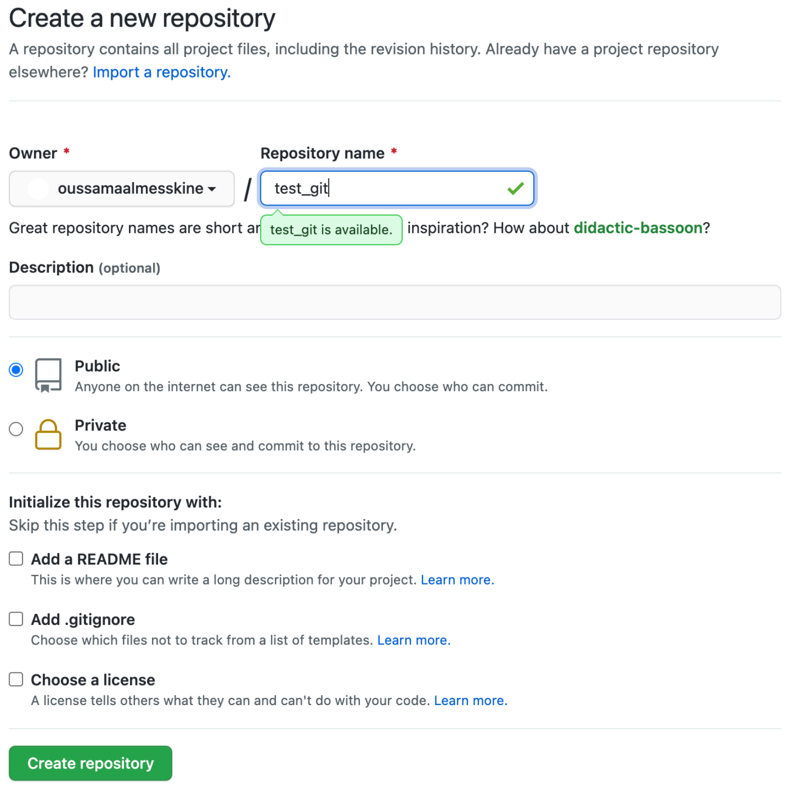
Workshop2: “Git & Github”

1)



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * **Repository name** *

oussamaalmesskine ▾ / test_git ✓

Great repository names are short ar test_git is available. inspiration? How about **didactic-bassoon**?

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:
Skip this step if you’re importing an existing repository.

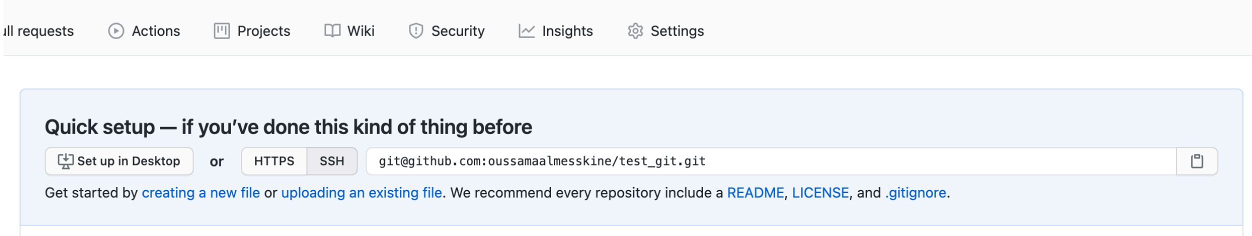
□ **Add a README file**
This is where you can write a long description for your project. Learn more.

□ **Add .gitignore**
Choose which files not to track from a list of templates. Learn more.

□ **Choose a license**
A license tells others what they can and can't do with your code. Learn more.

Create repository

2)

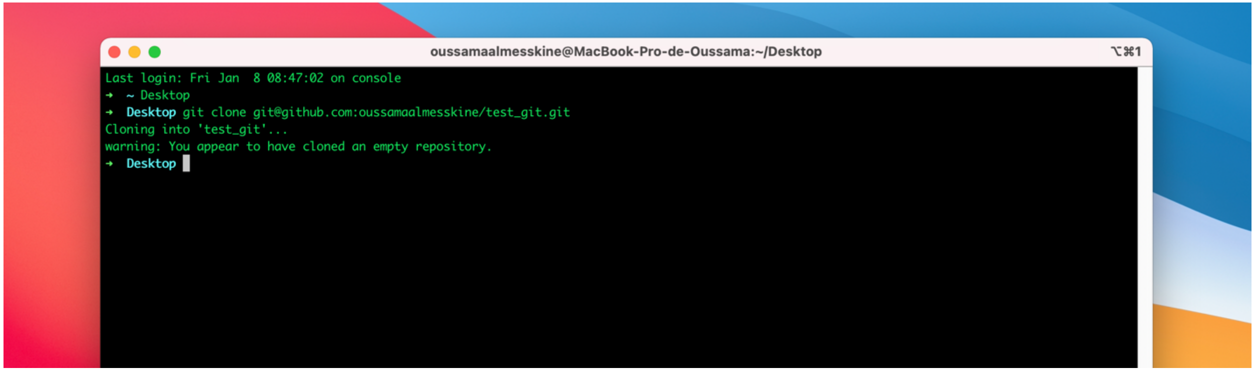


Quick setup — if you’ve done this kind of thing before

Set up in Desktop or HTTPS SSH `git@github.com:oussamaalmesskine/test_git.git`

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

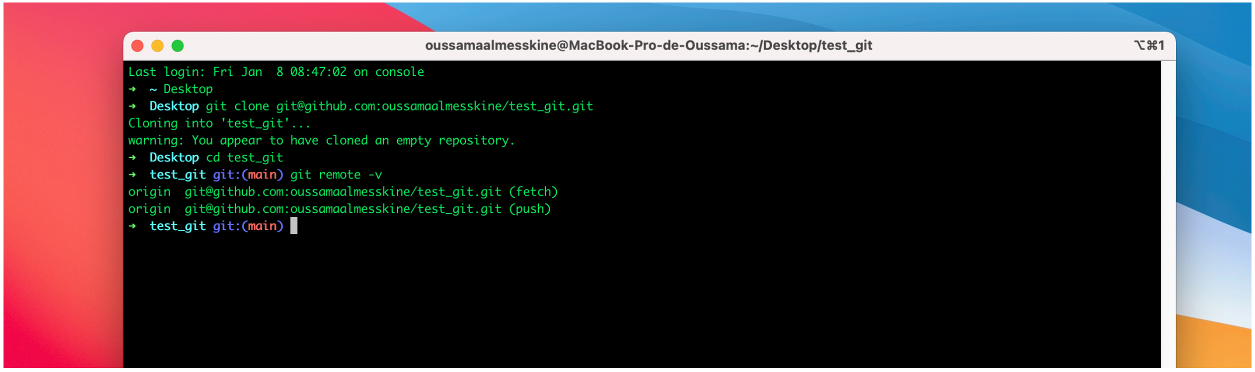
3)



oussamaalmesskine@MacBook-Pro-de-Oussama:~/Desktop ⌥⌘1

```
Last login: Fri Jan  8 08:47:02 on console
➜  ~ Desktop
➜  Desktop git clone git@github.com:oussamaalmesskine/test_git.git
Cloning into 'test_git'...
warning: You appear to have cloned an empty repository.
➜  Desktop
```

4)



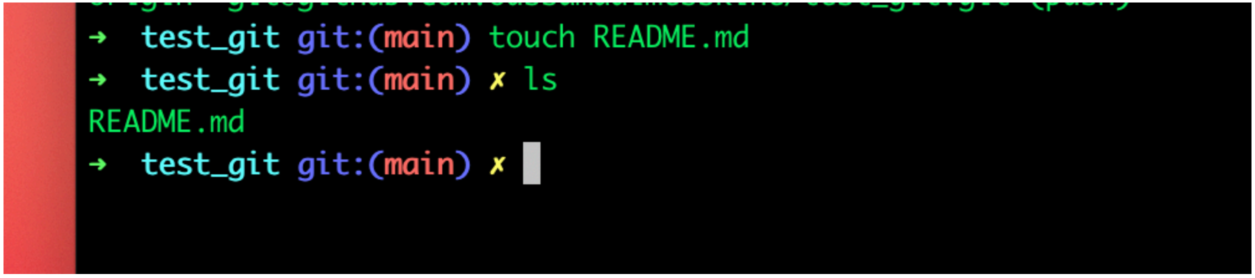
```
Last login: Fri Jan  8 08:47:02 on console
➜  ~ Desktop
➜  Desktop git clone git@github.com:oussamaalmesskine/test_git.git
Cloning into 'test_git'...
warning: You appear to have cloned an empty repository.
➜  Desktop cd test_git
➜  test_git git:(main) git remote -v
origin  git@github.com:oussamaalmesskine/test_git.git (fetch)
origin  git@github.com:oussamaalmesskine/test_git.git (push)
➜  test_git git:(main)
```

5)

-With "`git push`", you will send all the commits you've made in your local repository to GitHub.

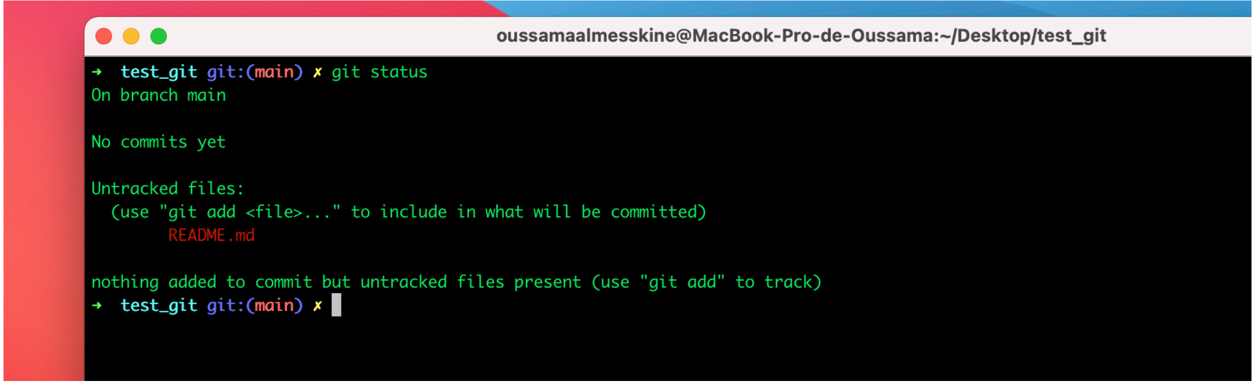
- The git fetch command downloads commits, files, and refs from a remote repository into your local repo

6)



```
➜  test_git git:(main) touch README.md
➜  test_git git:(main) ✗ ls
README.md
➜  test_git git:(main) ✗
```

7)

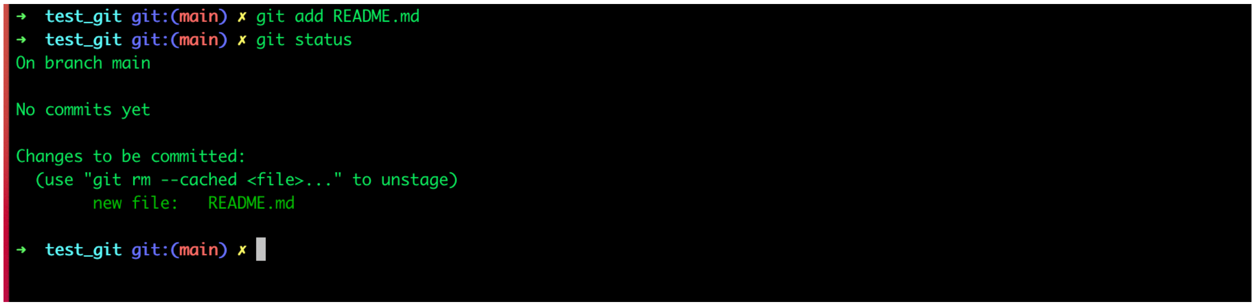


```
➜  test_git git:(main) ✗ git status
On branch main

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
➜  test_git git:(main) ✗
```

En faisant git status, on devrait voir les fichiers qu'on a modifiés en rouge. Cela signifie qu'ils ne seront pas pris en compte lorsque vous allez faire un commit.

Il faut explicitement préciser les fichiers qu'on voulait « commiter »

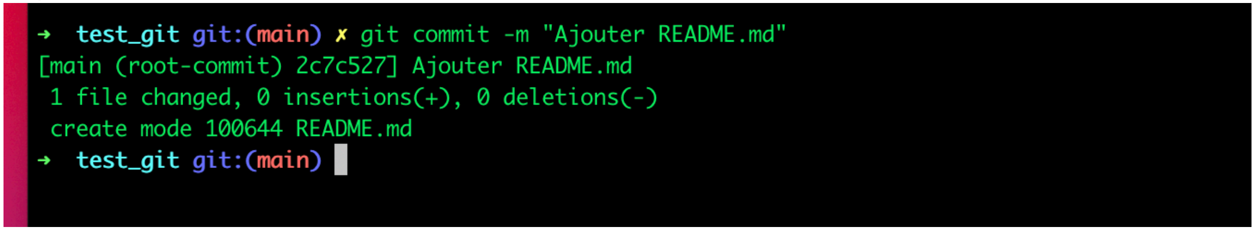
8/9)



```
➜  test_git git:(main) ✗ git add README.md
➜  test_git git:(main) ✗ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md

➜  test_git git:(main) ✗ 
```

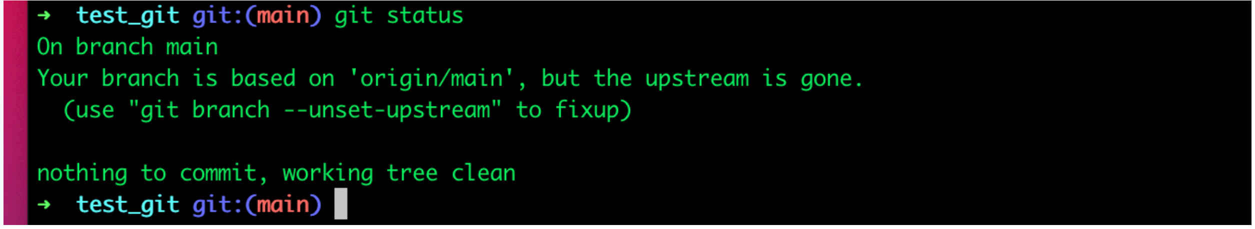
En faisant git status, README.md ils sont pris en compte maintenant

10)



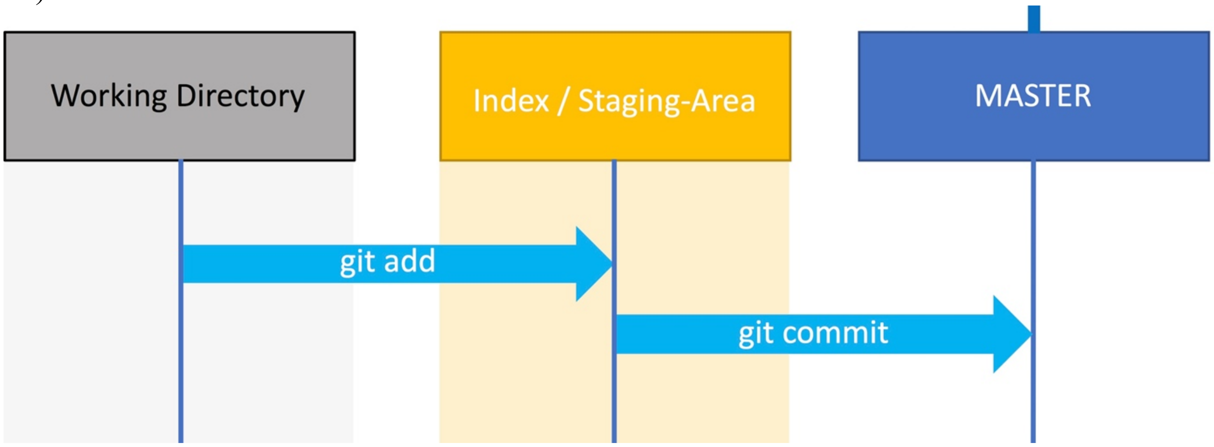
```
➜  test_git git:(main) ✗ git commit -m "Ajouter README.md"
[main (root-commit) 2c7c527] Ajouter README.md
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
➜  test_git git:(main) 
```

11)

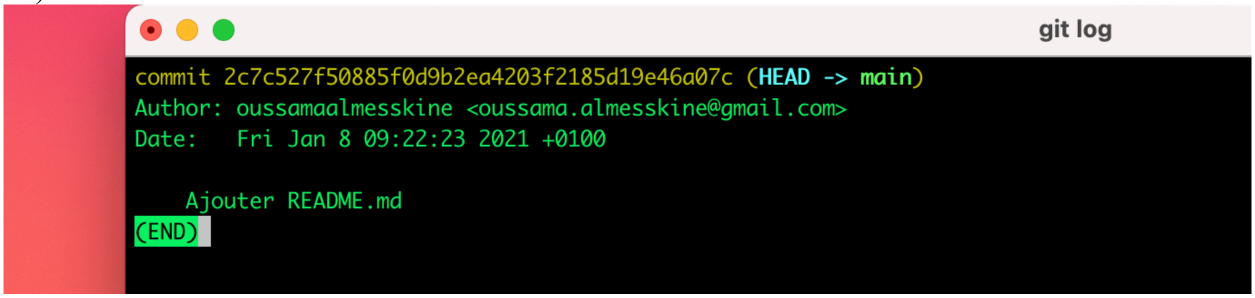


```
➜  test_git git:(main) git status
On branch main
Your branch is based on 'origin/main', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

nothing to commit, working tree clean
➜  test_git git:(main) 
```

12)



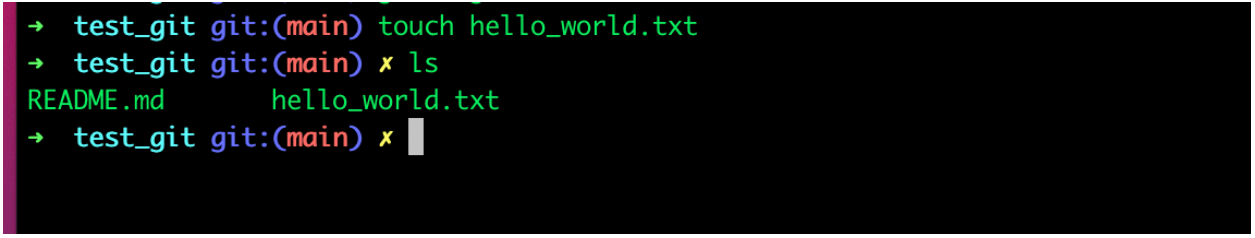
13)



```
commit 2c7c527f50885f0d9b2ea4203f2185d19e46a07c (HEAD -> main)
Author: oussamaalmesskine <oussama.almesskine@gmail.com>
Date:   Fri Jan 8 09:22:23 2021 +0100

    Ajouter README.md
(END)
```

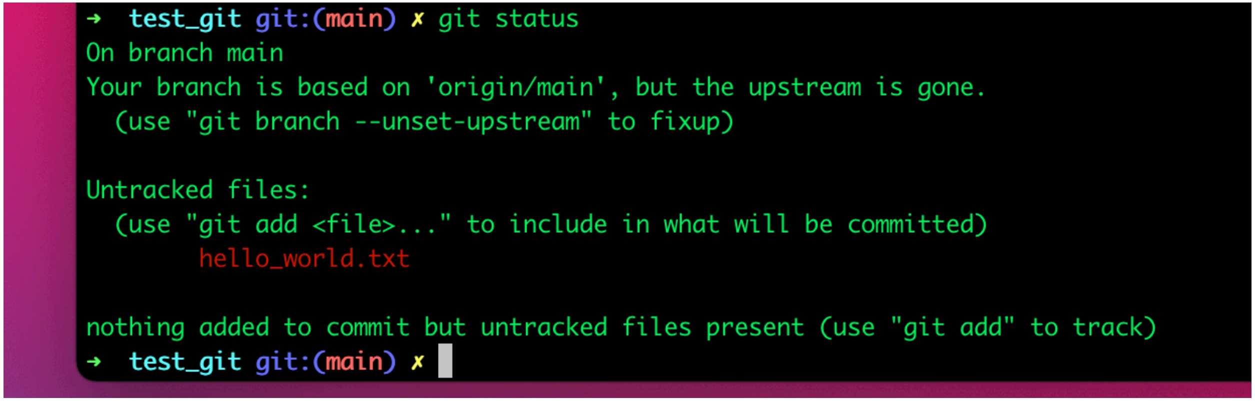
git log: consulter l'historique des commits ou on retrouve tout ce qui a été changé depuis les débuts du projet.

14)



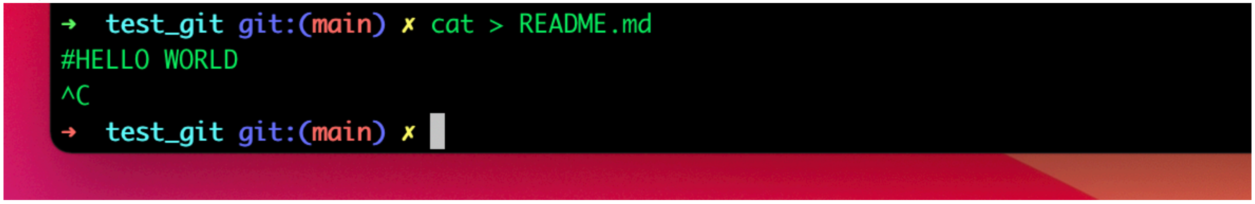
```
➜  test_git git:(main) touch hello_world.txt
➜  test_git git:(main) ✗ ls
README.md       hello_world.txt
➜  test_git git:(main) ✗
```

15)



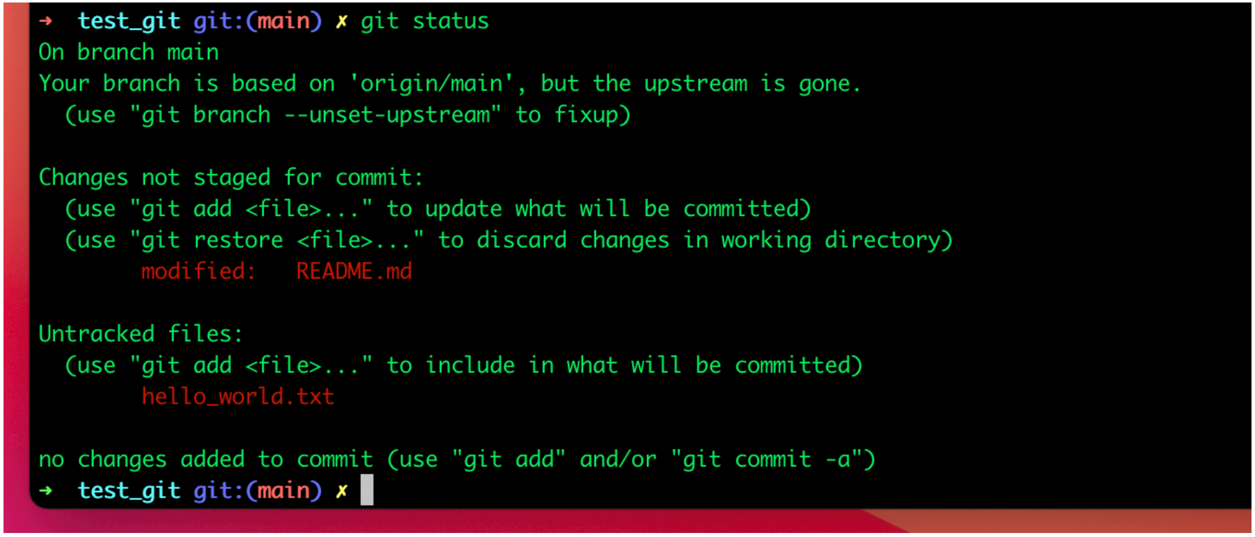
```
➜  test_git git:(main) ✗ git status
On branch main
Your branch is based on 'origin/main', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

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

nothing added to commit but untracked files present (use "git add" to track)
➜  test_git git:(main) ✗
```

16)



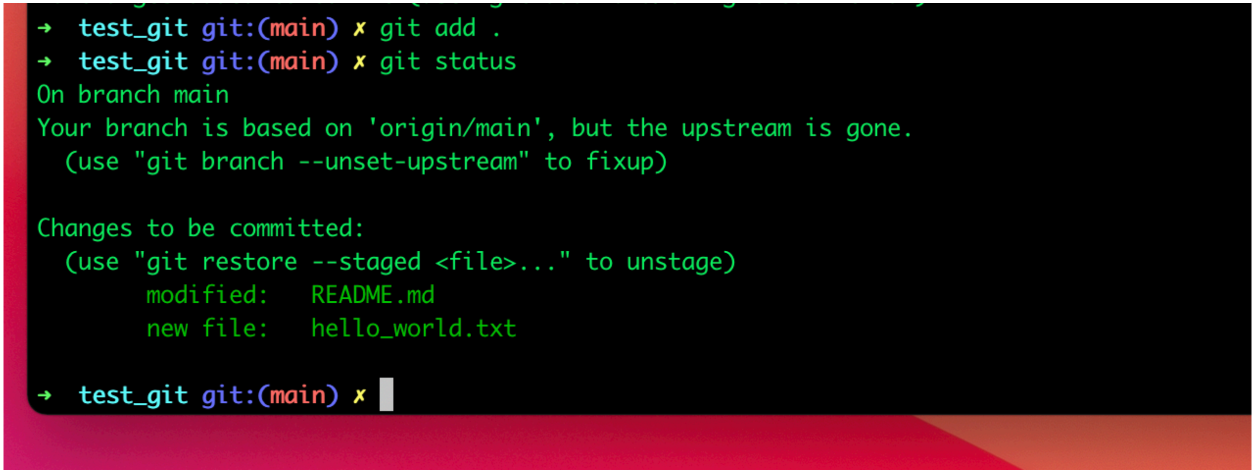
```
➜  test_git git:(main) ✗ cat > README.md
#HELLO WORLD
^C
➜  test_git git:(main) ✗
```

17)



```
➜  test_git git:(main) ✗ git status
On branch main
Your branch is based on 'origin/main', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

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:   README.md

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

no changes added to commit (use "git add" and/or "git commit -a")
➜  test_git git:(main) ✗ 
```

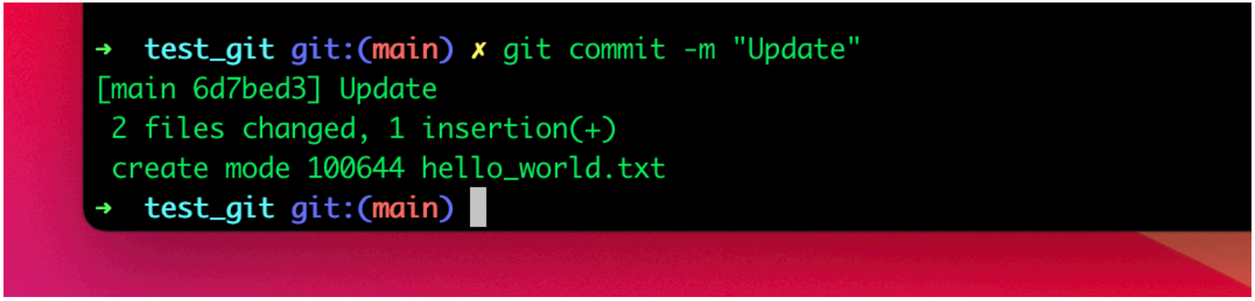
18/19)



```
➜  test_git git:(main) ✗ git add .
➜  test_git git:(main) ✗ git status
On branch main
Your branch is based on 'origin/main', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md
        new file:   hello_world.txt

➜  test_git git:(main) ✗ 
```

With the git add command, the files that you changed will be ready to commit.
And with the git add [file] we commit a specific file

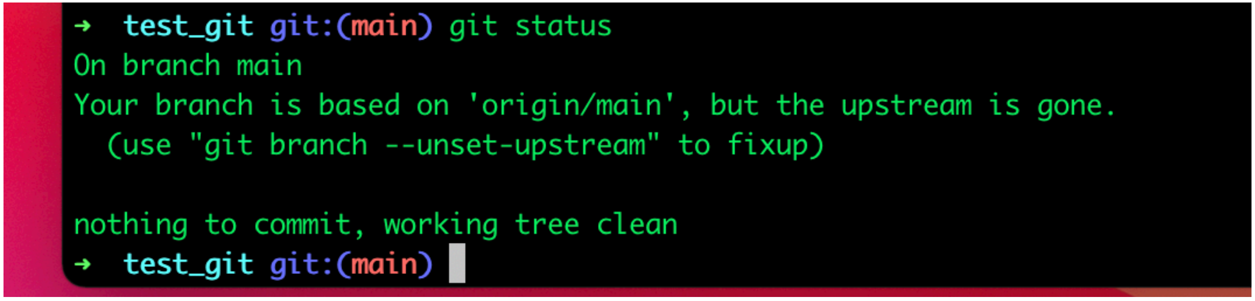
20)



```
➜  test_git git:(main) ✗ git commit -m "Update"
[main 6d7bed3] Update
 2 files changed, 1 insertion(+)
 create mode 100644 hello_world.txt
➜  test_git git:(main) 
```

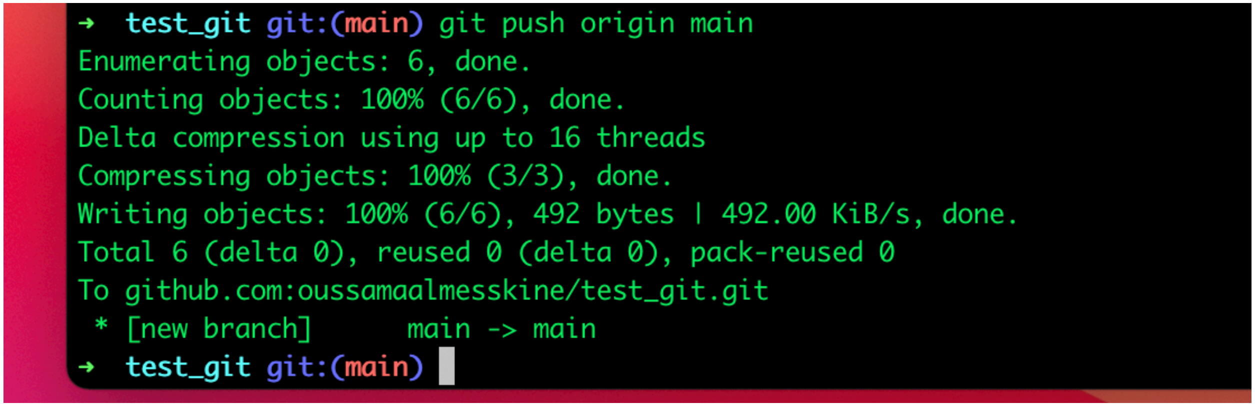
\

21)



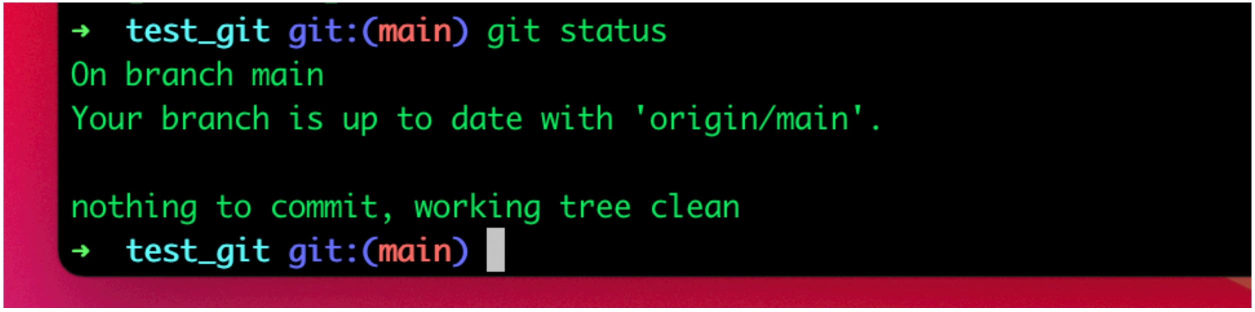
```
➜  test_git git:(main) git status
On branch main
Your branch is based on 'origin/main', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

nothing to commit, working tree clean
➜  test_git git:(main) 
```

22)



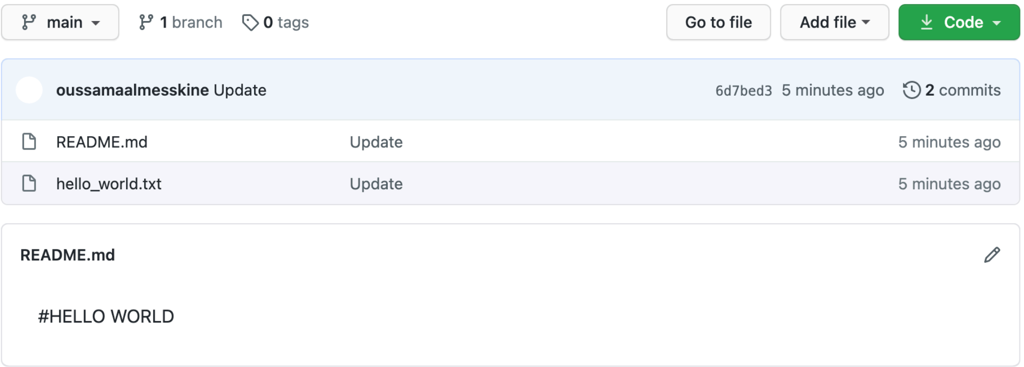
```
➜  test_git git:(main) git push origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 492 bytes | 492.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:oussamaalmesskine/test_git.git
 * [new branch]      main -> main
➜  test_git git:(main) 
```

23)



```
➜  test_git git:(main) git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
➜  test_git git:(main) 
```

24)



oussamaalmesskine Update 6d7bed3 5 minutes ago 2 commits

README.md	Update	5 minutes ago
hello_world.txt	Update	5 minutes ago

README.md

#HELLO WORLD