

Git and Github

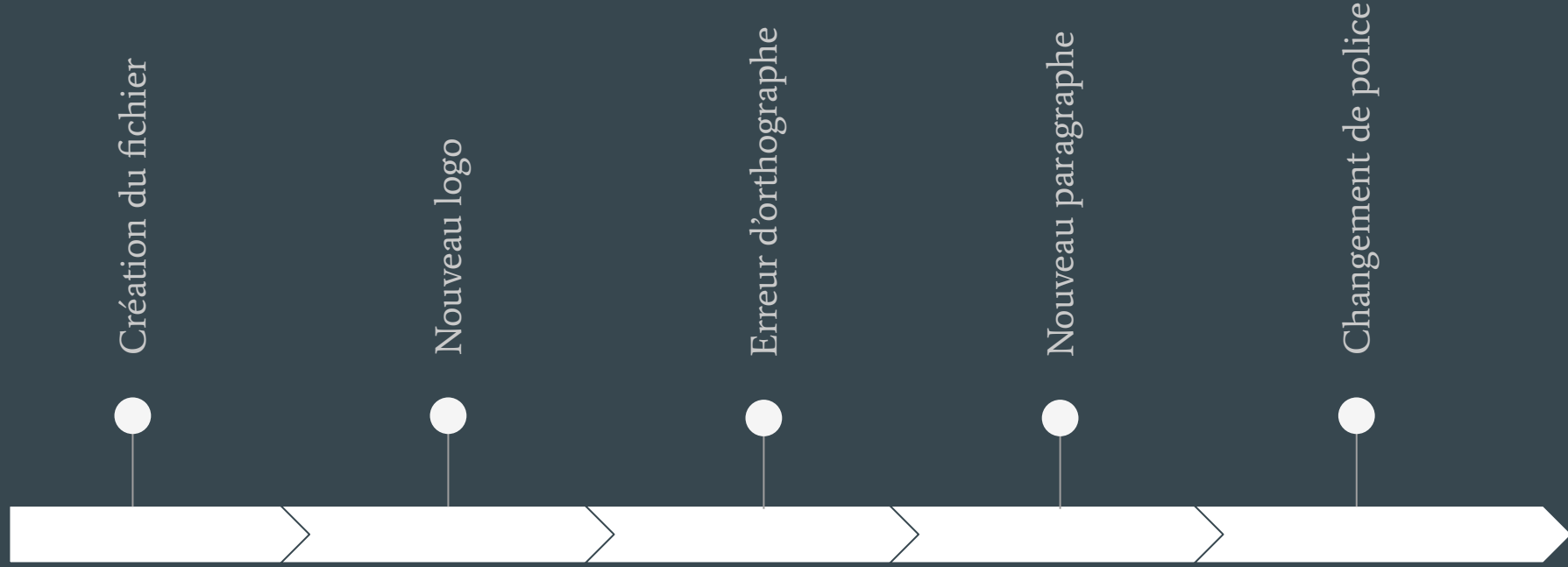


Guillaume Belleuvre - Juin 2022

Notre travail au quotidien

1. Créer un fichier
2. Sauvegarder le fichier
3. L'éditer
4. Le sauvegarder à nouveau
5. etc...

La vie d'un fichier



Contrôle de version manuel

- Rapport
- Rapport v2
- Rapport corrigé Julien
- Rapport + nouveau paragraphe
- Rapport v3
- Rapport FINAL VERSION
- Rapport FINAL 2
- ...

Peut-on automatiser cela ?

Pour chaque version du document,
nous avons besoin de savoir

1. **Quand** le fichier a été modifié ?
2. **Qu'est ce qui** a été changé ?
3. **Pourquoi** il a été modifié ?

Et nous travaillons une équipe



Création du fichier @Franck

Nouveau logo @Julien

Erreur d'orthographe @Franck

Nouveau paragraphe @Samir

Changement de police @Franck

Il y a une question de plus

Pour chaque version du document,
nous avons besoin de savoir

1. **Quand** le fichier a été modifié ?
2. **Qu'est ce qui** a été changé ?
3. **Pourquoi** il a été modifié ?
4. **Qui** a fait le changement ?

Ce dont nous avons besoin

Nous voulons un outil qui

1. Suit les versions du document
2. Garde un historique des changement du document
3. Permet de travailler en équipe

Voilà l'outil



Téléchargez et installer git



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

Donnez votre identité à git

Ouvrez un terminal et saisissez ceci en ligne de commande (CLI) :

```
git config --global user.name "Firstname Lastname"  
git config --global user.email "email@example.com"
```

Les commandes de base de git

...

Initialiser un repository

Dans le dossier de votre nouveau projet, exécutez la commande suivante :

```
git init
```

Le statut

Git peut vous dire si votre dossier a des fichiers modifiés.

```
git status
```

COMMIT

Une photographie de votre travail

Comment faire à un commit

Un commit = 2 étapes

```
git add index.html  
git commit -m "new html file"
```


Diff

Si git status vous dit que quelque chose a changé, vous pouvez savoir quoi grâce à diff :

```
git diff
```

Log

Vous pouvez connaître l'historique des commit comme ceci :

```
git log
```

BRANCH

Pour tester et travailler en équipe

UNE BRANCHE === UNE FONCTIONNALITÉ

Travailler sur une branche

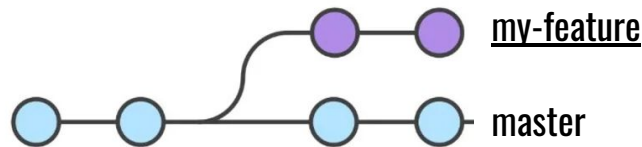
Créer une branche

```
git branch my-feature
```

Se positionner sur une branche

```
git checkout my-feature
```

On peut faire des commit sur my-feature

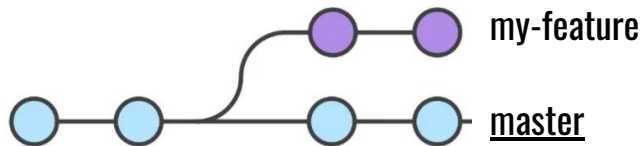


Travailler sur une branche ...

On revient sur master

```
git checkout master
```

On peut faire des commit sur master



Merge

Se positionner sur la branche de destination

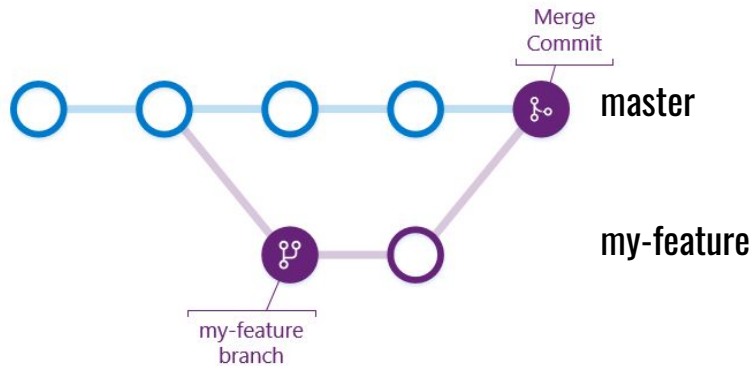
```
git checkout master
```

Fusionne la branche spécifié dans la branche courante.

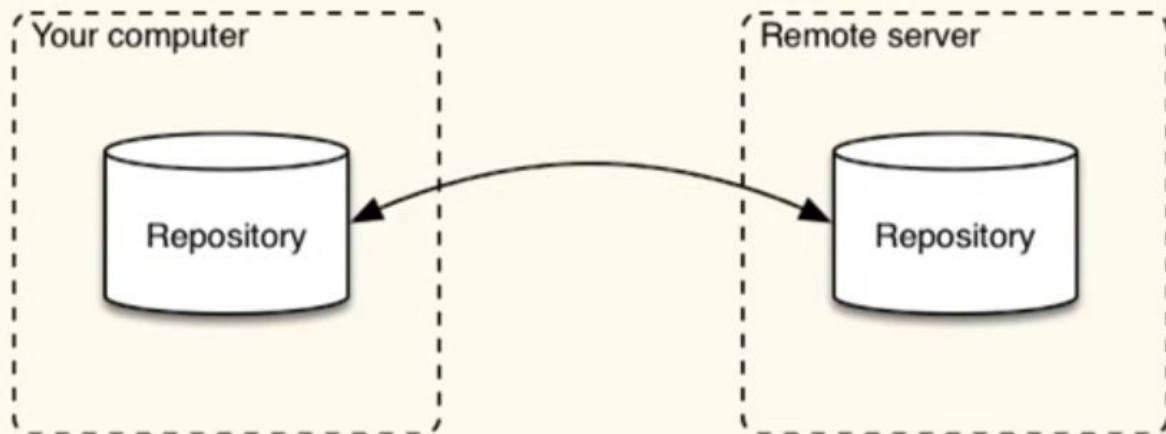
```
git merge my-feature
```

Supprime la branche

```
git branch -d my-feature
```



REMOTE



Il nous faut un repository distant

Allez sur [GitHub.com](https://github.com) vous créer un compte



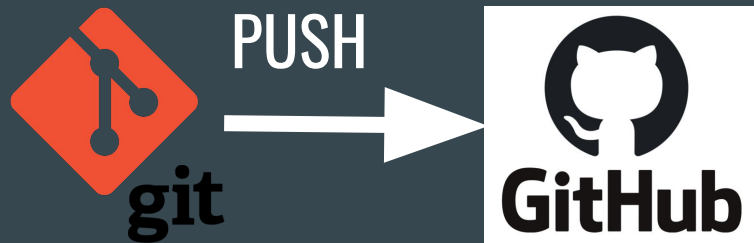
Connecter le repository local à un repository distant

1. Créer un repository public sur votre compte GitHub
2. Associer nos 2 repositories

```
git remote add origin https://github.com/...
```

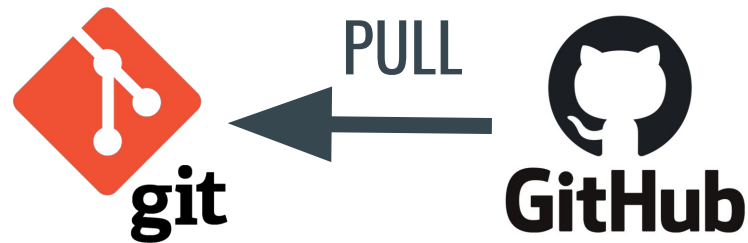


Envoyer le code sur GitHub



```
git push origin master
```

Récupérer le code depuis GitHub



```
git pull origin master
```

PULL REQUEST

Faire une PULL REQUEST pour

- Gérer le merge d'une branche sur le repository distant
- Informer son équipe de la fin du développement d'une fonctionnalité
- Demander une relecture de son code à son équipe
- Débattre entre développeurs sur l'utilité d'un code

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).



base: master ▾



compare: column-display ▾

✓ **Able to merge.** These branches can be automatically merged.

Discuss and review the changes in this comparison with others.

[Learn about pull requests](#)

Create pull request

1 commit

1 file changed

1 contributor



Commits on Jun 14, 2022

change flex properties en body



GBAfpa committed 29 seconds ago