

Outil Git pour usage en local

Enseignant : IALA Imad



Objectifs

- Connaitre le système de gestion de version « git », ses avantages et son principe
- Savoir installer Git sur une machine Windows et Linux
- Comprendre et pratiquer les commandes de gestion et création de dépôt git en local



Plan

❖ Références

❖ Git en quelques mots

❖ Avantages de Git

❖ Principe

❖ Démarrer Git

❖ Configurer Git

❖ Créer un dépôt local

❖ Indexer un fichier

❖ Voir les changements

❖ Envoyer un fichier au dépôt local

❖ Voir les commits

❖ Options d'indexation

❖ Fichier .gitignore

❖ Installation de Git

Références

Sur le site web officiel : <https://git-scm.com/>

- livre Pro Git - Scott Chacon
- [github-git-cheat-sheet.pdf](#)
- Git Manual Page «[git-htmldocs-2.11.1.tar](#) »
- <https://git-scm.com/videos>

Ailleurs :

- livre Pro Git - Scott Chacon version française
- <http://rogerdudler.github.io/git-guide/index.fr.html>



git en quelques mots

- ❖ multiplateformes
- ❖ Distribué
- ❖ Très populaire en 2016 avec +12 millions d'utilisateurs
- ❖ Développeur : Linus Torvalds auteur du noyau Linux.
- ❖ Première version en 2005.



Avantages de Git

Résistance aux pannes

- Aucun dépôt n'a des données plus que les autres dépôts

Rapidité d'exécution

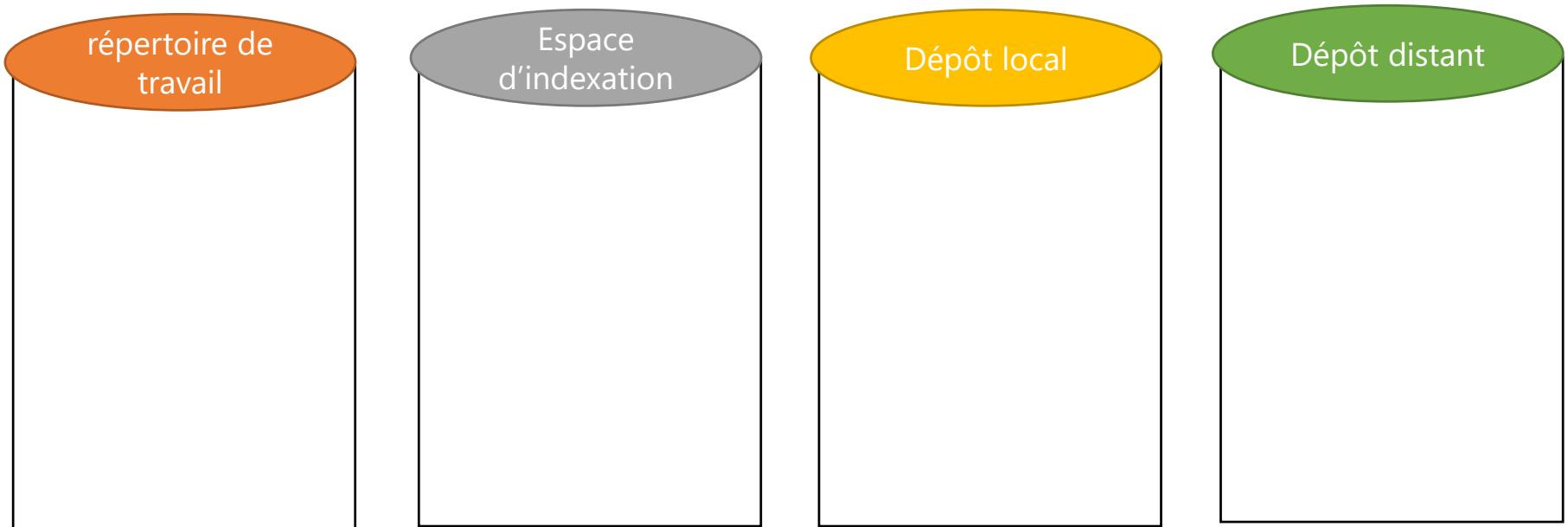
- Opérations très rapides par rapport aux autres CVS (comme Subversion)

Occupation réduite de l'espace disque

- Support de la compression Zlib pour les dépôts comme pour les fichiers.
- → Minimise la taille locale ainsi que les transferts de données

Principe

- Git manipule trois composantes locales plus une composante distante:



Démarrer Git

Pour démarrer la ligne de commande « Git » on peut:

- Dans la partition C , Créer un répertoire « Git »
- Clic droit sur ce dossier créée
- Ensuite Choisir « Git bash here »



Configuration de Git

- Avant de commencer, il est essentiel de configurer le nom et l'email du développeur qui utilise la machine (réponse à la question qui a fait ?)
 - **\$ git config --global user.name "votre nom"**
 - **\$ git config --global user.email "root@gmail.com"**
- Pour vérifier la correction du nom et l'email du développeur , il y a deux façons :
 1. **\$git config –list**
 2. **\$ cd puis \$cat .gitconfig**

Créer un dépôt local

Trois principales étapes :

1. Créer un répertoire :

```
$ mkdir formation_git
```

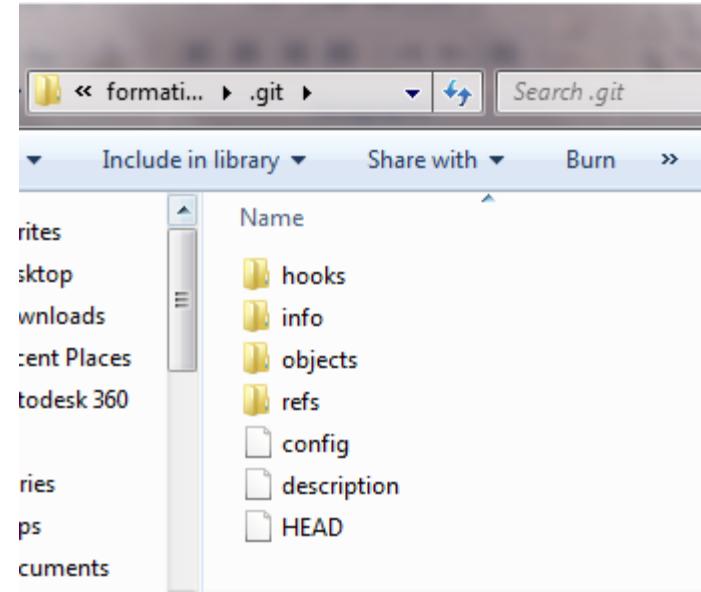
2. Se déplacer dans le répertoire créé :

```
$ cd formation_git
```

3. Initialiser un dépôt Git local dans le répertoire 'formation_git'

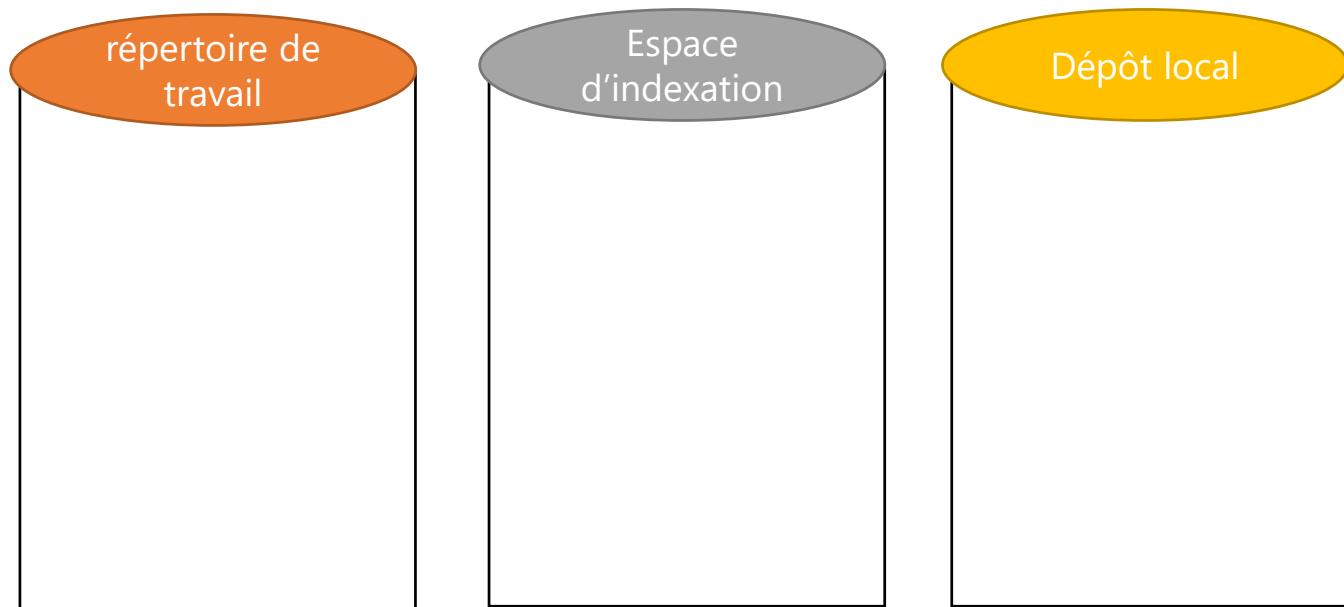
```
$ git init
```

Un répertoire « .git » est généré dans Le répertoire.



...

Une fois un dépôt est créée, les trois espaces **sont vides**



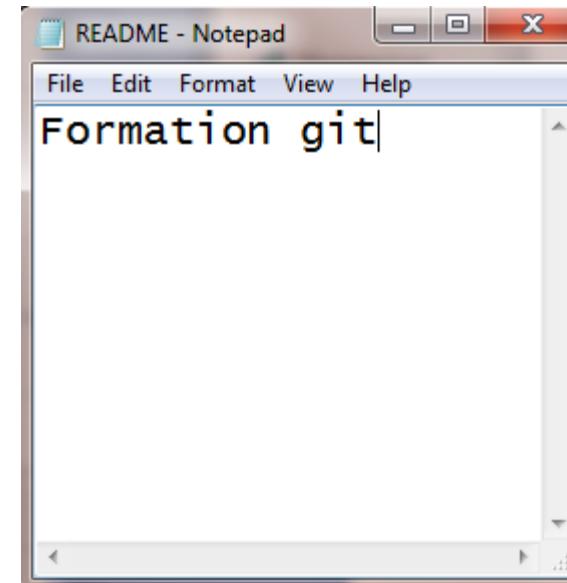
Créer un fichier README.txt

1. Sans changer de répertoire, créer un fichier « README.txt » dans l'invite de commande par touch :

```
$ touch README.txt
```

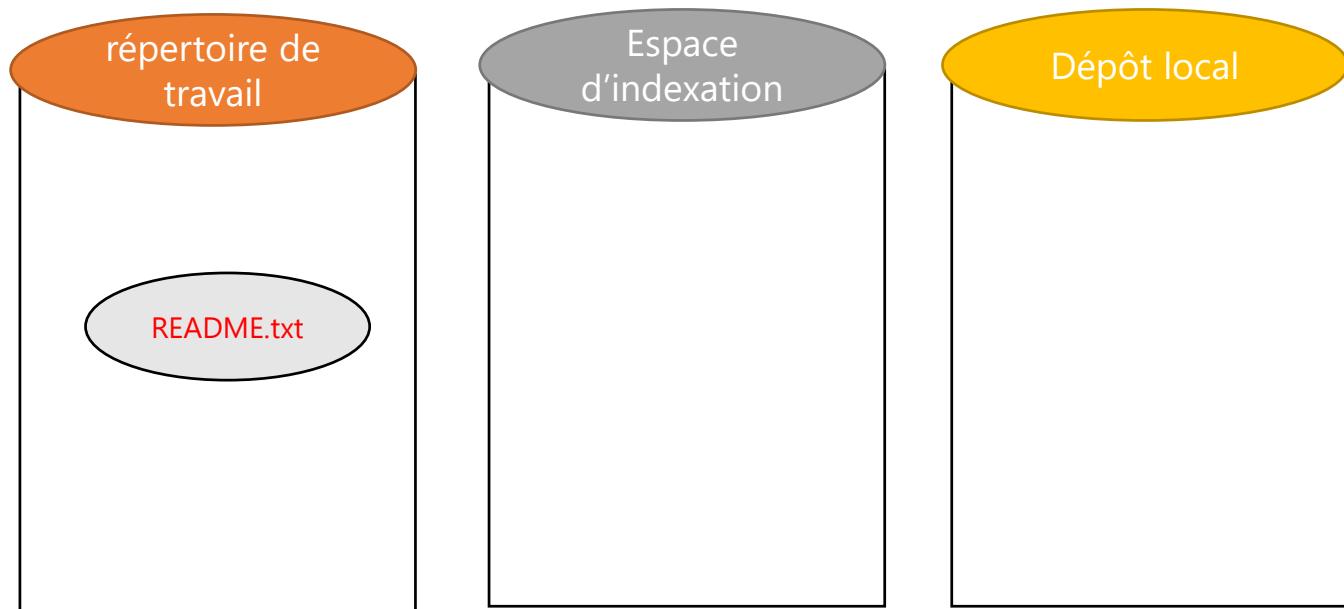
2. Ouvrir le fichier depuis l'interface Windows et ajouter ce contenu puis enregister :

« Formation Git »



...

Le fichier README.txt est ajouté au répertoire de travail.



Constater les changements

Pour voir l'état des changements dans le répertoire de travail, utiliser la commande :

```
$ git status
```

```
root@root-PC MINGW64 /c/Git/formation_git (master)
$ git status
On branch master

Initial commit

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

    README.txt

nothing added to commit but untracked files present (use "git add" to track)
root@root-PC MINGW64 /c/Git/formation_git (master)
$ |
```

README.txt n'est pas indexé (**untracked**) , dans le texte d'aide, on propose de l'indexer avec **\$git add**

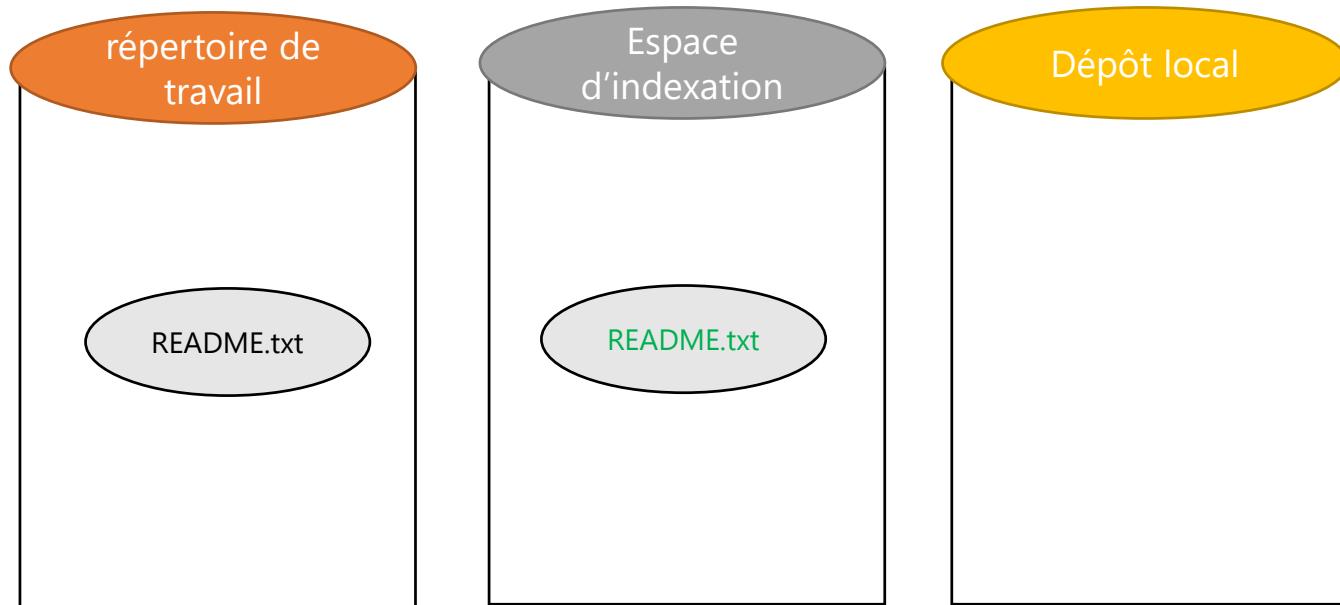
Indexer un fichier

Indexer (en anglais track or stage) un fichier avec la commande suivante :

```
$ git add README.txt
```

...

En indexant le fichier avec la commande **\$ git add README.txt**, le fichier passe à l'espace d'indexation. C-a-d on va le sélectionner pour entrer dans des prochaines versions (commits) du dépôt local.



Constater les changements

```
$ git status
```

```
root@root-PC MINGW64 /c/Git/formation_git (master)
$ git status
On branch master

Initial commit

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

    new file:   README.txt

root@root-PC MINGW64 /c/Git/formation_git (master)
$ |
```

README.txt est dans l'espace Index, distingué par la couleur Verte

Envoyer au dépôt local

Pour envoyer (ou mettre à jour) le dépôt local et laisser une trace de la création du fichier README.txt, on utilise :

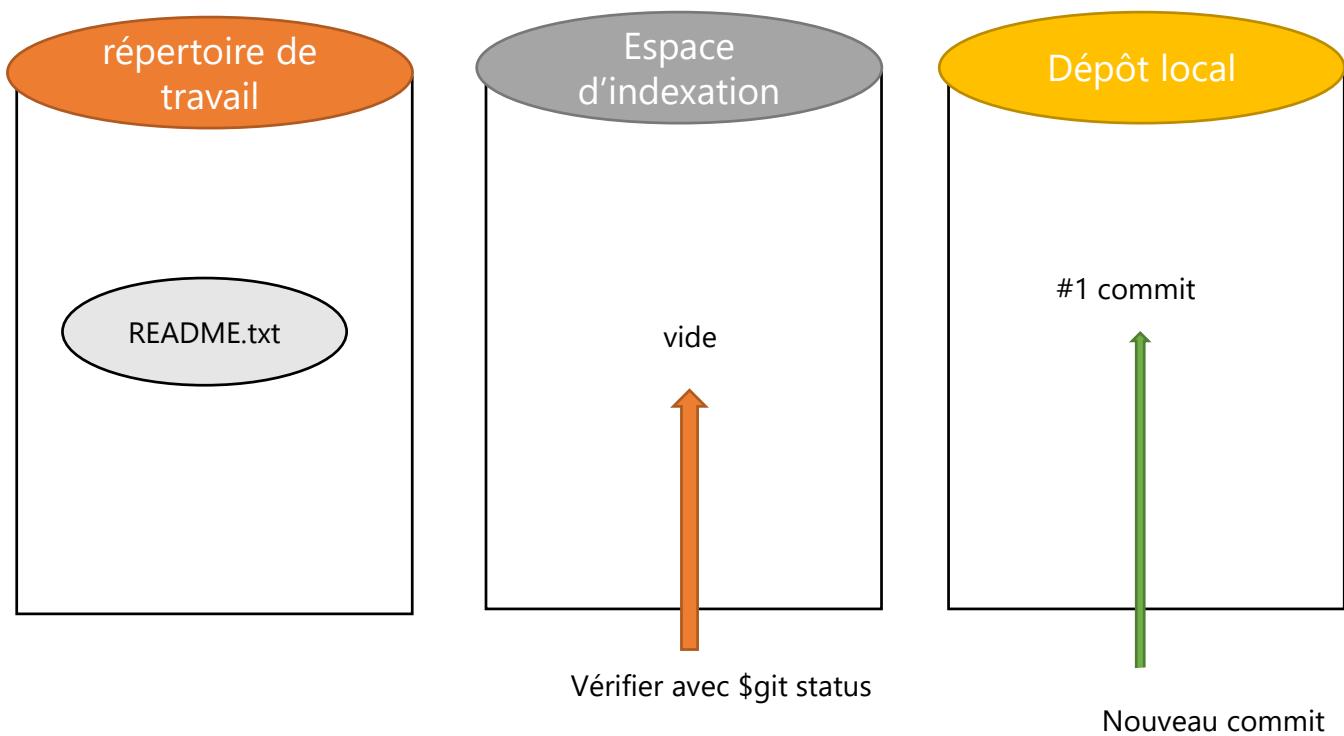
```
$git commit -m "premier ajout de README.txt"
```

```
root@root-PC MINGW64 /c/Git/formation_git (master)
$ git commit -m "premier ajout de README.txt"
[master (root-commit) 7752a9b] premier ajout de README.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.txt

root@root-PC MINGW64 /c/Git/formation_git (master)
$ |
```

Noter que tout commit est décrit par un message qui doit être **précis et concis**.

...



Historique des commits



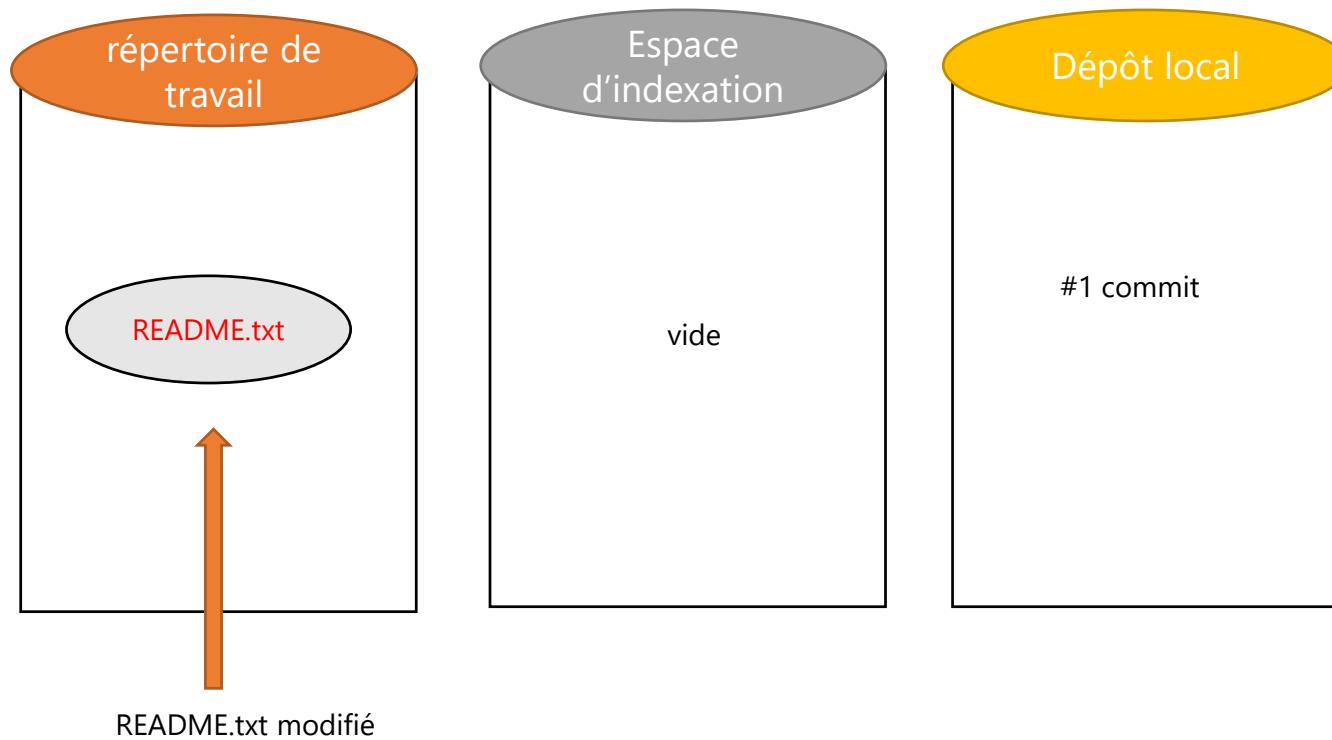
Poursuivons le travail

1. Ajouter une ligne à la fin du fichier README.txt :

« A l'école VINCI »

2. Enregistrer le fichier

...



Constater les changements

\$git status

```
root@root-PC MINGW64 /c/Git/formation_git (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

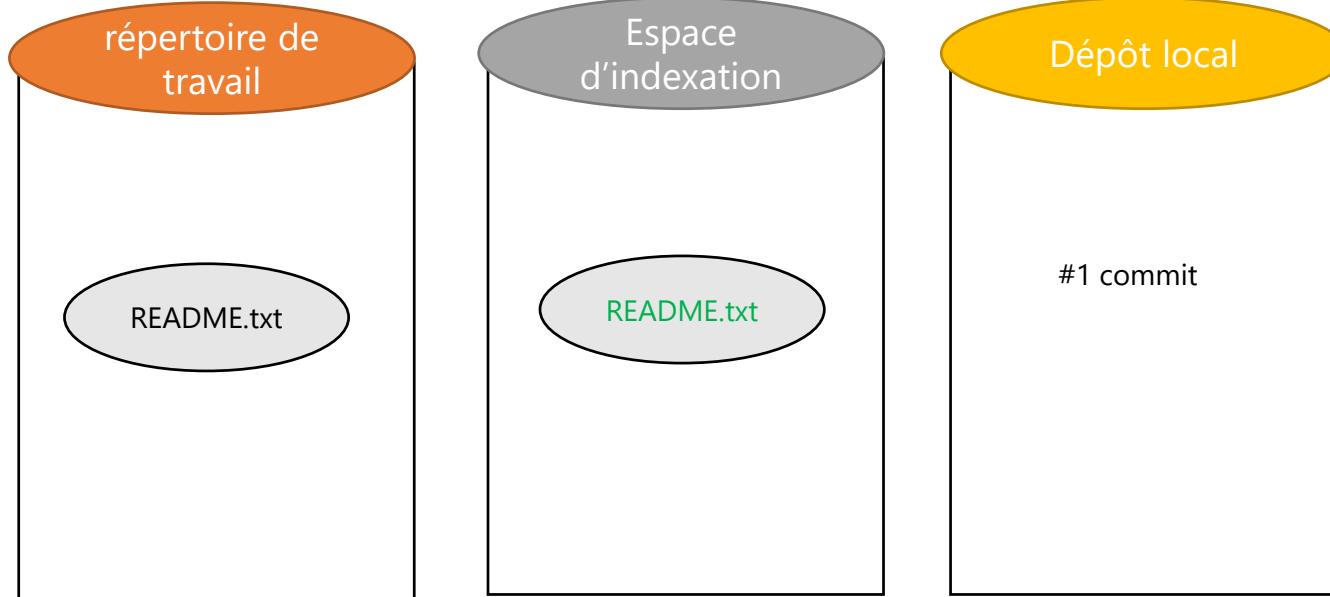
        modified:   README.txt

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

Indexer (stage)

\$git add README.txt

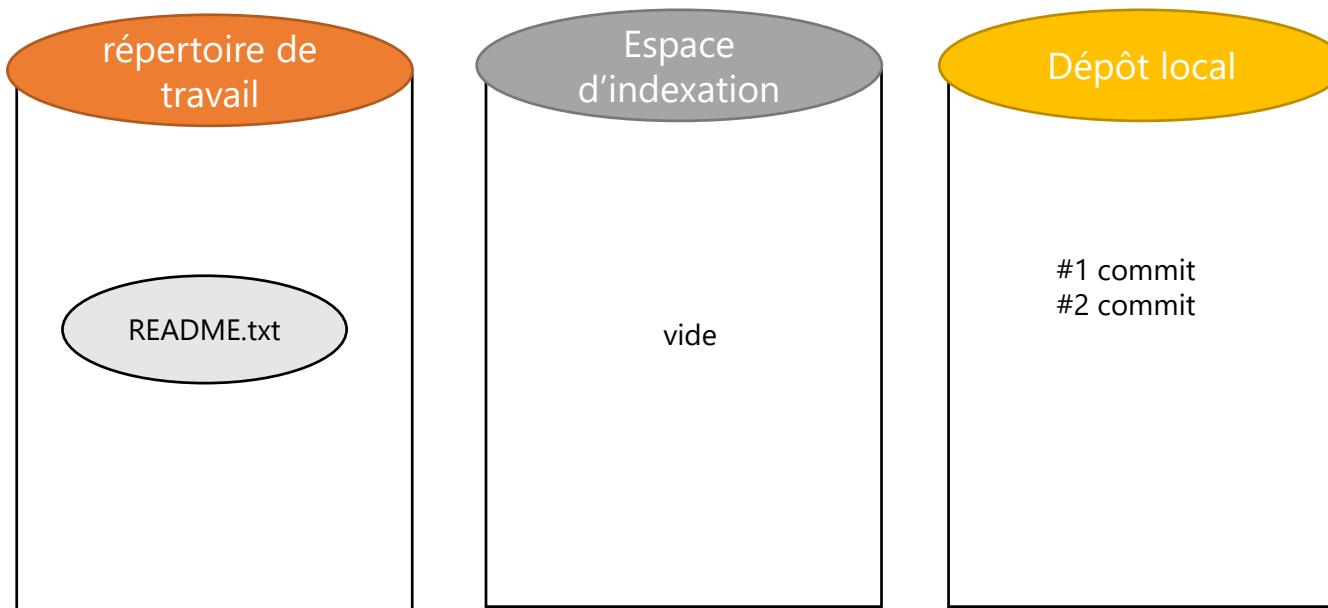
Une fois exécuter la commande, on va voir ce résultat (figure) :



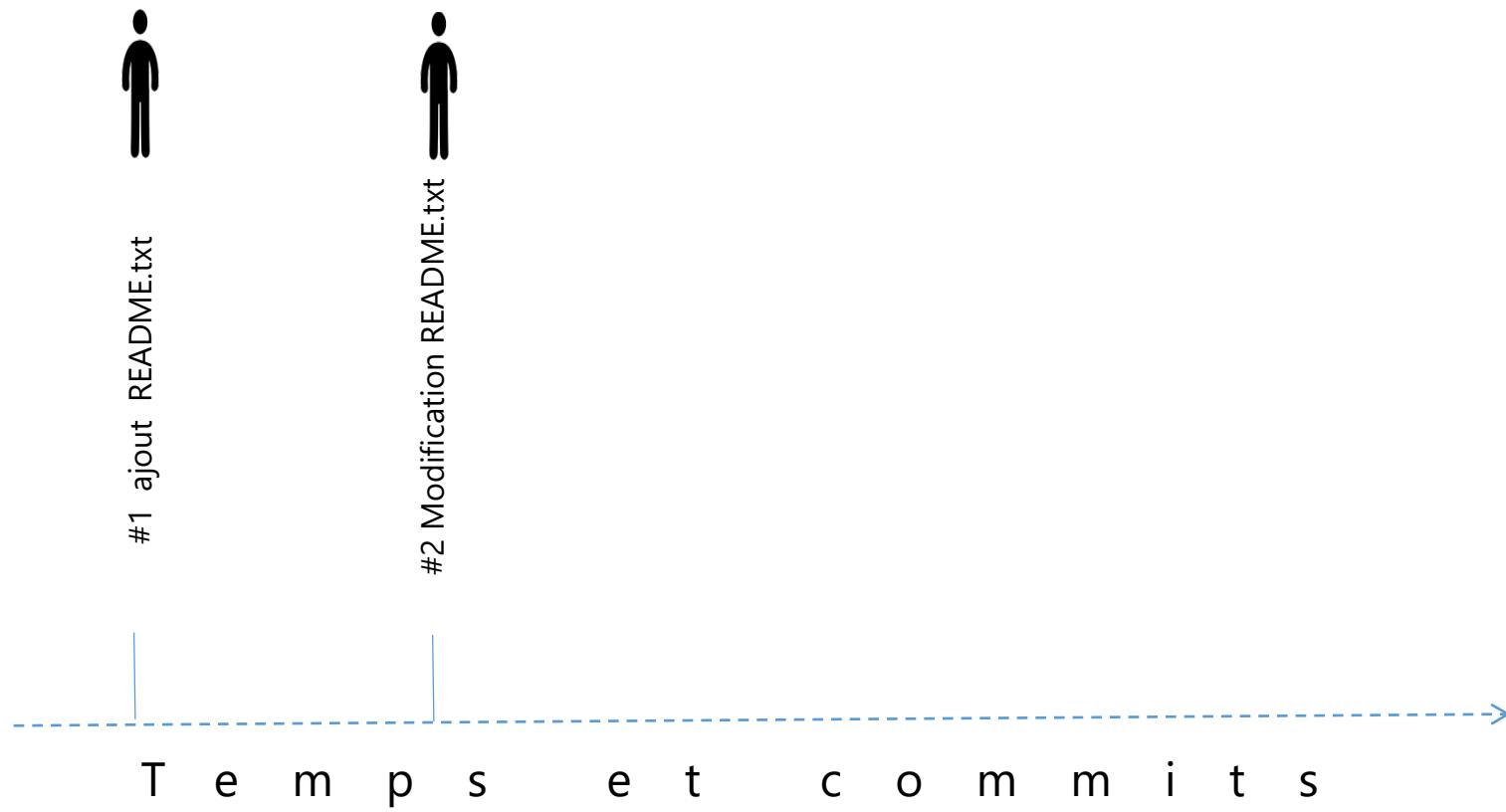
Envoyer au dépôt local

```
$ git commit -m "Modification README.txt"
```

Une fois exécuter la commande, on va avoir ce résultat (figure) :



Historique des commits



Historique des commits

```
$ git log
```

```
$ git log
commit 32a25f6252b5878879620918f54d4d214901259b
Author: arbi cours git <elalaouy.elarbi@gmail.com>
Date:   Thu Feb 9 17:56:06 2017 +0000

    Modification README.txt

commit 7752a9bacbc159e3c1616868161a51a32c8e00a5
Author: arbi cours git <elalaouy.elarbi@gmail.com>
Date:   Thu Feb 9 17:22:59 2017 +0000

    premier ajout de README.txt

root@root-PC MINGW64 /c/Git/formation_git (master)
```

En Résumé

Quelle commande utilise-t-on ?

1. Pour ajouter un fichier à l'espace d'indexation ;
2. Pour envoyer les modifications au dépôt local;
3. Pour constater l'état des changements du répertoire du travail
4. Pour voir l'historique des commits ;

En Résumé

Quelle commande utilise-t-on ?

1. Pour ajouter un fichier à l'espace d'indexation ; **\$ git add**
2. Pour envoyer les modifications au dépôt local; **\$git commit -m "message"**
3. Pour constater l'état des changements du répertoire du travail; **\$git status**
4. Pour voir l'historique des commits ; **\$git log**

Options d'indexation

La commande « **git add** » a des options intéressantes :

- Indexer tout les fichiers ayant été modifié avec une seule commande :

```
$git add .
```

- Indexer plusieurs fichiers ayant une extension bien précise (e.g. fichiers html)

```
$ git add *.html
```

Fichier .gitignore

Le fichier `.gitignore` contient une liste de noms de fichiers **confidentiels** à exclure de l'**indexation**, e.g., fichiers log, fichiers contenant les mots de passe etc.

➤ Pour créer un fichier « `.gitignore` », on utilise cette commande :

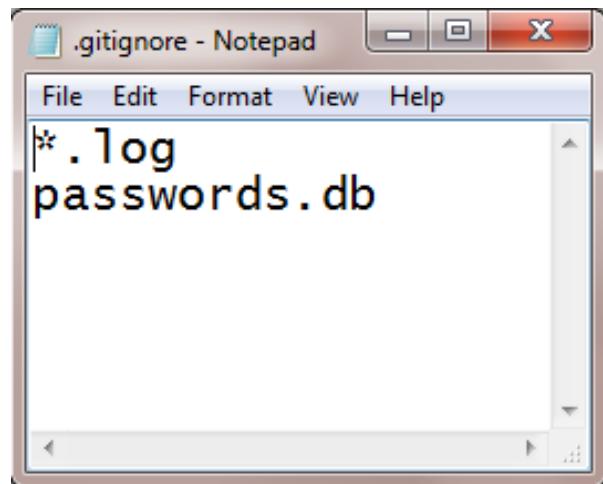
```
$ git touch .gitignore
```

➤ Ajouter le fichier « `.gitignore` », à l'espace d'indexation en utilisant cette commande :

```
$ git add .gitignore
```

...

- Ajout de la liste des noms de fichiers à exclure de l'indexation



- Même si maintenant on réalise l'ajout des fichiers avec « **\$git add .** » les fichiers ayant l'extension **.log** et le fichier « **passwords.db** » vont être exclus de l'indexation.