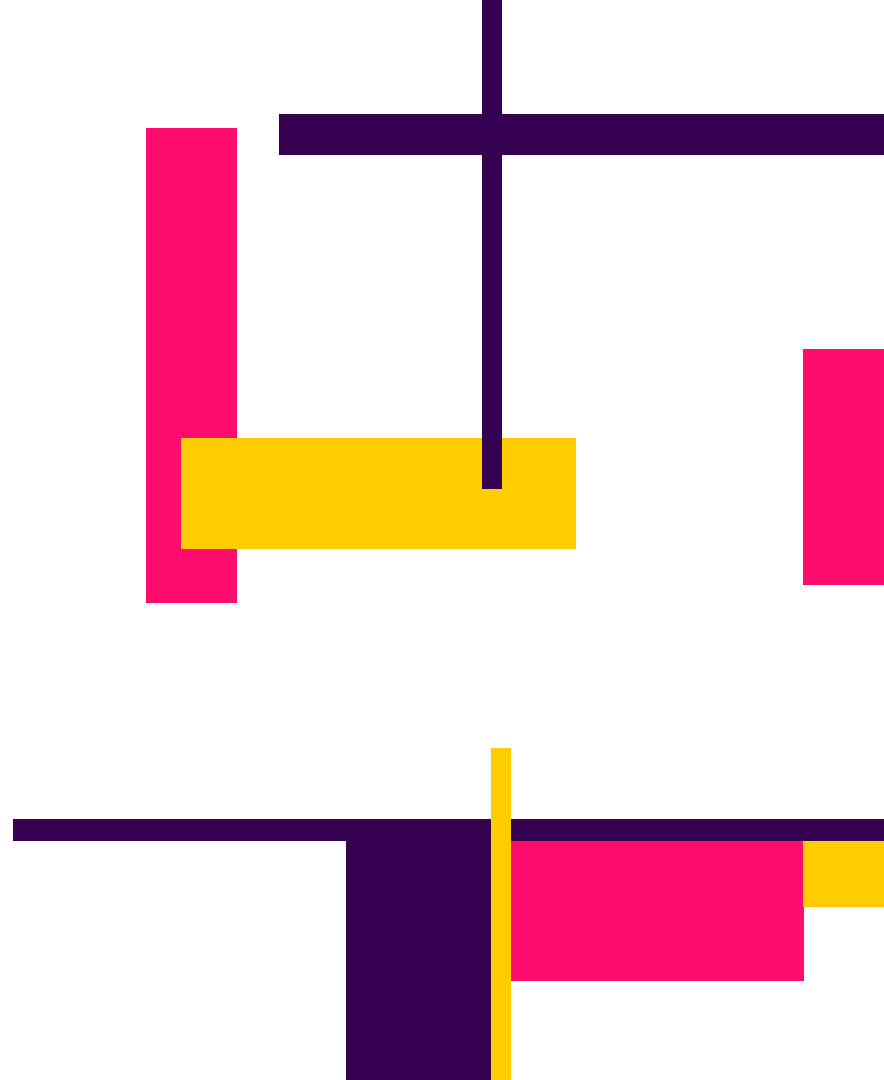


PyCharm



Il existe différentes versions de PyCharm

- Une version pour l'éducation (ce n'est pas pour développer)
- Une version communautaire (gratuite)
- Une version professionnelle (payante)

Bien qu'un peu plus limitée, la **version communautaire suffit amplement**

Peut s'exécuter sur Windows, MacOS et Linux



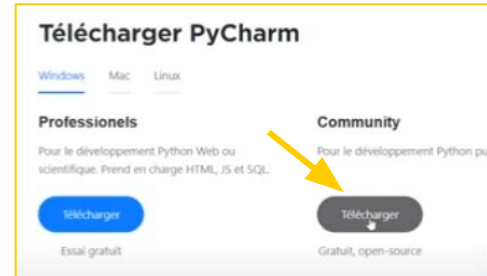
Voici le site officiel de téléchargement de PyCharm

<https://www.jetbrains.com/fr-fr/pycharm/>

Télécharger la version communautaire de cet IDE.

IDE: **Environnement de Développement Intégré.**

C'est une application qui permet aux développeurs de créer (développer) efficacement un code logiciel.



Lors de l'installation:

- Validez le chemin proposé pour installer PyCharm par défaut ou saisissez-en un autre.

Suivant

- Cochez la case pour avoir un raccourci sur le bureau

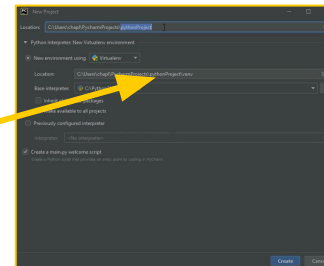
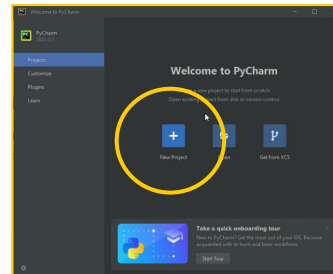
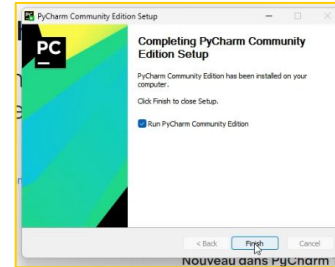
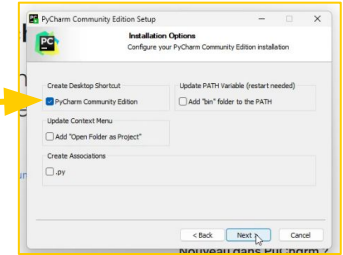
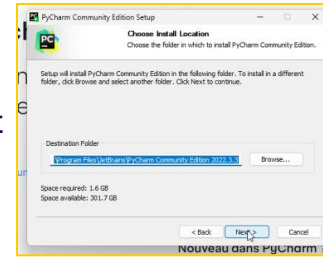
Suivant

- L'installation se charge puis ouvrez l'application

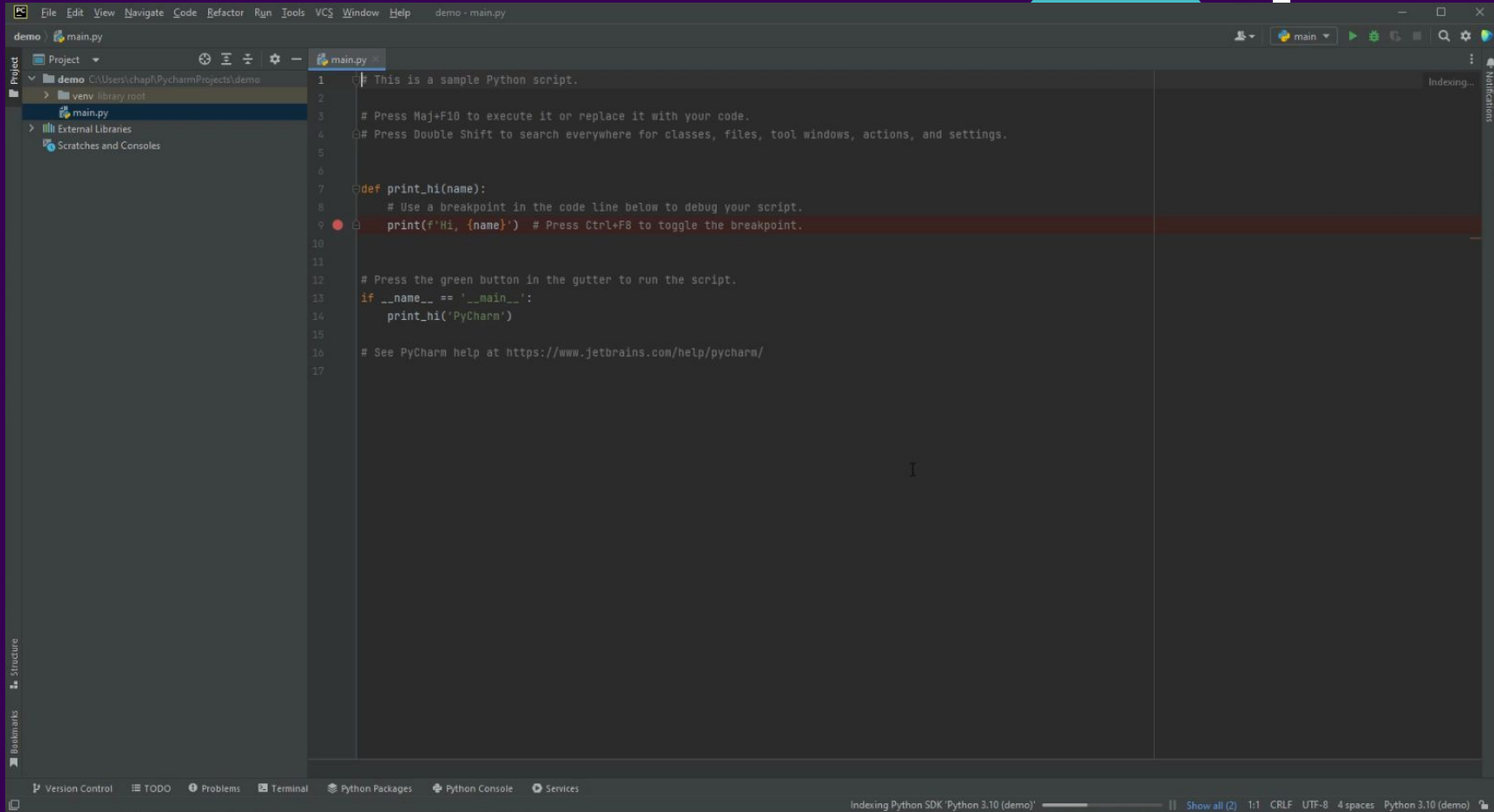
Suivant

- Cliquez sur *Nouveau Projet*

- Nommez votre projet

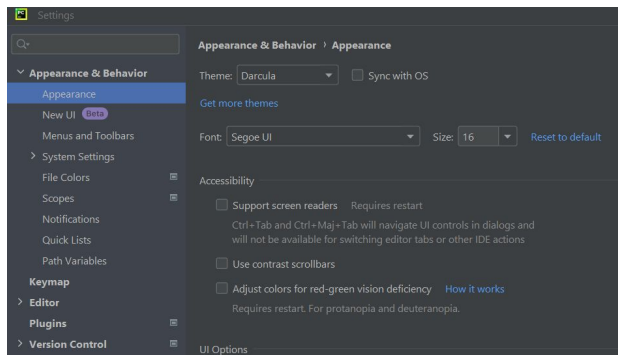


Voici l'interface de PyCharm



On peut modifier le thème de PyCharm

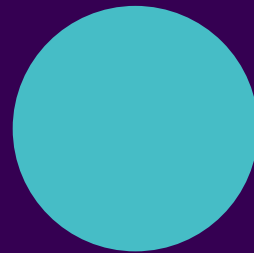
File / Settings / Appearance & Behavior / Appearance



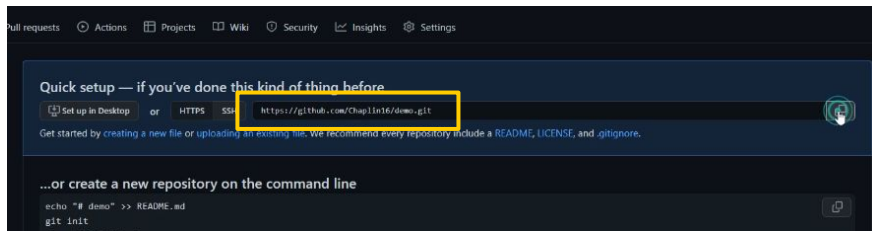
- 3 thèmes au choix : IntelliJ Light (très clair)
Darcula (sombre)
High Contrast (gros contraste)
- Dessous le thème: réglage de la police et de la taille de l'écriture



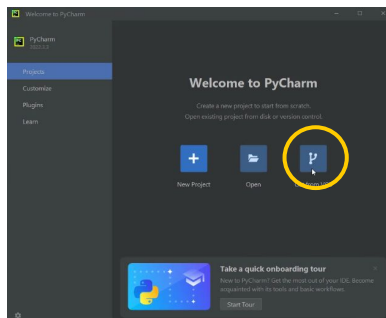
Utilisez Git avec PyCharm



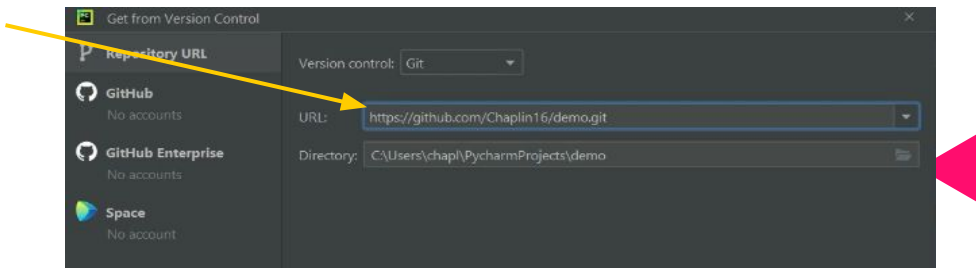
Créez un repo sur Github et copiez l'URL de votre dépôt distant

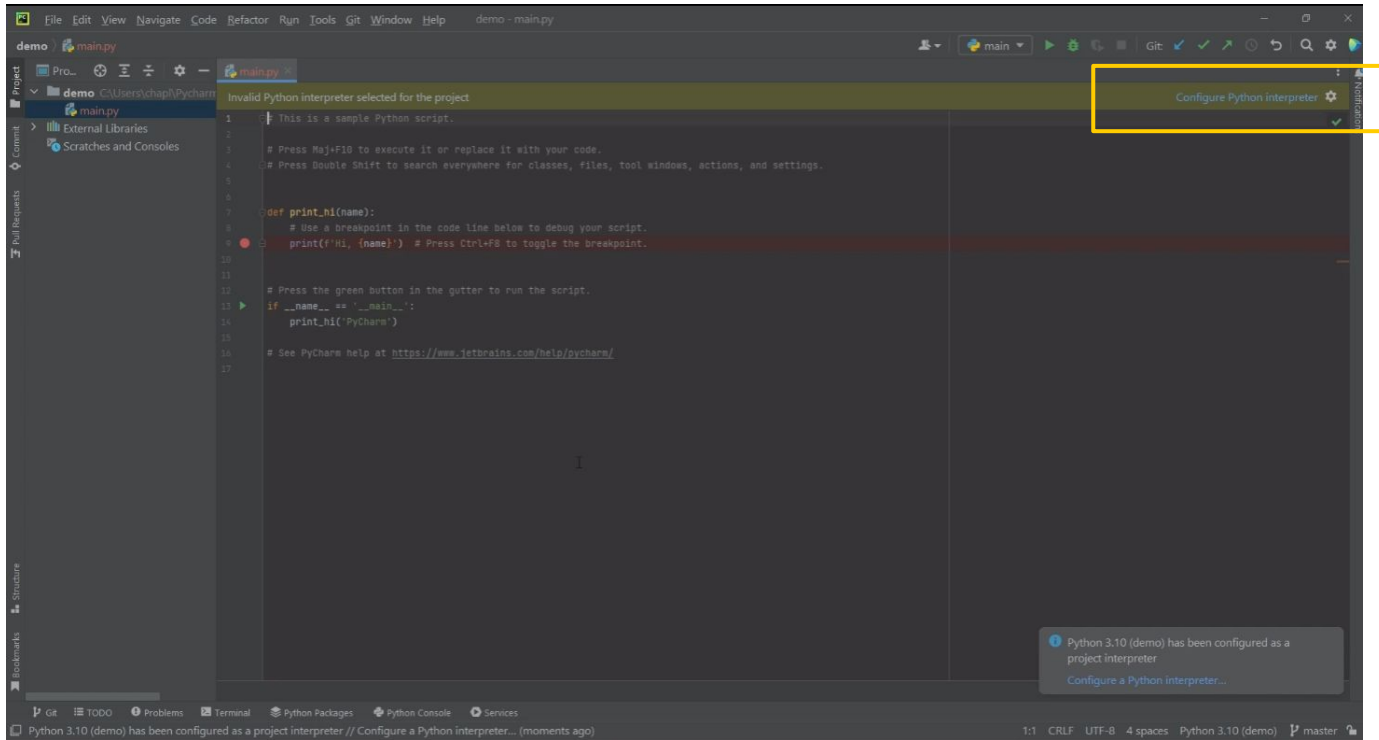


Ouvrir PyCharm avec *Get from VCS*



Coller le chemin de votre dépôt distant





S'il faut configurer Python:

Cliquez sur *Configure Python Interpreter*

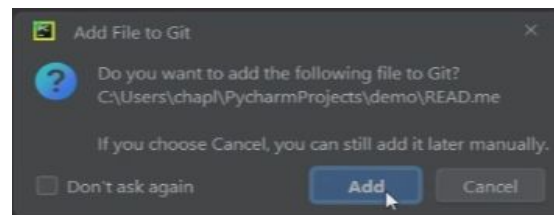
Ou

Allez dans *File / Settings / Project Demo / Python Interpreter*

Menu déroulant: *Add Interpreter / sélectionner Add Local Interpreter puis Valider*

Ajouter / suivre dans le dépôt distant

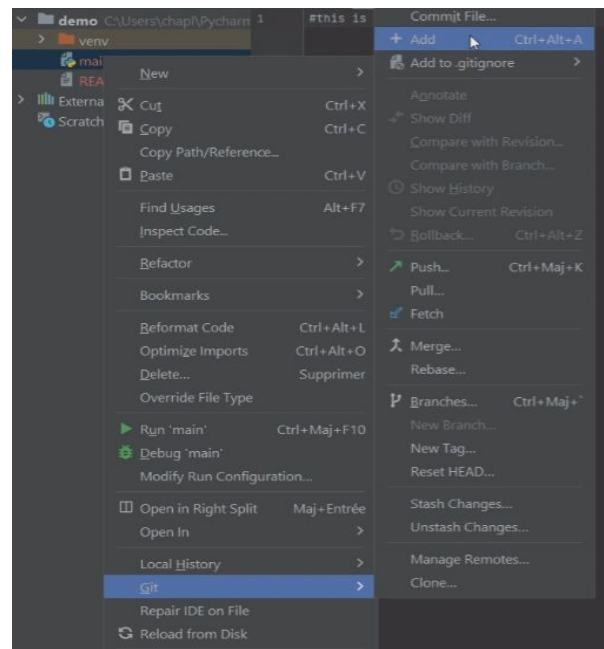
A la création d'un dossier, d'un fichier, PyCharm propose de l'ajouter / de le suivre (= git add)



Autrement :

Click droit / descendre dans le menu sur Git et add

Il existe un raccourci Ctrl + Alt + A

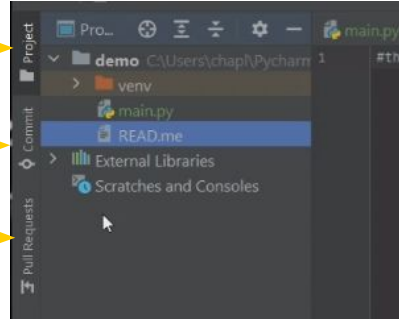


Commiter des changements

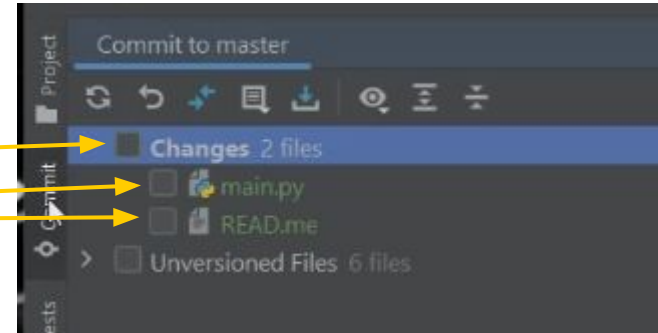
Explorateur de fichiers

Commiter

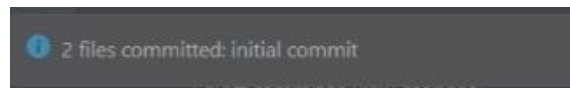
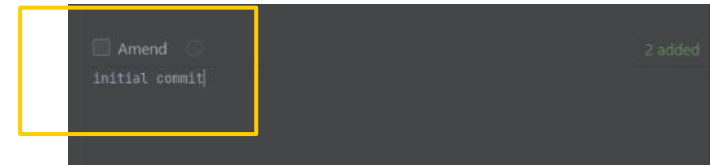
Pull Requests



En cliquant sur *Commit*, les changements s'affichent avec à gauche une case à cocher. Cochez les tous ensemble ou à l'unité



Inscrivez le nom du commit décrivant les actions que vous venez de réaliser puis cliquez sur *Commit*
Une pop-up de confirmation apparaît.

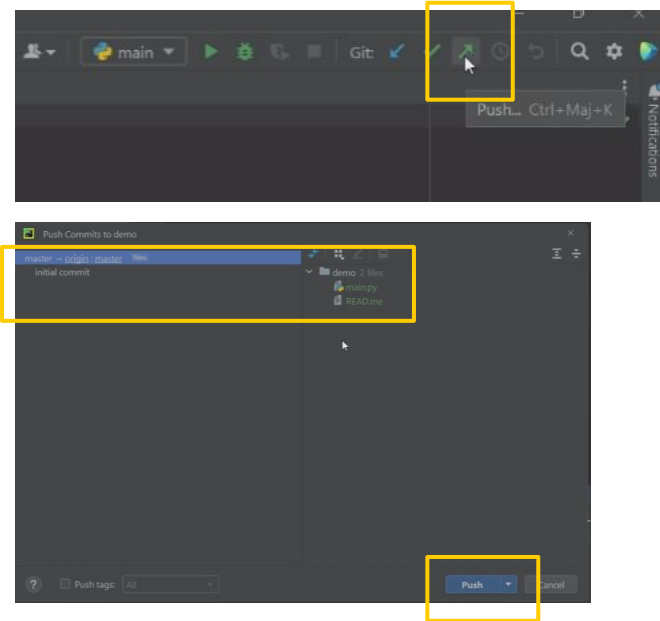
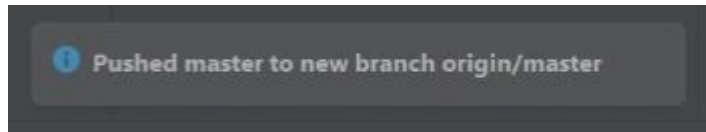


Pusher les commits

En haut, à gauche, la flèche verte permet de pusher

Une fenêtre récapitulant le dossier, le nom du commit, la branche et les éléments prêts à être pushés apparaît. Cliquez sur *Push*

Une pop-up de confirmation apparaît.



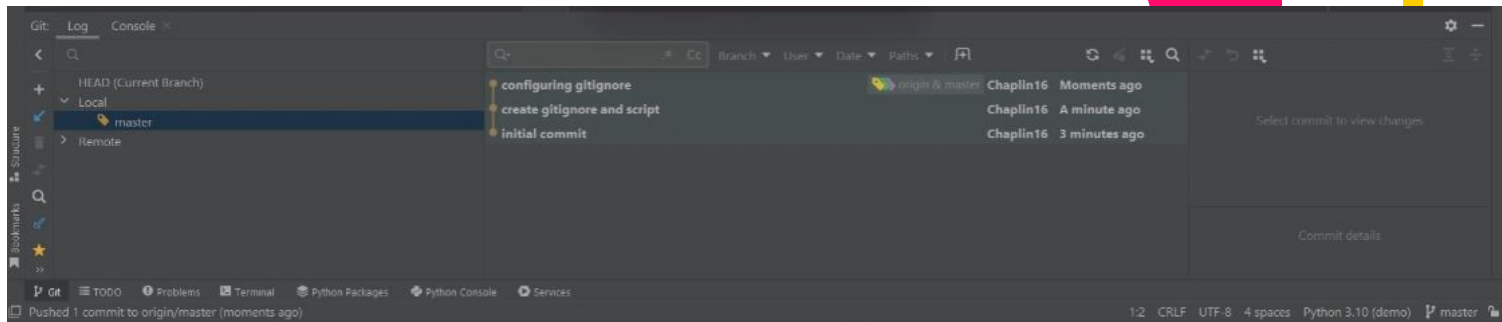
Récapitulatif rapide de tous les commits

En bas de PyCharm se trouve l'icône *Git*.
Cliquez dessus, l'interface du bas se modifie



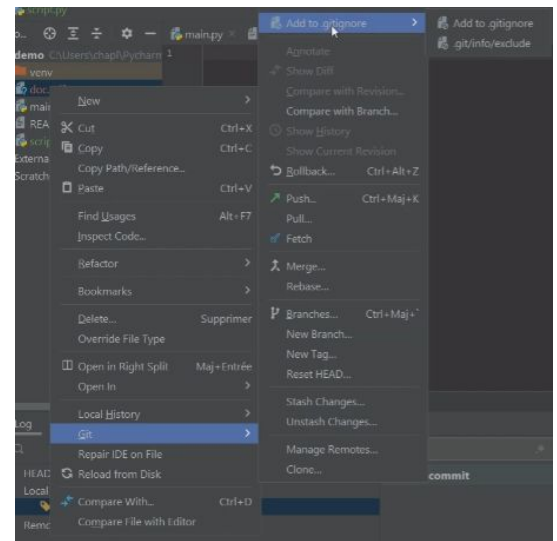
Les commits y sont répertoriés avec leur nom, la
branche, l'heure et leur auteur.

Rapide et pratique.

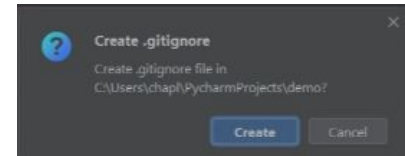


Créer un dossier .Gitignore

Cliquez droit sur le dossier qui ne doit pas aller sur *Git*
Sélectionnez *Git / Add to .gitignore /add to .gitignore*



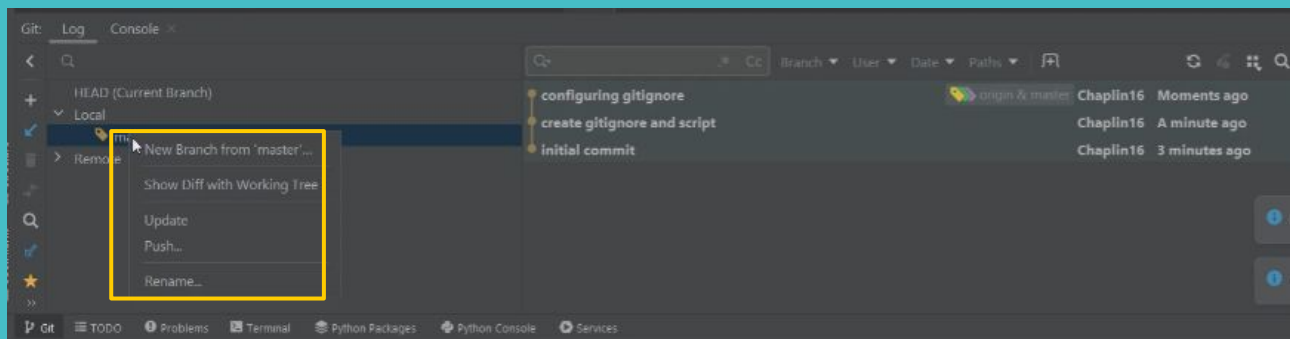
Si c'est le premier dossier, alors cela va générer la création de gitignore. Le dossier sélectionné s'inscrive automatiquement dans Gitignore



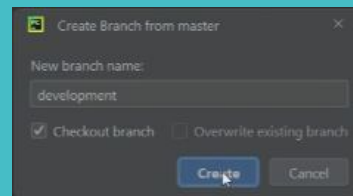
S'il s'agit du même type de format (ex: des pdf), alors mettez *.pdf dans gitignore et tous les pdf seront automatiquement exclus de Git



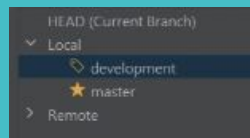
Les Branches



Dans la partie de Git sur Pycharm, cliquez droit sur la branche actuelle *master*
Sélectionnez *New Branch from 'master'*
Ecrire le nom de la nouvelle branche et *Create*

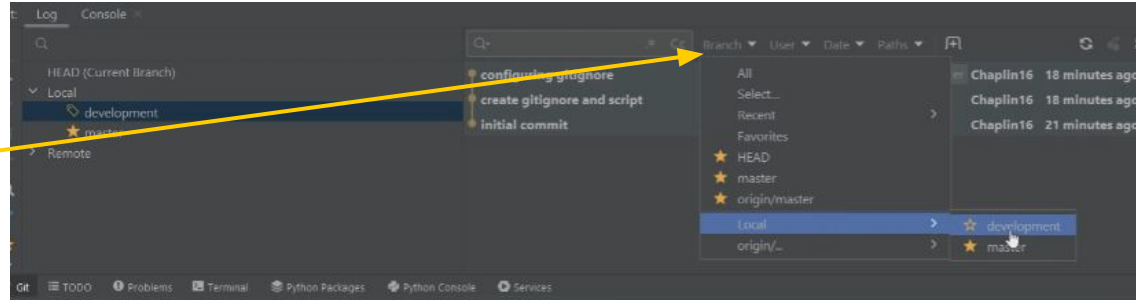


Les deux branches
apparaissent en local



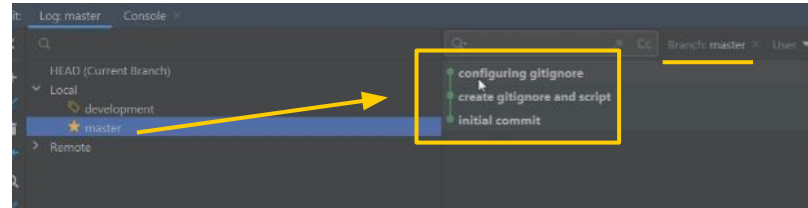
En cliquant droit sur le nom d'une des branches et en sélectionnant *checkout*,
on switch entre les branches

Sélectionnez dans le menu déroulant la branche sur laquelle vous souhaitez inscrire vos changements.



Et voilà, il suffit d'ajouter, de commiter et de pusher et cela apparaîtra sur la nouvelle branche et non sur master.

3 commits sur la branche *master*



4 commits sur la branche *Development*

