

Programmation Web – Avancé

JavaScript & Node.js

Introduction à Git



Attribution –
Partage dans les
Mêmes Conditions
4.0 International
(CC BY-SA 4.0)

*Presentation template
by [SlidesCarnival](#)*



Introduction à Git

Pour développer où que vous soyez tout en traçant vos versions de codes...



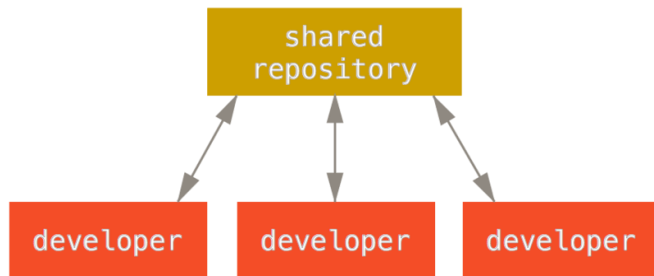
Git

- Open Source Distributed Version Control System
- Gestion de projets logiciels et de leurs fichiers via le tracking des changements
- Structure de données traquant les changements :
 - repository incluant des « commit objects » et des références vers ces « commit objects » (heads)
 - **.git** dans sous-dossier du projet



Git workflows

- Possibilité de nombreux workflows
 - Commandes de bases pour un workflow local
 - Distributed workflows



Workflow centralisé [103.]



Git

- Pro Git book [\[103.\]](#)
- Git Cheat Sheet [\[106.\]](#)
- Utilisation du contrôle de version dans VS Code [\[104.\]](#)



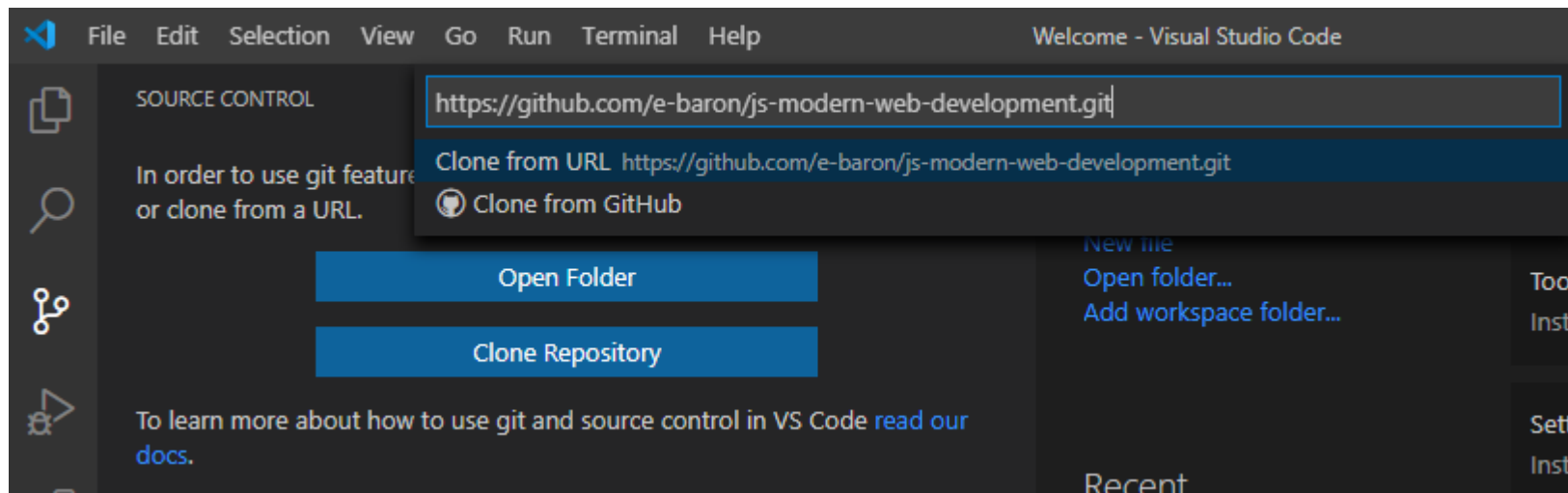


Repository

- Sous-dossiers & fichiers d'un dossier projet contenant :
 - toutes les versions des fichiers du projet
 - tout l'historique du projet



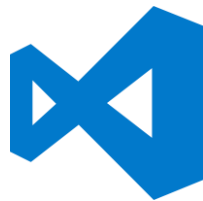
Obtenir la copie d'un projet existant





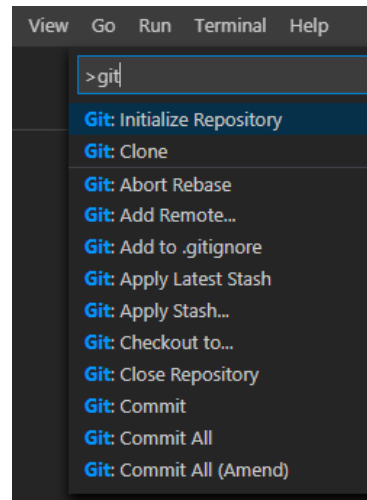
Obtenir la copie d'un projet existant

- `git clone https://gitlab.vinci.be/web2/js-modern-web-development.git`
`[your_project_name]`



Créer un repository local à partir d'un dossier d'un projet

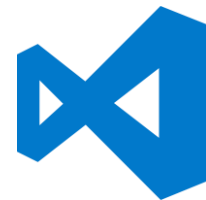
- Si projet non existant, création du répertoire du projet via VS Code ou l'Explorateur de Fichiers
- **Ctrl + Shift + P** ou **View, Command Palette...**
- **Git: Initialize Repository**



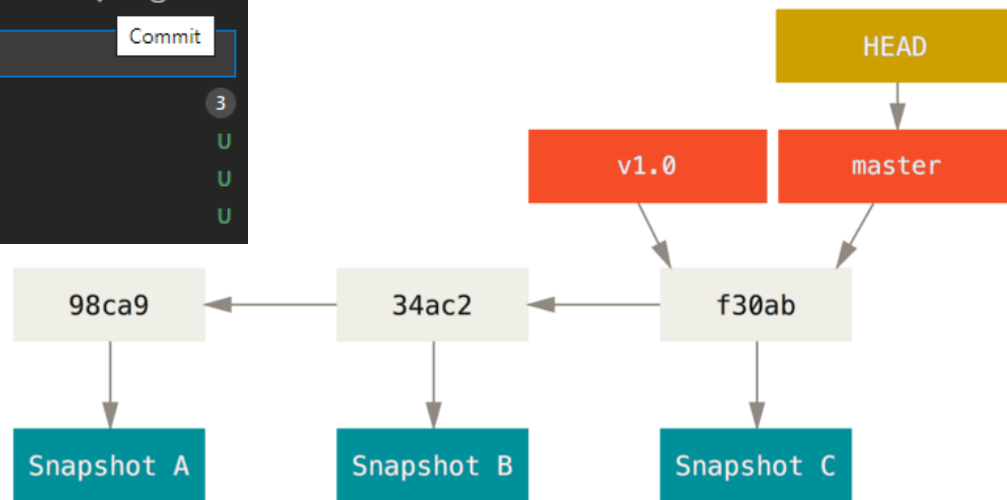
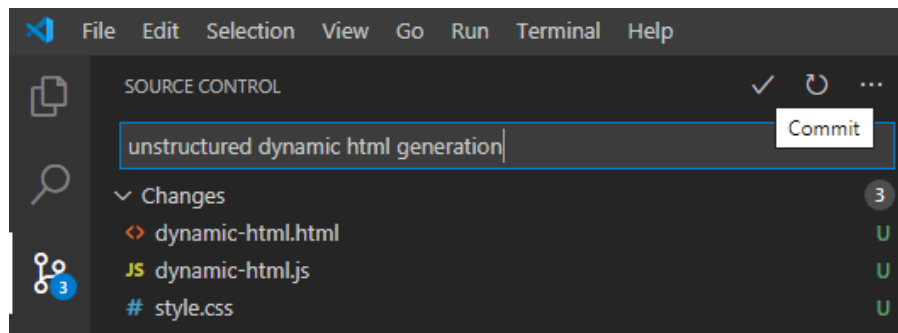


Créer un repository local à partir d'un dossier d'un projet

- Si projet non existant : **mkdir project_name**
- **cd project_name**
- Créer le repository (.git) : **git init**
- Configuration de l'information de l'utilisateur pour tous les repos locaux :
 - **git config [--global] user.name "yourName"**
 - **git config [--global] user.email "your@email"**



Enregistrements de changements dans le repository local



Historique des commits d'une branche [103.]



Enregistrements de changements dans le repository local

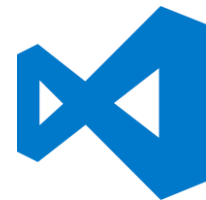
- Tracking de tous les nouveaux fichiers et « staging » de tous les fichiers modifiés : **git add .**
- Sauvegarde du groupe de changements dans votre repository local : **git commit -m "unstructured dynamic html generation"**



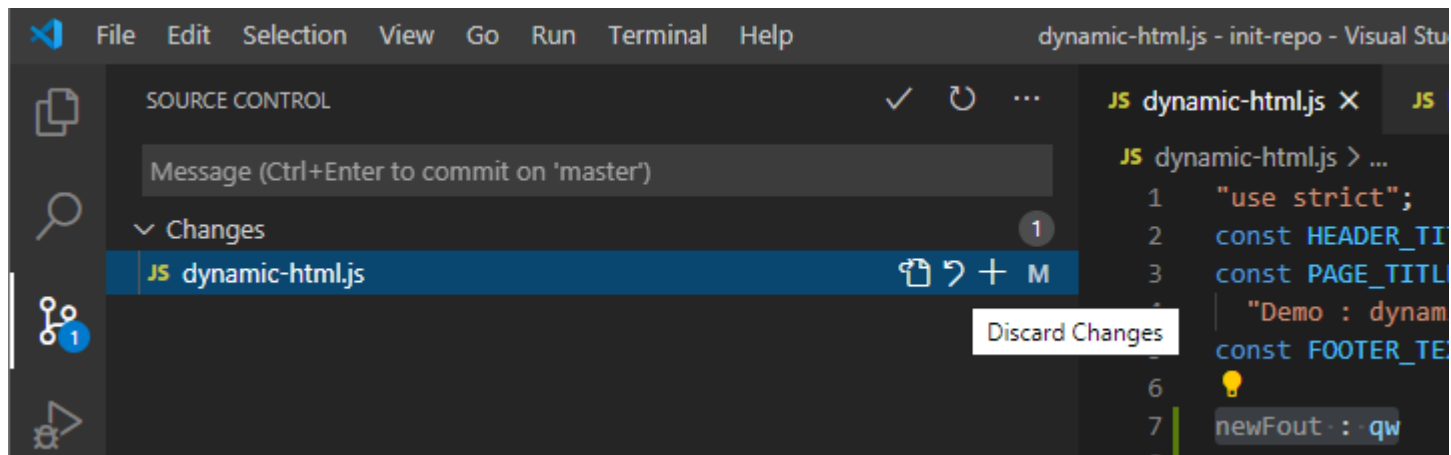
Ignorer des dossiers & fichiers pour les écarter du repository

- Indication des dossiers et fichiers à ignorer :
.gitignore

```
# ignore all node_modules folders and nested folders & files
node_modules
# ignore the users.json data file
users.json
# ignore the specific VS Code configuration parameters
.vscode
# ignore only the course-
input folder in the current directory, not subdir /course-website
/course-input
```



Revenir à la version du fichier au dernier commit





**Revenir à la version du fichier
au dernier commit**

◎ `git checkout -- dynamic-html.js`



Revenir à la version du fichier lors d'un commit éloigné

- **git log** : liste de tous les commits identifiés par leur **sha**
- **git checkout <sha> [-- filename]** :
git checkout 2d93 -- dynamic-html.js

```
...
commit 9f8443a03ef4ee394e4741291271e7d5b98bb1bd
Author: e-baron <raphael.baroni@vinci.be>
Date:   Fri Aug 14 11:42:42 2020 +0200

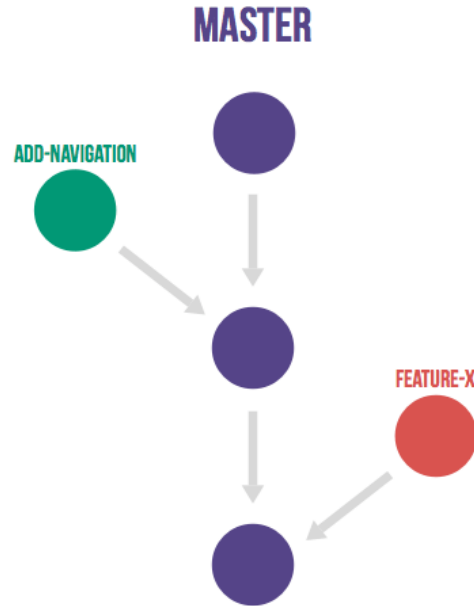
    2n fout added

commit 2d93c56308d45211c0db7a0538e7b7579e91f768
Author: e-baron <raphael.baroni@vinci.be>
Date:   Fri Aug 14 11:12:06 2020 +0200

    ignore .vscode directory
...
```



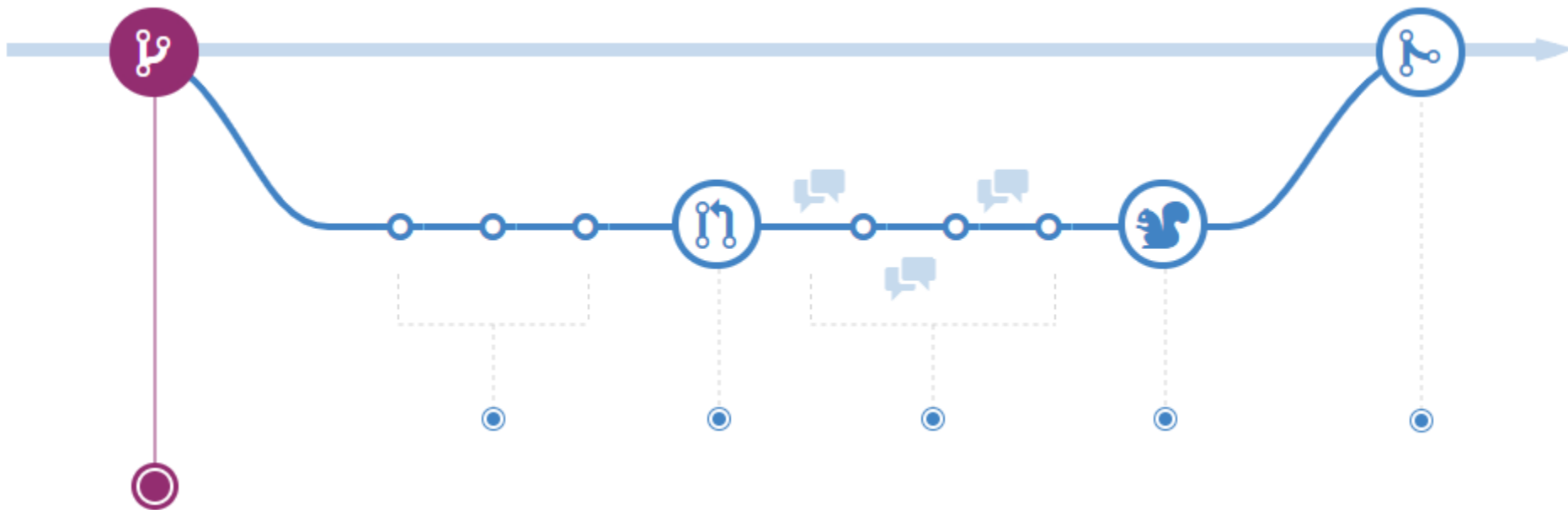

Git workflows et les branches



GitHub flow : feature branches & master branch [105.]



Git workflows et les branches



Comprendre le GitHub flow [106.]



Exemple de workflow pour collaborer autour d'une feature branch

- ① 1) Création d'un web repository à partir d'un repository local :
 - Synchronisation du web repository à partir du repository local
- ② 2) Création de feature branches locales par plusieurs développeurs
- ③ 3) Push des feature branches sur un web repo
 - Synchronisation du web repository à partir du repository local



Exemple de workflow pour collaborer autour d'une feature branch

- ◎ 4) **Merge Request** (GitLab) ou **Pull Request** (GitHub) de la feature branch dans le master
- ◎ 5) Discussions, code review... :
 - Synchronisation du local repository à partir d'un web repository
 - Synchronisation du web repository à partir du repository local
 - ...



Exemple de workflow pour collaborer autour d'une feature branch

- 6) Intégration d'une feature branche sur le master du web repo
 - Synchronisation du local repository à partir d'un web repository
 - Gestion des conflits localement
 - Push du master sur le web repo
- 7) Fermeture de la Merge Request (GitLab) ou de la Pull Request (GitHub)



Exemple de workflow pour collaborer autour d'une feature branch

- 1) Création d'un web repository à partir d'un repository local



Créer un web repository à partir d'un repository local

● Création d'un **Project** à partir de GitLab :

- <https://gitlab.vinci.be/>
- New project : **git-tuto**



● Création d'un Repository à partir de GitHub :

- <https://github.com/>
- New : **git-tuto**



Create a new repository

```
git clone git@gitlab.vinci.be:e-baron/git-tuto.git
cd git-tuto
touch README.md
git add README.md
git commit -m "add README"
git push -u origin master
```

Push an existing folder

```
cd existing_folder
git init
git remote add origin git@gitlab.vinci.be:e-baron/git-tuto.git
git add .
git commit -m "Initial commit"
git push -u origin master
```

Push an existing Git repository

```
cd existing_repo
git remote rename origin old-origin
git remote add origin git@gitlab.vinci.be:e-baron/git-tuto.git
git push -u origin --all
git push -u origin --tags
```



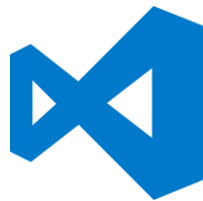
Créer un web repository à partir d'un repository local

- Paramétrer votre compte sous GitLab ou GitHub :
 - Pour connexion https :
 - Ajout d'un password
 - Pour connexion ssh :
 - Génération d'une paire de clé public / privé (Git Bash, ssh-keygen.exe)
 - Copie de la clé public au sein de GitLab ou GitHub



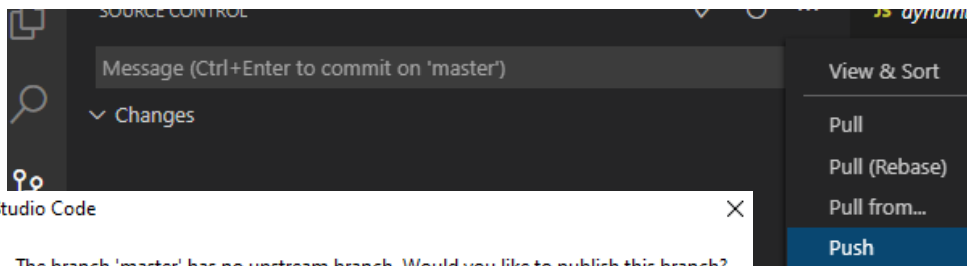
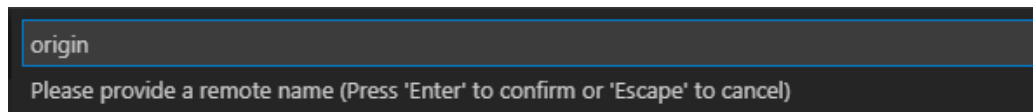
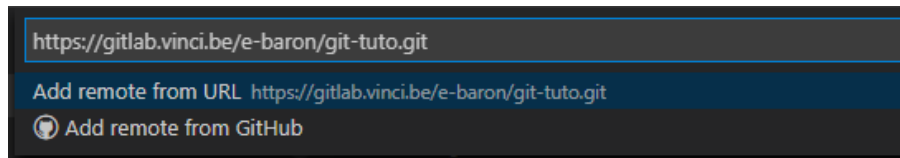
Exemple de workflow pour collaborer autour d'une feature branch

- 1) Création d'un web repository à partir d'un repository local :
 - Synchronisation du web repository à partir du repository local



Synchronisation du web repository à partir du repository local

- Ajout d'un **remote** repository (1^{ère} fois seulement) : **Ctrl + Shift + P**, **Git : Add Remote...**
- Push : **...**, **Push**



Visual Studio Code



The branch 'master' has no upstream branch. Would you like to publish this branch?

OK

Cancel



Synchronisation du web repository à partir du repository local

- Push : envoi des commit locaux dans le remote repo
- Code donné par GitLab ou GitHub



```
cd existing_repo
git remote rename origin old-origin
# SSH connexion : git remote add origin git@gitlab.vinci.be:e-baron/git-tuto.git
git remote add origin https://gitlab.vinci.be/e-baron/git-tuto.git
git push -u origin --all
git push -u origin --tags
```



```
git remote add origin https://github.com/e-baron/git-tuto
git git push -u origin master
```



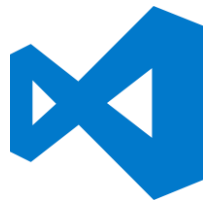
Synchronisation du web repository à partir du repository local

- Gestion d'un remote repository :
 - **remote** : version d'un projet hébergée sur le web ou un réseau
 - Ajout d'une remote : **git remote add mygithub**
<https://github.com/e-baron/git-tuto>
 - Liste des remotes : **git remote -vv**
 - Changement du nom : **git remote rename mygithub origin**
 - Suppression d'une remote : **git remote remove mygithub**
- Push de toutes les branches locales et tracking de toutes les branches distantes : **git push -u origin --all**



Exemple de workflow pour collaborer autour d'une feature branch

- 2) Création de feature branches locales par plusieurs développeurs



Création et utilisation d'une feature branche locale

- Soit ... , **Branch, Create Branch...**
- Soit Création & Utilisation de la branche : ... , **Checkout to...**
- NB : Création d'une branche : **Ctrl + Shift + P**, **Git : Create branche...**

add-navigation

Please provide a new branch name (Press 'Enter' to confirm or 'Escape' to cancel)

Select a ref to checkout

+ Create new branch...

+ Create new branch from...

add-navigation b8e271a4

master b8e271a4

origin/master Remote branch at b8e271a4



Création et utilisation d'une feature branche locale

- Option a :
 - Création et utilisation d'une nouvelle branche :
git checkout -b add-login
- Option b :
 - Création d'une branche : **git branch add-login**
 - Switch vers la nouvelle branche et MàJ des « working files » : **git checkout add-login**



Autres commandes utiles pour les branches

- Visualisation de toutes les branches
 - locales : **git branch -vv**
 - distantes : **git branch -rv**
- Effacer la branche spécifiée : **git branch -d add-login**



Exemple de workflow pour collaborer autour d'une feature branch

- 3) Push des feature branches sur un web repo
 - Synchronisation du web repository à partir du repository local



Synchronisation du web repository à partir du repository local

- ☉ Switch vers la feature branche locale : ... , **Checkout to...**
- ☉ Création d'une branche distante à tracker par votre repository local : ... , **Push...**



Synchronisation du web repository à partir du repository local

- Push d'une branche particulière :
 - `git checkout add-login`
 - `git push origin add-login`
- Configuration de la « upstream branch », la branche distante trackée par votre branche local [\[108.\]](#) : `git push -u origin add-login`
- Visualisation des branches trackées : `git branch -vv`
- Push de toutes les branches trackées : `git push [remote]`



Exemple de workflow pour collaborer autour d'une feature branch

- 4) **Merge Request** (GitLab) ou **Pull Request** (GitHub) de la feature branch dans le master



Créer une Merge Request ou Pull Request

- Sous GitLab :
 - Create merge request : e.g. From correct-bug into master
- Sous GitHub
 - Compare & pull request : base : master
compare : correct-bug



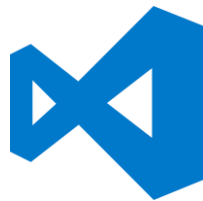
Exemple de workflow pour collaborer autour d'une feature branch

- 5) Discussions, code review... :
 - Synchronisation du local repository à partir d'un web repository
 - Synchronisation du web repository à partir du repository local
 - ...



Synchronisation du local repository à partir d'un web repository

- Pull : récupération de tous les changements et intégration des changements dans la branche en cours local



Synchronisation du local repository à partir d'un web repository

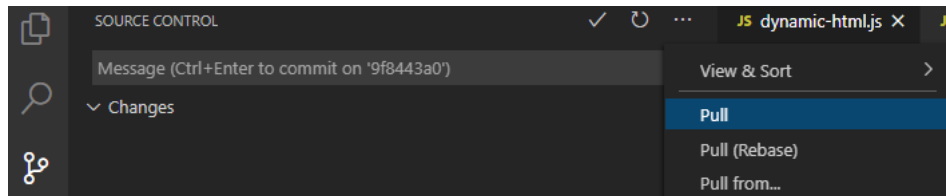
- Cas 1) Pull d'une branche distante (e.g. origin/master) dans la branche en cours portant le même nom (master)
- Pull (**fetch** et **merge** de la branche en cours) : ... , **Pull from...**
- Autre option : ... , **Pull**

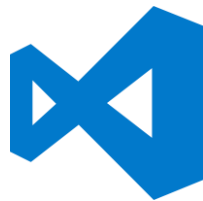
Pick a remote to pull the branch from

origin https://gitlab.vinci.be/e-baron/git-tuto.git

Pick a branch to pull from

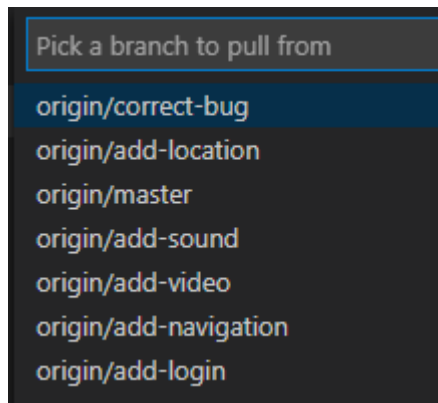
origin/master

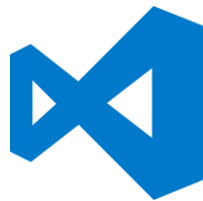




Synchronisation du local repository à partir d'un web repository

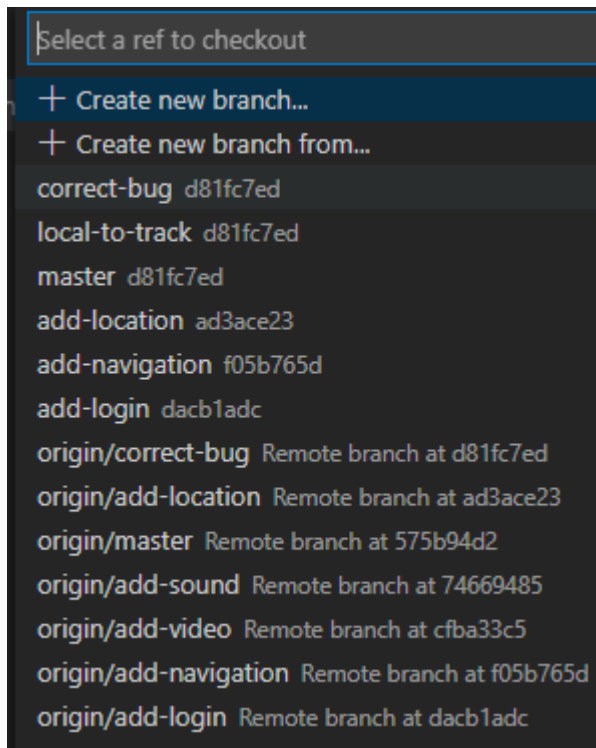
- Cas 2) Pull d'une branche distante (e.g. origin/correct-bug) non encore ajoutée localement & intégration dans la branche en cours
 - Obtenir tous les changements : ... , **Fetch**
 - Pull de la branche : ... , **Pull from...**





Synchronisation du local repository à partir d'un web repository

- Cas 3) Pull d'une branche distante (e.g. add-video) non encore ajoutée localement avec création d'une branche locale portant le même nom
 - Obtenir tous les changements : ... , **Fetch**
 - Création d'une nouvelle branche & intégration des changement : ... , **Checkout from...**





Synchronisation du local repository à partir d'un web repository

- Vérification que le lien vers le remote repository est fait : **git remote -v**



Synchronisation du local repository à partir d'un web repository

- Cas 1) Pull d'une branche distante (e.g. origin/master) dans la branche en cours portant le même nom (master)
 - Option 1 :
 - Récupération de tous les changements et intégration des changements dans la branche en cours du local repo :
 - `[git add . && git commit -m "prior to merge"]`
 - `git pull [origin master]`



Synchronisation du local repository à partir d'un web repository

- Cas 1) Pull d'une branche distante (e.g. origin/master) dans la branche en cours portant le même nom (master)
 - Option 2 :
 - Récupération de tous les changements sans mise à jour des « working files » : **git fetch**
 - Intégration des changements faits dans une branche distante vers la branche en cours :
 - **[git add . && git commit -m "prior to merge"]**
 - **git merge [master]**



Synchronisation du local repository à partir d'un web repository

- Cas 2) Pull d'une branche distante (e.g. origin/correct-bug) non encore ajoutée localement & intégration dans la branche en cours
 - Récupération des changements : **git fetch**
 - Intégration des changements faits dans une branche distante vers la branche en cours :
 - [**git add . && git commit -m "prior to merge"**]
 - [**git checkout master**]
 - **git merge origin/correct-bug**



Synchronisation du local repository à partir d'un web repository

- Cas 3) Pull d'une branche distante (e.g. add-video) non encore ajoutée localement avec création d'une branche locale portant le même nom
 - Récupération des changements : **git fetch**
 - Intégration des changements faits dans une branche distante vers une nouvelle branche :
 - **git checkout add-video**



Exemple de workflow pour collaborer autour d'une feature branch

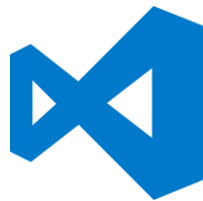
- 6) Intégration d'une feature branche sur le master du web repo
 - Synchronisation du local repository à partir d'un web repository
 - Gestion des conflits localement
 - Push du master sur le web repo



Intégration d'une feature branche sur le master du web repo

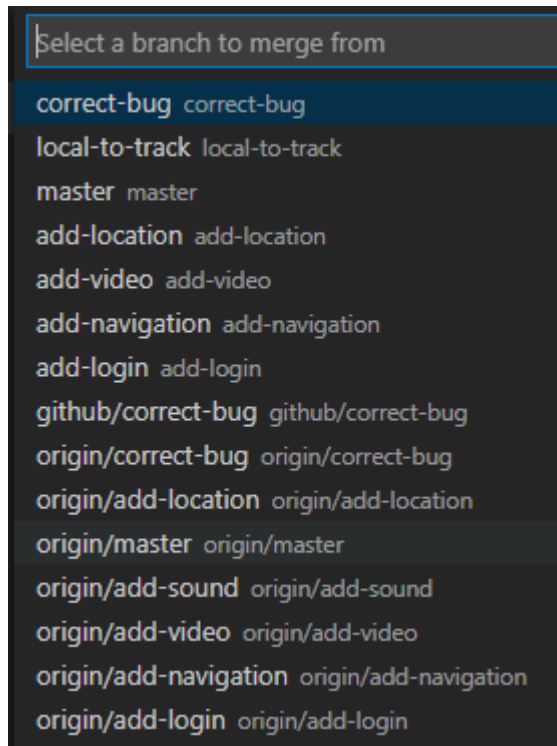
- Synchronisation du local repository à partir d'un web repository :
 - Synchro de la branche master pour avoir la dernière version distante
 - Synchro de la feature branche pour avoir la dernière version localement si pas votre branche
 - Intégration des changements de la feature branche locale dans la master locale :
 - **git checkout master**
 - **git merge correct-bug**





Intégration d'une feature branche sur le master du web repo

- Switch vers la branche master locale : ... , **Checkout from...**
- Intégration des changements : ... , **Branch** , **Merge Branch...**





Intégration d'une feature branche sur le master du web repo

● Gestion des conflits localement

- Exemple avec conflit :
 - Démarrage de deux nouvelles features en // :
add-home & **add-about**
 - **add-home** change le titre du site
 - Merge de **add-home** dans le master
 - **add-about** : change aussi le titre du site
 - Merge de **add-about** dans le master

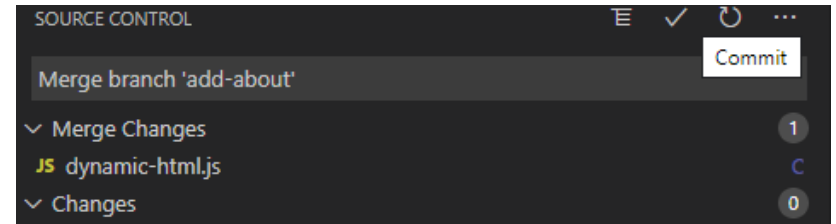




Gestion des conflits

- Etat du code sans merge de la feature branche : **Accept Current Change**
- Etat du code si merge de la feature branche : **Accept Incoming Change**
- Sauver les fichiers & commit des changements pour finalisation du merge

```
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes | Start Live Merge  
<<<<< HEAD (Current Change)  
nst HEADER_TITLE = "AMAZING JavaScript & Node.js full course with home menu";  
=====  
nst HEADER_TITLE = "JavaScript & Node.js full course : about added in title";  
>>>> add-about (Incoming Change)  
nst PAGE_TITLE
```





Intégration d'une feature branche sur le master du web repo

- Exemple sans conflit :
 - Démarrage de deux nouvelles features en // :
add-home & **add-about**
 - **add-home** change le titre du site
 - Merge de **add-home** dans le master
 - **add-about** : ne modifie pas le titre du site, ajoute simplement une description
 - Merge de **add-about** dans le master



Exemple de workflow pour collaborer autour d'une feature branch

- 6) Intégration d'une feature branche sur le master du web repo
 - Push du master sur le web repo



Exemple de workflow pour collaborer autour d'une feature branch

- 7) Fermeture de la Merge Request (GitLab) ou de la Pull Request (GitHub)