

TP N°2 : Git : v 1.0

Source d'aide : <http://git-scm.com/book/fr/v1/Démarrage-rapide>

Chapters		
1st Edition		
1. Démarrage rapide	5. Git distribué	9. Les tripes de Git
1.1 À propos de la gestion de version	5.1 Développements distribués	9.1 Plomberie et porcelaine
1.2 Une rapide histoire de Git	5.2 Contribution à un projet	9.2 Les objets Git
1.3 Rudiments de Git	5.3 Maintenance d'un projet	9.3 Références Git
1.4 Installation de Git	5.4 Résumé	9.4 Fichiers groupés
1.5 Paramétrage à la première utilisation de Git		9.5 Les références spécifiques
1.6 Obtenir de l'aide	6. Utilitaires Git	9.6 Protocoles de transfert
1.7 Résumé	6.1 Sélection des versions	9.7 Maintenance et récupération
	6.2 Indexation interactive	de données
	6.3 Le remisage	9.8 Résumé
2. Les bases de Git	6.4 Réécrire l'historique	. Notes on the french
2.1 Démarrer un dépôt Git	6.5 Déboguer avec Git	translation process
2.2 Enregistrer des modifications dans le dépôt	6.6 Sous-modules	
2.3 Visualiser l'historique des validations	6.7 Fusion de sous-arborescences	
2.4 Annuler des actions	6.8 Résumé	
2.5 Travailler avec des dépôts distants		1. Repository
2.6 Étiquetage	7. Personnalisation de Git	2. Check in
2.7 Trucs et astuces	7.1 Configuration de Git	3. To check in
2.8 Résumé	7.2 Attributs Git	4. Check out
3. Les branches avec Git	7.3 Crochets Git	5. To check out
3.1 Ce qu'est une branche	7.4 Exemple de politique gérée par Git	6. To commit
3.2 Brancher et fusionner : les bases	7.5 Résumé	7. Stage area
3.3 Gestion de branches		8. To stage
3.4 Travailler avec les branches	8. Git et les autres systèmes	9. To unstage
3.5 Les branches distantes	8.1 Git et Subversion	10. Change
3.6 Rebaser	8.2 Migrer sur Git	11. stash
3.7 Résumé	8.3 Résumé	12. To stash
4. Git sur le serveur		13. Checksum
4.1 Protocoles		14. Hash
4.2 Installation de Git sur un serveur		15. Workflow
4.3 Génération des clés publiques SSH		16. Snapshot
		17. Diff
		18. To track
		19. untracked
		20. unmodified
		21. To update

Enregistrer les modifications dans le dépôt en local

1. Initialiser un dépôt Git et créer le fichier *oiseaux.txt* qui contient la liste suivante (un oiseau par ligne)

```
Canard colvert
Nette rousse
Canard pilet
Canard chipeau
```

2. Taper `git status`

Vous remarquez que le fichier *oiseaux.txt* ne fait pas partie du suivi de version.

Untracked files:

```
(use "git add <file>..." to include in what will be committed)

oiseaux.txt
```

3. Ajouter ce fichier pour un suivi de version.

Verifier avec la commande `git status` qu'il a bien été ajouté...

4. Ajoutez au fichier les noms suivants :

```
Canard siffleur
Sarcelle d'hiver
```

TP N°2 : Git : v 1.0

5. Refaites un `git status`. Comment connaître les modifications ?
6. Faites un commit en indiquant « Ajout des canards »
7. Taper `git log`
8. Modifier le fichier en ajoutant les noms suivants :

Chevalier culblanc
Chevalier aboyeur
Courlis corlieu
Courlis cendré

9. Faites un commit en indiquant « Ajout des limicoles »
10. Faites un `git log`

```
pc6:oiseaux sebastien$ git log
commit 22cb8522dfd7328eabffb3a33ccaba68ae18cff7
Author: SebInfo <contact@inion.info>
Date:   Sat Nov 15 11:11:17 2014 +0100

    Ajout des limicoles

commit 2b5ed55ed83c9d4ad6ea051b0c36feed7a5cf554
Author: SebInfo <contact@inion.info>
Date:   Sat Nov 15 11:01:32 2014 +0100

    Ajout des Canards
```

11. Revenez au commit Ajout des Canards.
Vous devez obtenir ceci

```
pc6:oiseaux sebastien$ cat oiseaux.txt
Canard colvert
Nette rousse
Canard pilet
Canard chipeau
```

Travailler avec les branches

12. Vous allez créer une branche test et basculer vers cette branche.

13. Modifier le fichier oiseaux.txt en ajoutant les noms suivants :

```
Chevalier culblanc
Chevalier aboyeur
Courlis corlieu
Courlis cendré
```

14. Faire un commit avec comme texte 'Ajout limicoles branche test'

15. Retourner sur la branche master et ajouter
Rossignol

16. Committer

17. Repasser sur la branche et afficher le fichier oiseaux.txt
Vous ne devriez pas avoir Rossignol

18. Dans la branche test ajouter : Martinet

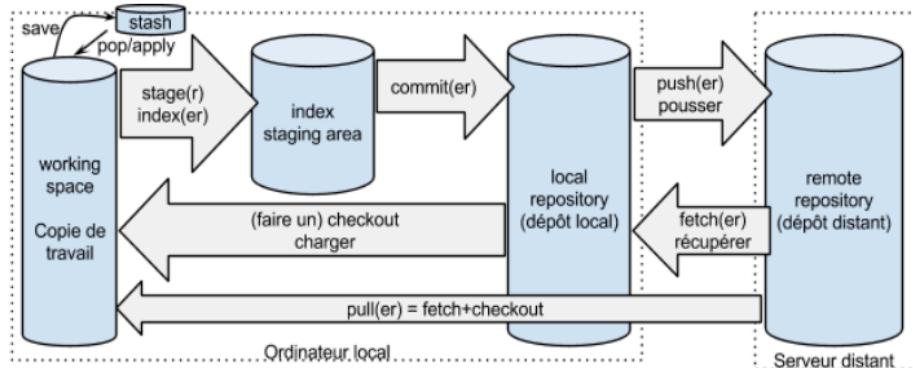
19. Fusionner les deux branches. Gérer les conflits.

20. Vérifier avec git status que tout va bien et faites un commit

TP N°2 : Git : v 1.0

Travailler avec un dépôt distant

Rappel :



Jusqu'à maintenant dans ce TP nous avons travaillé en local. Nous allons maintenant utiliser un dépôt distant afin de publier et partager nos fichiers.

Vous devez avoir un compte Github (<https://github.com>). Si ce n'est pas le cas crée le !

21. Sur Github créer un Repository du nom de TPGitOiseaux

22. Envoyer votre projet oiseau sur github.

TP N°2 BTS SIO — Edit

La capture d'écran montre la page d'édition d'un dépôt GitHub nommé « TPGitOiseaux ». On peut voir 7 commits et 1 branche. La branche actuelle est « master ». La description du dépôt est « fin du tp ». Un commit récent a été effectué par l'utilisateur « SebInfo » il y a 41 minutes, ajoutant le fichier « oiseaux.txt » avec la note « Ajout martinet branche test ».

23. Ajouter directement sur Github deux oiseaux :

Foulque macroule

Outarde canepetière

Commiter en indiquant "Ajout depuis Github"

24. Faire un fetch afin de récupérer dans le local dépository la dernière version sur Github.

25. Afficher le contenu de votre fichier oiseaux.txt

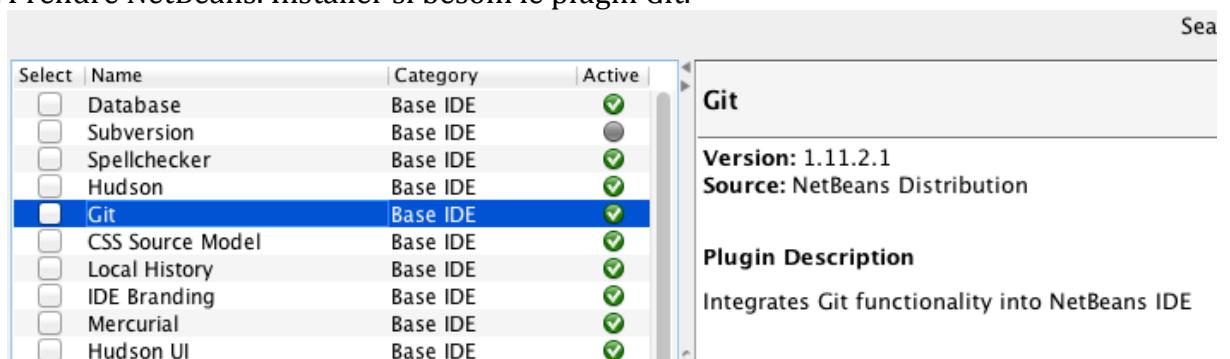
Comment expliquez vous qu'il n'y est pas les deux oiseaux ajoutés sur Github ?

TP N°2 : Git : v 1.0

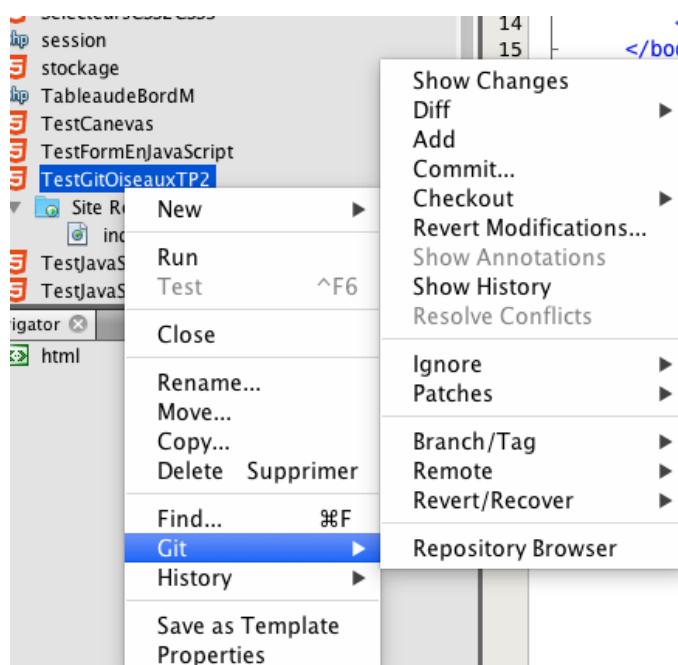
26. Faites un git pull et regarder à nouveau le fichier oiseaux.txt
27. Ajouter Echasse blanche
Committer
Afficher les différences avec le dépôt distant.
Vous deviez obtenir cela :

```
diff --git a/oiseaux.txt b/oiseaux.txt
index c501a70..11378f0 100644
--- a/oiseaux.txt
+++ b/oiseaux.txt
@@ -16,4 +16,5 @@ Courlis cendré
 Martinet
 Foulque macroule
 Outarde canepetière
+Echasse blanche
```

28. Prendre NetBeans. Installer si besoin le plugin Git.



29. Créer un projet de type HTML de base
Vous devez avoir ceci en option sur votre projet Netbeans



TP N°2 : Git : v 1.0

30. Créer sur Github un repository sur Github pour l'acceuillir.
31. Envoyer ce projet de base sur Github. Pour cela vous devrait paramétrer votre compte Github et faire un commit local avant d'avoir une branche master.
32. Travailler maintenant en binome sur une page de base de type HTML5.
Un est l'instigateur du projet l'autre doit contribuer afin d'améliorer ce mini site.
Je vous conseille de travailler avec des includes et des fichiers différents.
Exemple : Entete.php, PiedDePage.php
Gérer les conflits et amusez vous ;-)