



Atelier Git et Librairies KiCad

KiCad EDA

A Cross Platform and Open Source Electronics Design
Automation Suite

 DOCS

 DOWNLOAD   

 LIBRARIES



YouTube

Atelier Git appliqué au partage de librairies KiCad

Déroulement de l'atelier

- Tour de table et partage d'expérience autour des différents logiciels de CAO
- Introduction à l'utilisation de Git (installation et configuration)
- Utilisation dans le contexte des librairies officielles de KiCad
- Création d'un dépôt Git pour nos librairies KiCad
- Collaboration avec GitLab pour le partage des librairies
- Conclusion et perspectives de l'utilisation de Git dans nos projets



Tour de table

Partage d'expérience autour des différents logiciels de CAO



Guide pour votre intervention :

- Présentation : votre Prénom, Labo et Fonction
- Quel usage de Git avez-vous ?
 - je navigue sur GitHub et fais des téléchargements
 - j'utilise régulièrement le versioning pour mes projets
- Vous sentez vous à l'aise avec ces outils ?
- Avez vous une anecdote à partager ?



Recommandations d'ordre générale

Lors de l'installation de programmes, il est souvent préférable d'utiliser les paramètres par défauts, comme le chemin (PATH) d'installation que l'on vous propose.



C:\Program Files ou %localappdata%\programs →
C:\Users\username\AppData\Local\Programs

Rendre visible les extensions des noms de fichiers dont le type est connu.



README.md.txt



Paramètres avancés :

- ☒ Masquer les conflits de fusion de dossiers
- ☒ Masquer les extensions des fichiers dont le type est connu
- ☒ Masquer les fichiers protégés du système d'exploitation (recon

Préférez un nommage de vos dossiers et fichiers sans espace, ni caractères spéciaux.



C:\gitlab\in2p3\rdei\gt-kicad\libraries





Git vs GitLab, GitHub, Bitbucket, Forgejo

Programme à télécharger sur :
<https://gitforwindows.org/>



Git-2.49.0-64-bit.exe

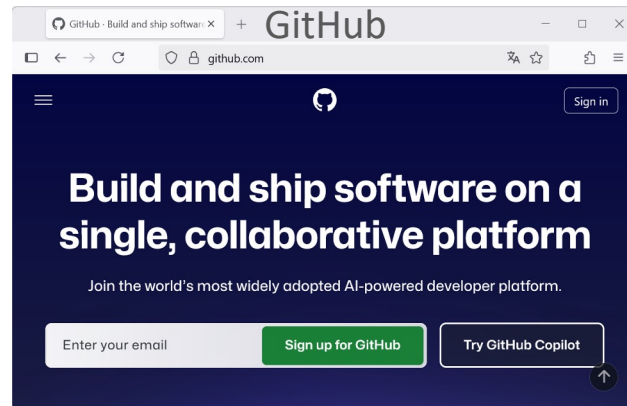
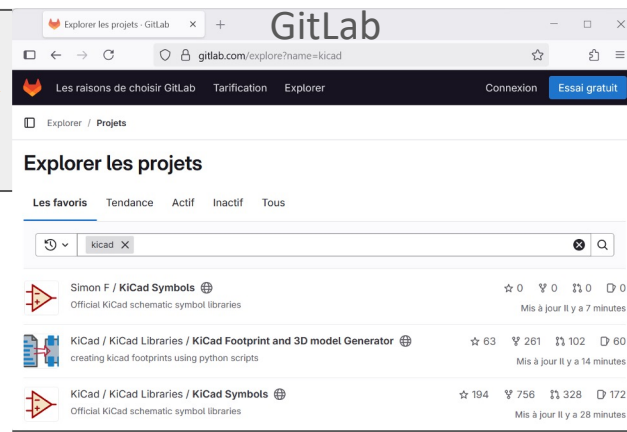
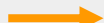
```
C:\Windows\System32\cmd.e  X  +  v  -  □  X
Microsoft Windows [version 10.0.22631.5335]
(c) Microsoft Corporation. Tous droits réservés.

C:\Windows\System32>where git
C:\Users\biganzoli\AppData\Local\Programs\Git\cmd\git.exe

C:\Windows\System32>git --version
git version 2.49.0.windows.1

C:\Windows\System32>git -v
git version 2.49.0.windows.1
```

Forges
Logicielles
en ligne





Prérequis

Git doit-être au préalable installé sur votre poste de travail ([tutoriel installation](#))
Pour vérifier l'installation et l'accès aux commandes Git, ouvrez une console système (DOS) :

[Win] + [R] (ouvre la fenêtre “Exécuter”), puis saisir : **cmd**

Essayez l'appel de git avec l'une des commandes suivantes, si une erreur apparaît vérifier votre PATH vers c:\Program Files\Git\cmd\git.exe ou installez git :

```
>git  
>git --version  
>git --help
```



```
usage: git [--version] [--help] [-C <path>] [-c  
<name>=<value>] [--exec-path[=<path>]] [--html-path] [-  
-man-path] [--info-path]
```

Utilisation de Git

dans le contexte des librairies officielles de KiCad



Suivez le guide et à vous de jouer...





Lien des dépôts officiels des librairies KiCad

KiCad BLOG DISCOVER COMMUNITY HELP CONTRIBUTE SPONSORS LIBRARIES STORE ABOUT Donate

GitLab Libraries

The KiCad libraries are community contributed and hosted on GitLab at gitlab.com/kicad/libraries. Users who wish to make contributions to the libraries can [fork](#) the library repositories. If you wish to contribute to the libraries, refer to the [contribution guide](#).

It is also possible to keep your system libraries up to date with the latest additions by [cloning](#) the library repositories using Git. Tracking the library repositories using Git means that only the *changes* to the libraries need to be downloaded, rather than retrieving the entire library set each time.

The libraries are actively maintained for the current stable version of KiCad. Libraries for previous versions of KiCad are available in the Git history and are tagged by point release, as described below.

Library Repositories for KiCad 9.x

For KiCad version 9, the libraries are organized into four separate repositories on GitLab:

- Schematic symbols: [kicad-symbols](#)
- PCB footprints: [kicad-footprints](#)
- 3D models: [kicad-packages3d](#)
 - Source files for 3D models: [kicad-packages3d-source](#)
- Project templates: [kicad-templates](#)

<https://gitlab.com/kicad/libraries>

Lien GitLab





Copiez le lien du dépôt de la librairie dans le presse papier

The screenshot shows the GitLab interface for the **KiCad Symbols** repository. The URL in the browser is `gitlab.com/kicad/libraries/kicad-symbols`. The page title is **KiCad Symbols**. The repository is owned by **KiCad** and has 193 stars. The current branch is **master** and the file path is **kicad-symbols**. A table of files is visible, including `CMakeModules`, `.gitattributes`, `.gitignore`, `.gitlab-ci.yml`, `4xxx.kicad_sym`, and `4xxx.IEEE.kicad_sym`. A dropdown menu is open, showing options to clone the repository using SSH or HTTPS, or to open it in an IDE. The **Code** button is highlighted with a red box and the number 1. The **Cloner avec HTTPS** option is highlighted with a red box and the number 2. The **Copier l'URL** button is also highlighted with a red box. The **Informations sur le projet** section on the right shows 8 150 validations, 228 branches, 83 étiquettes, 19 Releases, and 127 environnements.

1

2

Code

Cloner avec SSH

git@gitlab.com:kicad/libraries

Cloner avec HTTPS

Copier l'URL

https://gitlab.com/kicad/libra

Ouvrir dans votre IDE

Visual Studio Code (SSH)

Visual Studio Code (HTTPS)

IntelliJ IDEA (SSH)

IntelliJ IDEA (HTTPS)

Télécharger le code source

zip

Informations sur le projet

Official KiCad schematic symbol libraries

8 150 validations

228 branches

83 étiquettes

19 Releases

127 environnements

README

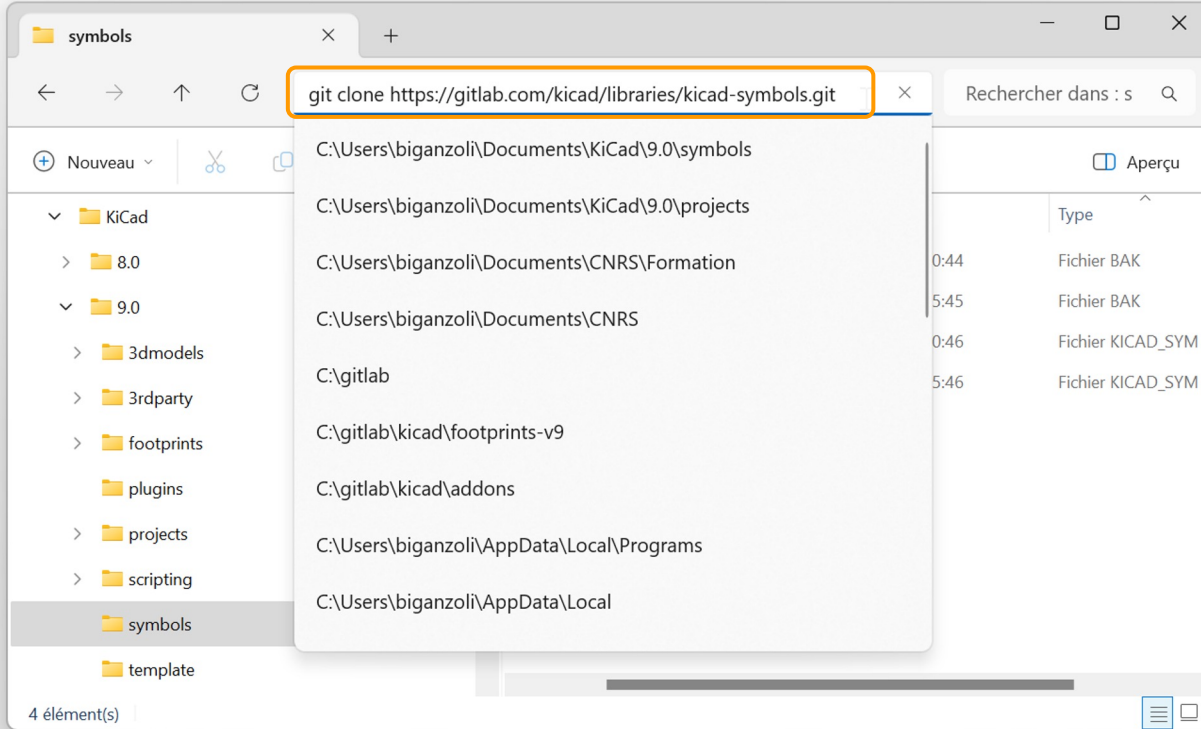
Creative Commons Attribution Share Alike 4.0 International

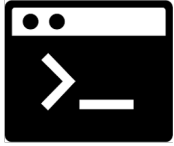
CONTRIBUTING



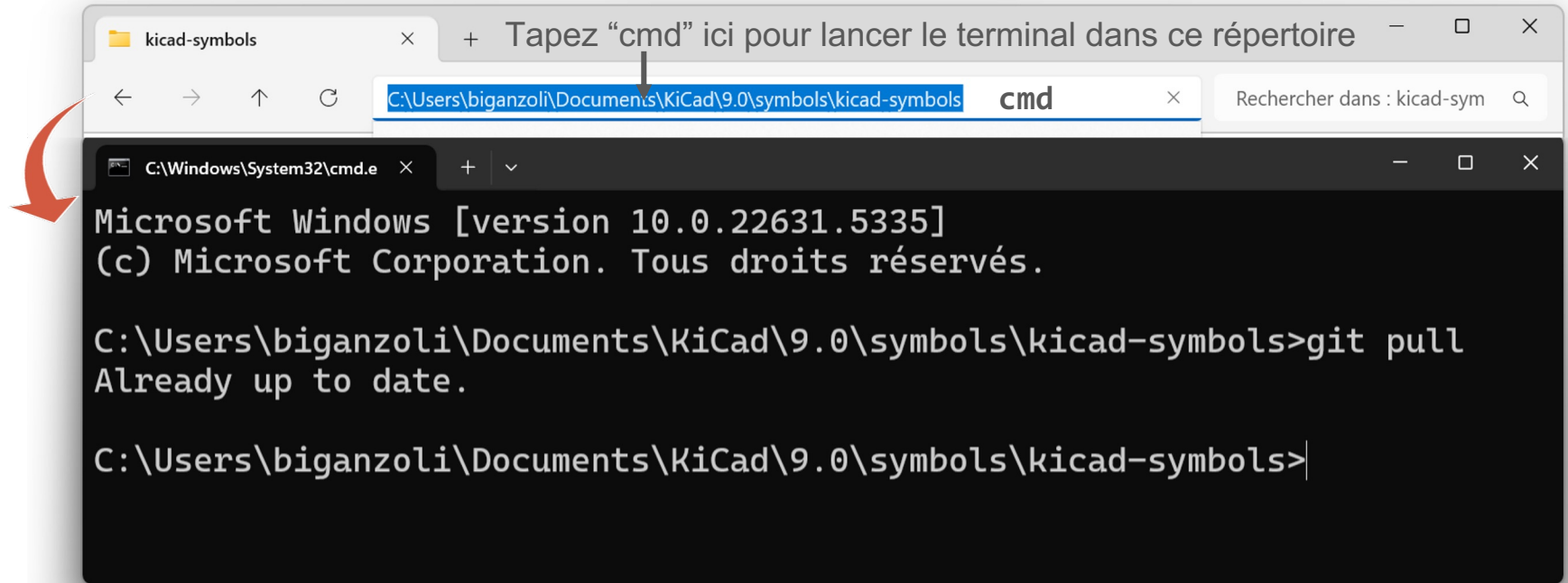


Collez le lien du dépôt dans la barre d'adresse de l'Explorateur Windows de votre dossier, précédé de la cmd : **git clone**





La copie du dépôt dans votre dossier pourra être mise à jour à votre initiative, avec la cmd : **git pull**



```
Microsoft Windows [version 10.0.22631.5335]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\biganzoli\Documents\KiCad\9.0\symbols\kicad-symbols>git pull
Already up to date.

C:\Users\biganzoli\Documents\KiCad\9.0\symbols\kicad-symbols>
```





Première utilisation de Git

Création d'un dépôt local Git pour son projet KiCad



Suivez le guide et à vous de jouer...



Objectifs

- ❖ Comprendre les principes de la gestion de version distribuée
- ❖ Pouvoir gérer des versions de documents textuels avec l'outil Git
- ❖ Apprendre à utiliser en local sur son poste de travail les commandes Git de base



Prérequis

Créez votre accès au GitLab du RdEI en vous connectant avec vos identifiants EduGAN : <https://gitlab.in2p3.fr/rdei/>

Gitlab IN2P3

- You already have an account on the IN2P3 Gitlab platform (read more...)
- You are a new user from IN2P3 or from another academic organization (read more...)
- You do not fit any of the situations above (read more...)

Please take a moment to read the [documentation](#).
[Terms of use](#) | [Privacy](#)

Nom d'utilisateur ou adresse de courriel principale

Mot de passe

[Mot de passe oublié ?](#)

☐ Se souvenir de moi

Connexion

ou connectez-vous avec

CC-IN2P3 Single Sign On

☐ Se souvenir de moi

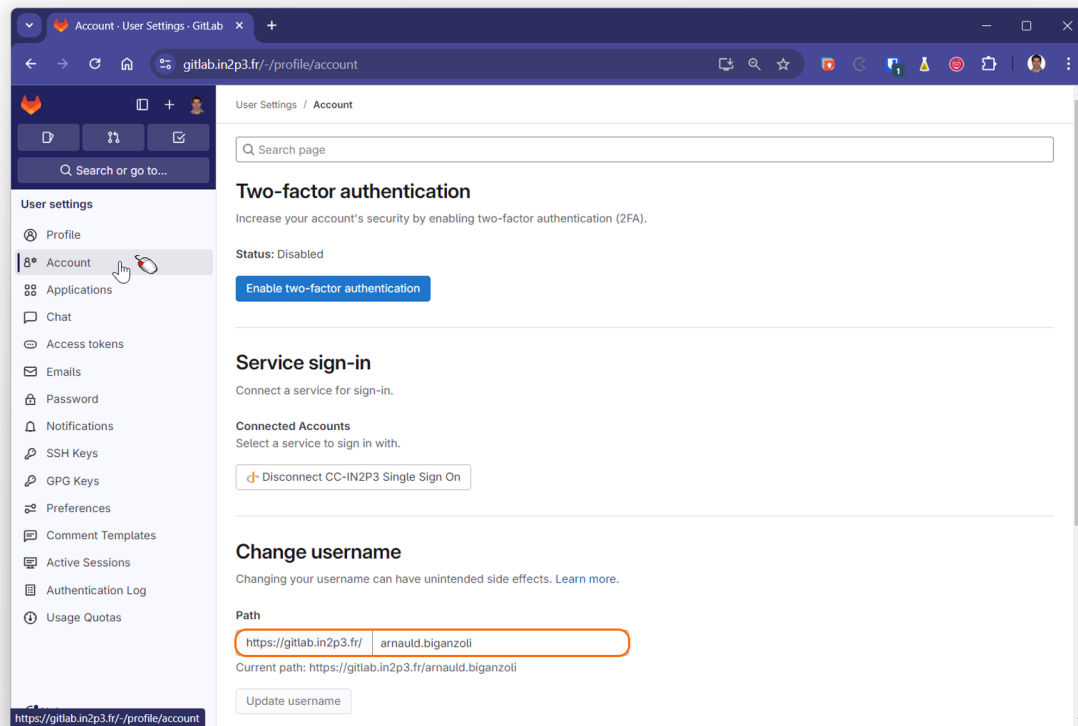
Explorer Aide À propos de GitLab Forum de la communauté

français



Prérequis

Définissez votre nom d'identifiant sur GitLab pour votre compte :





Prérequis

Créer un jeton d'accès au GitLab pour vous identifier lors du transfert de données avec les commandes git :

Sauvegardez votre clé du jeton

Personal access tokens

You can generate a personal access token for each application you use that needs access to the GitLab API. You can also use personal access tokens to authenticate against Git over HTTP. They are the only accepted password when you have Two-Factor Authentication (2FA) enabled.

Active personal access tokens 1 [Add new token](#)

Token name	Description	Scopes	Created	Last Used	Expires	Action
gitlab-rdel-arnauldddev-token		api_read_api, read_user, create_runner, manage_runner, k8s_proxy, read_repository, write_repository, read_registry, write_registry, ai_features	Feb 13, 2025	1 day ago	in 11 months	Reset

Feed token

Your feed token authenticates you when your RSS reader loads a personalized RSS feed or when your calendar application loads a personalized calendar. It is visible in those feed URLs. It cannot be used to access any other data.

Feed token

..... [Copy](#) [Reset](#)

Keep this token secret. Anyone who has it can read activity and issue RSS feeds or your calendar feed as if they were you. If that happens, [reset this token](#).



Vérifier la configuration de git





Configurer Git sur sa machine

> Créez votre compte personnel sur la plateforme GitLab de l'IN2P3 :

Create your account : <https://gitlab.in2p3.fr/rdei>

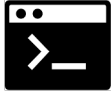
> et/ou configurez simplement dans git votre **Prénom** et votre **adresse mail** :

En faite, ceci est purement informatif pour les log de git, l'authentification se fera avec votre jeton de connexion pour le serveur GitLab.

```
>git config --global user.name "Mon_Prénom_ou_Pseudo"
```

```
>git config --global user.email "mon.adresse@labo-mail.fr"
```





Pour vérifier votre configuration du prog git.exe

Tapez simplement dans votre interpréteur de cmd (console) :

```
>git config --global user.name
```

```
Mon_Prénom_ou_Pseudo
```

```
>git config --global user.email
```

```
mon.adresse@labo-mail.fr
```

```
>git config --list
```

Git contient un outil appelé git config pour vous permettre de voir et modifier les variables de configuration qui contrôlent tous les aspects de l'apparence et du comportement de Git.





Explorer le GitLab du RdEI

Dossier

Repository

libraries · GitLab

gitlab.in2p3.fr/rdei/gt-kicad/libraries

Explorer Connexion S'inscrire

RdEI / GT-KiCad / libraries

libraries

Créer un repository sur un serveur GitLab (remote)

Groupe des bibliothèques KiCad du RdEI Voir également la page web sur les convention de nommage des bibliothèques...
[En savoir plus](#)

Sous-groupes et projets Projets partagés Groupes partagés Inactif

Recherche (3 caractères minimum) Nom

footprints

Chaque contributeur pourra avoir son propre dépôt sous ce sous-groupe s'il gère une librairie spécifique.

0 2 1

footprints_rdei

0 1 month ago

RdEI_Connector.pretty

0 1 month ago





Commencer un nouveau projet avec un dépôt git

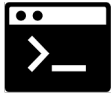
A titre d'exercice, nous allons créer un repository local, depuis un nouveau dossier de travail (ex: “demo-lib-occitanie-2025”) :

- lancer l'invite de commande “cmd” depuis ce dossier
- créer un nouveau dépôt avec la commande (en local uniquement)
- créer un fichier texte “README.md” contenant un descriptif du projet

```
>echo # Titre du projet > README.md
```

(Cette méthode avec l'outils de redirection “>” ou “>>” pour la concaténation permet la création et la complétion d'un fichier depuis la console, mais vous pouvez tout aussi bien créer et compléter le fichier README.md à votre manière.)





Réaliser son premier versionning

Toujours depuis l'invite de commande :

- vérifier le contenu et l'état de votre dépôt

```
>git status
```

- ajouter tous vos fichiers pour le versionning

```
>git add .
```

[ajouter tous les fichiers déjà présent dans l'index au commit]

```
>git status
```

```
>git commit -m "version initiale du projet"
```

ou

```
>git commit -a -m "version initiale du projet"
```

[ajouter un message au commit]

- visualiser l'historique des validations (commit) de votre versionning

```
>git log
```

voir aussi la commande "git log --graph" ou l'utilitaire gitk

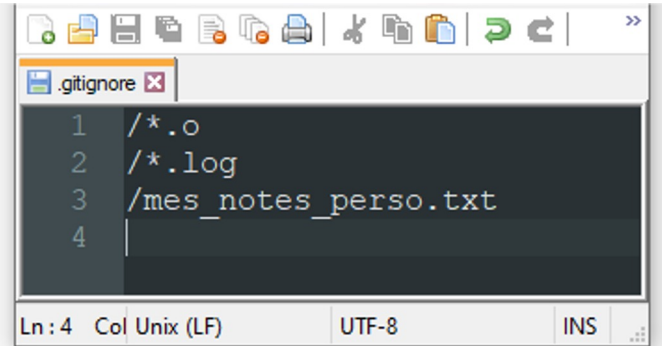


Exclure des fichiers du versionning

Pour exclure des fichiers du suivi, vous devez créer à la racine de votre dépôt un fichier sans extension dont le nom est **“.gitignore”**

Par exemple si l'on souhaite ne pas diffuser, ni suivre les modifications des fichiers objets dont l'extension est **“.o”**, il suffit de mettre **“/”** pour la racine du dépôt et **“*.o”** dans le fichier **.gitignore**.

Il est aussi possible d'exclure explicitement un seul fichier, exemple : **“/mes_notes_perso.txt”**



```
1 /*.o
2 /*.log
3 /mes_notes_perso.txt
4
```



Attention toutefois, car le fichier **“mes_notes_perso.txt”** ne sera pas sauvegardé dans l'historique de git, donc sa suppression sera définitive.



Les commandes de base de Git à retenir

Voir “[GIT - Ligne de commande principale](#)”

Créer un “repository” dans un dossier,
ce sera votre répertoire de travail local

Ajouter des fichiers et sauvegarder
votre “working directory”

Lire l’état du dépôt local et voir
l’historique des commits

C:\...\chemin-du-projet\

`git init`

`git add .`

`git commit -m "Mon message"`

`git status`

`git log`



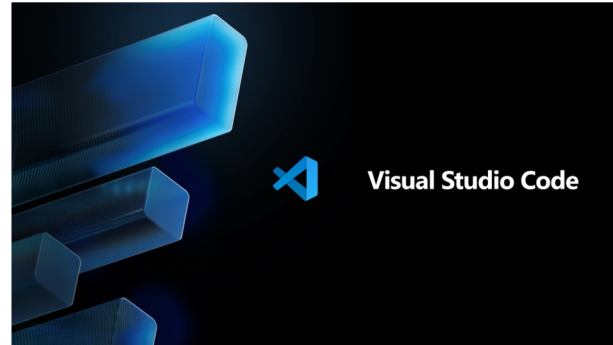
Outil de visualisation

Appelez l'outil de visualisation Git de votre dépôt depuis la ligne de commande :

```
C:\...\chemin-du-projet\
```

```
>gitk --all
```

Ou utilisez le plugin “[git graph](#)” pour une visualisation depuis [VS Code](#) :



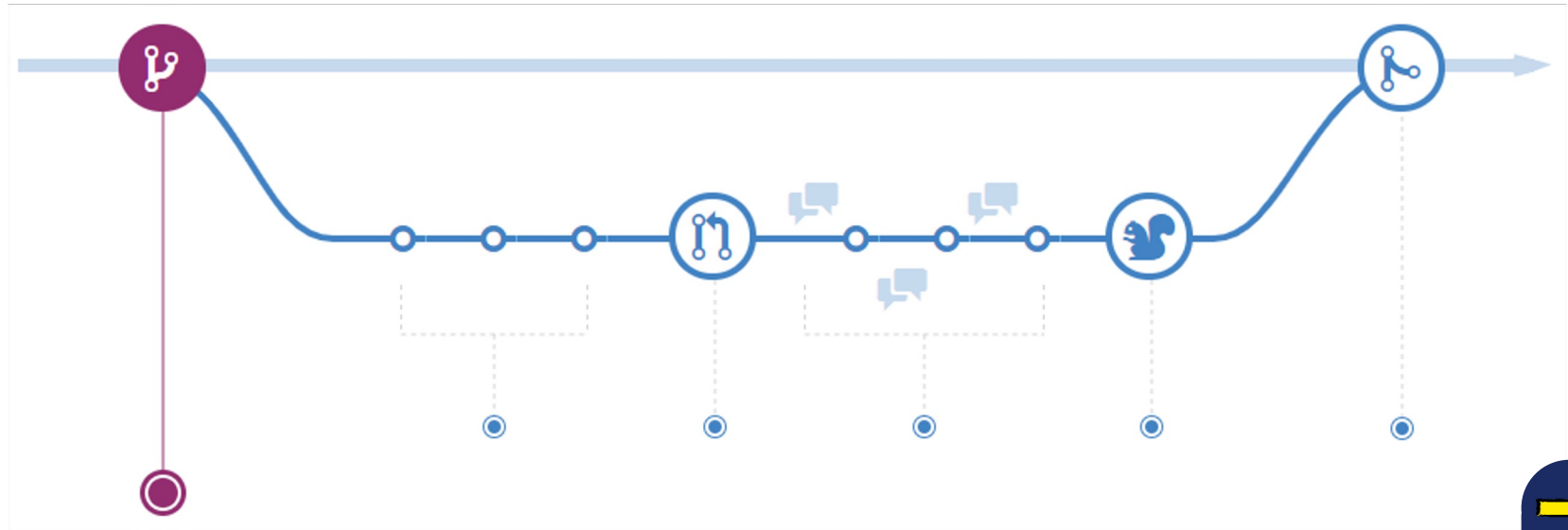
Comprendre le flux de travail avec GitLab ou GitHub



Understanding the GitHub flow

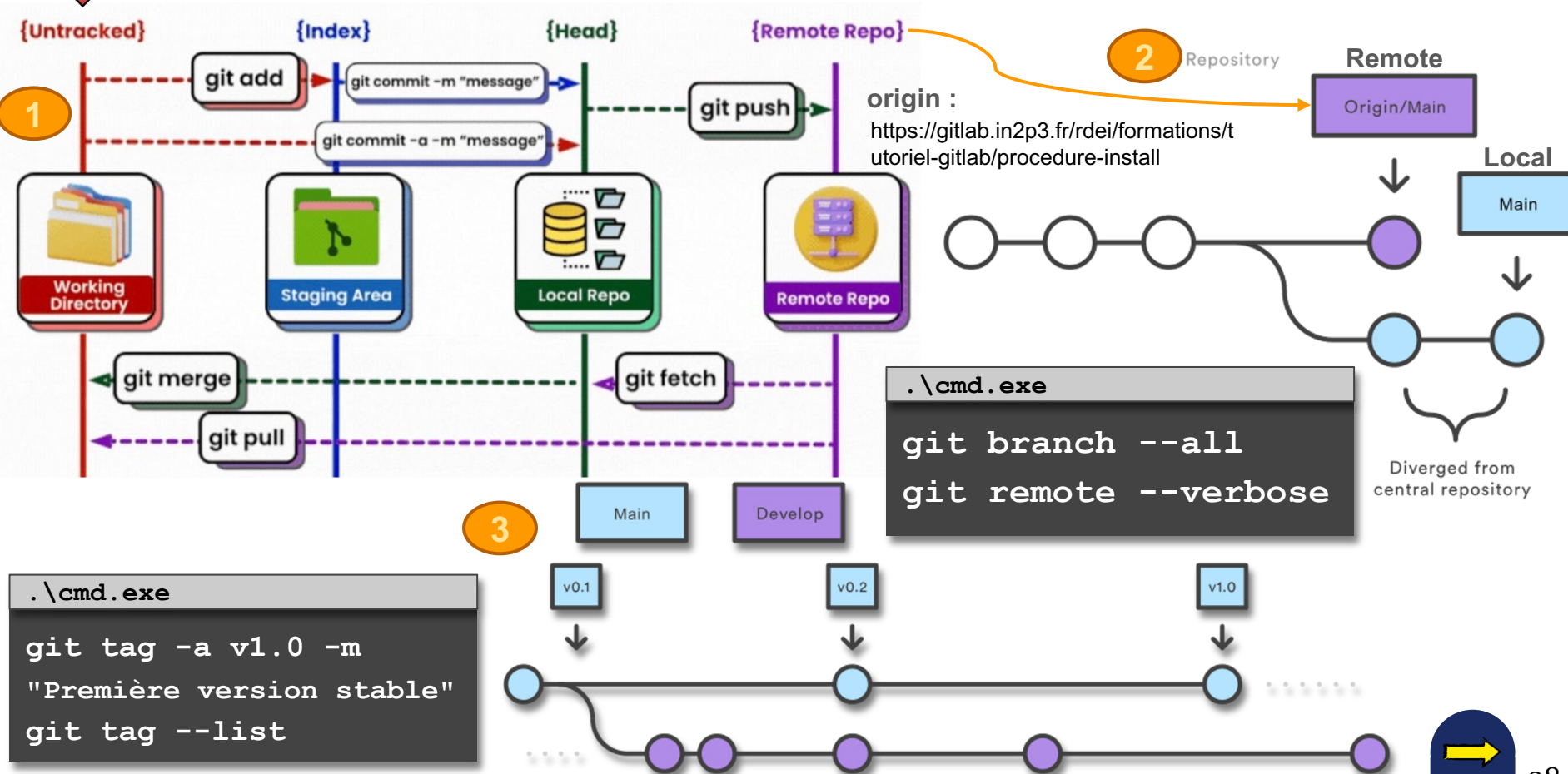
⌚ 5 minute read

📄 Download PDF version





Git Workflow



Bibliothèques de composants avec bases de données





Merci pour votre attention ;-)

Il ne vous reste plus qu'à passer à la pratique et à tester vos connaissances :

Bienvenue sur Learn Git Branching !

Et maintenant vous êtes prêt pour [contribuer sur GitLab](https://github.com/github/opensource.guide/blob/master/_articles/fr/how-to-contribute.md) ou GitHub :

https://github.com/github/opensource.guide/blob/master/_articles/fr/how-to-contribute.md

<https://guides.github.com/activities/hello-world/>