

TP1 (FAS-1001) ver2

KANGA Kouakou Auguste

2024-01-22

TP1 (introduction à GitHub et à Quarto)

Étapes pour le tp1:

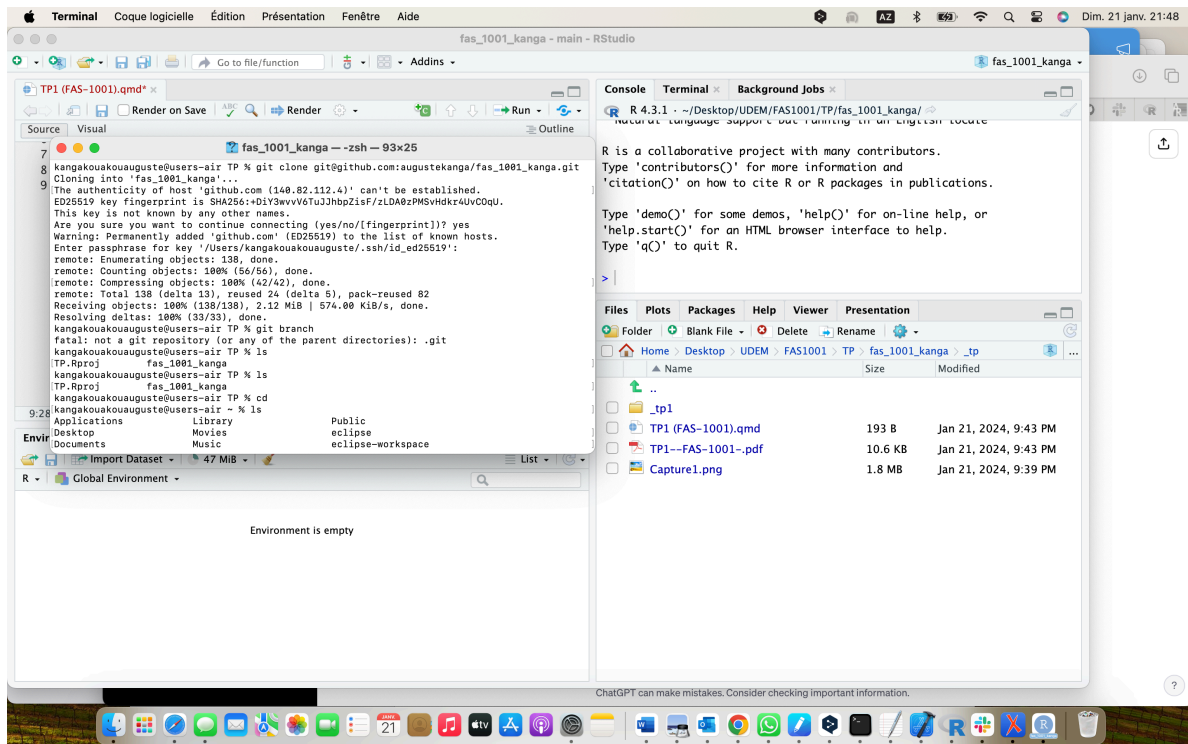


Figure 1: Cloner le Fork

1: J'ai “forké” (ou fork en anglais) et “cloné” (ou clone en anglais) le fichier fas_1001 du compte GitHub de l’enseignant Nadjim Fréchet, changé le nom en fas_1001_kanga. Ce

fichier sera celui où je déposerez mes travaux pratiques et mon travail de session.

Fork d'un Dépôt :

Je suis allé sur la page du dépôt de l'enseignant Nadjim Fréchet que je veux "forker" et j'ai cliqué sur le bouton "Fork" pour créer une copie du dépôt dans mon compte.

Cloner le Fork :

Une fois le fork créé, j'ai cloné le fork sur ma machine avec **git clone**

```
kangakouakouauguste@users-air TP % git clone git@github.com:augustekanga/fas_1001_kanga.git
```

2: Une fois le fichier "cloné", j'ai créé ma propre "branche" (ou branch en anglais) avant de faire mes modifications dans mon terminal. fas_1001_kanga.

j'ai utilisé la commande **git branch** pour créer ma branche **fas_1001_kanga**

```
kangakouakouauguste@users-air fas_1001_kanga % git branch fas_1001_kanga
```

j'ai vérifié en utilisant encore **git branch**

```
kangakouakouauguste@users-air fas_1001_kanga % git branch
```

```
fas_1001_kanga
```

```
* main.
```

j'ai utilisé **git checkout** pour choisir ma branche fas_1001_kanga

```
kangakouakouauguste@users-air fas_1001_kanga % git checkout fas_1001_kanga
```

j'ai ensuite fusionné les modifications de la branche main dans la branche fas_1001_kanga avec **git merge**

```
kangakouakouauguste@users-air fas_1001_kanga % git merge main
```

j'ai vérifié en utilisant encore **git branch**

```
kangakouakouauguste@users-air fas_1001_kanga % git branch
```

```
* fas_1001_kanga
```

```
main
```

3: Une fois ma "branche" créée, j'ai ouvert RStudio en cliquant sur le Rproj. fas_1001.Rproj et créé un document Quarto en format PDF que j'ai sauvegardé dans le fichier __tp à l'intérieur du fichier fas_1001_kanga. Le titre du document est TP1 (FAS-1001). J'ai ajouté mon nom dans la section author et la date de remise du TP1 à l'entête du document dans le code (aussi appelé YAML).

4: Ensuite, j'ai enregistré et ajouté mes modifications au document en cliquant sur le bouton Render dans la barre de commande. Une fois mes modifications enregistrées, ajoutez (add en anglais) ces dernières à votre historique Git avec votre terminal. Par la suite, enregistrez (commit en anglais) vos modifications sous le nom de: Premier commit toujours dans votre terminal.

5: J'ai créé un deuxième fichier Quarto en format PDF que j'ai ajouté dans le fichier __travail_session à l'intérieur du fichier fas_1001_kanga. J'ai ajouté mon nom dans la section

author et généré l'ensemble des sections de mon travail de session de la manière suivante:

- Introduction - gros entête (heading en anglais): #.
- Problématique de recherche - petit entête: ##.
- Données et méthodes - petit entête: ##.
- Résultats - petit entête: ##.
- Conclusion - gros entête (heading en anglais): #.

J'ai enregistré et ajouté mes modifications à mon document de travail de session en cliquant sur le bouton Render.

```
[TP:Rproj]          fas_1001_kanga
kangakouakouauguste@users-air TP % cd fas_1001_kanga
kangakouakouauguste@users-air fas_1001_kanga % ls
LICENSE             _tp                renv
README.md           _travail_session  renv.lock
_misc               fas_1001.Rproj
kangakouakouauguste@users-air fas_1001_kanga % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   _tp/.DS_Store

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    _tp/Capture1.png
    _tp/TP1 (FAS-1001).qmd
    _tp/TP1--FAS-1001-.pdf
    _travail_session/

no changes added to commit (use "git add" and/or "git commit -a")
kangakouakouauguste@users-air fas_1001_kanga % git add
kangakouakouauguste@users-air fas_1001_kanga % git status
fatal: cannot handle Status as a builtin
kangakouakouauguste@users-air fas_1001_kanga % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   _tp/.DS_Store
    new file:   _tp/Capture1.png
    new file:   _tp/Capture2.png
    new file:   _tp/TP1 (FAS-1001).qmd
    new file:   _tp/TP1--FAS-1001-.pdf
    new file:   _travail_session/travail_session_hiver24.pdf
    new file:   _travail_session/travail_session_hiver24.qmd

kangakouakouauguste@users-air fas_1001_kanga % git commit -m "Création du document du travail de session"
[main d2d1b7d] Création du document du travail de session
Committer: KANGA KOUAKOU AUGUSTE <kangakouakouauguste@Users-MacBook-Air.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

7 files changed, 65 insertions(+)
create mode 100644 _tp/Capture1.png
create mode 100644 _tp/Capture2.png
create mode 100644 _tp/TP1 (FAS-1001).qmd
create mode 100644 _tp/TP1--FAS-1001-.pdf
create mode 100644 _travail_session/travail_session_hiver24.pdf
create mode 100644 _travail_session/travail_session_hiver24.qmd
kangakouakouauguste@users-air fas_1001_kanga %
```

Figure 2: Utilisation de add et commit

6: Une fois mes modifications enregistrées, j'ai ajouté (add) ces dernières à mon historique Git avec mon terminal. Par la suite, enregistrez (commit) vos modifications sous le nom de: Création du document du travail de session toujours dans votre terminal.

Avant d'ajouter des fichiers, j'ai vérifié l'état actuel de mon dépôt Git en utilisant **git status** *kangakouakouauguste@users-air fas_1001_kanga % git status*

Ensuite j’ai ajouté tous les fichiers modifiés avec **git add** .
kangakouakouauguste@users-air fas_1001_kanga % git add .
 Après avoir ajouté les fichiers, j’ai enregistré mes modifications avec **git commit** et un message
git commit -m “Création du document du travail de session”
 J’ai vérifié encore une fois le statut de mes fichiers avec **git status**
kangakouakouauguste@users-air fas_1001_kanga % git Status

```
commit de61a46ff3269f94850442581194daaddc75e9f5e (HEAD -> fas_1001_kanga)
Author: KANGA KOUAKOU AUGUSTE <kangakouakouauguste@Users-MacBook-Air.local>
Date: Mon Jan 22 00:16:36 2024 -0500

    Création du document du travail de session

commit f0327ce89cb8265c93503b7180a09c74fd21caf (main)
Author: KANGA KOUAKOU AUGUSTE <kangakouakouauguste@Users-MacBook-Air.local>
Date: Sun Jan 21 23:35:56 2024 -0500

    Création du document du travail de session

commit d2d1b7da6820b5ab1a358b2299ba69609846ff90
Author: KANGA KOUAKOU AUGUSTE <kangakouakouauguste@Users-MacBook-Air.local>
Date: Sun Jan 21 22:43:47 2024 -0500

    Création du document du travail de session

commit 9dd2e9f1876f27b2e980cfe235444ddc45427800 (origin/main, origin/HEAD)
Author: Nadjim Frechet <31970826+Nadjim10@users.noreply.github.com>
Date: Tue Jan 16 04:57:37 2024 -0500

    Update README.md

commit e8a8aaa5696951770dad30f380cc52361ddc5e50
Author: Nadjim10 <nadjimfrechet@gmail.com>
Date: Tue Jan 16 04:06:59 2024 -0500

    removing tp2 - test done

commit e039c124311b5a0951b80837552e0c0e0a0e5eca
Author: Nadjim10 <nadjimfrechet@gmail.com>
Date: Tue Jan 16 03:46:30 2024 -0500

    ignore ds store

commit 7e3f65a14778d0fd2bfa9c9989564c325b977b3
Author: Nadjim10 <nadjimfrechet@gmail.com>
Date: Tue Jan 16 03:43:12 2024 -0500

    test with tp2

commit d54e0f87732db8978f298e20e125b0ff3847d31b
Author: Nadjim Frechet <31970826+Nadjim10@users.noreply.github.com>
Date: Tue Jan 16 03:40:05 2024 -0500

    Update README.md

commit ebe435b42925782e0a756985fb02a7c5a56a9cde
Author: Nadjim10 <nadjimfrechet@gmail.com>
Date: Tue Jan 16 03:36:44 2024 -0500

    modif need to revert

commit 5ecfb40521c090956f664a083da41edad3c334ca
Author: Nadjim10 <nadjimfrechet@gmail.com>
Date: Tue Jan 16 03:33:43 2024 -0500

    remove rproj from gitignore

commit ea7d6798138001aa6e94d109a6eb388074410bb2
Author: Nadjim Frechet <31970826+Nadjim10@users.noreply.github.com>
Date: Tue Jan 16 03:33:43 2024 -0500
```

Figure 3: Utilisation de git log

7: J’ai utilisé la commande **git log** pour voir l’historique de mes premiers pas sur Git

8: Finalement, j’ai rentré ou “pushé” (push en anglais) mes modifications sur GitHub avec **git push origin**.

kangakouakouauguste@users-air fas_1001_kanga % git push origin fas_1001_kanga