

GIT

GIT

DURE TOTALE DU MODULE : 21H00

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

GIT

Table des matières

Objectifs pédagogiques :	3
Présentation et installation de GIT.....	4
Présentation de GIT :	4
Installation de GIT :	4
Création d'un compte github et du premier repository (dépôt distant) :	7
Création d'un repository sur github :	8
Cloner un dépôt github (dépôt distant) sur votre repository (local)	9
Le système de sauvegarde (commit) :	10
Présentation du système de sauvegarde de git :	10
Gestion des fichiers et commits (sauvegarde) :	11
Les Remises (Stash) :	14
Interaction avec un repository distant (github) :	16
Téléchargement des données distantes vers votre dépôt local (dossier)	16
Envoi des données locales vers votre dépôt distant (repository github).....	17
Le système de branches :	18
Présentation du système de branches :	18
Gestion des branches :	18
Correction de problèmes Git :	20
Les 3 types de réinitialisation de GIT :	23
Scénarios d'erreurs :	25
Fusion (merge):	32

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

GIT

Objectifs pédagogiques :

Etre capable de comprendre les enjeux du système de branches (versionning, collaboration)

Etre capable de comprendre et utiliser le workflow Git Flow

Etre capable d'initialiser et de restaurer un dépôt local ou distant

Etre capable de réaliser des commandes commit, push et pull sur des dépôts distants (github)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

GIT

Présentation et installation de GIT

Présentation de GIT :



Qu'est-ce que GIT ?

Git est un outil de gestion de projets et de versions collaboratif.

Git est un [logiciel de gestion de versions décentralisé](#). C'est un [logiciel libre](#) créé par [Linus Torvalds](#), auteur du [noyau Linux](#), et distribué selon les termes de la [licence publique générale GNU](#) version 2. En 2016, il s'agit du [logiciel de gestion de versions](#) le plus populaire qui est utilisé par plus de douze millions de personnes.

Installation de GIT :

Télécharger l'outil à cette adresse :

<https://git-scm.com/downloads>

1 Lancer Git Bash :

2 Enregistrer le login :

```
$ git config --global user.name "nom_utilisateur"
```

3 Enregistrer son adresse mail :

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

GIT

```
$ git config --global user.email mail_utilisateur
```

4 Pour vérifier sa configuration :

```
$ git config --list
```

5 Activation de la coloration (optionnel):

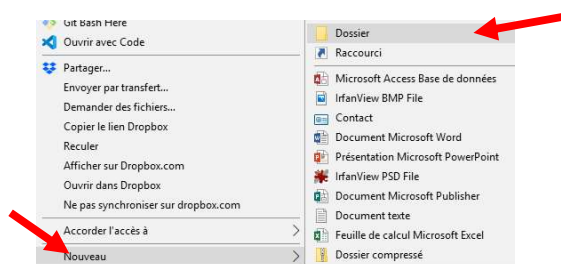
```
$ git config --global color.diff auto
$ git config --global color.status auto
$ git config --global color.branch auto
```

6 Configuration de son éditeur de texte par défaut et Vimdiff comme outil de merge (optionnel):

```
$ git config --global core.editor notepad++
$ git config --global core.editor code
$ git config --global merge.tool vimdiff
```

7 Création d'un répertoire sur votre ordinateur :

Clic droit nouveau dossier.



Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

GIT

8 Accéder à votre répertoire (dépôt local) :

Taper cd puis l'url de votre répertoire ou clic droit sur votre dossier (ou bien git bash here dans le menu) ex :

```
$ cd C:/Users/nom_utilisateur/Desktop/git
```

9 Initialisation du dépôt (local) :

```
$ git init
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

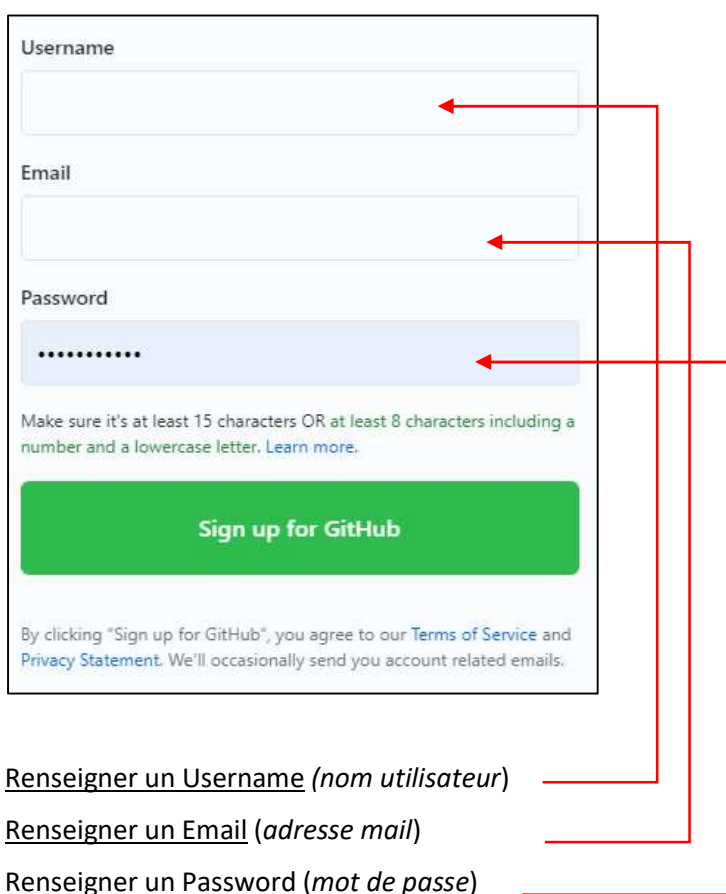
GIT

Création d'un compte github et du premier repository (dépôt distant) :

1 Se connecter sur github.com :

<https://github.com>

2 Cliquer sur Sign up en haut à droite :



Username

Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.

Renseigner un Username (nom utilisateur)

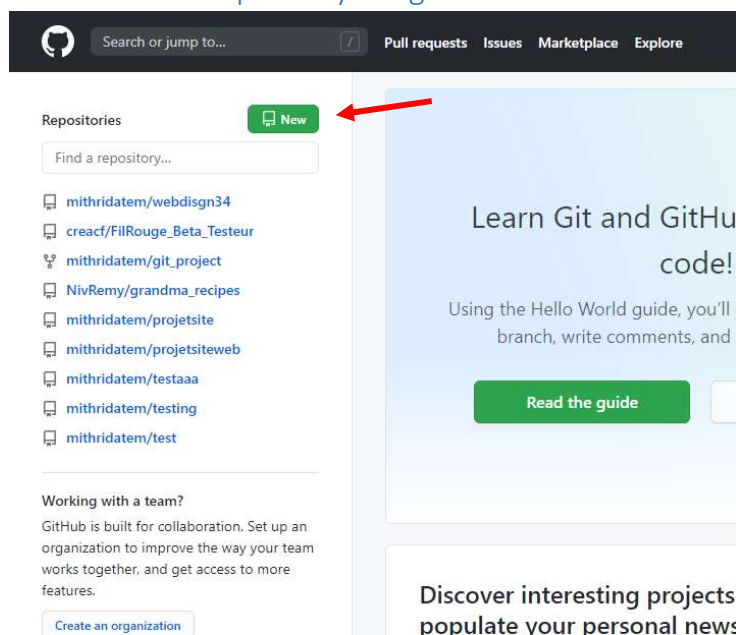
Renseigner un Email (adresse mail)

Renseigner un Password (mot de passe)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

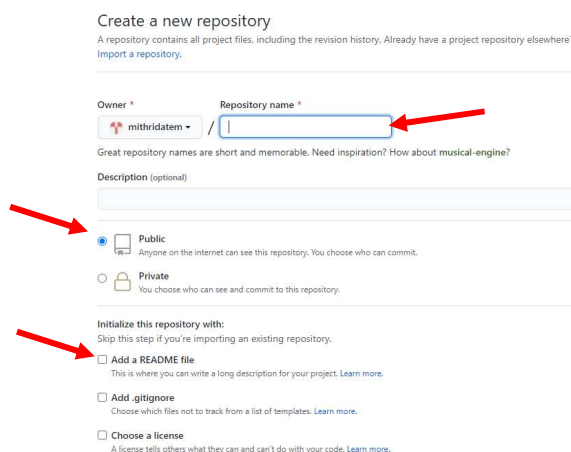
GIT

Création d'un repository sur github :



1 Cliquer sur new.

2 Choisir un nom pour votre repository (nom_repository, cocher public et cocher add readme file):



3 Cliquer sur create repository bouton en bas de la page.

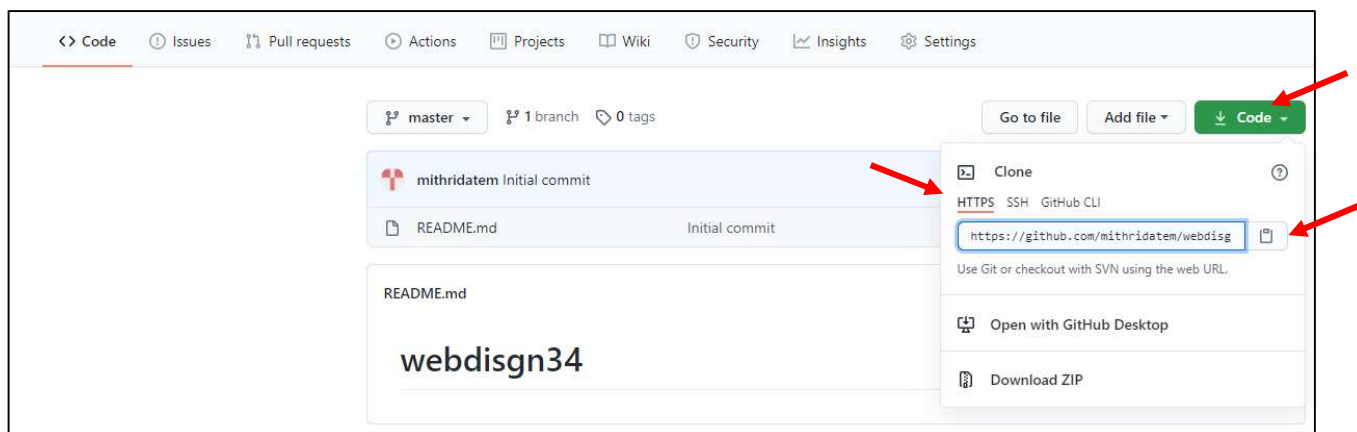
Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

GIT

Cloner un dépôt github (dépôt distant) sur votre repository (local)

1 Cliquer sur code (interface de github) :

2 Récupérer l'url de votre dépôt (https) :



3 Taper la ligne de commande suivante dans la console git :

```
$ git remote add clone https://github.com/votre_url.git
```

4 méthode alternative :

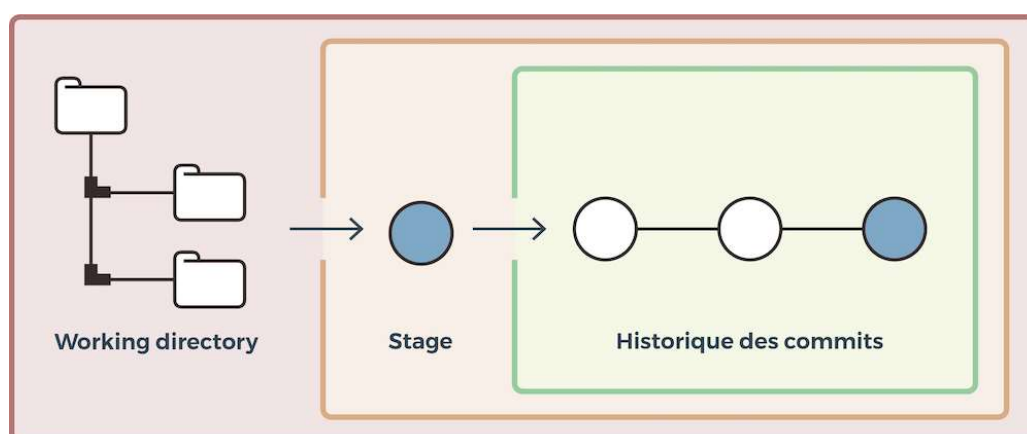
```
$ git remote add origin https://github.com/votre_url.git
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

GIT

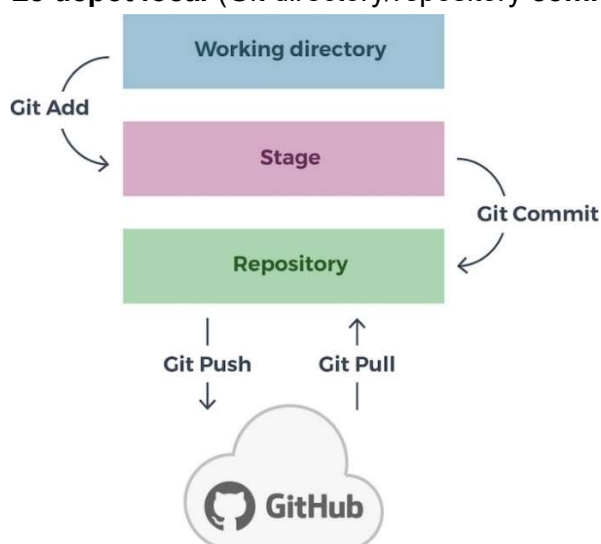
Le système de sauvegarde (commit) :

Présentation du système de sauvegarde de git :



Git gère les versions de nos travaux locaux à travers 3 zones locales majeures :

- **Le répertoire de travail** (working directory/WD) ;
- **L'index, ou stage** (File d'attente) ;
- **Le dépôt local** (Git directory/repository **commit**).



Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

GIT

Gestion des fichiers et commits (sauvegarde) :

1 Créer un fichier dans votre répertoire :

Ex : test.txt

2 Ajouter le fichier à git (Stage) :

Utiliser la commande ci-dessous :

```
$ git add test.txt
```

3 Sauvegarde des modifications (commit) :

Vous avez réalisé des évolutions sur la branche **nom de la branche** et il va falloir maintenant demander à Git de les enregistrer.

Pour cela on va utiliser la commande commit avec -m (pour ajouter un message).

```
$ git commit -m "message de la modification"
```

4 Sauvegarde des modifications (commit simple) :

```
$ git commit
```

Cette commande va ouvrir l'éditeur Vim.

On vous demande alors d'indiquer le message du commit puis de valider. Pour valider le message, une fois que vous l'avez écrit, appuyez sur Echap (votre curseur va basculer sur la dernière ligne) et

Saisir

```
:x!
```

Cette commande va sauvegarder et quitter l'éditeur des messages de commit.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

GIT

5 Sauvegarder plusieurs création, modifications :

Plutôt que faire plusieurs commits individuel on va utiliser le paramètre `-am` sur la commande `commit` voir exemple ci-dessous :

On va ajouter le/les fichiers (stage) avec la commande suivante :

```
$ git add * ou git add -A
```

Sauvegarde (commit) avec la commande suivante :

```
$ git commit -am "message modifications multiples"
```

6 sauvegarder des modifications de fichiers (édition de fichiers) :

Pour sauvegarder des modifications dans des fichiers on va utiliser la même méthode que pour ajouter des fichiers :

Éditer un fichier en local :

-Ouvrir un fichier et l'éditer (par ex : ajouter du contenu) puis sauvegarder le.

Ajouter le fichier à git avec la commande suivante :

```
$ git add nom_fichier.extension (ex : test.txt)
```

Sauvegarder la modification avec la commande suivante (commit) :

```
$ git commit -m "message de la modification"
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

GIT

7 Suppression de fichiers :

Pour supprimer des fichiers on va utiliser la commande suivante (stage):

```
$ git rm nom_fichier.extension
```

On répète l'opération si on n'a plusieurs fichiers à supprimer.

Sauvegarde des suppressions (commit) on va utiliser la commande suivante :

```
$ git commit -m ou -am "message de la modification"
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

GIT

Les Remises (Stash) :

Les remises permettent de mettre de côté des modifications (ajout, modifications ...) pour pouvoir les copier sur la branche actuelle ou une autre branche à n'importe quel moment. Cela fonctionne comme une liste d'attente.

1 Voir l'état des fichiers :

```
$ git status
```

2 Création d'une remise :

Pour pouvoir effectuer une remise il faut avoir précédemment avoir ajouter les nouveaux fichiers ou modifications avec la commande ci-dessous :

```
$ git add *
```

Cette commande va permettre d'ajouter tous les fichiers à la fois (*modifications et nouveaux fichiers*)

Pour effectuer la remise nous allons utiliser la commande ci-dessous :

```
$ git stash
```

Cela va créer une remise et mettre en attente les modifications (*les fichiers ou modifications disparaîtront de votre dossier local*).

3 Application de la remise :

Cette commande va appliquer toutes les modifications précédentes (*Créer une remise*) sur la branche sélectionnée.

(BrancheCommit).

```
$ git stash apply
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

GIT

4 Voir la liste des remises :

Nous avons la possibilité de faire plusieurs remises afin de voir la liste des remises nous allons utiliser la commande ci-dessous :

```
$ git stash list
```

Cela permet de voir la liste des remises avec un identifiant associé à chaque remise.

5 Application d'une remise avec son identifiant :

```
$ git stash apply stash@{0}
```

Cette commande applique sur la branche sélectionné la remise « **stash@{0}** » (la première dans ce cas).

6 Supprimer une remise avec son identifiant :

Récupérer l'identifiant de la remise à supprimer avec la commande suivante :

```
$ git stash list
```

7 Supprimer une remise individuelle utiliser la commande suivante :

```
$ git stash stash@{id de la remise} drop
```

8 Vider la liste des remises :

```
$ git stash drop
```

Cela aura pour effet de vider complètement la liste des remises.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

GIT

Interaction avec un repository distant (github) :

Téléchargement des données distantes vers votre dépôt local (dossier)

1 La première des choses à faire quand on souhaite travailler sauvegarder son travail en ligne est de créer un dossier dans son ordinateur.

2 Sélectionner le dossier : clic droit -> Git Bash here ou cd url du dépôt local (ex :
 c:/users/test/desktop/depot_local/.

3 Initialiser le dossier avec la commande ci-dessous :

```
$ git init
```

4 Clonage du dépôt distant (github) :

Pour plus d'information consulter la page 8 (Cloner un dépôt github (dépôt distant) sur votre repository (local)

Saisir la commande ci-dessous :

```
$ git remote add clone https://github.com/votre_url.git
```

Cela va connecter votre dépôt local(dossier) à votre dépôt distant(github).

5 Téléchargement des données distantes :

Pour télécharger le contenu d'un repository (github) nous allons utiliser la commande ci-dessous :

```
$ git pull https://github.com/votre_url.git
```

L'adresse du dépôt distant se trouve sur le site de github (bouton code -> https)

NB : Pour pouvoir télécharger des données un fichier doit se trouver sur le dépôt distant (à la création veuillez penser à cocher la case add à readme file ou créer un fichier sur le repository github distant)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

GIT

Envoi des données locales vers votre dépôt distant (repository github)

Il faut avoir téléchargé les données du dépôt distant en Amon (`git pull url_dépôt_github`).

1 Créer ou plusieurs fichiers dans votre dossier local puis ajoutez les a Git avec la commande :

```
$ git add nom_fichier.extension ou git add * (si plusieurs)
```

2 Sauvegarder les modifications avec la commande ci-dessous :

```
$ git commit -m "message modifications multiples" ou -am
```

3 Envoie des données sur le repository (github) avec la commande ci-dessous :

```
$ git push https://github.com/votre_url.git nom_branche
```

Par défaut on utilisera la branche **master**.

La première fois que vous envoyez des données à distance la console git va vous demander de valider vos identifiants de connexion de github et d'autoriser la connexion. Une fenêtre popup apparaîtra elle va vous guider pour autoriser la connexion (cela se fait une seule fois). Suivant la taille, le nombre de fichiers, votre connexion internet cela sera plus ou moins long.

NB : (Optionnel) :

Si on a créé une autre branche et que l'on souhaite l'envoyer en ligne (repository github on devra en premier lieu se positionner sur la branche désirée avec la commande :

```
$ git checkout nom_branche
```

On réutilisera la commande `git push url_github nom_de_la_branche`. La branche se créera automatiquement sur github.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

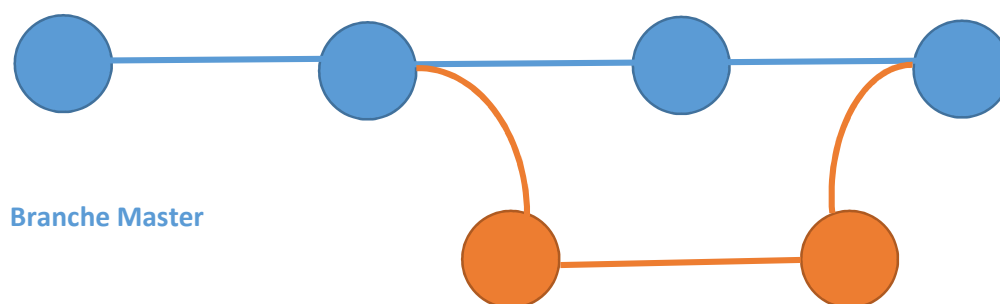
GIT

Le système de branches :

Présentation du système de branches :

Le principal atout de Git est son système de branches. Il faut voir le système de branche comme des « dossiers virtuel ».

La branche principale est appelée la branche master. C'est celle-ci, où au final, vous aurez à la fin toutes vos modifications. Le but est de ne surtout pas réaliser les modifications directement sur cette branche, mais de réaliser les modifications sur d'autres branches, et après tests, les intégrer sur la branche master.



Branche Master

Branche Secondaire

Gestion des branches :

1 Afficher la liste des branches :

```
$ git branch
```

Cela va s'afficher comme ci-dessous : (*git branch*) :

```
git branch
*master
```

Cette commande va afficher la liste des branches avec un astérisque à côté de la branche sélectionnée dans ce cas-là la branche **master**.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

GIT

2 Créer une branche :

```
$ git branch nom de la branche
```

Ex : git branch secondaire (création d'une nouvelle branche qui s'appelle **secondaire**).

3 Basculer sur la branche secondaire :

```
$ git checkout secondaire
```

La branche va fonctionner comme un dossier virtuel. Avec la commande **Git checkout**, on va être téléporté dans le dossier virtuel nom de la branche « (secondaire) ».

4 Publier une branche :

```
$ git push origin secondaire
```

Il faut être sur la branche à publier (si elle n'existe pas elle va être créée automatiquement).

5 Renommer une branche :

```
$ git branch -m old_branch new_branch  
$ git push origin :old_branch  
$ git push origin new_branch
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		