

Université Toulouse Paul Sabatier - KEAT9AA1

2024-09-03

Guilhem Saurel

Available at

**[https://homepages.laas.fr/gsaurel/teach/
2024-2025/M2_ISTR/4-git.pdf](https://homepages.laas.fr/gsaurel/teach/2024-2025/M2_ISTR/4-git.pdf)**

Under License



<https://creativecommons.org/licenses/by-sa/4.0/>

Source

**<https://gitlab.laas.fr/gsaurel/teach> :
2024-2025/M2_ISTR/4-git.md**

Contact

Matrix: @gsaurel:laas.fr
Mail: gsaurel@laas.fr

Introduction

```
$ cat liste-v1.txt
```

```
1
```

```
2
```

```
3
```

```
$ cat liste-v2.txt
```

```
1
```

```
9
```

```
3
```

```
$ diff -u liste-v1.txt liste-v2.txt
--- liste-v1.txt
+++ liste-v2.txt
@@ -1,3 +1,3 @@
  1
-2
+9
  3
```

```
$ diff -u liste-v1.txt liste-v2.txt \  
>> v1-to-v2.patch
```

```
$ diff -u liste-v1.txt liste-v2.txt \  
    >> v1-to-v2.patch  
  
$ patch -i v1-to-v2.patch  
patching file liste-v1.txt
```



```
$ diff -u liste-v1.txt liste-v2.txt \  
>> v1-to-v2.patch
```

```
$ patch -i v1-to-v2.patch  
patching file liste-v1.txt
```

```
$ cat liste-v1.txt
```

```
1  
9  
3
```

Git Pratique

```
$ cat hello.py
```

```
#!/usr/bin/env python
```

```
if __name__ == "__main__":  
    print("hello")
```

```
$ cat hello.py
```

```
#!/usr/bin/env python
```

```
if __name__ == "__main__":  
    print("hello")
```

```
$ git init
```

```
Dépôt Git vide initialisé  
dans ~/code/mon-projet/.git/
```

\$ git status

Sur la branche main

Aucun commit

Fichiers non suivis:

(utilisez "git add <fichier>..." pour inclure dans ce qui sera validé)

hello.py

aucune modification ajoutée à la validation mais des fichiers non suivis sont présents

(utilisez "git add" pour les suivre)

```
$ git add hello.py  
$ git status  
Sur la branche main
```

Aucun commit

Modifications qui seront validées :
(utilisez "git rm --cached <fichier>..."
pour désindexer)
nouveau fichier : hello.py

```
$ git commit -m "hello world"  
[main (commit racine) f5cc82f] hello world  
1 file changed, 4 insertions(+)  
create mode 100644 hello.py
```

```
$ vim hello.py && cat hello.py  
#!/usr/bin/env python  
"""Affiche "hello" sur la sortie standard."""  
  
if __name__ == "__main__":  
    print("hello")
```



```
$ vim hello.py && cat hello.py
#!/usr/bin/env python
"""Affiche "hello" sur la sortie standard."""

if __name__ == "__main__":
    print("hello")

$ git add hello.py
$ git commit -m "documentation"
[main 56148a5] documentation
 1 file changed, 1 insertion(+)
```

```
$ chmod +x hello.py  
$ git commit -am "executable"  
[main ab3723c] executable  
1 file changed, 0 insertions(+), 0 deletions(-)  
mode change 100644 => 100755 hello.py
```

```
$ git remote add origin \  
    https://github.com/nim65s/example.git  
$ git push -u origin main  
Énumération des objets: 8, fait.  
Décompte des objets: 100% (8/8), fait.  
Compression par delta en utilisant jusqu'à 4 fils d'exéc  
Compression des objets: 100% (5/5), fait.  
Écriture des objets: 100% (8/8), 803 octets | 200.00 Kio,  
Total 8 (delta 0), réutilisés 0 (delta 0), réutilisés du  
To github.com:nim65s/example.git  
* [new branch]      main -> main  
la branche 'main' est paramétrée pour suivre 'origin/main'
```

(sur un autre ordinateur / dans un autre dossier)

```
git clone https://github.com/nim65s/example.git
```

Git Théorique

Merci Matthieu :)

<https://homepages.laas.fr/matthieu/talks/git.pdf>

Références

https://learngitbranching.js.org/?locale=fr_FR