

# GitHub et GitHub classroom pour la gestion des TP en CPGE

Jean-Julien FLECK

Lycée Kléber

Luminy 2019

## Plan de l'exposé

- 1 Gestion des TP, ancien protocole
- 2 git, GitHub et GitHub Classroom
- 3 Gestion des TP, nouveau protocole
- 4 Diverses utilisation possibles
- 5 Conclusions
- 6 Mise en place de GitHub Classroom

## Gestion des TP, ancien protocole

# Gestion des TP

## Organisation avant GitHub

- Dossier de chaque TP distribué (depuis un ordi du lycée)  
sur les sessions des élèves (sur les ordis du lycée)
- Travail des élèves sur les ordis du lycée et récupération sur  
clé USB pour bosser à la maison
- Ramassage des dossiers des élèves (depuis un ordi du lycée)

# Gestion des TP

## Organisation avant GitHub

- Dossier de chaque TP distribué (depuis un ordi du lycée) sur les sessions des élèves (sur les ordis du lycée)
- Travail des élèves sur les ordis du lycée et récupération sur clé USB pour bosser à la maison
- Ramassage des dossiers des élèves (depuis un ordi du lycée)

# Gestion des TP

## Organisation avant GitHub

- Dossier de chaque TP distribué (depuis un ordi du lycée) sur les sessions des élèves (sur les ordis du lycée)
- Travail des élèves sur les ordis du lycée et récupération sur clé USB pour bosser à la maison
- Ramassage des dossiers des élèves (depuis un ordi du lycée)

# Gestion des TP

## Problèmes divers

- Besoin de penser chaque semaine à se connecter sur un ordi du lycée pour mettre en place le TP
- Défilé des élèves le jour de la deadline (le mardi) pour accéder aux ordis du lycée et y mettre leur travail du WE.
- Pas de sauvegarde régulière du travail (bug Pyzo d'enregistrement du fichier temporaire)
- Nombreux mails d'élèves qui ont oublié de mettre leur travail sur les ordis du lycée
- Pas d'archivage du côté élève (les dossiers sont effacés après récupération)
- etc.

# Gestion des TP

## Problèmes divers

- Besoin de penser chaque semaine à se connecter sur un ordi du lycée pour mettre en place le TP
- Défilé des élèves le jour de la deadline (le mardi) pour accéder aux ordis du lycée et y mettre leur travail du WE.
- Pas de sauvegarde régulière du travail (bug Pyzo d'enregistrement du fichier temporaire)
- Nombreux mails d'élèves qui ont oublié de mettre leur travail sur les ordis du lycée
- Pas d'archivage du côté élève (les dossiers sont effacés après récupération)
- etc.

# Gestion des TP

## Problèmes divers

- Besoin de penser chaque semaine à se connecter sur un ordi du lycée pour mettre en place le TP
- Défilé des élèves le jour de la deadline (le mardi) pour accéder aux ordis du lycée et y mettre leur travail du WE.
- Pas de sauvegarde régulière du travail (bug Pyzo d'enregistrement du fichier temporaire)
- Nombreux mails d'élèves qui ont oublié de mettre leur travail sur les ordis du lycée
- Pas d'archivage du côté élève (les dossiers sont effacés après récupération)
- etc.

# Gestion des TP

## Problèmes divers

- Besoin de penser chaque semaine à se connecter sur un ordi du lycée pour mettre en place le TP
- Défilé des élèves le jour de la deadline (le mardi) pour accéder aux ordis du lycée et y mettre leur travail du WE.
- Pas de sauvegarde régulière du travail (bug Pyzo d'enregistrement du fichier temporaire)
- Nombreux mails d'élèves qui ont oublié de mettre leur travail sur les ordis du lycée
- Pas d'archivage du côté élève (les dossiers sont effacés après récupération)
- etc.

# Gestion des TP

## Problèmes divers

- Besoin de penser chaque semaine à se connecter sur un ordi du lycée pour mettre en place le TP
- Défilé des élèves le jour de la deadline (le mardi) pour accéder aux ordis du lycée et y mettre leur travail du WE.
- Pas de sauvegarde régulière du travail (bug Pyzo d'enregistrement du fichier temporaire)
- Nombreux mails d'élèves qui ont oublié de mettre leur travail sur les ordis du lycée
- Pas d'archivage du côté élève (les dossiers sont effacés après récupération)
- etc.

# Gestion des TP

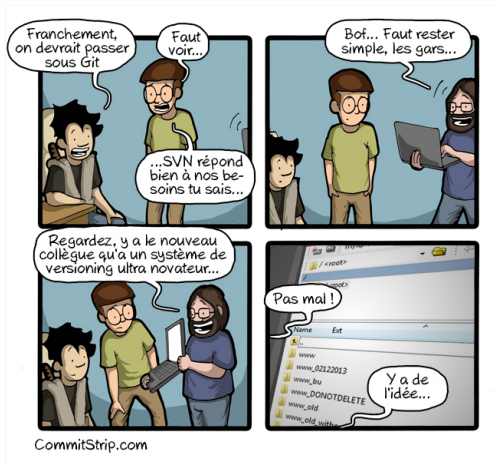
## Problèmes divers

- Besoin de penser chaque semaine à se connecter sur un ordi du lycée pour mettre en place le TP
- Défilé des élèves le jour de la deadline (le mardi) pour accéder aux ordis du lycée et y mettre leur travail du WE.
- Pas de sauvegarde régulière du travail (bug Pyzo d'enregistrement du fichier temporaire)
- Nombreux mails d'élèves qui ont oublié de mettre leur travail sur les ordis du lycée
- Pas d'archivage du côté élève (les dossiers sont effacés après récupération)
- etc.

# git, GitHub et GitHub Classroom

# git

Logiciel de gestion de versions décentralisé



# git

## Logiciel de gestion de versions décentralisé

- Logiciel libre créé par Linus Torvald en 2005
- Permet des sauvegardes successives mais aussi le travail en parallèle de plusieurs personnes sur un même jeu de fichier (penser aux multiples versions des programmes de simulation en TIPE)
- Accessible par la ligne de commande (pratique pour scripter mais pas trop pour les élèves)
- Fonctionne en terme de « commit » qui rassemble des changements atomiques
- Permet de multiples actions et un suivi historique des modifications successives.

# git

Logiciel de gestion de versions décentralisé

- Logiciel libre créé par Linus Torvald en 2005
- Permet des sauvegardes successives mais aussi le travail en parallèle de plusieurs personnes sur un même jeu de fichier (penser aux multiples versions des programmes de simulation en TIPE)
- Accessible par la ligne de commande (pratique pour scripter mais pas trop pour les élèves)
- Fonctionne en terme de « commit » qui rassemble des changements atomiques
- Permet de multiples actions et un suivi historique des modifications successives.

# git

Logiciel de gestion de versions décentralisé

- Logiciel libre créé par Linus Torvald en 2005
- Permet des sauvegardes successives mais aussi le travail en parallèle de plusieurs personnes sur un même jeu de fichier (penser aux multiples versions des programmes de simulation en TIPE)
- Accessible par la ligne de commande (pratique pour scripter mais pas trop pour les élèves)
- Fonctionne en terme de « commit » qui rassemble des changements atomiques
- Permet de multiples actions et un suivi historique des modifications successives.

# git

Logiciel de gestion de versions décentralisé

- Logiciel libre créé par Linus Torvald en 2005
- Permet des sauvegardes successives mais aussi le travail en parallèle de plusieurs personnes sur un même jeu de fichier (penser aux multiples versions des programmes de simulation en TIPE)
- Accessible par la ligne de commande (pratique pour scripter mais pas trop pour les élèves)
- Fonctionne en terme de « commit » qui rassemble des changements atomiques
- Permet de multiples actions et un suivi historique des modifications successives.

# git

Logiciel de gestion de versions décentralisé

- Logiciel libre créé par Linus Torvald en 2005
- Permet des sauvegardes successives mais aussi le travail en parallèle de plusieurs personnes sur un même jeu de fichier (penser aux multiples versions des programmes de simulation en TIPE)
- Accessible par la ligne de commande (pratique pour scripter mais pas trop pour les élèves)
- Fonctionne en terme de « commit » qui rassemble des changements atomiques
- Permet de multiples actions et un suivi historique des modifications successives.

# GitHub

## Une interface web pour git

- Site web permettant un accès centralisé aux repository git (<https://github.com>)
- Sorte de «facebook» des développeurs
- Modèle commercial : repositories publics illimités mais privés payant au-delà de 3 collaborateurs (sauf organisations à visées pédagogiques)

# GitHub

## Une interface web pour git

- Site web permettant un accès centralisé aux repository git (<https://github.com>)
- Sorte de «facebook» des développeurs
- Modèle commercial : repositories publics illimités mais privés payant au-delà de 3 collaborateurs (sauf organisations à visées pédagogiques)


# GitHub

## Une interface web pour git

- Site web permettant un accès centralisé aux repository git (<https://github.com>)
- Sorte de «facebook» des développeurs
- Modèle commercial : repositories publics illimités mais privés payant au-delà de 3 collaborateurs (sauf organisations à visées pédagogiques)

# GitHub

## Exemple du noyau Linux


 [torvalds / linux](#) Watch 6,694 Star 72,315 Fork 25,612

[Code](#) [Pull requests 259](#) [Projects 0](#) [Insights](#)

Linux kernel source tree

[825,768 commits](#) [1 branch](#) [598 releases](#) [∞ contributors](#) [View license](#)

Branch: master [New pull request](#) [Create new file](#) [Upload files](#) [Find File](#) [Clone or download](#)

 [torvalds](#) Merge branch 'akpm' (patches from Andrew) [...](#) Latest commit f654f0f 5 hours ago

# GitHub

## Exemple de visualisation de compte



**Jean-Julien  
Joseph Fleck**  
jjfPCSI1

Block or report user

- 📍 Strasbourg
- ✉ [Sign in to view email](#)
- 🌐 [pcsi.kleber.free.fr/IPT](#)

Overview   Repositories **21**   Projects **0**   Stars **2**   Followers **7**   Following **1**

### Pinned

[py4phys](#)

Projet de programmes Python pour la physique en CPGE

🟠 Jupyter Notebook   ★ 14   🍷 7

[animations\\_Jatex](#)

● TeX   ★ 1

[codes\\_concours\\_CPGE](#)

Propositions d'implémentations pour les sujets proposés aux différentes épreuves d'informatique des concours en CPGE

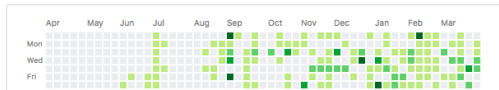
● Python   ★ 1   🍷 1

[beamer-TIPE-2019](#)

Présentation Beamer pour TIPE 2019

● TeX

5,479 contributions in the last year



2019

2018

2017

2016

# GitHub Desktop

Interface graphique pour interagir avec GitHub

- Installation facile depuis <https://desktop.github.com/>
- Permet de rattachier les repositories depuis GitHub.com sur l'ordinateur en local (Clone Repository...)
- Permet de faire les commits depuis l'ordinateur local.
- Aucune connexion requise pour les commits (on peut bosser dans le train par exemple), il suffit de penser à «pousser» les modifications une fois la connexion rétablie.
- Très loin d'utiliser toutes la puissance de git mais justement utile pour ne pas effrayer les élèves

# GitHub Desktop

Interface graphique pour interagir avec GitHub

- Installation facile depuis <https://desktop.github.com/>
- Permet de rattachier les repositories depuis GitHub.com sur l'ordinateur en local (**Clone Repository...**)
- Permet de faire les commits depuis l'ordinateur local.
- Aucune connexion requise pour les commits (on peut bosser dans le train par exemple), il suffit de penser à «pousser» les modifications une fois la connexion rétablie.
- Très loin d'utiliser toutes la puissance de git mais justement utile pour ne pas effrayer les élèves

# GitHub Desktop

Interface graphique pour interagir avec GitHub

- Installation facile depuis <https://desktop.github.com/>
- Permet de rattachier les repositories depuis GitHub.com sur l'ordinateur en local (Clone Repository...)
- Permet de faire les commits depuis l'ordinateur local.
- Aucune connexion requise pour les commits (on peut bosser dans le train par exemple), il suffit de penser à «pousser» les modifications une fois la connexion rétablie.
- Très loin d'utiliser toutes la puissance de git mais justement utile pour ne pas effrayer les élèves

# GitHub Desktop

Interface graphique pour interagir avec GitHub

- Installation facile depuis <https://desktop.github.com/>
- Permet de rattachier les repositories depuis GitHub.com sur l'ordinateur en local (Clone Repository...)
- Permet de faire les commits depuis l'ordinateur local.
- Aucune connexion requise pour les commits (on peut bosser dans le train par exemple), il suffit de penser à «pousser» les modifications une fois la connexion rétablie.
- Très loin d'utiliser toutes la puissance de git mais justement utile pour ne pas effrayer les élèves

# GitHub Desktop

Interface graphique pour interagir avec GitHub

- Installation facile depuis <https://desktop.github.com/>
- Permet de rattachier les repositories depuis GitHub.com sur l'ordinateur en local (Clone Repository...)
- Permet de faire les commits depuis l'ordinateur local.
- Aucune connexion requise pour les commits (on peut bosser dans le train par exemple), il suffit de penser à «pousser» les modifications une fois la connexion rétablie.
- Très loin d'utiliser toutes la puissance de `git` mais justement utile pour ne pas effrayer les élèves

# GitHub Desktop

## Exemple de visualisation d'historique

The screenshot shows the GitHub Desktop application interface. At the top, it displays the current repository as 'info\_sem2\_2018-2019-feldel7', the current branch as 'master', and the last fetch time as 'Last fetched 2 days ago'. The main area is split into two panes. The left pane shows the commit history, listing several commits with their authors and dates, such as 'Gauss' by feldel7 on Mar 10, 2019, and 'Erreur pour créer l'histogramme' by feldel7 on Mar 9, 2019. The right pane shows a diff view for the file 'TP/TP09/TP0\_t\_probab.py'. The diff highlights changes between two versions of the file. The top part of the diff shows a change from a function 'Ftri\_decaxe(x)' to 'Ftriangle(x)'. The bottom part shows a change in the 'tirage\_triangle()' function, where the range parameter in 'plt.hist' was updated from '(0,1)' to '(2,4)'. The interface also includes search and refresh icons in the bottom right corner.

Current Repository: info\_sem2\_2018-2019-feldel7  
Current Branch: master  
Fetch origin: Last fetched 2 days ago

Changes: No Branches to Compare

History:

- Gauss  
feldel7 committed Mar 10, 2019
- tirage triangle  
feldel7 committed Mar 10, 2019
- moy et var  
feldel7 committed Mar 9, 2019
- On voit pas les PCSH  
feldel7 committed Mar 9, 2019
- Erreur pour créer l'histogramme  
feldel7 committed Mar 9, 2019
- Update TP09\_integrat...  
feldel7 committed Mar 7, 2019
- On verra  
feldel7 committed Mar 7, 2019
- Ca  
feldel7 committed Mar 7, 2019
- Arrivée TP10  
Jean-Julien Fleck committed Mar 5, 2019
- Dichotomie  
feldel7 committed Feb 27, 2019
- error

Erreur pour créer l'histogramme  
feldel7 committed fe229bc 1 changed file

@jjfps11 il doit y avoir un problème dans la génération de mon histogramme sur triangle, pourriez-vous m'aider s'il vous plaît?

TP/TP09/TP0\_t\_probab.py

```
104 -def Ftri_decaxe(x):  
104 +def Ftriangle(x):  
105     return integration_trapeze(triangle,0,x,10)  
106  
107 def tirage_triangle():  
107 @@ -109,10 +109,10 @@ def tirage_triangle():  
109     P=[]  
110     for i in range(nb):  
111         alea=random.random()  
112 -     def f(x):  
112 +     def F(x):  
113 -         return Ftri_decaxe(x)-alea  
113 +         return Ftriangle(x)-alea  
114 -         P.append(zero_dichotomie(F,2,4))  
114 +         P.append(zero_dichotomie(F,2,4))  
115 -         plt.hist(P,bins=16,range=(0,1))  
115 +         plt.hist(P,bins=16,range=(2,4))  
116  
116         plt.title('Loi de distribution triangle')  
117         plt.savefig('TP09_tirage_triangle_Delattre.png')  
118         plt.show()  
118  
118 @@ -122,7 +122,7 @@ def tirage_triangle():  
122  
122  
123 # Ligne suivante a decommenter uniquement quand vous voulez faire votre tira
```

# GitHub Desktop

## Exemple de visualisation de commit

The screenshot shows the GitHub Desktop interface. At the top, it displays the repository name 'GitHub-luminy', the current branch 'master', and the push status 'Never fetched'. Below this, the 'Changes' tab is active, showing a list of 2 changed files: 'figures/githubdesktop\_commit.png' and 'slides/01\_premiere\_partie.tex'. The 'History' tab is also visible, showing the commit history for 'slides/01\_premiere\_partie.tex'. The commit visualization shows a commit with 2 changed files. The file 'slides/01\_premiere\_partie.tex' is highlighted in blue, and its content is shown in a light green background. The content is a LaTeX document with a frame and a diapo (slide) titled 'github Desktop'. The commit message is 'Rajout de la diapo commit'.

Current Repository: GitHub-luminy

Current Branch: master

Push origin: Never fetched

Changes: 2 changed files

- figures/githubdesktop\_commit.png
- slides/01\_premiere\_partie.tex

History: slides/01\_premiere\_partie.tex

```
@@ -102,6 +102,19 @@
102 | 102 |
103 | 103 | \end{frame}
104 | 104 |
105 | 105 | +~~~~~
106 | 106 | +% Diapo github Desktop
107 | 107 | +~~~~~
108 | 108 | +\begin{frame}
109 | 109 | + \frametitle{GitHub Desktop}
110 | 110 | + \framesubtitle{Exemple de visualisation de commit}
111 | 111 | +
112 | 112 | + \begin{center}
113 | 113 | + \includegraphics[width=\linewidth]{figures/githubdesktop_commit.png}
114 | 114 | + \end{center}
115 | 115 | +
116 | 116 | +\end{frame}
117 | 117 | +
118 | 118 | +~~~~~
119 | 119 | +% Diapo classroom
120 | 120 | +~~~~~
```

Rajout de la diapo commit

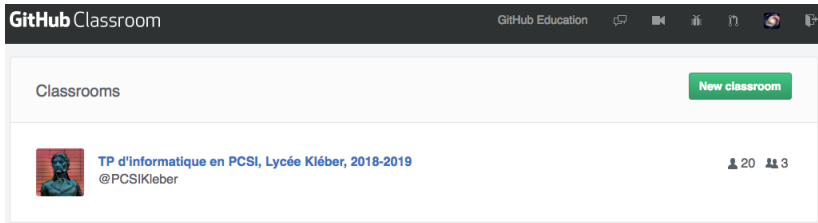
Il s'agit de montrer comment se fait un commit avec une remarque

Commit to master

Committed 4 minutes ago  
Avancée jusqu'à GitHubDesktop

# GitHub Classroom <https://classroom.github.com/>

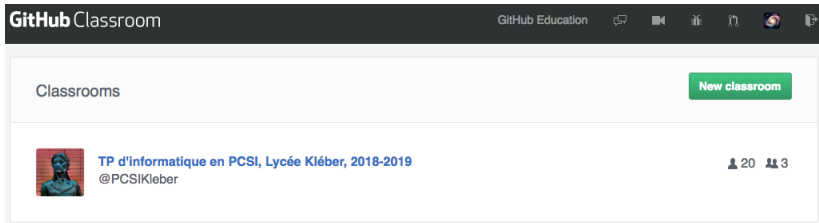
Un service annexe de GitHub



- Automatise le processus de création de repositories pour les élèves
- Permet de rattachier les repositories sur l'ordi prof via «classroom assistant»  
(<https://classroom.github.com/assistant>)

# GitHub Classroom <https://classroom.github.com/>


Un service annexe de GitHub



GitHub Classroom

GitHub Education

Classrooms New classroom

 **TP d'informatique en PCSI, Lycée Kléber, 2018-2019**  
@PCSIKleber 20 3

- Automatise le processus de création de repositories pour les élèves
- Permet de rattachier les repositories sur l'ordi prof via «classroom assistant»  
(<https://classroom.github.com/assistant>)

## Gestion des TP, nouveau protocole

# Gestion des TP

## Organisation après GitHub

- Dossier de chaque TP distribué via GitHub depuis n'importe où (souvent la maison), le tout pouvant être préparé à l'avance facilement.
- Travail des élèves sur les ordis du lycée ou leur propre ordi en ayant toujours la dernière version à jour (à condition d'avoir pensé à «pousser» leurs derniers commits)
- Ramassage des dossiers des élèves de n'importe où (et à n'importe quelle heure) avec possibilité d'imposer une deadline indépendante de l'aspect matériel (on prend le dernier commit avant la deadline, sans compter les suivants).

```
git checkout $(git rev-list -n 1 -before="2018-09-15 13:37" master)
```

# Gestion des TP

## Organisation après GitHub

- Dossier de chaque TP distribué via GitHub depuis n'importe où (souvent la maison), le tout pouvant être préparé à l'avance facilement.
- Travail des élèves sur les ordis du lycée ou leur propre ordi en ayant toujours la dernière version à jour (à condition d'avoir pensé à «pousser» leurs derniers commits)
- Ramassage des dossiers des élèves de n'importe où (et à n'importe quelle heure) avec possibilité d'imposer une deadline indépendante de l'aspect matériel (on prend le dernier commit avant la deadline, sans compter les suivants).

```
git checkout $(git rev-list -n 1 -before="2018-09-15 13:37" master)
```

# Gestion des TP

## Organisation après GitHub

- Dossier de chaque TP distribué via GitHub depuis n'importe où (souvent la maison), le tout pouvant être préparé à l'avance facilement.
- Travail des élèves sur les ordis du lycée ou leur propre ordi en ayant toujours la dernière version à jour (à condition d'avoir pensé à «pousser» leurs derniers commits)
- Ramassage des dossiers des élèves de n'importe où (et à n'importe quelle heure) avec possibilité d'imposer une deadline indépendante de l'aspect matériel (on prend le dernier commit avant la deadline, sans compter les suivants).

```
git checkout `git rev-list -n 1 -before="2018-09-15 13:37" master`
```

# Gestion des TP

## Avantages élèves

- Sauvegarde et archivage automatique du travail des élèves.



### STOP GitHub


Allez sur Github Desktop pour faire un commit. Choisissez vous-même (avec pertinence) le résumé.  
Pensez aussi à appuyer sur le bouton «Push origin» en haut à droite pour mettre à jour sur le web.

- Possibilité de poser des questions lors des commits.
- Feedback possible depuis n'importe où directement dans le code des commits des élèves.

# Gestion des TP

## Avantages élèves

- Sauvegarde et archivage automatique du travail des élèves.

 **STOP GitHub**  
Allez sur Github Desktop pour faire un commit. Choisissez vous-même (avec pertinence) le résumé.  
Pensez aussi à appuyer sur le bouton «Push origin» en haut à droite pour mettre à jour sur le web.

- Possibilité de poser des questions lors des commits.

@jjfPCS11


Comment trouver la position de Eo (cad wo) dans la liste ?  
Comment parcourir la liste depuis la fin?

- Feedback possible depuis n'importe où directement dans le code des commits des élèves.

# Gestion des TP

## Avantages élèves

- Sauvegarde et archivage automatique du travail des élèves.

 **STOP GitHub**  
Allez sur Github Desktop pour faire un commit. Choisissez vous-même (avec pertinence) le résumé.  
Pensez aussi à appuyer sur le bouton «Push origin» en haut à droite pour mettre à jour sur le web.

- Possibilité de poser des questions lors des commits.

@jjfPCSI1

Comment trouver la position de Eo (cad wo) dans la liste ?  
Comment parcourir la liste depuis la fin?

- Feedback possible depuis n'importe où directement dans le code des commits des élèves.

```
15 +for k in range (len(f)):  
16 +   if f[k]==Eo/2**((1/2):
```



jjfPCSI1 on 28 Jan

Il n'y a aucune chance que l'on soit tombé *pile* sur la bonne valeur. Au lieu de cela, vérifiez une inégalité qui fasse intervenir `i[k]` et `i[k+1]`.

# Gestion des TP

## Avantages élèves

```
29 +   if mesures[0] < mesures[1]:  
30 +       L.append(1)  
31 +   elif mesures[-1] > mesures[-2]:
```



jjfPCS11 on 19 Jan

Vous devez conserver l'ordre: cette vérification est donc à faire en dernier.



AminalInfo on 29 Jan Author

Parce qu'il y avait marqué de renvoyer "dans l'ordre" !



Reply...

```
32 +       L.append(t-1)  
33 +   for i in range(len(mesures)-1):
```



jjfPCS11 on 19 Jan

Vous devriez commencer à 1 et non à 0



AminalInfo on 29 Jan Author

là du coup tu savais mais comme avant c'était faux bah c'était faux aussi

# Gestion des TP

## Avantages profs

- Suivi facilité du travail des élèves durant les TP (et au-dehors)
- Conseils localisés et personnalisés
- Accès direct au code pour identifier les problèmes
- Détermination facilitée des fraudeurs potentiels
- Gestion entière scriptée, moins de manipulations «manuelles» nécessaires

# Gestion des TP

## Avantages profs

- Suivi facilité du travail des élèves durant les TP (et au-dehors)
- Conseils localisés et personnalisés
- Accès direct au code pour identifier les problèmes
- Détermination facilitée des fraudeurs potentiels
- Gestion entière scriptée, moins de manipulations «manuelles» nécessaires

# Gestion des TP

## Avantages profs

- Suivi facilité du travail des élèves durant les TP (et au-dehors)
- Conseils localisés et personnalisés
- Accès direct au code pour identifier les problèmes
- Détermination facilitée des fraudeurs potentiels
- Gestion entière scriptée, moins de manipulations «manuelles» nécessaires

# Gestion des TP

## Avantages profs

- Suivi facilité du travail des élèves durant les TP (et au-dehors)
- Conseils localisés et personnalisés
- Accès direct au code pour identifier les problèmes
- Détermination facilitée des fraudeurs potentiels
- Gestion entière scriptée, moins de manipulations «manuelles» nécessaires

# Gestion des TP

## Avantages profs

- Suivi facilité du travail des élèves durant les TP (et au-dehors)
- Conseils localisés et personnalisés
- Accès direct au code pour identifier les problèmes
- Détermination facilitée des fraudeurs potentiels
- Gestion entière scriptée, moins de manipulations «manuelles» nécessaires

## Diverses utilisation possibles

# Un devoir par TP par élève

## Avantages et inconvénients

- Avantages :
  - Bien pour démarrer
  - Pas d'action à faire sur le repository après avoir distribué le devoir
  - Possibilité de corriger si les «bêta-testeurs» trouvent des bugs : sera pris en compte pour les connexions ultérieures (mais pas pour ceux qui ont déjà accepté le devoir)
- Inconvénients :
  - Accumulation de nombreux dossiers dans le répertoire GitHub/
  - Nécessite de cliquer sur un lien à chaque nouveau TP (dur dur pour les élèves...)
  - Demande de rappatrier systématiquement les dossiers par «classroom assistant» via l'interface web.

# Un devoir par TP par élève

## Avantages et inconvénients

- Avantages :
  - Bien pour démarrer
  - Pas d'action à faire sur le repository après avoir distribué le devoir
  - Possibilité de corriger si les «bêta-testeurs» trouvent des bugs : sera pris en compte pour les connexions ultérieures (mais pas pour ceux qui ont déjà accepté le devoir)
- Inconvénients :
  - Accumulation de nombreux dossiers dans le répertoire GitHub/
  - Nécessite de cliquer sur un lien à chaque nouveau TP (dur dur pour les élèves...)
  - Demande de rappatrier systématiquement les dossiers par «classroom assistant» via l'interface web.

# Un devoir par TP par élève

## Avantages et inconvénients

- Avantages :
  - Bien pour démarrer
  - Pas d'action à faire sur le repository après avoir distribué le devoir
  - Possibilité de corriger si les «bêta-testeurs» trouvent des bugs : sera pris en compte pour les connexions ultérieures (mais pas pour ceux qui ont déjà accepté le devoir)
- Inconvénients :
  - Accumulation de nombreux dossiers dans le répertoire GitHub/
  - Nécessite de cliquer sur un lien à chaque nouveau TP (dur dur pour les élèves...)
  - Demande de rappatrier systématiquement les dossiers par «classroom assistant» via l'interface web.

# Un devoir par TP par élève

## Avantages et inconvénients

- Avantages :
  - Bien pour démarrer
  - Pas d'action à faire sur le repository après avoir distribué le devoir
  - Possibilité de corriger si les «bêta-testeurs» trouvent des bugs : sera pris en compte pour les connexions ultérieures (mais pas pour ceux qui ont déjà accepté le devoir)
- Inconvénients :
  - Accumulation de nombreux dossiers dans le répertoire GitHub/
  - Nécessité de cliquer sur un lien à chaque nouveau TP (dur dur pour les élèves...)
  - Demande de rappatrier systématiquement les dossiers par «classroom assistant» via l'interface web.

# Un devoir par TP par élève

## Avantages et inconvénients

- Avantages :
  - Bien pour démarrer
  - Pas d'action à faire sur le repository après avoir distribué le devoir
  - Possibilité de corriger si les «bêta-testeurs» trouvent des bugs : sera pris en compte pour les connexions ultérieures (mais pas pour ceux qui ont déjà accepté le devoir)
- Inconvénients :
  - Accumulation de nombreux dossiers dans le répertoire GitHub/
  - Nécessite de cliquer sur un lien à chaque nouveau TP (dur dur pour les élèves...)
  - Demande de rattraper systématiquement les dossiers par «classroom assistant» via l'interface web.

# Un devoir par TP par élève

## Avantages et inconvénients

- Avantages :
  - Bien pour démarrer
  - Pas d'action à faire sur le repository après avoir distribué le devoir
  - Possibilité de corriger si les «bêta-testeurs» trouvent des bugs : sera pris en compte pour les connexions ultérieures (mais pas pour ceux qui ont déjà accepté le devoir)
- Inconvénients :
  - Accumulation de nombreux dossiers dans le répertoire `GitHub/`
  - Nécessite de cliquer sur un lien à chaque nouveau TP (dur dur pour les élèves...)
  - Demande de rattraper systématiquement les dossiers par «classroom assistant» via l'interface web.

# Un devoir par TP par élève

## Avantages et inconvénients

- Avantages :
  - Bien pour démarrer
  - Pas d'action à faire sur le repository après avoir distribué le devoir
  - Possibilité de corriger si les «bêta-testeurs» trouvent des bugs : sera pris en compte pour les connexions ultérieures (mais pas pour ceux qui ont déjà accepté le devoir)
- Inconvénients :
  - Accumulation de nombreux dossiers dans le répertoire GitHub/
  - Nécessite de cliquer sur un lien à chaque nouveau TP (dur dur pour les élèves...)
  - Demande de rattraper systématiquement les dossiers par «classroom assistant» via l'interface web.

# Un devoir par TP par élève

## Avantages et inconvénients

- Avantages :
  - Bien pour démarrer
  - Pas d'action à faire sur le repository après avoir distribué le devoir
  - Possibilité de corriger si les «bêta-testeurs» trouvent des bugs : sera pris en compte pour les connexions ultérieures (mais pas pour ceux qui ont déjà accepté le devoir)
- Inconvénients :
  - Accumulation de nombreux dossiers dans le répertoire GitHub/
  - Nécessite de cliquer sur un lien à chaque nouveau TP (dur dur pour les élèves...)
  - Demande de rappatrier systématiquement les dossiers par «classroom assistant» via l'interface web.

# Un dossier semestriel par élève

## Avantages et inconvénients

- Avantages :

- Possibilité d'organiser les dossiers (par TP, par PP, etc.)
- Un seul rattachement global, puis des «pull» suffisent dans chaque dossier (scriptable)
- Les «pull» récupèrent en parallèles tous les projets en cours
- Ajouts automatisés des nouveaux dossiers, modifications des instructions possible par après

- Inconvénients :

- Nécessité de savoir scripter l'ajout automatique dans tous les dossiers

# Un dossier semestriel par élève

## Avantages et inconvénients

- Avantages :
  - Possibilité d'organiser les dossiers (par TP, par PP, etc.)
  - Un seul rattachement global, puis des «pull» suffisent dans chaque dossier (scriptable)
  - Les «pull» récupèrent en parallèles tous les projets en cours
  - Ajouts automatisés des nouveaux dossiers, modifications des instructions possible par après
- Inconvénients :
  - Nécessité de savoir scripter l'ajout automatique dans tous les dossiers

# Un dossier semestriel par élève

## Avantages et inconvénients

- Avantages :
  - Possibilité d'organiser les dossiers (par TP, par PP, etc.)
  - Un seul rattachement global, puis des «pull» suffisent dans chaque dossier (scriptable)
  - Les «pull» récupèrent en parallèles tous les projets en cours
  - Ajouts automatisés des nouveaux dossiers, modifications des instructions possible par après
- Inconvénients :
  - Nécessité de savoir scripter l'ajout automatique dans tous les dossiers

# Un dossier semestriel par élève

## Avantages et inconvénients

- Avantages :
  - Possibilité d'organiser les dossiers (par TP, par PP, etc.)
  - Un seul rattachement global, puis des «pull» suffisent dans chaque dossier (scriptable)
  - Les «pull» récupèrent en parallèles tous les projets en cours
  - Ajouts automatisés des nouveaux dossiers, modifications des instructions possible par après
- Inconvénients :
  - Nécessité de savoir scripter l'ajout automatique dans tous les dossiers

# Un dossier semestriel par élève

## Avantages et inconvénients

- Avantages :
  - Possibilité d'organiser les dossiers (par TP, par PP, etc.)
  - Un seul rattachement global, puis des «pull» suffisent dans chaque dossier (scriptable)
  - Les «pull» récupèrent en parallèles tous les projets en cours
  - Ajouts automatisés des nouveaux dossiers, modifications des instructions possible par après
- Inconvénients :
  - Nécessité de savoir scripter l'ajout automatique dans tous les dossiers

# Un dossier semestriel par élève

## Avantages et inconvénients

- Avantages :
  - Possibilité d'organiser les dossiers (par TP, par PP, etc.)
  - Un seul rattachement global, puis des «pull» suffisent dans chaque dossier (scriptable)
  - Les «pull» récupèrent en parallèles tous les projets en cours
  - Ajouts automatisés des nouveaux dossiers, modifications des instructions possible par après
- Inconvénients :
  - Nécessité de savoir scripter l'ajout automatique dans tous les dossiers

# Un dossier semestriel par élève

## Avantages et inconvénients

- Avantages :
  - Possibilité d'organiser les dossiers (par TP, par PP, etc.)
  - Un seul rattachement global, puis des «pull» suffisent dans chaque dossier (scriptable)
  - Les «pull» récupèrent en parallèles tous les projets en cours
  - Ajouts automatisés des nouveaux dossiers, modifications des instructions possible par après
- Inconvénients :
  - Nécessité de savoir scripter l'ajout automatique dans tous les dossiers

# Un dossier semestriel par élève

## Récupération globale

```
daenerys Info_sem2_2018-2019/Info_sem2_2018-2019-zeinila68>git pull
remote: Enumerating objects: 19, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (3/3), done.
Dépaquetage des objets: 100% (10/10), fait.
remote: Total 10 (delta 7), reused 10 (delta 7), pack-reused 0
Depuis github.com:PCSIKleber/Info_sem2_2018-2019-zeinila68
   f566ea1..9aabd3f  master    -> origin/master
Mise à jour f566ea1..9aabd3f
Fast-forward
 PP/PP16/PP16_disque_tournant_serie.py | 36 ++++++-----
 TP/TP10/TP10_diagramme_HR.py          |  1 +
 TP/TP11/TP11_euler_verlet.py          | 38 ++++++-----
 3 files changed, 71 insertions(+), 4 deletions(-)
```

## Exemples de script

«Pull» sur tous les repositories

```
1 base = "Info_sem2_2018-2019"
2 import glob,sys,os,subprocess
3 dossiers = glob.glob(base + '/' + base + '*')
4 racine = os.getcwd()
5 for d in sorted(dossiers):
6     print("On s'occupe de {}".format(d))
7     os.chdir(d)
8     subprocess.call(['git','checkout','master'])
9     subprocess.call(['git','pull'])
10    os.chdir(racine)
```

# Exemples de script

## Ajout sur tous les repositories, partie 1

```
1 import glob, subprocess, os
2 base = "Info_sem2_2018-2019"
3 origin = '/chemin/vers/PP17'
4 target = 'PP/'
5 commit_message = "Arrivée du PP17"
6 dossiers = glob.glob(base + '/' + base + '*')
7 racine = os.getcwd()
```

## Exemples de script

### Ajout sur tous les repositories, partie 2

```
1  for d in sorted(dossiers):
2      print("On s'occupe de {}".format(d))
3      os.chdir(d)
4      print('PULL')
5      subprocess.call(['git','pull'])
6      print('CP and COMMIT')
7      subprocess.call(['cp','-R',origin,target])
8      subprocess.call(['git','add','.'])
9      subprocess.call(['git','commit','-m',commit_message])
10     print('PUSH')
11     subprocess.call(['git','push'])
12     os.chdir(racine)
```

# Autres utilisations

## Les TIPE

Voir [https://github.com/PCSIKleber/TIPE\\_2020](https://github.com/PCSIKleber/TIPE_2020)

- Travail en groupe
- Dossier commun (mieux pour suivre les versions des simulations informatiques)
- Travail en simultané via l'interface web

# Autres utilisations

## Les TIPE

Voir [https://github.com/PCSIKleber/TIPE\\_2020](https://github.com/PCSIKleber/TIPE_2020)

- Travail en groupe
- Dossier commun (mieux pour suivre les versions des simulations informatiques)
- Travail en simultané via l'interface web

# Autres utilisations

## Les TIPE

Voir [https://github.com/PCSIKleber/TIPE\\_2020](https://github.com/PCSIKleber/TIPE_2020)

- Travail en groupe
- Dossier commun (mieux pour suivre les versions des simulations informatiques)
- Travail en simultané via l'interface web

# Autres utilisations

## Les TIPE

PCSIKleber / TIPE\_2020-fellan-dellon Private Unwatch 1 Star 0 Fork 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Insights](#) [Settings](#)

TIPE\_2020-fellan-dellon created by GitHub Classroom [Edit](#)

[Manage topics](#)

42 commits 1 branch 0 releases 3 contributors

Branch: master [New pull request](#) [Create new file](#) [Upload files](#) [Find File](#) [Clone or download](#)

SwannIn Update README.md Latest commit 233ecbb 18 days ago

DOT	Initial commit	3 months ago
Prise de Note	Update Parce que j'ai pas Word.txt	25 days ago
experiences	Update README.md	18 days ago
simulations	Update README.md	25 days ago
.gitignore	Initial commit	3 months ago
2019-Attendus_pedagogiques_Livrables.pdf	Initial commit	3 months ago
README.md	Update README.md	24 days ago

# Autres utilisations

## Résumés de cours en L<sup>A</sup>T<sub>E</sub>X

🔒 PCSIKleber / `resume_physique-pcs1` Private

👁 Unwatch 2   ★ Star 0   🍴 Fork 0

🔼 Code   ⓘ Issues 5   🔀 Pull requests 0   📁 Projects 0   📖 Wiki   📊 Insights   ⚙ Settings

Travail collaboratif pour produire des résumés du cours de physique, PCS1, 2018-2019 Edit

[Manage topics](#)

📄 441 commits   🌿 1 branch   📦 0 releases   👤 39 contributors

- Travail en groupe à l'échelle de la classe
- Travail en simultané via l'interface web (pour éviter les problèmes de fusion à gérer)
- Seul le prof a la main sur la génération du pdf via OverLeaf (beaucoup de boulot pour corriger les erreurs de syntaxe, mais le résultat est plutôt pas mal!)

# Autres utilisations

## Résumés de cours en L<sup>A</sup>T<sub>E</sub>X

The screenshot shows the GitHub interface for the repository 'PCSIKleber / resume\_physique-pcs1'. At the top, there are buttons for 'Unwatch' (2), 'Star' (0), and 'Fork' (0). Below this is a navigation bar with 'Code' selected, and links for 'Issues' (5), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Insights', and 'Settings'. The repository description is 'Travail collaboratif pour produire des résumés du cours de physique, PCS1, 2018-2019' with an 'Edit' button. Below the description is a 'Manage topics' link. At the bottom of the repository header, there are statistics: '441 commits', '1 branch', '0 releases', and '39 contributors'.

- Travail en groupe à l'échelle de la classe
- Travail en simultané via l'interface web (pour éviter les problèmes de fusion à gérer)
- Seul le prof a la main sur la génération du pdf via OverLeaf (beaucoup de boulot pour corriger les erreurs de syntaxe, mais le résultat est plutôt pas mal!)

## Autres utilisations

### Résumés de cours en $\text{\LaTeX}$

🔒 PCSIKleber / `resume_physique-pcs1` Private

👁 Unwatch 2   ★ Star 0   🍴 Fork 0

🏠 Code   ⓘ Issues 5   🔀 Pull requests 0   📁 Projects 0   📖 Wiki   📊 Insights   ⚙ Settings

Travail collaboratif pour produire des résumés du cours de physique, PCS1, 2018-2019 Edit

[Manage topics](#)

📄 441 commits   🌿 1 branch   📦 0 releases   👤 39 contributors

- Travail en groupe à l'échelle de la classe
- Travail en simultané via l'interface web (pour éviter les problèmes de fusion à gérer)
- Seul le prof a la main sur la génération du pdf via OverLeaf (beaucoup de boulot pour corriger les erreurs de syntaxe, mais le résultat est plutôt pas mal!)

# Autres utilisations

## Boîte à outils pour TP de physique

### Cahier de TP numérique

---

Dans ce dossier, vous allez pouvoir vous constituer un véritable petit couteau suisse des fonctions les plus utiles en TP de physique. N'hésitez pas à le compléter, ça vous ressortira, notamment en TIPE.

De la même manière que pour les TP d'info, vous pouvez demander conseil à vos professeurs ans vos différents commits.

### Organisation

---

Le fichier `biblio_TP_phys.py` a pour vocation de contenir toutes les fonctions que vous serez amenés à programmer. Vous pouvez y avoir accès en incluant l'instruction

```
from biblio_TP_phys import *
```

au début de chaque fichier de TP.

## Autres utilisations

### Boîte à outils pour TP de physique

- Travail en groupe (mais problème des binômes non stables)
- Permet de mettre en commun les mesures et scripts de traitement
- L'an prochain, dossier unique pour toute la classe avec des sous-dossiers par élève pour que tout le monde puisse accéder aux mesures de tout le monde (échange plus facile si les binômes changent)

## Autres utilisations

### Boîte à outils pour TP de physique

- Travail en groupe (mais problème des binômes non stables)
- Permet de mettre en commun les mesures et scripts de traitement
- L'an prochain, dossier unique pour toute la classe avec des sous-dossiers par élève pour que tout le monde puisse accéder aux mesures de tout le monde (échange plus facile si les binômes changent)

## Autres utilisations

### Boîte à outils pour TP de physique

- Travail en groupe (mais problème des binômes non stables)
- Permet de mettre en commun les mesures et scripts de traitement
- L'an prochain, dossier unique pour toute la classe avec des sous-dossiers par élève pour que tout le monde puisse accéder aux mesures de tout le monde (échange plus facile si les binômes changent)

## Conclusions

# Conclusions

- Une expérience très positive
- Accueil favorable des élèves : pas plus compliqué de leur point de vue que prendre un IDE en main (mais pour les collègues, c'est une autre histoire)
- Permet un suivi bien meilleur des élèves
- Structure choisie pour l'année prochaine :
  - Un unique repository pour l'année avec rajouts réguliers de nouveaux répertoires
  - Cahier de TP de physique introduit tout de suite et partagé sur la classe

## Conclusions

- Une expérience très positive
- Accueil favorable des élèves : pas plus compliqué de leur point de vue que prendre un IDE en main (mais pour les collègues, c'est une autre histoire)
- Permet un suivi bien meilleur des élèves
- Structure choisie pour l'année prochaine :
  - Un unique repository pour l'année avec rajouts réguliers de nouveaux répertoires
  - Cahier de TP de physique introduit tout de suite et partagé sur la classe

## Conclusions

- Une expérience très positive
- Accueil favorable des élèves : pas plus compliqué de leur point de vue que prendre un IDE en main (mais pour les collègues, c'est une autre histoire)
- Permet un suivi bien meilleur des élèves
- Structure choisie pour l'année prochaine :
  - Un unique repository pour l'année avec rajouts réguliers de nouveaux répertoires
  - Cahier de TP de physique introduit tout de suite et partagé sur la classe

# Conclusions

- Une expérience très positive
- Accueil favorable des élèves : pas plus compliqué de leur point de vue que prendre un IDE en main (mais pour les collègues, c'est une autre histoire)
- Permet un suivi bien meilleur des élèves
- Structure choisie pour l'année prochaine :
  - Un unique repository pour l'année avec rajouts réguliers de nouveaux répertoires
  - Cahier de TP de physique introduit tout de suite et partagé sur la classe

# Conclusions

- Une expérience très positive
- Accueil favorable des élèves : pas plus compliqué de leur point de vue que prendre un IDE en main (mais pour les collègues, c'est une autre histoire)
- Permet un suivi bien meilleur des élèves
- Structure choisie pour l'année prochaine :
  - Un unique repository pour l'année avec rajouts réguliers de nouveaux répertoires
  - Cahier de TP de physique introduit tout de suite et partagé sur la classe

# Conclusions

- Une expérience très positive
- Accueil favorable des élèves : pas plus compliqué de leur point de vue que prendre un IDE en main (mais pour les collègues, c'est une autre histoire)
- Permet un suivi bien meilleur des élèves
- Structure choisie pour l'année prochaine :
  - Un unique repository pour l'année avec rajouts réguliers de nouveaux répertoires
  - Cahier de TP de physique introduit tout de suite et partagé sur la classe



## Mise en place de GitHub Classroom

## Ensemble du processus

Il est décrit dans cette petite vidéo

[https://www.youtube.com/watch?v=ChA\\_zph7aao](https://www.youtube.com/watch?v=ChA_zph7aao)

# Création de l'organisation

The screenshot shows the GitHub user settings page. On the left is a sidebar with navigation links: Personal settings, Profile, Account, Emails, Notifications, Billing, SSH and GPG keys, Security, Sessions, Blocked users, Repositories, Organizations (highlighted), and Saved replies. The main content area is titled 'Organizations' and includes a 'New organization' button. Below this, there is a list of organizations the user is a member of: 'Campus-Advisors' (member and collaborator on 3 repositories) and 'PCSIKleber' (member and collaborator on 6 repositories), each with a 'Leave' button. Underneath is a section titled 'Transform account' with a message: 'You cannot transform this account into an organization until you leave all organizations that you're a member of.' and a button that says 'Turn jfPCSI1 into an organization'.

- Depuis `github.com`, aller dans «Settings» (en haut à droite sur l'avatar de votre compte)
- Puis «Organizations» (onglet en bas à gauche)
- Puis «New organisation» (bouton)

# Création de l'organisation

The screenshot shows the GitHub user interface. At the top, there are navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. On the left, a sidebar under 'Personal settings' includes links for Profile, Account, Emails, Notifications, Billing, SSH and GPG keys, Security, Sessions, Blocked users, Repositories, Organizations (highlighted), and Saved replies. The main area is titled 'Organizations' and features a 'New organization' button. It lists two organizations: 'Campus-Advisors' (member and collaborator on 3 repositories) and 'PCSIKleber' (member and collaborator on 6 repositories), each with a 'Leave' button. Below the list is a 'Transform account' section with a warning message: 'You cannot transform this account into an organization until you leave all organizations that you're a member of.' and a button labeled 'Turn jfPCS11 into an organization'.

- Depuis `github.com`, aller dans «Settings» (en haut à droite sur l'avatar de votre compte)
- Puis «Organizations» (onglet en bas à gauche)
- Puis «New organisation» (bouton)

# Création de l'organisation

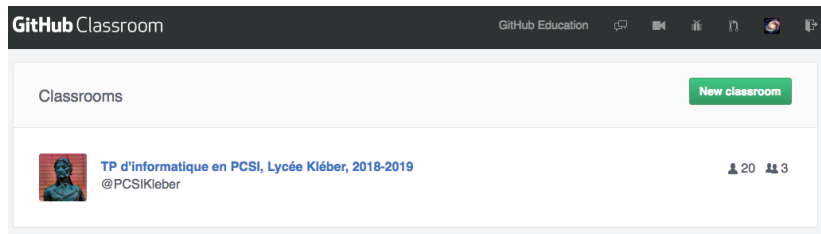
The screenshot shows the GitHub user settings interface. On the left is a sidebar menu with options: Personal settings, Profile, Account, Emails, Notifications, Billing, SSH and GPG keys, Security, Sessions, Blocked users, Repositories, Organizations (highlighted), and Saved replies. The main content area is titled 'Organizations' and includes a 'New organization' button. Below this, there is a list of organizations the user is a member of: 'Campus-Advisors' (member and collaborator on 3 repositories) and 'PCSIKleber' (member and collaborator on 6 repositories), each with a 'Leave' button. A section titled 'Transform account' contains a message: 'You cannot transform this account into an organization until you leave all organizations that you're a member of.' and a button labeled 'Turn jfPCSI1 into an organization'.

- Depuis `github.com`, aller dans «Settings» (en haut à droite sur l'avatar de votre compte)
- Puis «Organizations» (onglet en bas à gauche)
- Puis «New organisation» (bouton)

# Création de la « classroom »

## Première étape

Depuis <https://classroom.github.com/classrooms> (après s'être connecté, bien entendu), cliquer sur le bouton «New classroom» (en haut à droite)

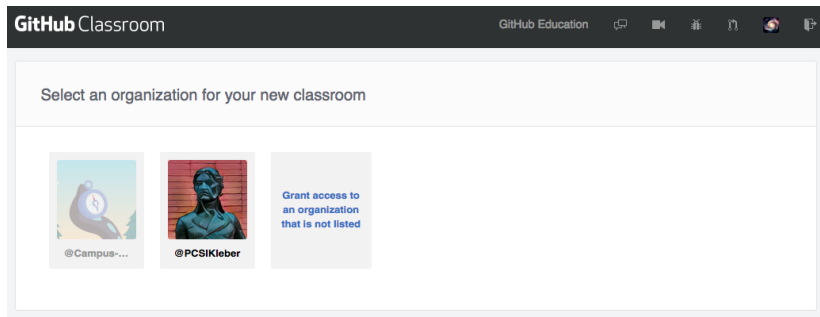


The screenshot shows the GitHub Classroom web interface. At the top, there is a dark header with the text "GitHub Classroom" on the left and "GitHub Education" followed by several icons on the right. Below the header, the main content area is titled "Classrooms" on the left and features a prominent green button labeled "New classroom" on the right. Underneath, a list of classrooms is displayed. The first entry is for a classroom titled "TP d'informatique en PCSI, Lycée Kléber, 2018-2019" with the handle "@PCSIKleber". To the left of the title is a small profile picture of a bust. To the right of the title, it shows "20" people and "3" avatars.

# Création de la « classroom »

## Sélection de l'organisation

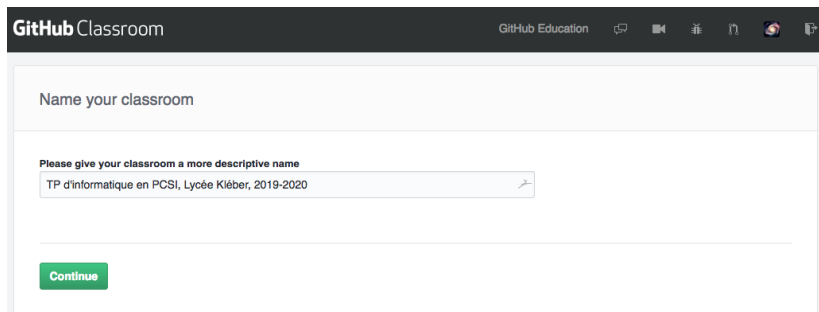
Sélectionner l'«organization» adéquate



# Création de la « classroom »

## Nomination

Donner un nom explicite à la classroom



The screenshot shows the GitHub Classroom creation interface. At the top, the header reads "GitHub Classroom" on the left and "GitHub Education" on the right, followed by several utility icons. The main content area is titled "Name your classroom". Below this title, there is a text input field containing the text "TP d'informatique en PCSI, Lycée Kléber, 2019-2020". Above the input field, a prompt reads "Please give your classroom a more descriptive name". At the bottom left of the form, there is a green button labeled "Continue".

# Création de la « classroom »

## Invitation d'autres administrateurs

The screenshot shows the GitHub Classroom interface. At the top, there's a header with 'GitHub Classroom' on the left and 'GitHub Education' and various utility icons on the right. Below the header is a green banner with the course title 'TP d'informatique en PCSI, Lycée Kléber, 2019-2020' and the instructor's name 'PCSIKleber'. A 'Manage classroom' button is visible on the right side of the banner.

Below the banner, the main content area is titled 'Invite other administrators'. It contains the following instructions:

- 1. Add them to your GitHub Organization**  
You must add them to the Organization as an Owner in order for them to have access.  
You can do so at <https://github.com/orgs/PCSIKleber/people>
- 2. Give them this URL**  
After signing in, they will automatically be added to this classroom.

At the bottom of the instructions, there is a text input field containing the URL: `https://classroom.github.com/classrooms/PCSIKleber-tp-d-informatique`. To the right of the input field is a 'Copy classroom link' button. A circular profile picture of the instructor is also visible next to the URL.

# Création de la « classroom »

## Préparation du roster

### Add students

Next, let's set up a "roster", so you can easily track student progress on your dashboard.

To create your roster, you'll need to share a way to identify your students. These "identifiers" can be:

- Student last names
- Student email addresses
- Any other unique piece of information

You can always come back and complete this step later.

What information would you like to use to identify your students?  
(emails, student ID #, etc.)

Paste your list of identifiers below (one per line, please!):

OR upload a CSV or text file:

Upload file



# Premier «Assignment»

## Création

GitHub Classroom

GitHub Education

## TP d'informatique en PCSI, Lycée Kléber, 2019-2020

PCSIKleber [Manage classroom](#)

### Assignments

This organization doesn't have any assignments yet.

[Create your first assignment](#)

# Premier «Assignment»

## Type

The screenshot shows the GitHub Classroom interface for a course titled "TP d'informatique en PCSI, Lycée Kléber, 2019-2020". The interface is in French and offers two options for creating a new assignment: "Individual assignment" and "Group assignment".

**GitHub Classroom** GitHub Education

### TP d'informatique en PCSI, Lycée Kléber, 2019-2020

PCSIKleber [Manage classroom](#)

#### New assignment

**Individual assignment**

Each participant works individually on their own repository.

[Create an individual assignment](#)

**Group assignment**

Participants form teams that work together on a shared repository.

[Create a group assignment](#)

# Premier «Assignment»

## Réglages



### Your assignment title

TP01, introduction à l'environnement de travail

### Your assignment repository prefix


TP01


This will prefix each GitHub repository that is created for this assignment. May only contain alphanumeric characters, underscores or hyphens.

-  **Public**  
Submit assignments using public repositories. All submissions will be visible to the world.
-  **Private**  
Submit assignments using private repositories. Submissions will only be visible to the submitter and organization owners. Editing this after assignments are created will not retroactively change their visibility.
- Give students Admin permissions on their repository. Editing this after assignments are created will not retroactively change their permissions.**
- Enable assignment invitation URL**

### Add your starter code from GitHub (optional)

code\_concours\_CPGE

 `jjfPCSI1/codes_concours_CPGE` Propositions d'implémentations pour les sujets proposés aux différentes épreuves d'informatique des concours en CPGE

 `lcarada/codes_concours_CPGE` Propositions d'implémentations pour les sujets proposés aux

### Unlimited free private repositories

If you have not already, you may apply for unlimited free private repositories for this classroom organization [here](#).

# Premier «Assignment»

## Récupération des dossiers

GitHub Classroom

GitHub Education

## TP d'informatique en PCSI, Lycée Kléber, 2019-2020

PCSIKleber

Manage classroom

"TP01, premier contact" has been created

TP01, premier contact  
Individual assignment

Download Repositories ▾

Assignment settings

Give this to your students

<https://classroom.github.com/a/NyAvMYIM>

Copy invitation link

Download Student Repositories

Open Assignment in Classroom Assistant to clone student submissions.

Open in Classroom Assistant

"TP01, premier contact" does not have any repositories.

Share the invitation link with your students to get started.