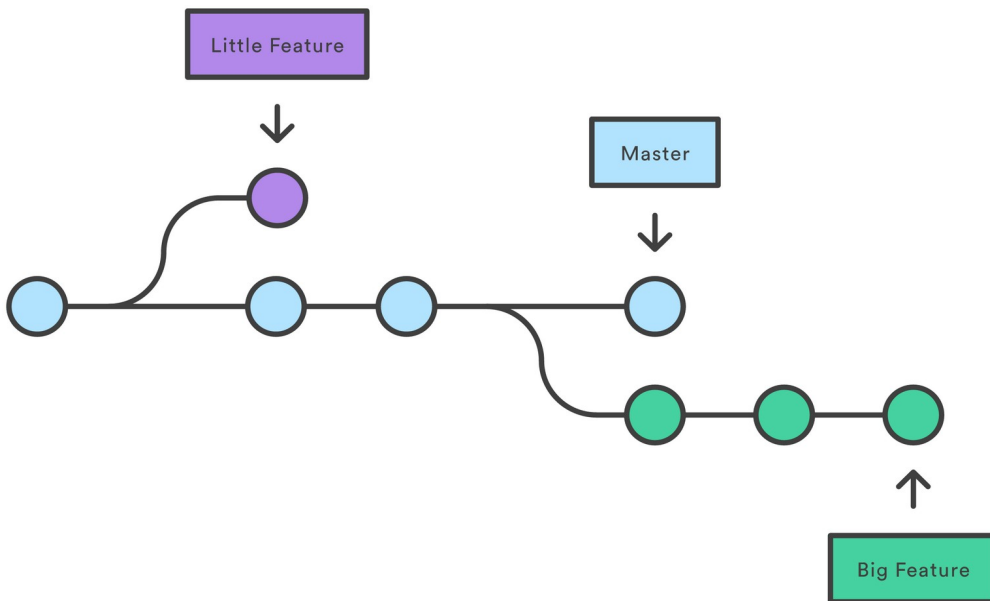


Git Branch

Une branche est une version du repository qui diverge du projet principal. C'est une fonctionnalité disponible dans la plupart des systèmes de contrôle de version (VCS) modernes. Un projet Git peut avoir plusieurs branches. Ces branches indiquent un instantané de vos modifications. Lorsque nous souhaitons ajouter une nouvelle fonctionnalité ou corriger un bug, nous générons une nouvelle branche pour mettre en place vos modifications. Noté qu'il est complexe de fusionner un code instable avec la base principale de votre code, d'où l'utilité des branches, qui nous permettra également de nettoyer notre futur historique avant de la fusionner avec notre branche principale (master).



Git Master Branch

La branche principale (master) est une branche par défaut dans Git. Elle est instanciée lors de la première validation effectuée sur le projet. Lorsque nous effectuons le premier commit, nous recevons une branche principale au point de commit de départ. Lorsque nous commençons à effectuer une validation, le pointeur de votre branche master avance automatiquement. Un repository ne peut avoir qu'une seule branche principale.

La branche principale est la branche dans laquelle tous les changements sont finalement fusionnés. C'est généralement la branche que nous exploitons pour la mise en production.

Opérations sur les branches

Nous pouvons effectuer diverses opérations sur les branches Git. La commande `git branch` nous permet de créer, répertorier, renommer et supprimer des branches. De nombreuses opérations sur les branches sont appliquées par la commande `git checkout` et `git merge`. Ainsi, les branches git sont étroitement liées aux commandes `git checkout` et `git merge`.

Ces opérations pouvant être effectuées sur une branche.

Créer une branche

Nous pouvons créer une nouvelle branche à l'aide de la commande `git branch`. Cette commande sera utilisée comme:

```
git branch $nomBranche
```

Cette commande créera la branche `$nomBranche` localement dans le répertoire Git.

Liste des Branches (Branch List)

Nous pouvons répertorier toutes les branches disponibles dans notre repository à l'aide de la commande suivante.

Nous pouvons utiliser `git branch --list` ou la commande `git branch` pour lister les branches disponibles dans le repository.

```
git branch --list
```

ou

```
git branch
```

Ici, les deux commandes répertorient les branches disponibles dans le repository. Le symbole * qui apparaît dans la liste représente la branche actuellement active.

Supprimer la branche

Nous pouvons supprimer une branche spécifiée. C'est une opération sûre. Dans cette commande, Git nous empêche de supprimer la branche si elle a des modifications non fusionnées à master. Nous trouverons ci-dessous la commande pour ce faire.

```
git branch -d $nomBranche
```

Cette commande supprimera la branche \$nomBranche existante du repository.

La commande `git branch -d` peut être utilisée dans deux formats. Un autre format de cette commande est `git branch -D`. La commande '`git branch D`' est utilisée pour supprimer la branche spécifiée.

```
git branch -D $nomBranche
```

Supprimer une branche distante

Nous pouvons supprimer une branche distante de Github. La commande ci-dessous est utilisée pour supprimer une branche distante:

```
git branch origin -delete $nomBranche
```

Nous pourrions constater après l'utilisation de cette commande que la branche distante nommée \$nomBranche de notre compte GitHub est supprimée.

Switch Branch

Git nous permet de basculer entre les branches sans faire de commit. Nous pouvons basculer entre deux branches avec la commande `git checkout`. Pour basculer entre les branches, nous utilisons la commande ci-dessous :

```
git checkout $nomBranche
```

Passer à une autre branche depuis la branche principale

Nous pouvons passer de la master à toute autre branche disponible dans notre repository sans effectuer de validation.

```
git checkout $nomBranche
```

Nous passons ici de la branche master à une autre branche \$nomBranche sans faire de commit.

Passer à la branche principale (master)

Nous pouvons basculer vers la branche principale à partir de n'importe quelle autre branche à l'aide de la commande ci-dessous.

```
git branch -m master  
ou
```

```
git checkout master
```

Renommer la branche

Nous pouvons renommer la branche à l'aide de la commande `git branch`. Pour renommer une branche, utilisez la commande ci-dessous:

```
git branch -m $ancienNomBranche $nouveauNomBranche
```

Nous renommons `$ancienNomBranche` par `$nouveauNomBranche`.

Merge Branch (fusionner des branches)

Git nous permet de fusionner l'autre branche avec la branche actuellement active. Nous pouvons fusionner deux branches à l'aide de la commande `git merge`.

```
git merge $nomBranche
```

Cette commande nous permettra de fusionner les branches.

Après utilisation, nous pouvons constater que la branche principale a fusionné avec la branche `$nomBranche`. Si nous n'avons fait aucun commit avant la fusion, l'output dans notre terminal nous affichera que nous sommes à jour.

