Les Branches de Git .

Réaliser par : Zaimi Zakia

Plan :

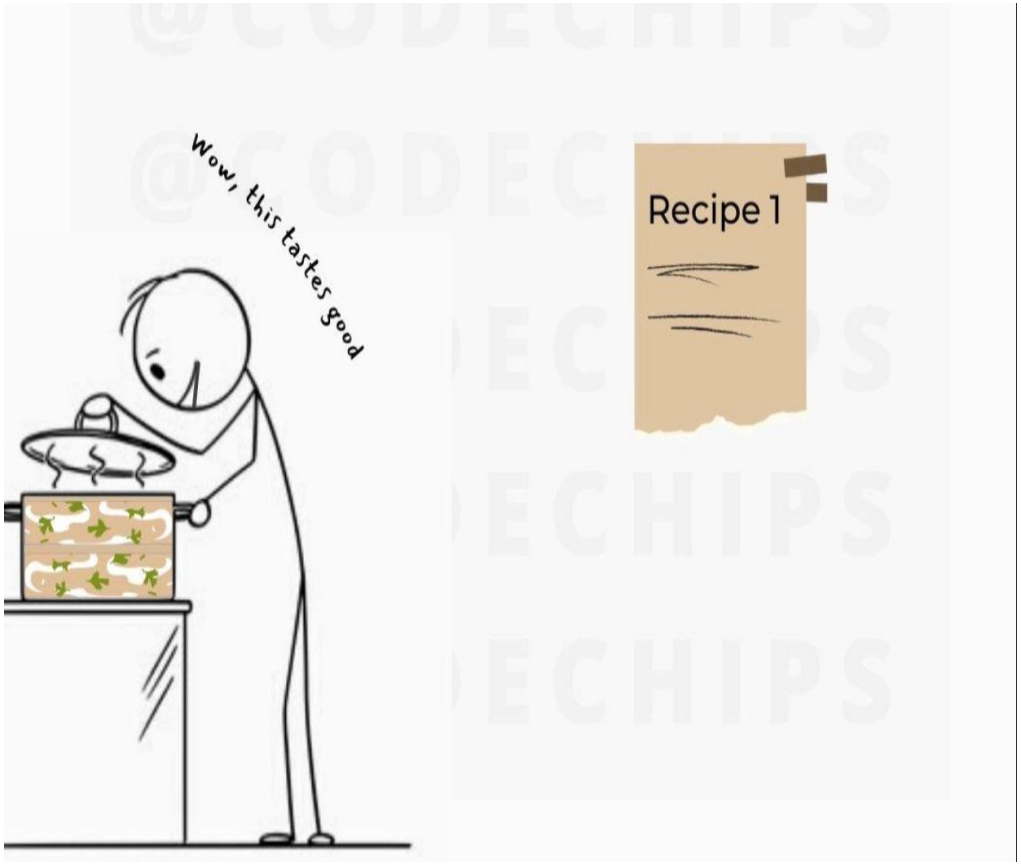
Introduction Branches

Définition branches

Création d'une branche

Basculer entre les branches

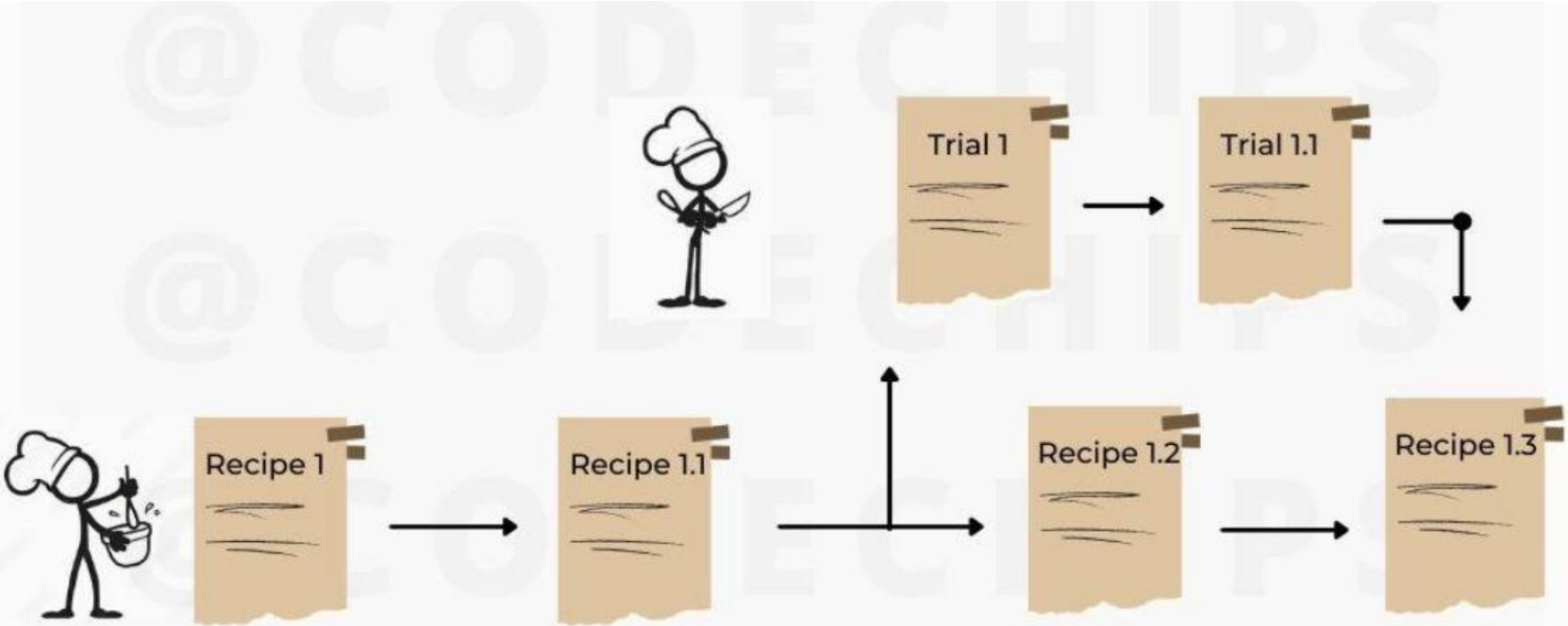
Les Commandes Git des Branches


Wow, this tastes good
Recipe 1

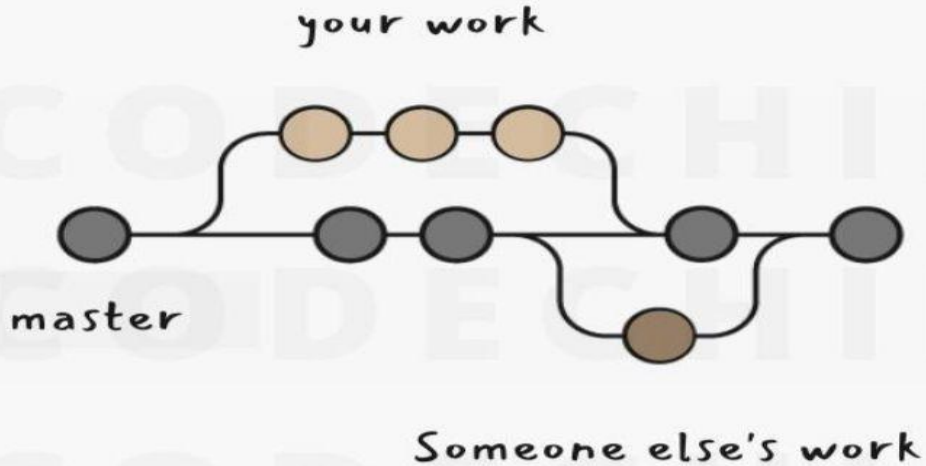

This is good too
Recipe 1.2
Add Chicken


uh oh
May be I shouldn't have mixed donuts with chicken
Recipe 1.3
Add Donuts


Add some chocolates
pizzas pls


Trial 1
Trial 1.1
Recipe 1
Recipe 1.1
Recipe 1.2
Recipe 1.3

changes without losing each other's work.



Créer une branche, c’est en quelque sorte comme créer une “copie” de votre projet pour développer et tester de nouvelles fonctionnalités sans impacter le projet de base.

Une branche, dans Git, est simplement un pointeur vers un commit (une branche n’est qu’un simple fichier contenant les 40 caractères de l’empreinte SHA-1 du commit sur lequel elle pointe).

Créer une nouvelle branche

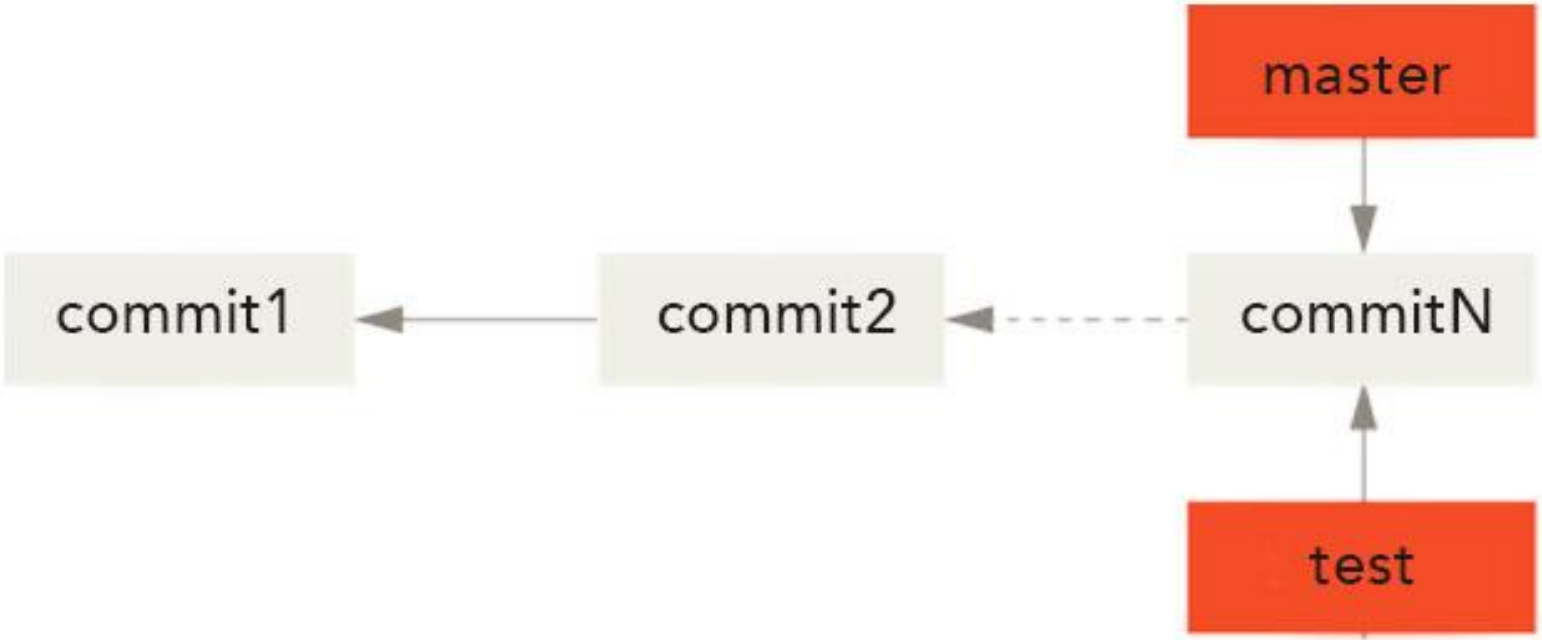
Pour créer une nouvelle branche, on utilise la commande git branch nom-de-la-branche. Cette syntaxe va créer un nouveau pointeur vers le dernier commit effectué (le commit courant).

```
pierres-macbook-pro:~ pierre$ cd desktop/git
pierres-macbook-pro:git pierre$ cd projet-git
pierres-macbook-pro:projet-git pierre$ git branch test
pierres-macbook-pro:projet-git pierre$ git status
On branch master
nothing to commit, working tree clean
```

La branche par défaut dans Git s’appelle master.

Cette branche master va se déplacer automatiquement à chaque nouveau commit pour pointer sur le dernier commit effectué tant qu’on reste sur cette branche.

Notez que la branche master n’est pas une branche spéciale pour Git : elle est traitée de la même façon que les autres branches. L’idée est que lorsqu’on tape une commande git init, une branche est automatiquement créée et que le nom donné à cette branche par Git par défaut est “master”. On pourrait très bien la renommer .

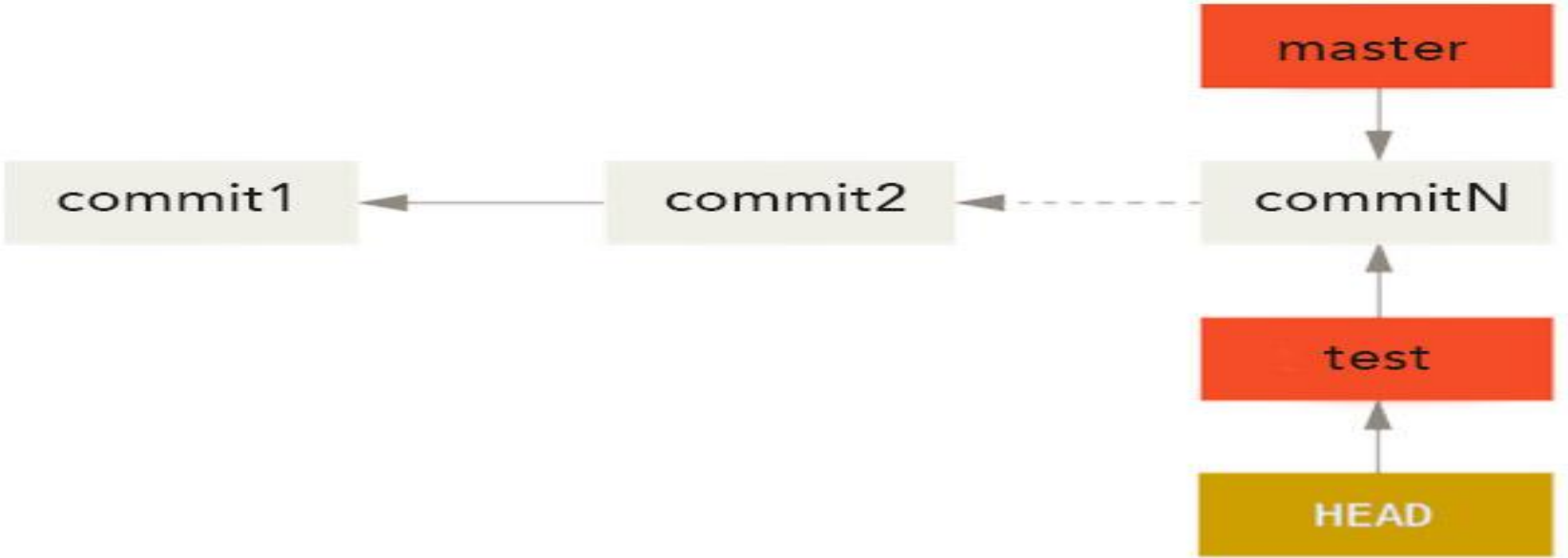

master
commit1
commit2
commitN
test

Basculer entre les branches

Pour déterminer quel pointeur vous utilisez, c'est-à-dire sur quelle branche vous vous trouvez, Git utilise un autre pointeur spécial appelé HEAD. HEAD pointe sur la branche master par défaut. Notez que la commande git branch permet de créer une nouvelle branche mais ne déplace pas HEAD.

Nous allons donc devoir déplacer explicitement HEAD pour indiquer à Git qu'on souhaite basculer sur une autre branche. On utilise pour cela la commande git checkout suivi du nom de la branche sur laquelle on souhaite basculer.

```
pierres-macbook-pro:projet-git pierre$ git checkout test
Switched to branch 'test'
pierres-macbook-pro:projet-git pierre$ git status
On branch test
nothing to commit, working tree clean
pierres-macbook-pro:projet-git pierre$
```



```
$ git checkout <existing_branch>

$ git checkout -b <new_branch>


$ git checkout -b non-existing-branch

error :Switched to a new branch 'non-existing'

Equivalent de chekout
$ git switch <existing_branch>

$ git switch -c <non_existing_branch>

$ git fetch --> switcher du local au remote

$ git checkout -t <remote_name>/<branch_name> //on prends Branch du remote en local

$ git checkout -B <branch> <start_point> // si on veut deplacer vers un commit precis
```

Merci Pour Votre Attention .