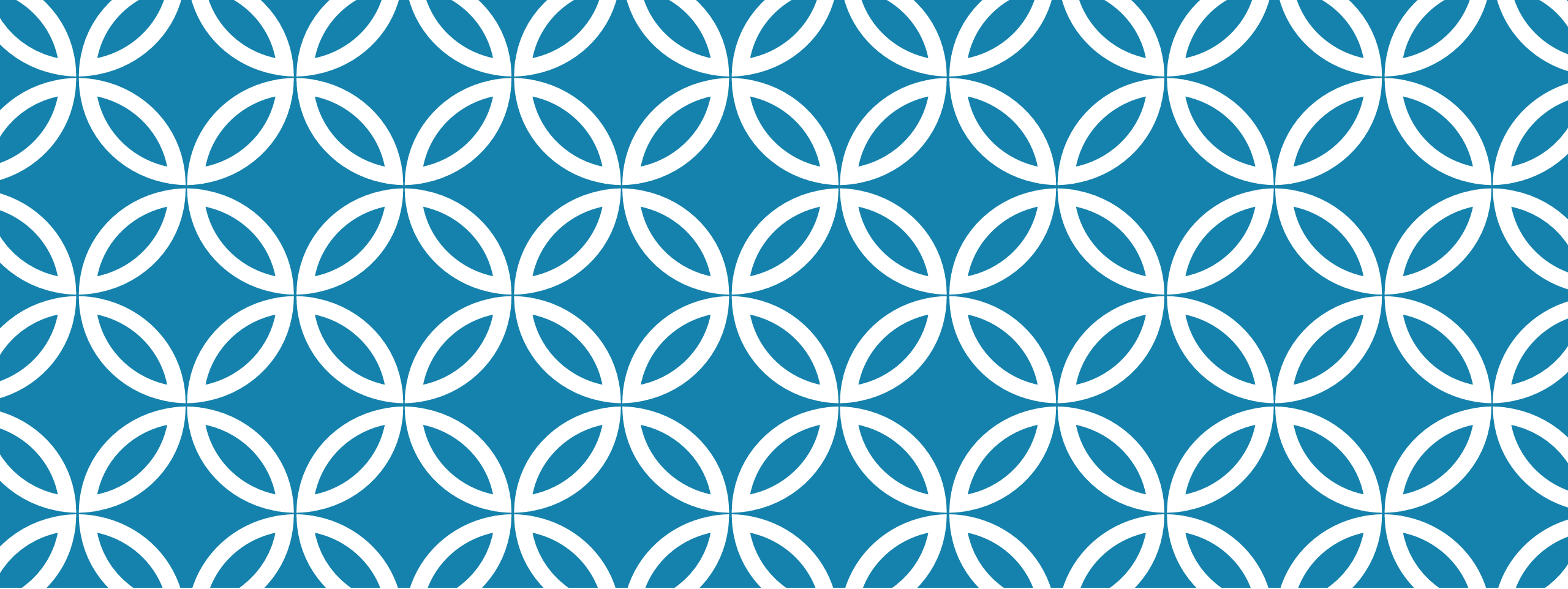


FORMATION GIT

Jordan ABID



PRÉSENTATION



GESTION DES VERSIONS DANS UN PROJET

Workflow classique d'un fichier



QU'EST CE QUE GIT ?

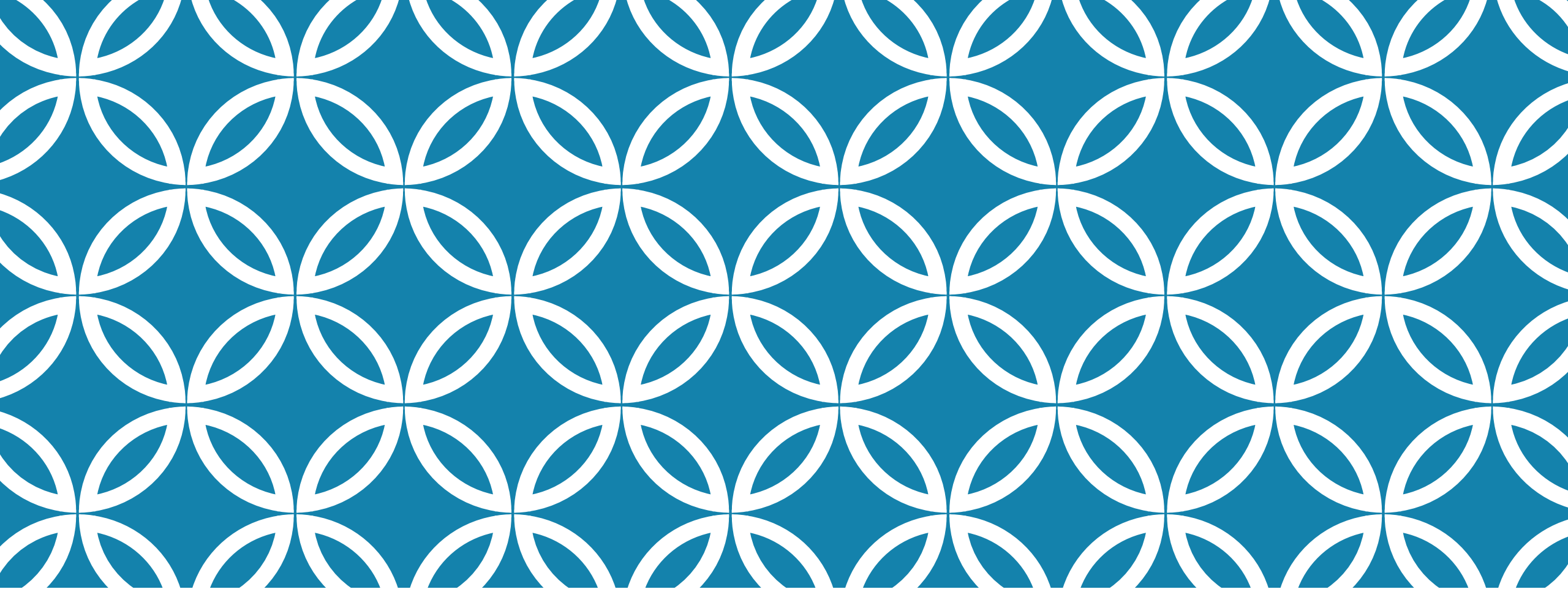
GIT est un outil aidant à la gestion des versions d'un projet collaboratif.

Pour chaque versions, il répond aux questions suivantes :

- Quels sont les fichiers modifiés
- Qui a modifier les fichiers
- Quand est-ce que la modification a eu lieu
- Pourquoi les fichiers sont modifiés

GIT permet de revenir rapidement à une ancienne version

GIT permet de fusionner les modifications de plusieurs personnes sur un même fichier



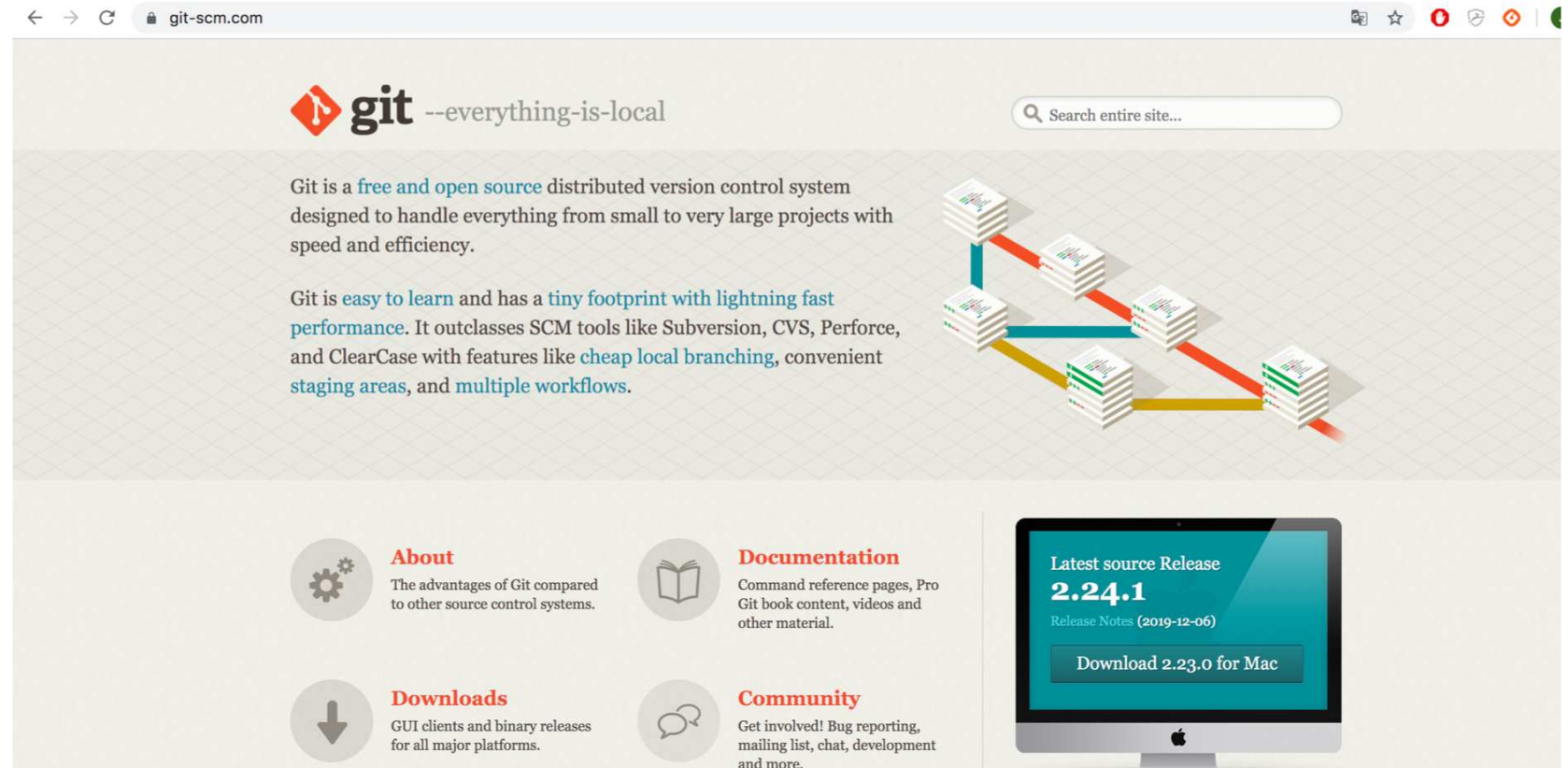
INSTALLATION DE GIT



TÉLÉCHARGER GIT

Pour télécharger GIT, il suffit de se rendre sur le site officiel et de suivre les recommandations d'installation.

<https://git-scm.com/>



VALIDER L'INSTALLATION ET VÉRIFIER LA VERSION

Check install

Verifier la version de GIT



```
git --version
```

PARAMÉTRAGE DE GIT

Pour chaque ajout de version, il faudra renseigner son identité. On peut fixer ses valeurs dès le début

```
git config --global user.name "Jordan Abid"  
git config --global user.email "j.abid@ajc-ingenierie.fr"
```


INITIALISATION DE GIT DANS UN DOSSIER EXISTANT

Initialisation

Initialise GIT dans le projet

```
git init
```



STATUT DU REPOSITORY

Etat des lieux

Voir à tout moment le statut des fichiers



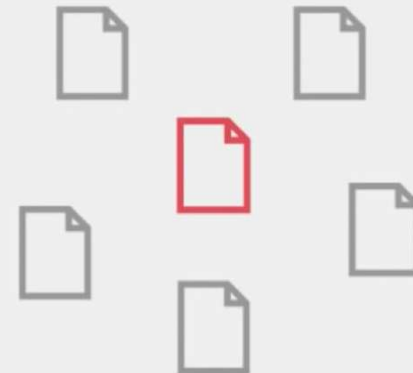
```
git status
```

GESTION D'UN FICHIER

GIT gère les fichiers modifiés à l'aide d'une liste. Pour prendre en compte une modification, il faudra ajouter les fichiers modifiés dans cette liste

1. Sélection des fichiers

```
git add fileName
```

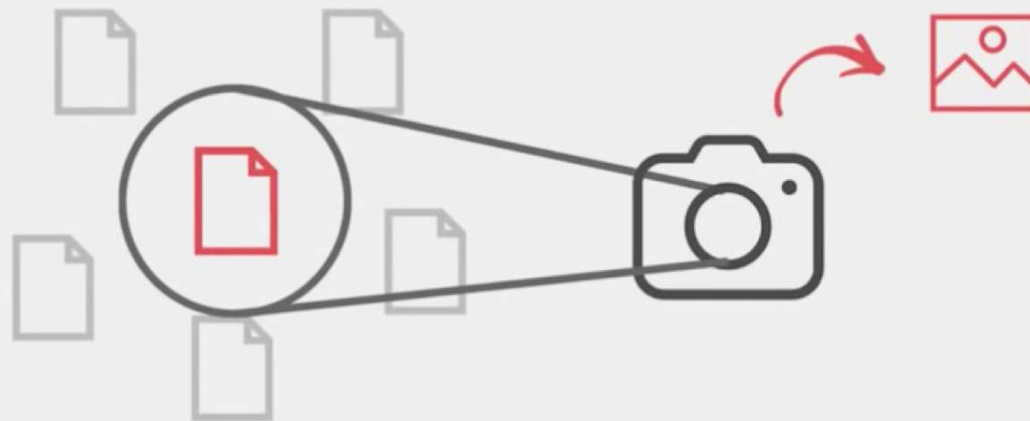


VALIDATION DE LA NOUVELLE VERSION

2. Photo des fichiers

```
git commit -m "first commit"
```

commentaire



VÉRIFIER TOUTES LES VERSIONS

Time line des commits

```
git log
```



CRÉATION D'UNE BRANCHE

Par défaut, GIT place notre projet sur une branche appelée « master »

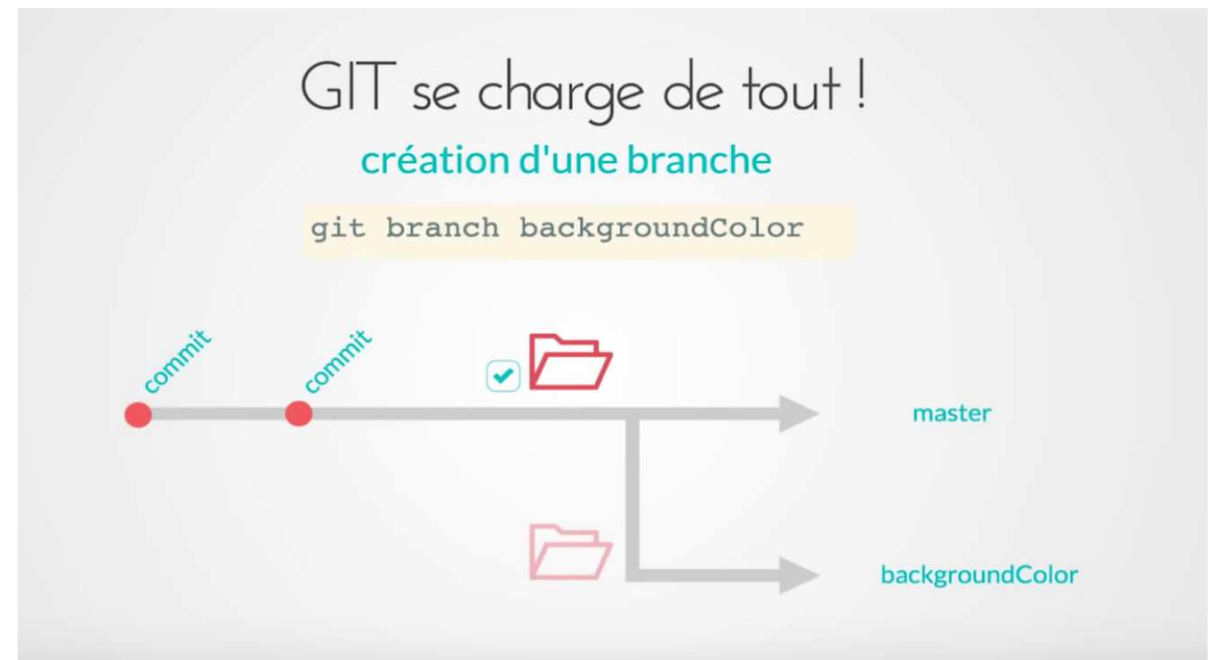
La création d'une nouvelle branche est une copie du projet initial dans laquelle on pourra travailler sans affecter le master.

On pourra par la suite fusionner une branche avec le master, ou tout simplement la supprimer

Il es possible de lister l'ensemble des branches avec la commande `git branch`

Sur une branche, il est possible de masquer / afficher les fichiers modifiés pour revenir à l'état du master

`git stach / git stach apply`



COMMIT SUR UNE BRANCHE

Branches indépendantes

Le commit s'applique uniquement sur la branche active



APPLIQUER LES MODIFICATIONS À LA BRANCHE MASTER

1. Sélectionner la branche principale

```
git checkout master
```



APPLIQUER LES MODIFICATIONS À LA BRANCHE MASTER

2. Fusionner les commits

```
git merge colorBackground
```



SUPPRIMER UNE BRANCHE

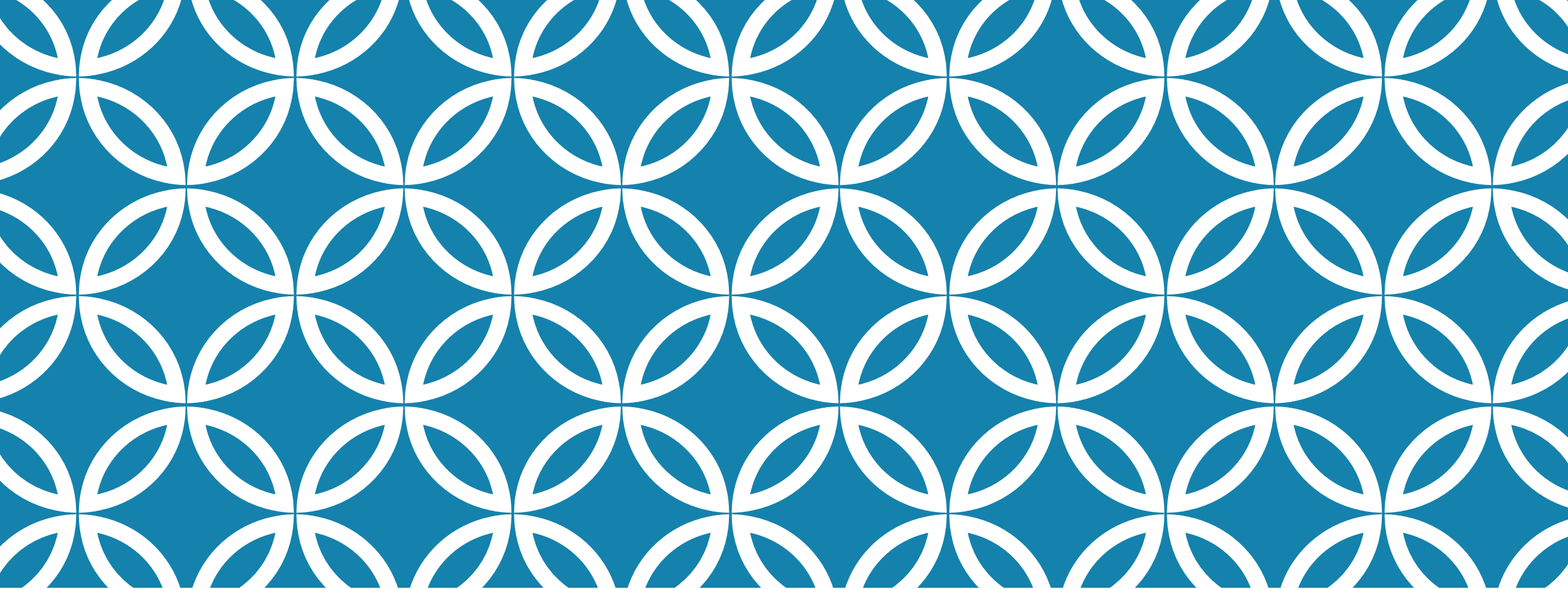
3. Suppression de la branche

```
git branch -d backgroundColor
```



REVENIR À UNE ANCIENNE VERSION

GIT propose de revenir à une certaine version en ne prenant plus en compte les modifications suivantes, pour cela on utilise la commande `git reset bddbbe34be...`



GITHUB



DISTRIBUTION DU PROJET

Distribuer le projet

Déposons le projet sur un espace partagé



TRAVAIL EN ÉQUIPE

Travail collaboratif
Chaque membre peut participer au projet



TRAVAIL EN ÉQUIPE

Push : pour déposer le projet

Commande générique

```
git push origin master
```

label du dépôt

nom de la branche à
deposer



TRAVAIL EN ÉQUIPE

Pull : pour récupérer le projet

Commande générique

```
git pull origin master
```

label du dépôt

nom de la branche à
recupérer

