GITHUB

* Qu’on on travaille sur un projet en local, il faut toujours avoir une « remote », c’est-à-dire une machine qui peut garder un backup de notre code source, pour éviter de rester en panne si par exemple notre machine tombe en panne.
* Une autre machine peut être :
  + Interne (si on a plusieurs ordinateurs)
  + Externe (grâce à des services comme github)
* Github est un service en ligne qui permet d’héberger des repositories (dossier) de code open source

Avec github, Vous allez pouvoir :

* Communiquer avec d'autres développeurs et signaler des problèmes de code en déclarant des "issues" ;
* Partager des morceaux de code en ligne à l'aide de "gists" ;
* Proposer des modifications de code à d'autres repos en faisant des "pull requests" ;
* Et même récupérer du code depuis un autre repository.

Une dernière petite astuce pour mieux vous y retrouver dans votre code en ligne : appuyez sur la touche t, vous pourrez alors faire une recherche dans vos noms de fichiers en tapant un mot / des lettres clé !

Copier un repository distant

* Pour cloner (copier) un repository distant dans notre machine local, on utilise la commande « git clone [lienFourniParGitHub] ».

Exemple : git clone <https://github.com/facebook/react.git>

Créer un repository sur mon compte github

* Lors de la création de notre repository github, il ya des paramètres à définir comme le nom, la description, le statut (public ou privée) et l’option Initialise with readme.

Vous avez également une option "Initialise with a README", qui vous permet de cloner votre repository sur votre machine. Cette option est à cocher uniquement dans le cas où vous n'avez **pas encore créé le repository en question sur votre machine** (ce qui est bien notre cas ici, cochez donc !)

... et en parlant de la commande **git clone**, notez que vous pouvez cloner votre repo avec deux options :

1. **L'option HTTPS** que je vous ai  invité à utiliser dans ce cours : c'est l'option la plus simple, qui demande de fournir vos identifiants GitHub (nom d'utilisateur et mot de passe) à chaque fois que vous faites un git clone.
2. **L'option SSH** qui est plus pratique car elle ne vous demande pas vos identifiants à chaque fois. Par contre, pour l'utiliser, il faut générer une clé SSH, ce qui est un peu plus compliqué pour ce cours d'initiation. Mais une fois que vous vous sentez plus à l'aise, je vous invite à consulter la [marche à suivre](https://help.github.com/articles/generating-ssh-keys) dans la documentation de GitHub et à utiliser cette option SSH.

Envoyer votre code sur github

Généralement, ça dépend des cas :

* + Cas 1 : je veux mettre mon code source sur le repository distant (github) sans besoin de cloner le repository distant dans ma machine local avant

On lance les commandes suivants :

* git remote add origin https://github.com/ShadowBinaryCode/JavaScript.git
* git push -u origin master
  + Case 2 : j’ai cloné le repository github dans ma machine local

Une fois qu’on fait le clone, on peut envoyer tout les modifications sur le repository local en lançant la commande « git push origin master », c’est-à-dire envoie mes modifications dans la branche master de mon remote origin

La branche master contient le code courant de notre repository github

Le remote sur lequel on envoie notre code est « origin », dans notre cas, ce remote est github

* Parfois on trouve la commande « git push –u origin master », The -u option do the following: For every branch that is up to date or successfully pushed, add upstream (tracking) reference, used by argument-less git-pull and other commands.

So, after pushing your local branch with -u option, this local branch will be automatically linked with remote branch, and you can use git pull without arguments.

Remarque : chaque repository doit contenir les README, LICENCE et « .gitignore »

* Quand on ecrit un repository sur github, github nous donne 3 choix :

### …or create a new repository on the command line

echo "# JavaScript" >> README.md

git init

git add README.md

git commit -m "first commit"

git remote add origin https://github.com/ShadowBinaryCode/JavaScript.git

git push -u origin master

### …or push an existing repository from the command line

git remote add origin https://github.com/ShadowBinaryCode/JavaScript.git

git push -u origin master

### …or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[**Import code**](https://github.com/ShadowBinaryCode/JavaScript/import)

Récuperer les modifications

Pour récuperer en local les modifications du repository remote, on utilise la commande :

« git pull origin master »