
Travaux pratiques Git

Une grande partie de ce TP consiste à lire et comprendre des documents sur moodle. Le but n'est pas de le faire le plus rapidement possible, mais de comprendre au mieux une architecture beaucoup plus complexe que ce que vous avez vu jusque là.

Présentation

Au cours de ce TP, on va mettre en place les comptes sur différents sites Web externes qui nous permettront de gérer le code C qui sera écrit.

I Exercice : Création d'une clef RSA

On va commencer par travailler uniquement avec le terminal pour s'entraîner.

(a) On va commencer par créer une clef `rsa`. Cette clef sert à nous identifier de manière sûre auprès d'autres sites. Pour cela, utilisez la commande :

Code bash 1 – Terminal

```
1 ssh-keygen -t rsa
```

et ne mettez pas de passphrase (elle est utile si on veut plus de sécurité). Affichez le contenu du répertoire `.ssh`.

Code bash 2 – Terminal

```
1 ls .ssh
```

II Exercice : Création d'un compte sur GitHub

On continue sur le terminal, mais on ouvre en plus un navigateur Web. En appuyant sur la touche Windows, vous pouvez mettre le navigateur en plein écran sur un autre Bureau en le déplaçant.

(a) Ouvrez un nouvel onglet sur le navigateur. Pour créer un compte sur <https://github.com/>, il faut utiliser le lien **sign up** (Fig. 1). Il vous sera demandé une adresse mail à laquelle sera envoyée un e-mail de confirmation d'inscription. Choisissez le compte gratuit (Fig. 2).

(b) Il vous est ensuite demandé de créer un répertoire (un dépôt). Choisissez un nom en lien avec le cours ! **Il faut choisir un répertoire privé lors de la création** (Fig. 3). Initialisez le répertoire avec un fichier `README` et un fichier `.gitignore` pour le langage C.

III Exercice : Liaison répertoire local—GitHub

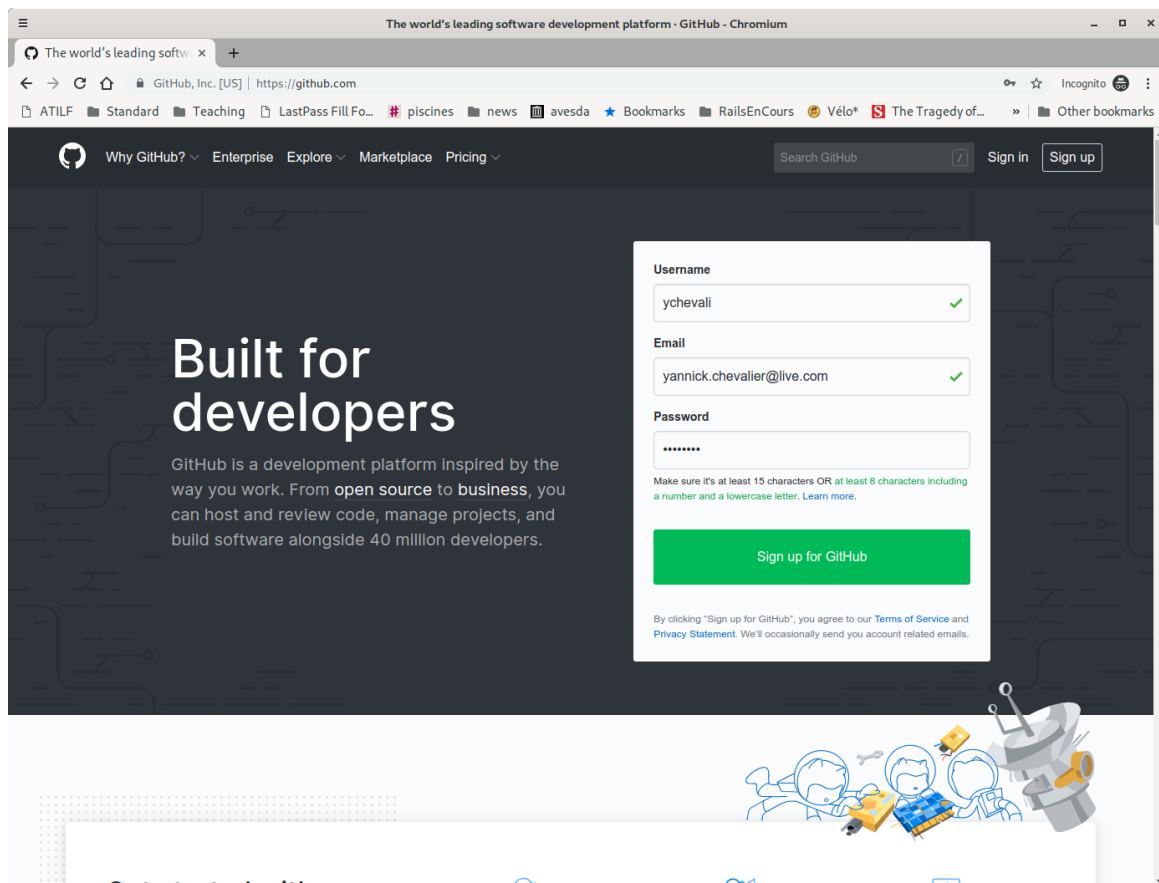


FIGURE 1 – Création du compte sur GitHub

On va relier notre espace de travail à l'Université en utilisant la clef publique RSA qu'on a créée au premier exercice. Pour cela, sur GitHub, allez sur votre espace personnel (Fig. 4) et choisissez dans le menu à gauche **SSH and GPG keys**, puis **SSH keys** (Fig. 5). Dans le terminal, affichez et copiez le contenu du fichier `~/.ssh/id_rsa.pub` (Fig. 6, Shift-Ctrl-C avec Firefox), et collez-le (Shift-Ctrl-V) dans la fenêtre sur la site `github.com` (Fig. 7).

Cette manipulation, avec des clefs publiques RSA différentes, sera à refaire chez vous pour pouvoir accéder facilement à vos dépôts.

IV Exercice : *Organisation du compte Linux*

On revient sur le terminal.

(a) Dans votre répertoire home (`~`), créez un répertoire **ProgrammationS3**.

Code bash 3 – Terminal

```
1 cd
2 mkdir ProgrammationS3
```

Ce répertoire contiendra 2 sous-répertoires hébergés sur github :

- le répertoire `~/ProgrammationS3/énoncés`, qui contiendra les sujets de TP et du projet ;
- le répertoire `~/ProgrammationS3/TP`, qui contiendra votre travail en TP.

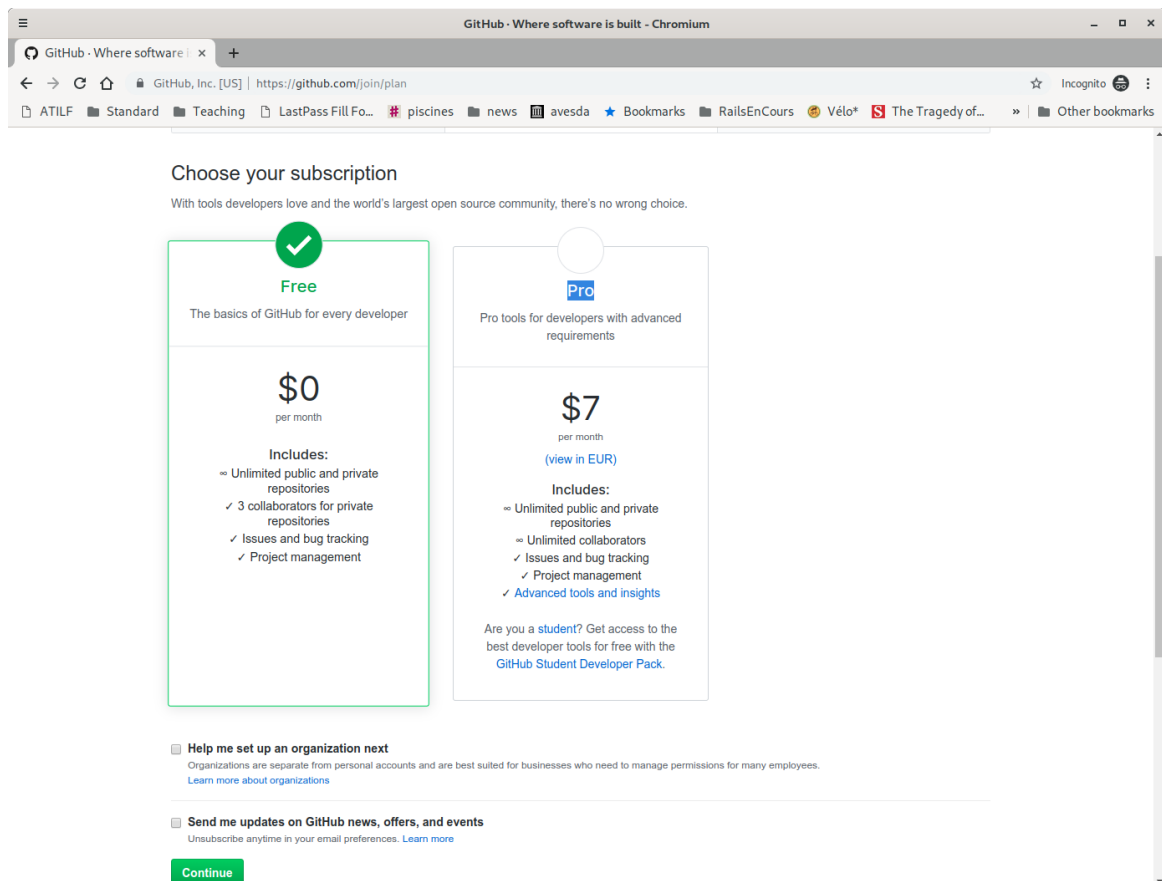


FIGURE 2 – Choix du type de compte sur GitHub

Plus tard, il y aura aussi un répertoire `~/ProgrammationS3/Projet`, qui contiendra un projet à faire en groupe. Le dépôt sera partagé par les membres du groupe, et hébergé sur un autre site (pour les comptes gratuits, **bitbucket.com** autorise jusqu'à 5 personnes sur un projet privé au lieu de 3 pour **github.com**), et le projet ne sera pas "rendu" mais téléchargé par votre enseignant de TP.

V Exercice : Récupération des énoncés

On va utiliser la structure usuelle pour l'écriture de programmes.

(a) Commencez par récupérer le répertoire que vous avez créé sur GitHub. Pour cela, récupérez l'adresse SSH du répertoire sur GitHub (Fig. 8), et tapez dans le terminal :

Code bash 4 – Terminal

```
1 cd ~/ProgrammationS3/TP
2 git clone ``adresse copiée``
```

(b) Après avoir cloné votre projet sur GitHub, créez les sous-répertoires suivants de `~/ProgrammationS3/TP` :

- `src` : qui contiendra les fichiers sources des programmes ;
- `include` : qui contiendra des déclarations que vous pourrez utiliser dans vos programmes ;
- `lib` : qui contiendra des fonctions que vous aurez écrites et compilées ;
- `bin` : qui contiendra vos programmes principaux.

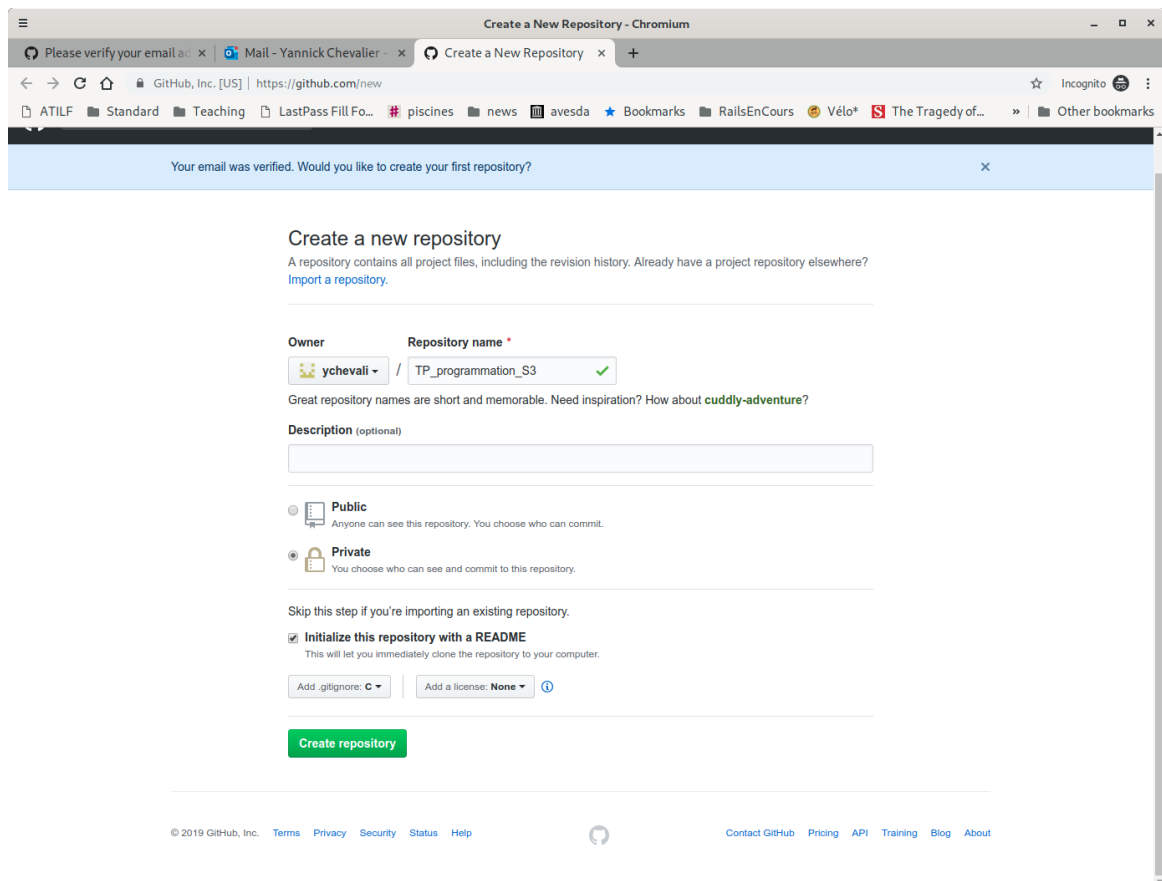


FIGURE 3 – Création d'un premier répertoire sur github

Ensuite, pour chaque Thème (le premier qu'on abordera est *EntreesSorties*), vous créerez un sous-répertoire de `src`.

Il est important de bien utiliser cette structure dès le début. Les énoncés de TPs ne sont pas linéaires (on passe d'un thème à l'autre sans avoir terminé), avec des réutilisations de résultats d'exercices d'un thème dans un autre thème.

(c) Dans chaque répertoire que vous aurez créé, créez un fichier `.keep` avec la commande :

Code bash 5 – Terminal

```
1 touch .keep
```

Ce fichier est nécessaire car `git` refuse d'ajouter des répertoires vides.

(d) Éditez le fichier `README.md` (par exemple avec **kate** ou **emacs**), sauvegardez, et committez, et transférez sur GitHub. Vérifiez que les transferts marchent correctement.

Code bash 6 – Terminal

```
1 echo "..."  
2 # découverte des opérations depuis la dernière sauvegarde  
3 git status  
4 # ajout des fichiers modifiés dans la liste de fichiers à transférer  
5 git add .  
6 # vérification de la liste de fichiers à modifier  
7 git status  
8 git commit -m "Modification_du_fichier_README.md"
```

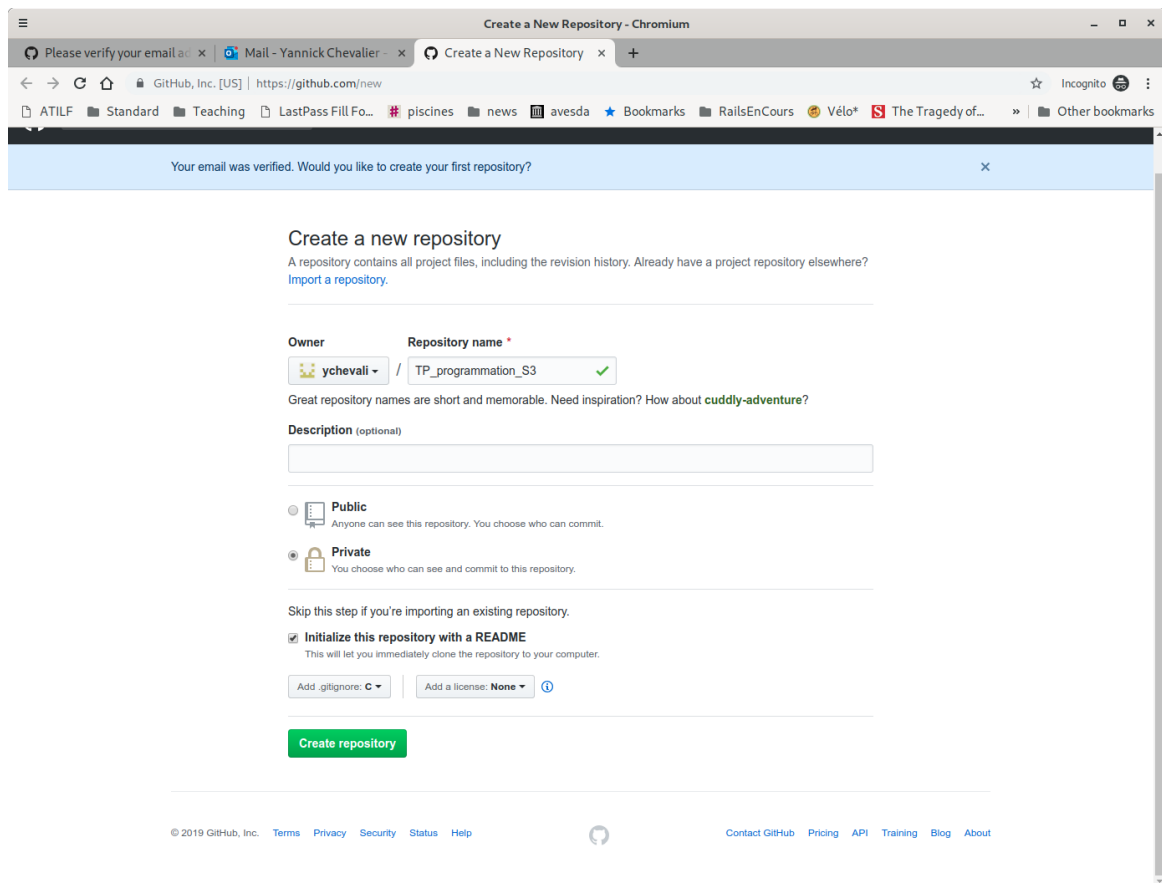


FIGURE 4 – Aller sur les informations personnelles sur GitHub

9 `git push`

Vous pouvez maintenant continuer avec le premier exercice du thème “Entrées/Sorties”, puis ceux du thème “Types”.

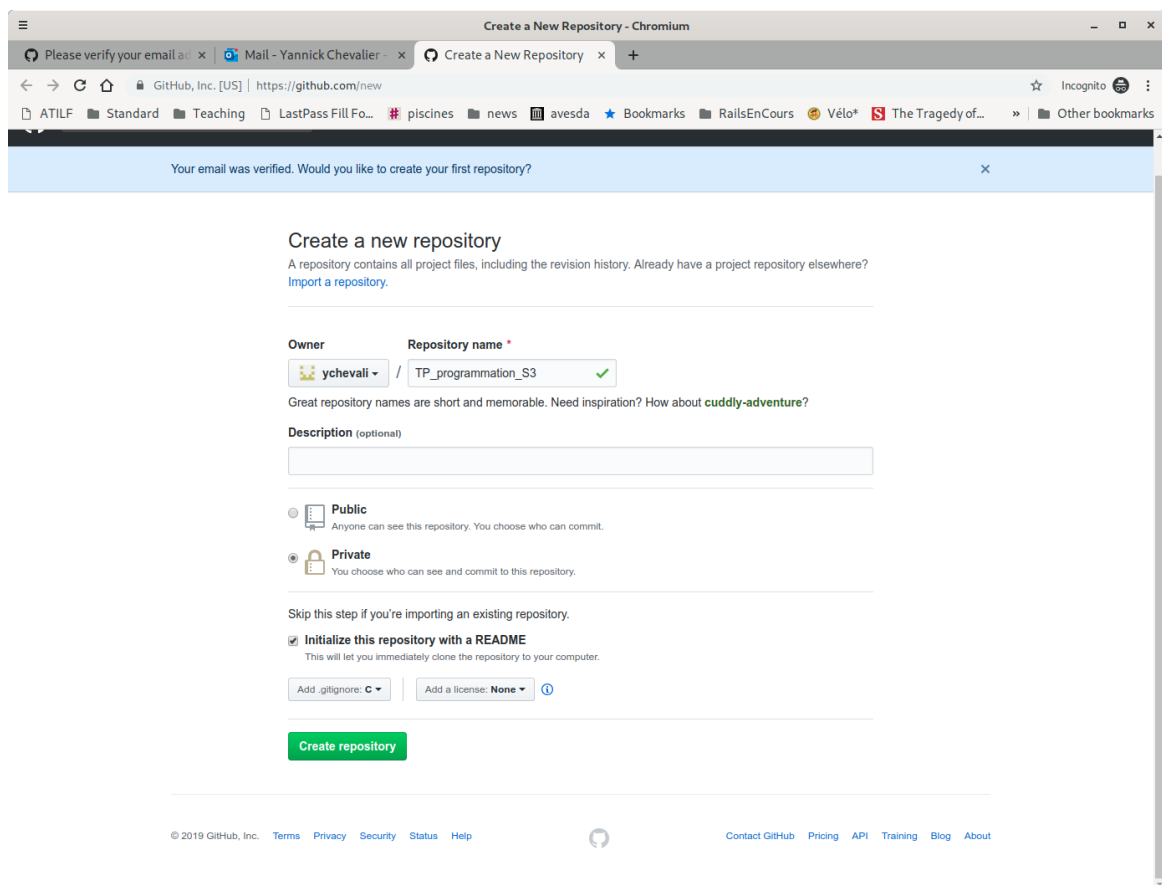


FIGURE 5 – Menu de création d'une clef SSH sur GitHub

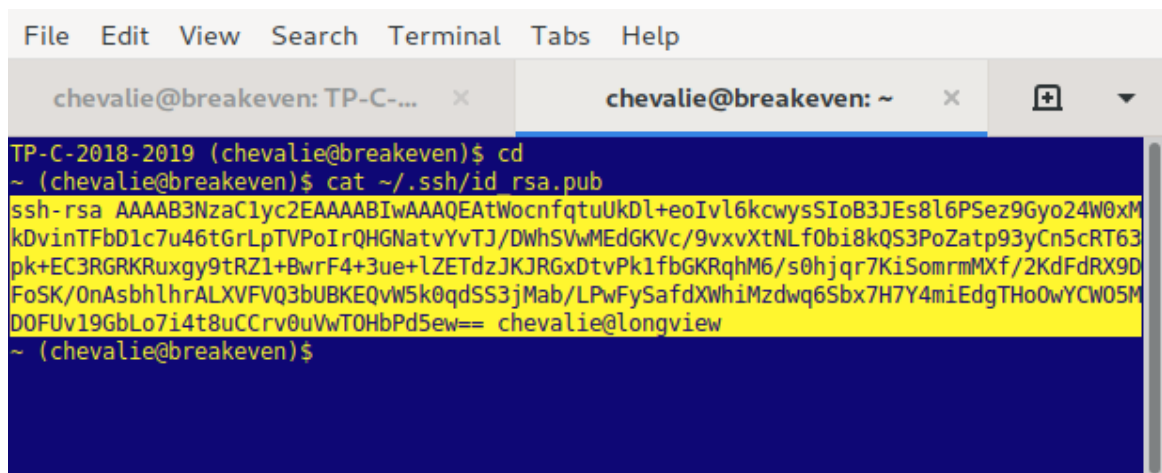


FIGURE 6 – Afficher et copier la clef RSA sur votre terminal

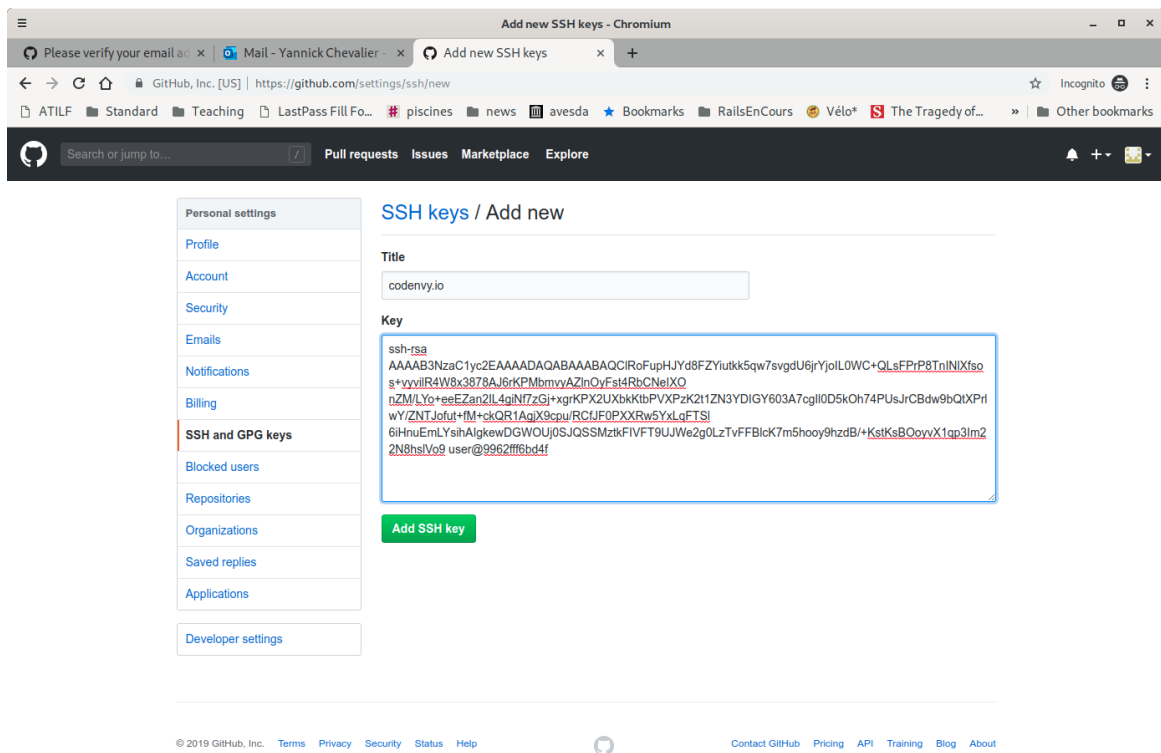


FIGURE 7 – Mettre la clef RSA sur GitHub

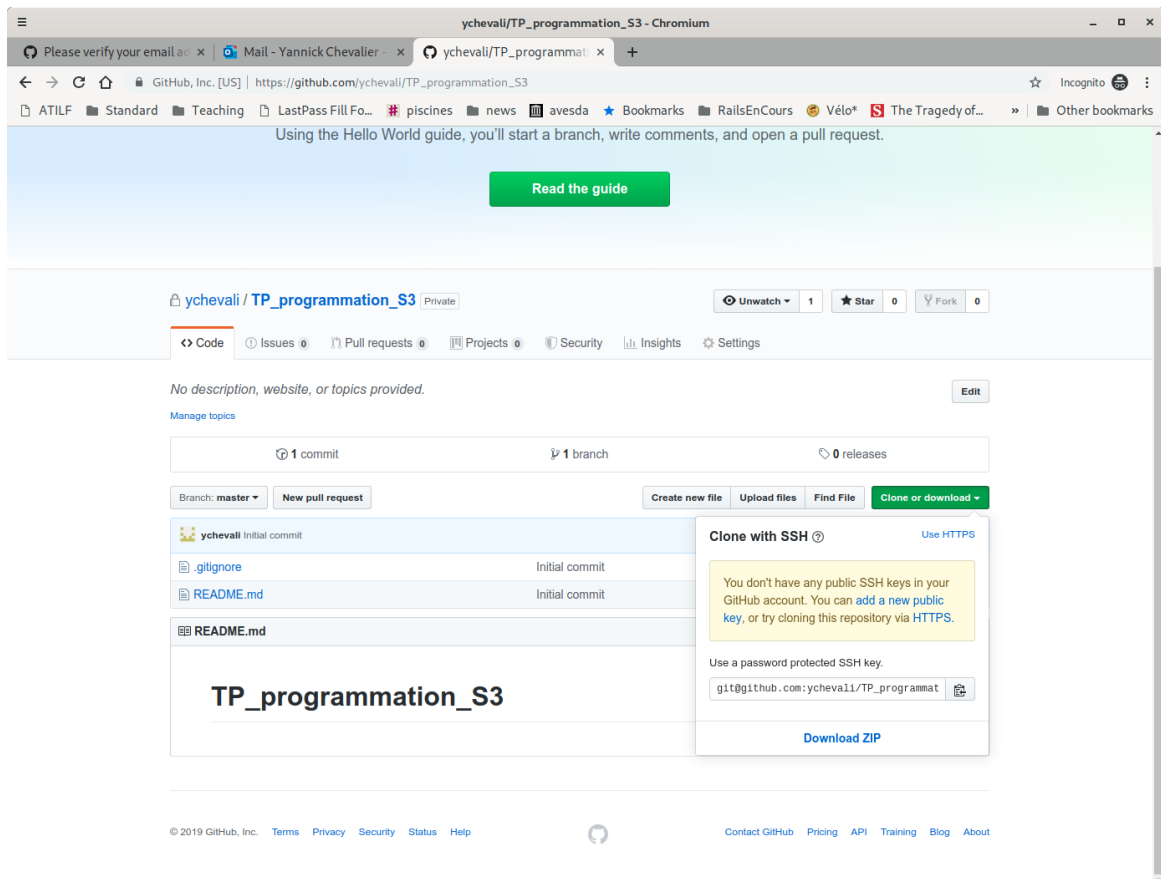


FIGURE 8 – Récupérez l'adresse pour cloner le dépôt avec ssh