

TP : Utilisation de l'outil GIT

TIAGO GOMES

Bts SIO option SISR



Toute image illisible possède un lien si celui-ci ne fonctionne pas me contactez par email : mwood7149@gmail.com

```
tiagomes@hpnul: $ git init /home/tiagomes/Documents/sandwich
astuce: Utilisation de 'master' comme nom de la branche initiale. Le nom
       de la branche
astuce: par défaut peut changer. Pour configurer le nom de la branche in
       itiale
astuce: pour tous les nouveaux dépôts, et supprimer cet avertissement, l
       ancez :
astuce:
astuce:      git config --global init.defaultBranch <nom>
astuce:
astuce: Les noms les plus utilisés à la place de 'master' sont 'main', '
       trunk' et
astuce: 'development'. La branche nouvellement créée peut être renommée
       avec :
astuce:
astuce:      git branch -m <nom>
Dépôt Git vide initialisé dans /home/tiagomes/Documents/sandwich/.git/
```

2.1 : Le fichier burger.txt a été créé à l'aide de la commande de base nano, marquant ainsi la création du répertoire Git.

```
tiagomes@hpnul:~/Documents/sandwich$ nano burger.txt
tiagomes@hpnul:~/Documents/sandwich$ git status
Sur la branche master

Aucun commit

Fichiers non suivis:
  (utilisez "git add <fichier>..." pour inclure dans ce qui sera validé)
   burger.txt

aucune modification ajoutée à la validation mais des fichiers non suivis
sont présents (utilisez "git add" pour les suivre)
tiagomes@hpnul:~/Documents/sandwich$
```

2.2 : Comme le montrent les captures d'écran, le fichier burger.txt apparaît en rouge, indiquant qu'il se trouve dans l'espace de travail et n'est pas encore ajouté à l'index.

```

tiagoges@hpnul: ~/Documents/sandwich$ git add burger.txt
tiagoges@hpnul:~/Documents/sandwich$ git diff --cached
diff --git a/burger.txt b/burger.txt
new file mode 100644
index 0000000..760f26f
--- /dev/null
+++ b/burger.txt
@@ -0,0 +1,6 @@
+steak
+salade
+tomate
+cornichon
+fromage
+
tiagoges@hpnul:~/Documents/sandwich$

```

2.3 : Après avoir ajouté le fichier burger.txt à l'index, les modifications qui lui ont été apportées sont désormais visibles.

```

tiagoges@hpnul:~/Documents/sandwich$ git commit -m "creation du fichier sandwich"
[master (commit racine) 306276c] creation du fichier sandwich
1 file changed, 6 insertions(+)
create mode 100644 burger.txt
tiagoges@hpnul:~/Documents/sandwich$

```

2.4 : À la suite de cette commande, j'ai compris qu'une version de mon dossier "sandwich" a été enregistrée par Git.

```

tiagoges@hpnul:~/Documents/sandwich$ git status
Sur la branche master
rien à valider, la copie de travail est propre
tiagoges@hpnul:~/Documents/sandwich$

```

2.5 : Dans cet exemple, il est clair que mon répertoire surveillé par Git ne présente aucun problème et est à jour.

```

tiagoges@hpnul:~/Documents/sandwich$ git log
commit 306276cf6a31f3f9e69d876cb4ded99249f4d6f5 (HEAD -> master)
Author: TiTi <mwood7149@gmail.com>
Date: Tue Feb 20 17:15:18 2024 +0100

    creation du fichier sandwich

```

2.6 : Le numéro "306276cf6a31f3f9e69d876cb4ded99249f4d6f5" correspond au dernier commit que j'ai effectué, comme indiqué en rouge dans l'image ci-dessus.

```

tiagoges@hpnul1:~/Documents/sandwich$ git add hot_dog.txt
tiagoges@hpnul1:~/Documents/sandwich$ git commit hot_dog.txt
[master a8599a2] création d'un nouveau sandwich : hot_dog.txt
1 file changed, 5 insertions(+)
create mode 100644 hot_dog.txt
tiagoges@hpnul1:~/Documents/sandwich$

```

```

commit 597848046e9b91f377996292f57f072fe3b532b (HEAD -> master)
Author: TiTi <mwood7149@gmail.com>
Date: Tue Feb 20 17:35:53 2024 +0100

    Sur la branche master
    Modifications qui seront validées :
        modifié : burger.txt

+ la sauce bruger

commit a6225552f943f9e5951b920c69329122a9f2666f
Author: TiTi <mwood7149@gmail.com>
Date: Tue Feb 20 17:34:59 2024 +0100

    tmtc l'ingrédient qui manquais pour l'hot-dog

commit 6432d28f8770283229b80b29c4de3d48f01af5f9
Author: TiTi <mwood7149@gmail.com>
Date: Tue Feb 20 17:33:42 2024 +0100

    Ajout de la recette croque Mr

commit a8599a2f68e34e4c5ff5854187958802e7235b07
Author: TiTi <mwood7149@gmail.com>
Date: Tue Feb 20 17:24:24 2024 +0100

    création d'un nouveau sandwich : hot_dog.txt

commit 306276cf6a31f3f9e69d876cb4ded99249f4d6f5
Author: TiTi <mwood7149@gmail.com>
Date: Tue Feb 20 17:15:18 2024 +0100

    creation du fichier sandwich

```

2.7 : J'ai effectué plusieurs actions dans le répertoire, telles que l'ajout et la modification de fichiers existants. Chaque modification était suivie d'un commit [nom du fichier], laissant ainsi une trace de ma modification pour informer les participants imaginaires de mon projet "sandwich" des éventuelles erreurs que j'ai commises.

```

tiagoges@hpnul1:~/Documents/sandwich$ git diff
tiagoges@hpnul1:~/Documents/sandwich$ git diff --cached
tiagoges@hpnul1:~/Documents/sandwich$

```

2.8 : À la fin de l'exercice, j'ai exécuté les commandes git diff et git diff --cached, ce qui m'a permis de constater que tous les fichiers ont été correctement commités, comme en témoigne le manque d'informations renvoyé par la commande.

```

tiagoges@hpnul1:~/Documents/sandwich$ git add burger.txt
tiagoges@hpnul1:~/Documents/sandwich$ git status
Sur la branche master
Modifications qui seront validées :
  (utilisez "git restore --staged <fichier>..." pour désindexer)
    modifié : burger.txt

tiagoges@hpnul1:~/Documents/sandwich$ git reset burger.txt
Modifications non indexées après reset :
M    burger.txt
tiagoges@hpnul1:~/Documents/sandwich$ git status
Sur la branche master
Modifications qui ne seront pas validées :
  (utilisez "git add <fichier>..." pour mettre à jour ce qui sera validé
  )
  (utilisez "git restore <fichier>..." pour annuler les modifications da
  ns le répertoire de travail)
    modifié : burger.txt

aucune modification n'a été ajoutée à la validation (utilisez "git add"
ou "git commit -a")
tiagoges@hpnul1:~/Documents/sandwich$

```

2.9 : En utilisant la commande git reset, j'ai retiré le fichier burger.txt de l'index dans mon cas.

```

tiagoges@hpnul1:~/Documents/sandwich$ git checkout burger.txt
1 chemin mis à jour depuis l'index
tiagoges@hpnul1:~/Documents/sandwich$ git status
Sur la branche master
rien à valider, la copie de travail est propre
tiagoges@hpnul1:~/Documents/sandwich$ █

```

2.10 : En exécutant la commande git checkout, mon répertoire est revenu à l'état du dernier commit effectué. La commande git status indique que le fichier est propre et à jour, montrant ainsi que les modifications apportées dans la copie de travail ont été supprimées.

```

tiagoges@hpnul1:~/Documents/sandwich$ git checkout 306276cf6a31f3f9e69d876cb4ded99249f4d6f5
Note : basculement sur '306276cf6a31f3f9e69d876cb4ded99249f4d6f5'.

Vous êtes dans l'état « HEAD détachée ». Vous pouvez visiter, faire des modifications
expérimentales et les valider. Il vous suffit de faire un autre basculement pour
abandonner les commits que vous faites dans cet état sans impacter les autres branches

Si vous voulez créer une nouvelle branche pour conserver les commits que vous créez,
il vous suffit d'utiliser l'option -c de la commande switch comme ceci :

    git switch -c <nom-de-la-nouvelle-branche>

Ou annuler cette opération avec :

    git switch -

Désactivez ce conseil en renseignant la variable de configuration advice.detachedHead à false

HEAD est maintenant sur 306276c creation du fichier sandwich
tiagoges@hpnul1:~/Documents/sandwich$ git statut
git : 'statut' n'est pas une commande git. Voir 'git --help'.

La commande la plus ressemblante est
    statut
tiagoges@hpnul1:~/Documents/sandwich$ git status
HEAD détachée sur: 306276c
rien à valider, la copie de travail est propre
tiagoges@hpnul1:~/Documents/sandwich$ █

```

2.11 : En exécutant la commande git checkout avec l'ID "306276cf6a31f3f9e69d876cb4ded99249f4d6f5", correspondant au premier commit, mon répertoire "sandwich" est revenu à cette version. La commande git status me permet de vérifier sur quelle version de mon répertoire je me trouve.

```

tiagoges@hpnul1:~/Documents/sandwich$ git status
Sur la branche master
rien à valider, la copie de travail est propre
tiagoges@hpnul1:~/Documents/sandwich$ █

```

2.12 : En utilisant la commande git checkout master, j'ai effectué un retour à la version la plus récente de mon répertoire. La commande git status confirme que je suis de nouveau sur la branche master.