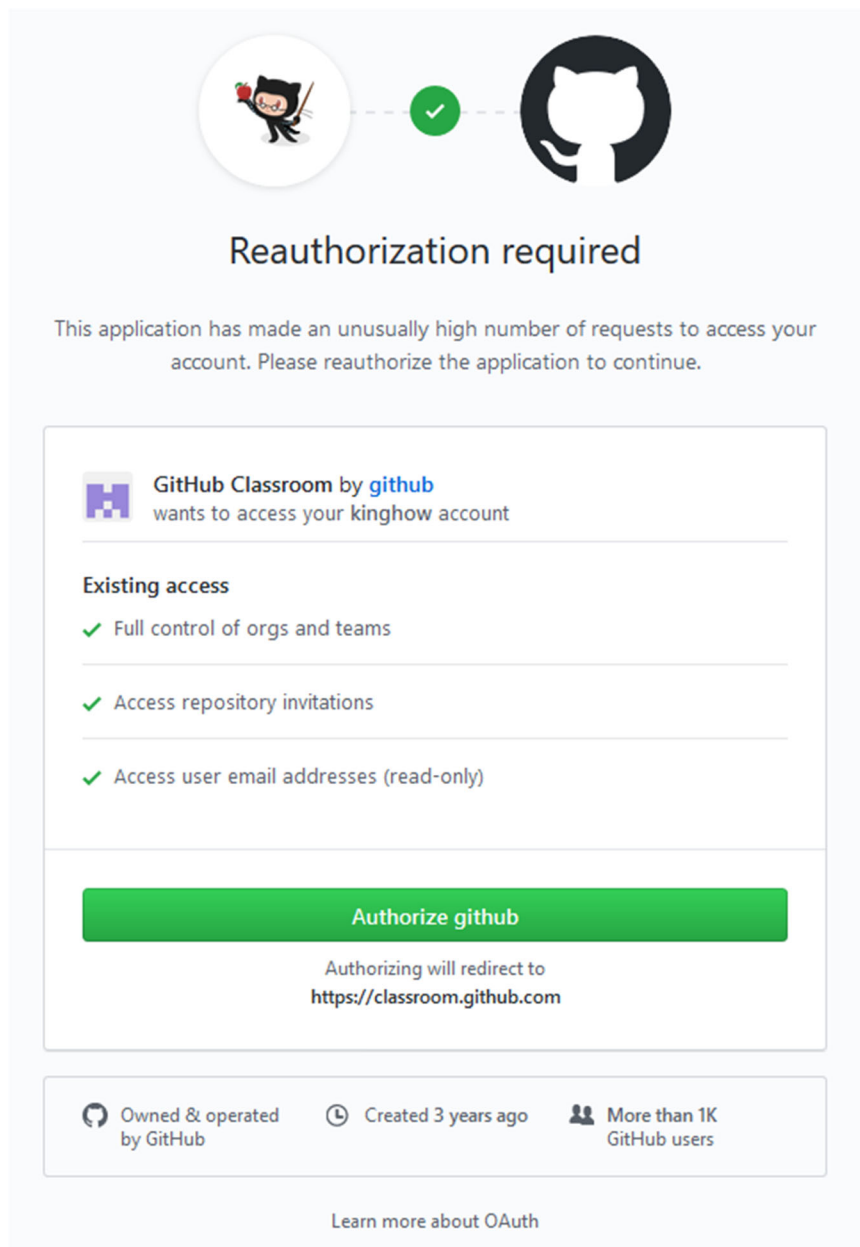


Instructions TP 631-1

Comment récupérer mon TP ?


- 1) Suivez ce lien : <https://classroom.github.com/a/Qfd4PuUS> afin de lier votre compte GitHub et GitHub Classroom.
- 2) Connectez-vous à l'aide de votre compte GitHub et Autorisez de faire le lien entre les deux.



The image shows a GitHub Classroom authorization interface. At the top, there are two circular icons: the GitHub Classroom logo (a cat with a red apple) and the GitHub logo (a white cat on a black background), connected by a dashed line with a green checkmark in the middle. Below this, the text "Reauthorization required" is displayed. A message states: "This application has made an unusually high number of requests to access your account. Please reauthorize the application to continue." The main content area is a white box with a purple header bar. The header bar contains the GitHub Classroom logo and the text "GitHub Classroom by github wants to access your kinghow account". Below the header, there is a section titled "Existing access" with three items, each preceded by a green checkmark: "Full control of orgs and teams", "Access repository invitations", and "Access user email addresses (read-only)". At the bottom of the white box is a large green button labeled "Authorize github". Below the button, it says "Authorizing will redirect to https://classroom.github.com". At the very bottom of the interface, there is a footer section with three items: "Owned & operated by GitHub", "Created 3 years ago", and "More than 1K GitHub users". A link "Learn more about OAuth" is also present.

Reauthorization required

This application has made an unusually high number of requests to access your account. Please reauthorize the application to continue.


 **GitHub Classroom** by **github**
wants to access your kinghow account


Existing access


- ✓ Full control of orgs and teams
- ✓ Access repository invitations
- ✓ Access user email addresses (read-only)

Authorize github

Authorizing will redirect to
<https://classroom.github.com>

 Owned & operated by GitHub

 Created 3 years ago


 More than 1K GitHub users

[Learn more about OAuth](#)

- 3) Choisissez votre Nom Prénom dans la liste (Si vous vous êtes trompés, ou qu'il n'y est pas contactez moi au plus vite !)

631-1-PROG

631-1-PROG

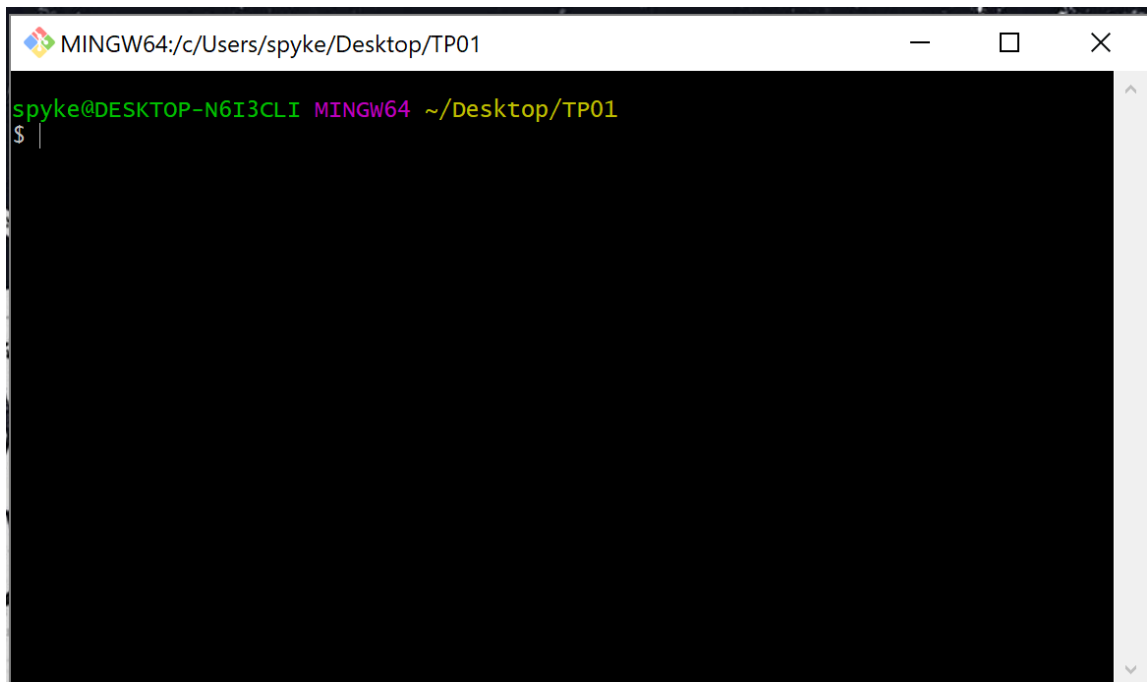
 Join the classroom roster

Your teacher has configured this classroom to pair GitHub accounts with identifiers. Please select yourself from the list below. You can also [skip](#) this step for now.

Identifiers
Ahmed Rafat
Almiral Renie
Alouan Bayan
Alvaipillai Anushan
Alvarez Méline
Apothéloz Luca
Aydin Fatma

- 4) Un message de confirmation s'affichera vous confirmant que le lien est fait et vous donnant le lien vers votre Repo Git créé avec le TP01 déjà à l'intérieur. (Il faut cliquer sur le lien 631-1-PROG/tp01-« votre username GitHub » pour y accéder)
- 5) Pour pouvoir travailler sur ces fichiers il va falloir les récupérer depuis GitHub. Créez un dossier où vous mettrez vos TPs et travaillerez si vous n'en n'avez pas déjà un. Par exemple sur votre Z:/ ou sur le bureau.
Créez un dossier pour la programmation et à l'intérieur de celui-ci un dossier pour le TP01.

- 6) Clic droit sur le dossier TP01 puis « Git Bash Here » et obtenir cette fenêtre. (Si vous n'avez pas cette option il faut installer Git : <https://git-scm.com/downloads>)



- 7) Tapez les commandes suivantes :

```
git init
git remote add origin votre_lien_git
git pull votre_lien_git master
```




votre_lien_git doit ressembler à ceci : <https://github.com/631-1-PROG/tp01-AurelieSauge.git>



Il s'agit de l'URL de votre Repo Git + .git

Ne pas oublier d'avoir le .git à la fin du lien !!

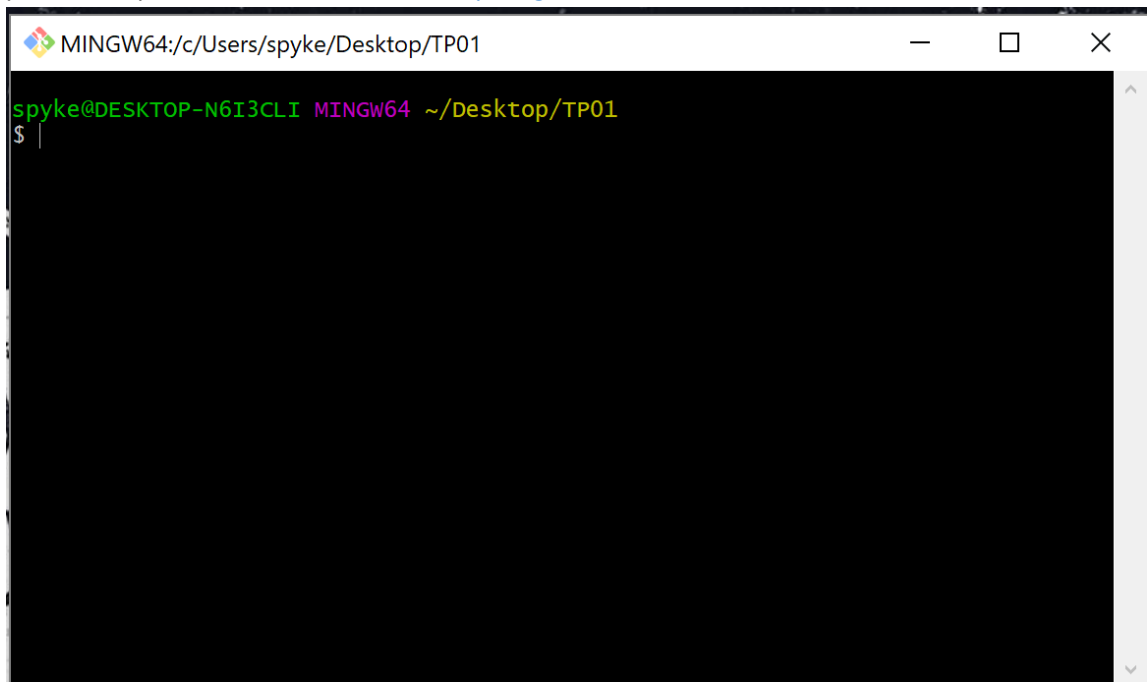
- 8) Entrez vos identifiants.
- 9) Les fichiers devraient se trouver dans votre dossier TP01

 TP01.docx	25.09.2019 14:06	Document Microsoft ...	89 Ko
 tp01_ex1.py	25.09.2019 14:06	Python File	1 Ko
 tp01_ex2.py	25.09.2019 14:06	Python File	2 Ko
 tp01_ex3.py	25.09.2019 14:06	Python File	1 Ko
 tp01_ex4.py	25.09.2019 14:06	Python File	2 Ko
 tp01_ex5.py	25.09.2019 14:06	Python File	1 Ko

- 10) Vous pouvez commencer à travailler ☺ ! En cas de problème, n'hésitez surtout pas à me contacter : jerome.humbert@hesge.ch

Comment déposer ma sauvegarde sur GitHub ?

Clic droit sur le dossier où vous travaillez puis « Git Bash Here » et obtenir cette fenêtre. (Si vous n'avez pas cette option il faut installer Git : <https://git-scm.com/downloads>)



Tapez les commandes suivantes :

```
git add .
```

```
git commit -m "Le nom de votre sauvegarde"
```

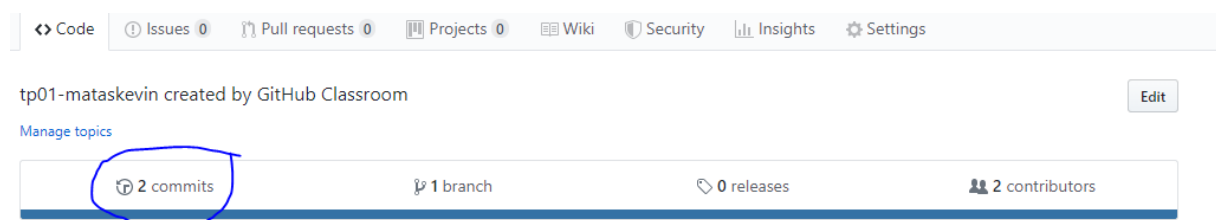
```
git push
```

Si cela ne marche pas :

```
git push votre_lien_git master
```

votre_lien_git doit ressembler à ceci : <https://github.com/631-1-PROG/tp01-AurelieSauge.git> avec votre nom de Repo à la place.

Vous pouvez ensuite vérifier que votre sauvegarde s'est bien déroulée sur votre Repo :



Comment récupérer la dernière sauvegarde que j'ai mise sur GitHub ?

- Si c'est dans un autre dossier de travail (vierge), répétez « Comment récupérer mon TP » à partir de l'étape 5
- Si c'est dans le même dossier de travail qu'auparavant, `git pull votre_lien_git master` devrait suffire

Erreurs fréquentes

« On branch master nothing to commit, working directory clean »

Essayez de recréer un autre dossier local (sur votre machine) et déplacez vos fichiers à l'intérieur (sans le dossier .git !). Faites la commande `git init` suivi des commandes que vous souhaitez faire.

« error: src refspec master does not match any. error: failed to push some refs... »

Cela apparaît lorsque vous n'avez pas la « branch » master. Pour la créer il faut lancer un premier commit. Pour cela entrez la commande : `git commit -m "Mon nom de sauvegarde"`

<< remote: Invalid username or password. fatal: Authentication failed for >>

Vous avez fait une erreur dans votre username ou mot de passe. Réessayez !

<< error: pathspec 'save' did not match any file(s) known to git.>>

Il faut utiliser les doubles guillemets lors du commit :

```
git commit -m "ma sauvegarde"
```

Et non

```
git commit -m 'ma sauvegarde'
```