



BONNES PRATIQUES POUR LA GESTION DES CODES ET MÉTADONNÉES

Ateliers techniques - Indicateurs territoriaux pour un développement durable au Maroc

24 mai 2022

Centre pour l'entrepreneuriat, les PME, les régions et les villes (CFE)



Avant de commencer

- Veuillez éteindre votre microphone pendant la présentation et la démonstration.
- Vous pouvez poser des questions dans le **chat**, ou en **levant la main** sur Zoom.
- Les exemples présentés aujourd'hui seront disponibles sur :

<https://github.com/oecd-cfe-eds/ateliers-bp>



Les objectifs de cet atelier

1. Donner une introduction à la gestion de versions de code avec Git et RStudio
2. Démontrer leur utilité dans le processus de création et mise à jour d'indicateurs territoriaux
3. Présenter les fonctionnalités de GitHub relatives à la documentation des codes, des données et métadonnées



Qu'est-ce que les **bonnes pratiques** ?

- Ce sont des **conventions** qui visent à rendre les projets d'analyse statistique plus reproductibles et pérennes
- Elles visent également à faciliter la collaboration et la transmission du savoir-faire au sein d'une équipe
- Parfois, leur mise en place peut entraîner l'adoption de **nouveaux outils et processus de travail**





La gestion de versions avec Git

Git est un logiciel *open-source* utilisé pour gérer les versions des codes

Que peut-on faire avec Git ?

- Sauvegarder différentes versions des codes dans un dossier
- Créer un historique documenté des modifications faites sur un dossier
- Revenir facilement en arrière sur des anciennes versions

Installer Git sur Windows: <https://gitforwindows.org/>





Git, en pratique...



analyse.R



analyse-2.R



analyse-3-revMP.R



analyse-FINAL.R



analyse.R



Git, en pratique...



analyse.R



analyse-2.R



analyse-3-revMP.R



analyse-FINAL.R



analyse.R



analyse.R



Git, en pratique...



analyse.R



analyse-2.R



analyse-3-revMP.R



analyse-FINAL.R



analyse.R



analyse.R



analyse.R



Git, en pratique...



analyse.R



analyse-2.R



analyse-3-revMP.R



analyse-FINAL.R



analyse.R



analyse.R



analyse.R



analyse.R



Git, en pratique...

c6163cf19b479140b0801262db51332a10300bf5

Marcos Díaz
2022-05-24 09:00

Analyse des indicateurs
territoriaux au niveau régional



analyse.R



Git, en pratique...

4a00f558faf1e5af6f26c00996c27f0b489c1080

Marcos Díaz

2022-05-25 09:34

Élargir l'analyse aux provinces



analyse.R



analyse.R



Git, en pratique...

18d97d69da0a637f6f01e50fdf4ff1ef6dd6dfbb

María Paula Caldas

2022-05-25 14:20

Ajouter les indicateurs de qualité
de la connexion internet



analyse.R



analyse.R



analyse.R



Git, en pratique...

07e9073c3bb9809f5f4b12c2a3f737768a49d5d1

María Paula Caldas

2022-05-26 10:10

Dernière version des figures pour
intégration au rapport



analyse.R



analyse.R



analyse.R

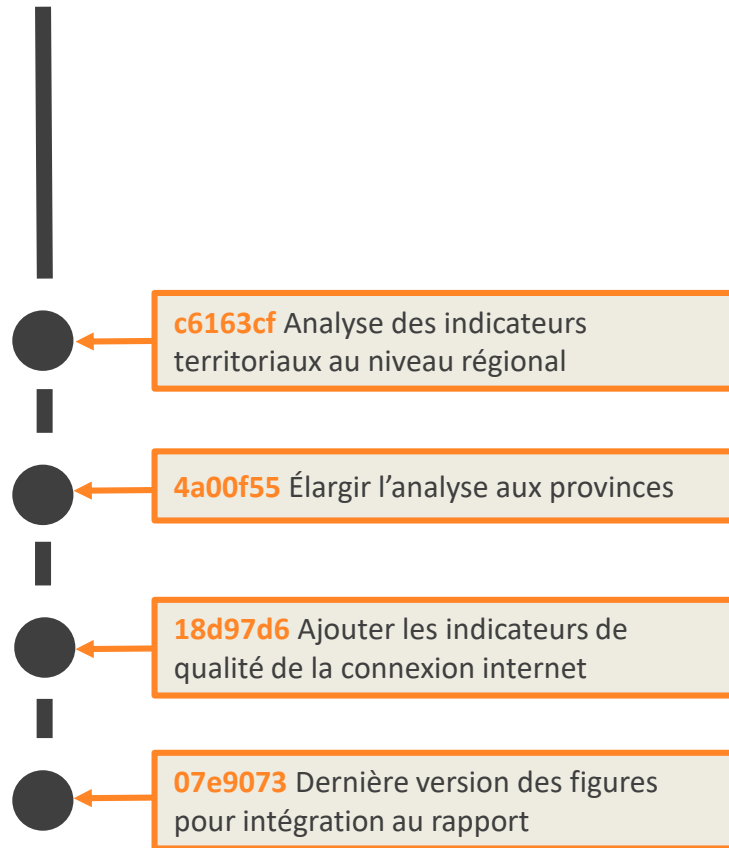


analyse.R



Git, en pratique...

- Git permet de créer un **historique documenté** des changements faits sur les codes dans un dossier
- Ces « codes » peuvent être des fichiers texte n'importe quel type (ex. R, Stata, SAS, Python)
- On peut faire un commit sur un ou plusieurs fichiers





Démonstration

Initialiser un dépôt et faire ses premiers *commits*

```
# Naviguer vers le dossier d'intérêt
cd chemin/vers/mon/dossier

# Initialiser Git et faire son premier commit
git init
git add mon-fichier.R
git commit -m "Mon premier commit"
git status
```



Le partage et la documentation avec **GitHub**

GitHub est un site web et un service d'hébergement qui aide les utilisateurs Git à stocker, partager, et collaborer sur leur code

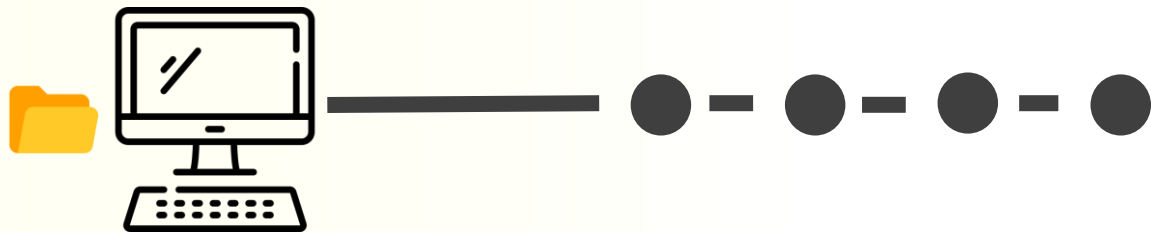
Que peut-on faire avec GitHub ?

- Stocker et synchroniser ses codes
- Partager les codes avec son équipe ou son public
- Mieux collaborer et gérer ses projets
- Automatiser ses flux de travail avec l'intégration et la livraison continues (CI/CD)



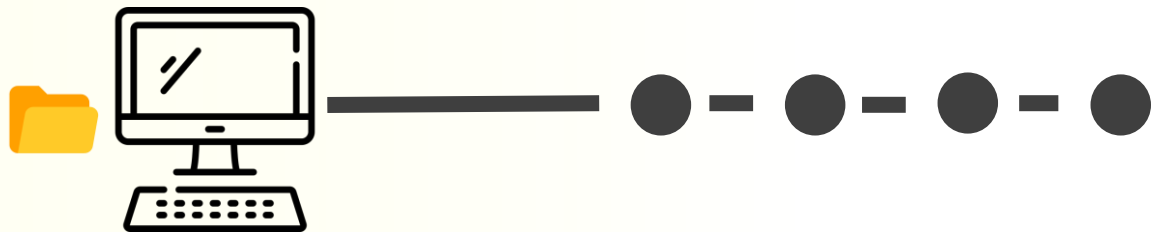


Créer un dépôt sur GitHub



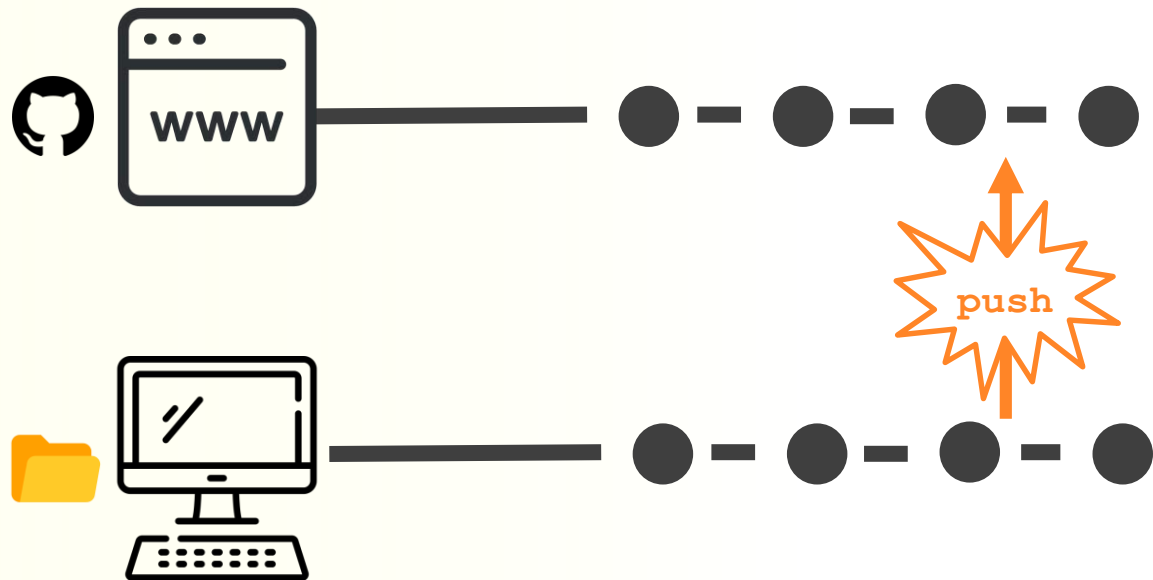


Créer un dépôt sur GitHub



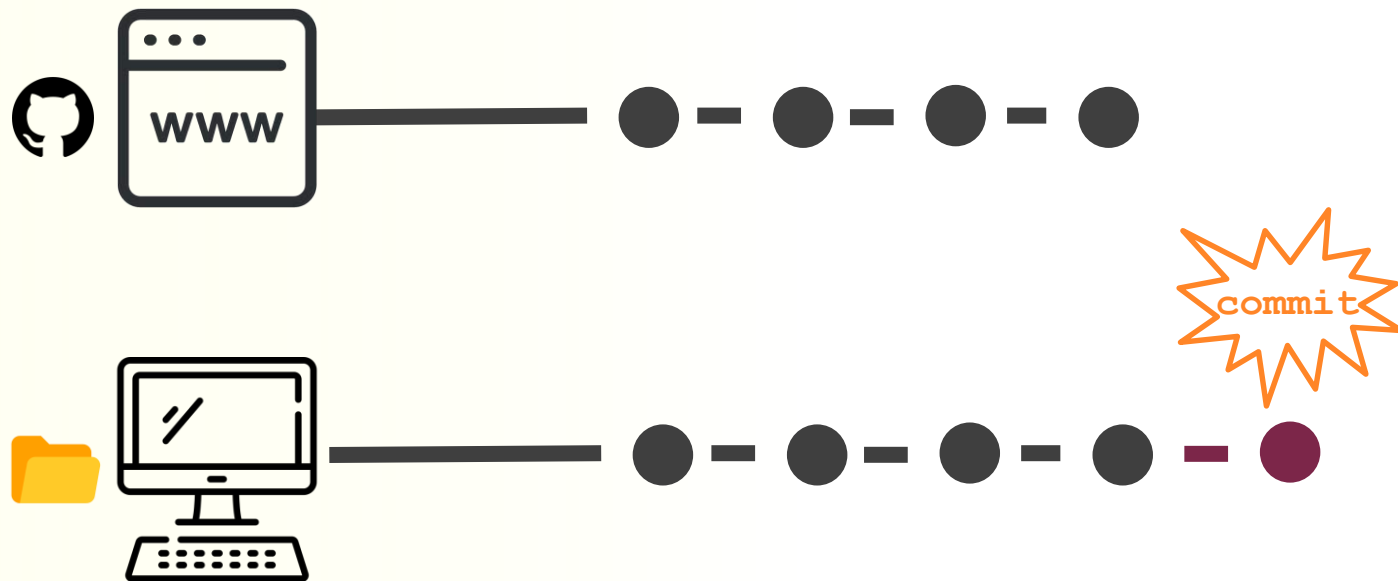


Créer un dépôt sur GitHub



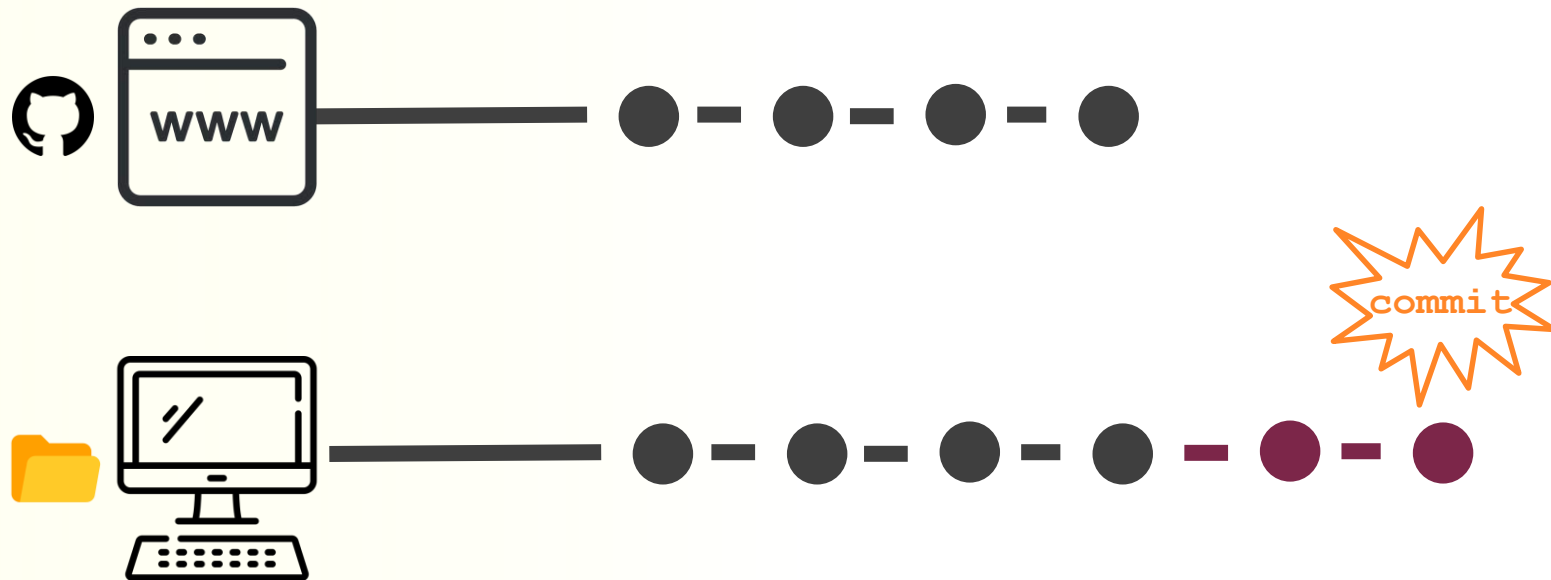


Créer un dépôt sur GitHub



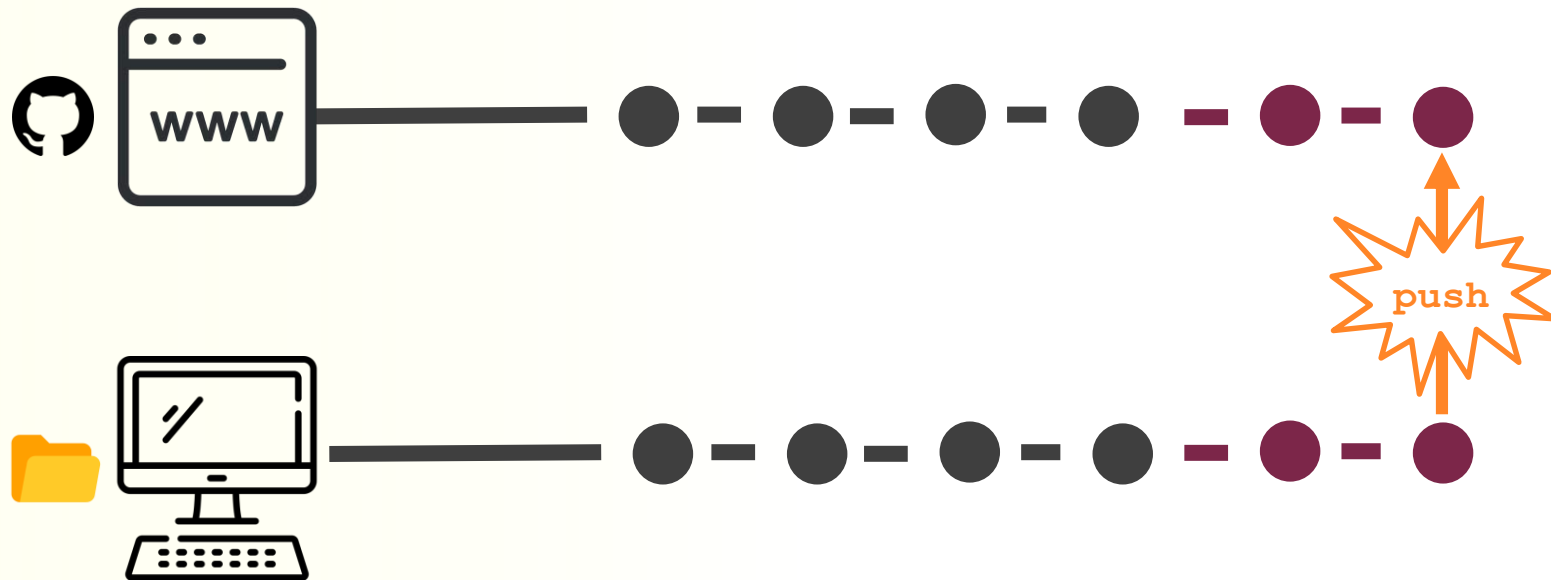


Créer un dépôt sur GitHub





Créer un dépôt sur GitHub






Démonstration

Publier mon dépôt sur GitHub

Create a new repository


A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)


Owner ^{*} Repository name ^{*}

 oecd-cfe-eds /

Great repository names are short and memorable. Need inspiration? How about [ubiquitous-waddle?](#)

Description (optional)

☐  Public
Anyone on the internet can see this repository. You choose who can commit.

☒  Private
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

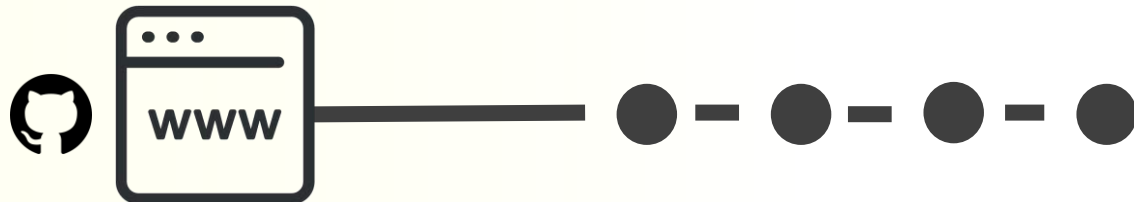
```
# Créer un nouveau dépôt sur Github
# (sans initialiser le README)

# Lier son dépôt local avec celui de Github
git remote add origin https://github.com/moi/mondepot.git
git push -u origin main

# Créer des nouveaux commits et les télécharger sur Github
git add fichier-qui-a-change.py
git commit -m "Update d'un nouveau script"
git status
git push
```

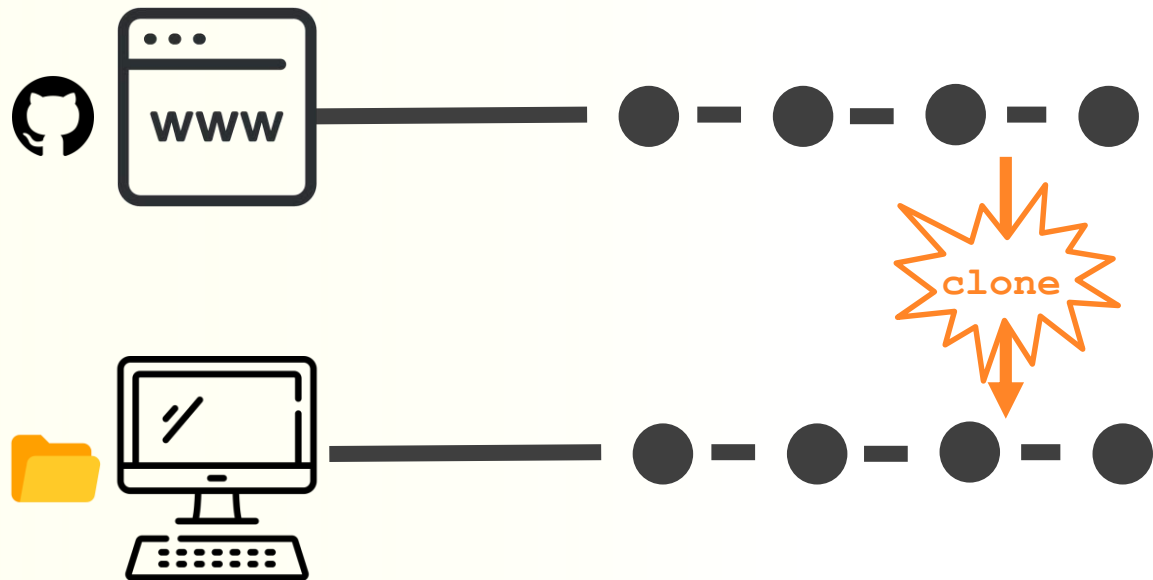


Télécharger un dépôt GitHub





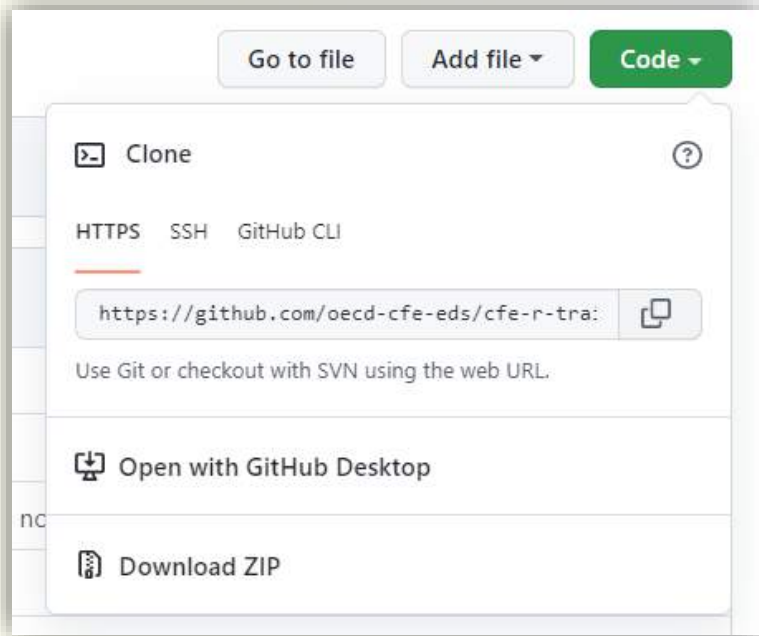
Télécharger un dépôt GitHub





Démonstration

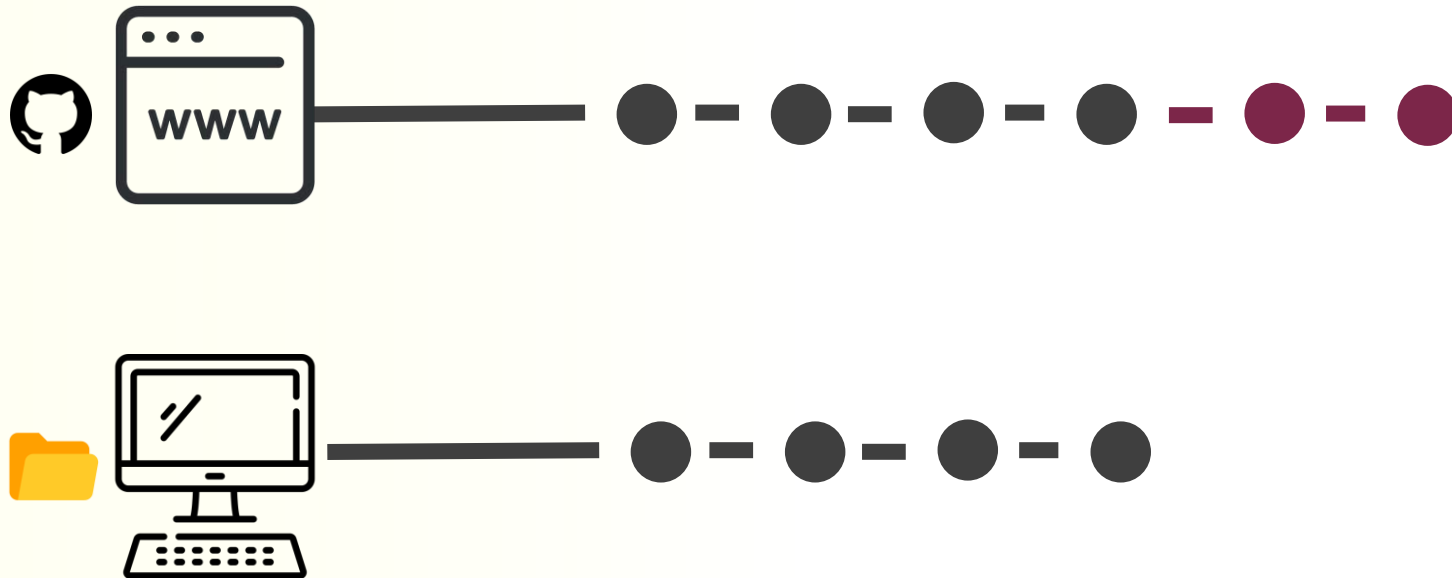
Télécharger un dépôt GitHub



```
# Récupérer le lien HTTPS du dépôt Github et  
# faire son clone  
git clone https://github.com/moi/mondepot.git  
cd mondepot  
  
# Vérifier que Git marche bien, et que j'ai  
# accès à tout l'historique  
git status  
git log
```

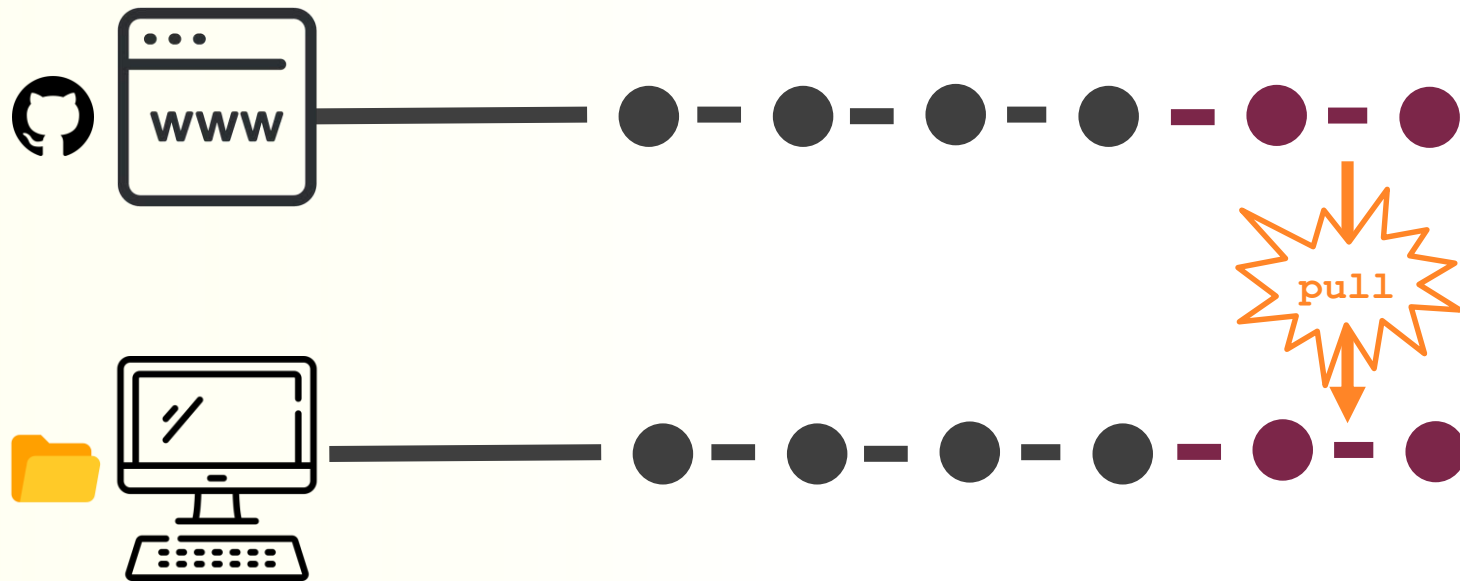


Télécharger les changements d'un dépôt GitHub





Télécharger les changements d'un dépôt GitHub





Les bonnes pratiques

La documentation et les README

- Le **README.md** est un guide avec une description brève de votre projet
- Les fichiers **md** (ou *Markdown*) sont des fichiers texte qui s'affichent comme des pages d'accueil sur GitHub

Que peut-on mettre dans le README ?

- Instructions d'installation
- Documentation des métadonnées
- Conditions de collaboration sur le projet



» Les bonnes pratiques

Les noms des fichiers

Les trois règles des « bons noms » de fichiers :

1. **Faciles à lire par les machines:** Les accents, les espaces, et les caractères spéciaux sont à éviter. On évite de mélanger les majuscules et les minuscules
2. **Faciles à lire par les humains:** On privilégie les noms informatifs aux noms courts. Pour faciliter la lisibilité, on utilise les tirets « - » pour séparer les mots
3. **Faciles à ordonner:** Les dates devraient suivre le format AAAA-MM-JJ. Pensez à rajouter un « 0 » au début des fichiers numérotés



Les bonnes pratiques

Prévenir le partage de données confidentielles

- On évite de stocker les données confidentielles dans le dossier où on va créer un dépôt Git
- On rajoute systématiquement un fichier **.gitignore** à nos dépôts, en précisant les fichiers qu'on veut *ignorer* sur Git
- Dans le .gitignore, on rajoute aussi les fichiers lourds (ex. Powerpoint)

```
# Ignorer les fichiers selon  
# leur extention  
*.csv  
*.xlsx  
*.rds  
*.RData  
*.png  
  
# Ignorer un sous-dossier  
datas-confidentielles/
```



Démonstration

Appliquer les bonnes pratiques

1. Ajouter un README
2. Renommer les fichiers
3. Ajouter un .gitignore

Questions ?

Commentaires ?



Twitter: @OECD_local

LinkedIn: www.linkedin.com/company/oecd-local

Website: www.oecd.org/cfe





Annexe

Liens utiles

[utilitR – Chapitre 4](#) – Guide écrite par les agents de l’Insee (France)

[Travail collaboratif avec R – Chapitre 3](#)