



# TP I - Installer Git

- Rendez-vous à la page de téléchargement : <http://git-scm.com/download>
- Suivez les instructions suivant votre plateforme (Windows, Linux, Mac)
- Vérification de l'installation : `git --version`

# TP 2 - A propos de vous

Objectif : Paramétrage initial

# git config --list

```
user.name=Laurent Kappel
user.email=lkappel.ext@orange.com
alias.lol=log --graph --pretty=format:"%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset"
alias.st=status
alias.ci=commit
alias.co=checkout
alias.br=branch
alias.unstage=reset HEAD --
alias.last=log -1 HEAD
alias.lg1=log --graph --all --format=format:"%C(bold blue)%h%C(reset) - %C(bold green)(%ar)%C(reset) %C(white)%s%C(reset) %C(bold white). %an%C(reset)%C(bold yellow)%d%C(reset)" --abbrev-commit --date=relative
alias.lg2=log --graph --all --format=format:"%C(bold blue)%h%C(reset) - %C(bold cyan)%aD%C(reset) %C(bold green)(%ar)%C(reset)%C(bold yellow)%d%C(reset)%n" %C(white)%s%C(reset) %C(bold white). %an%C(reset)" --abbrev-commit
alias.lg=lg1
alias.stash-unapply=lgit stash show -p | git apply -R
color.diff=auto
color.diff.meta=yellow bold
color.diff.frag=magenta bold
color.diff.old=red bold
color.diff.new=green bold
gui.fontdiff=-family Courier -size 7 -weight normal -slant roman -underline 0 -overstrike 0
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
core.hidedotfiles=dotGitOnly
remote.origin.url=ssh://lkappel@localhost:29418/gitproject.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.dev_DNS.remote=origin
branch.dev_DNS.merge=refs/heads/dev_DNS
branch.master.remote=origin
branch.master.merge=refs/heads/master
branch.ticket/1.remote=origin
branch.ticket/1.merge=refs/heads/ticket/1
```

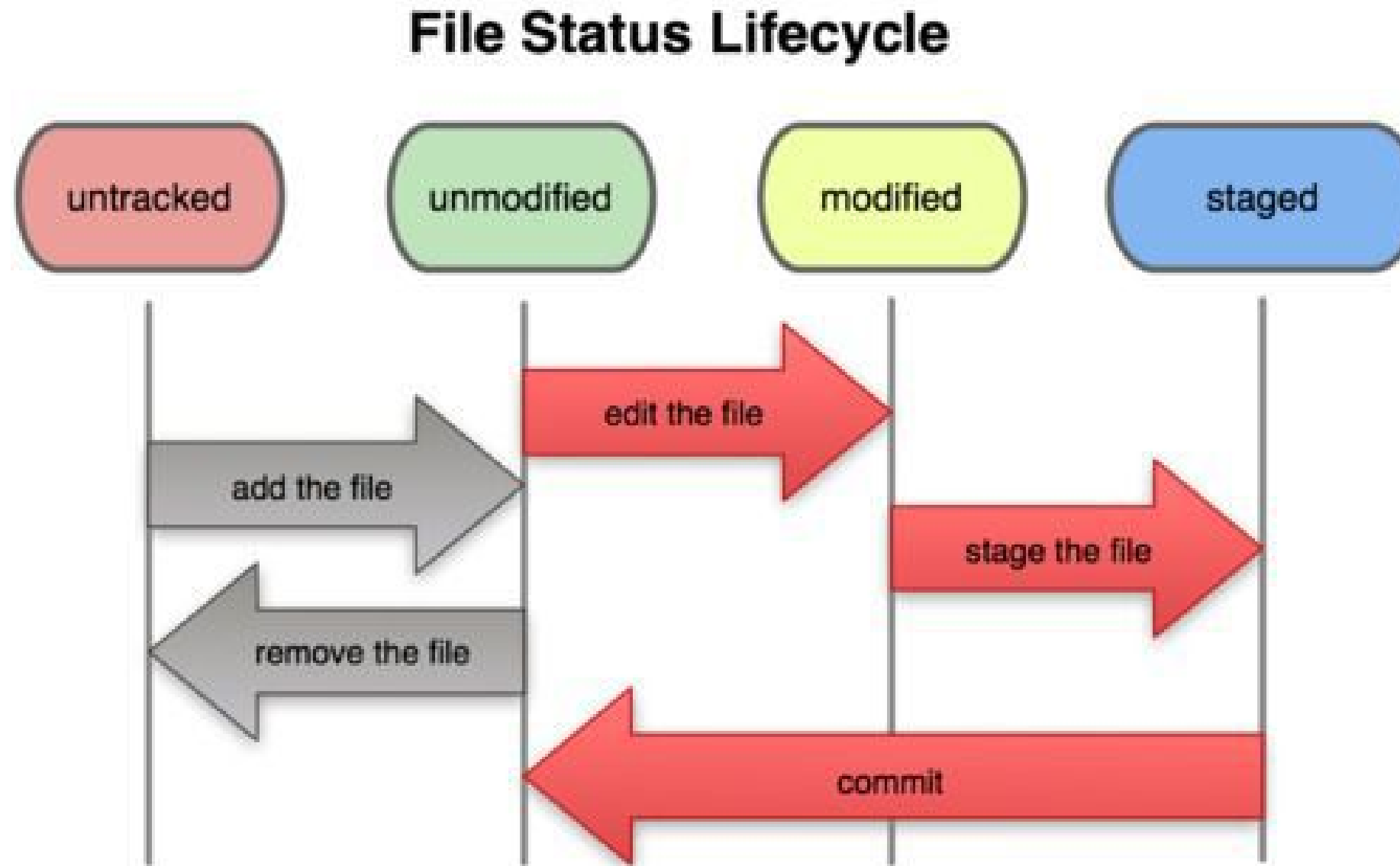
# Présentez-vous !

```
git config --global user.name "Kappel Laurent"  
git config --global user.email "lkappel@free.fr"
```

# TP 3 - Initialiser un projet

- Décompresser archive.zip
- Se positionner dans le répertoire container
- git init

# Cycle de vie d'un fichier



# TP 4 - Versionnez votre projet

Cas d'utilisation : versionner les fichiers du projet



# git status

```
# On branch master
#
# Initial commit
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#   README.md
#   pom.xml
#   src/
#   target/
```

Le répertoire “target” est un répertoire généré par Maven, nous ne souhaitons pas le versionner

# Créer un fichier .gitignore

Lister les fichiers/  
répertoires à ne pas  
versionner



# git status

```
# On branch master
#
# Initial commit
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#   .gitignore
#   README.md
#   pom.xml
#   src/
```

Le répertoire “target” n’apparaît plus

# git add .

Ajout de tous les fichiers à l'état “staged”  
=> candidat au prochain commit

# git status

**git commit -m “Ajout de toutes les  
ressources du projet”**

```
[master (root-commit) 9bd8c8a] Ajout de toutes les ressources du projet
118 files changed, 7431 insertions(+)
create mode 100644 .gitignore
create mode 100644 README.md
create mode 100644 pom.xml
...
```

# git log

```
commit 51d7ebaabe10d491fa76613e1090a4a96e0afa24
Author: Laurent Kappel <lkappel.@free.fr>
Date: Tue Sep 9 15:12:44 2014 +0200
    Ajout de toutes les ressources du projet
```

# Modifier .gitignore

Exclusion des fichiers  
pouvant être générés par  
Eclipse →

**.gitignore**

```
target  
.project  
.settings  
.classpath
```



# git status

```
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   .gitignore
#
no changes added to commit (use "git add" and/or "git commit -a")
```

# git commit -a -m “Exclusion des fichiers Eclipse”

```
[master 22180e6] Exclusion des fichiers Eclipse  
1 file changed, 3 insertions(+)
```

attention, commit automatique des fichiers modifiés  
les nouveaux fichiers ne sont pas ajoutés au commit

# git log

commit 525120581d01e885e182ef72f8c61cd9b467f374  
Author: Laurent Kappel <lkappel.ext@orange.com>  
Date: Tue Sep 9 15:14:29 2014 +0200

Exclusion des fichiers Eclipse

commit 51d7ebaabe10d491fa76613e1090a4a96e0afa24  
Author: Laurent Kappel <lkappel.ext@orange.com>  
Date: Tue Sep 9 15:12:44 2014 +0200

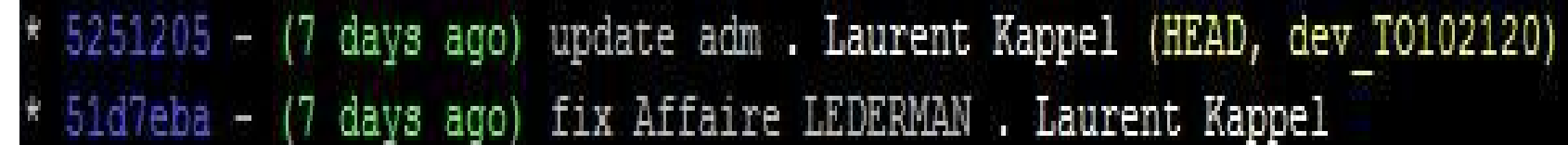
Ajout de toutes les ressources du projet

# git status

```
# On branch master  
nothing to commit (working directory clean)
```

# Exemple de customisation git log

```
log --graph --all --format=format:'%C(bold blue)%h%C(reset) - %C(bold green)(%ar)%C(reset)
%C(white)%s%C(reset) %C(bold white). %an%C(reset)%C(bold yellow)%d%C(reset)' --abbrev-commit --date=relative
```



```
* 5251205 - (7 days ago) update adm . Laurent Kappel (HEAD, dev_T0102120)
* 51d7eba - (7 days ago) fix Affaire LEDERMAN . Laurent Kappel
```

# TP 5 - Oops ! Je me suis trompé

Objectif : voir comment se sortir d'une manipulation malheureuse

**Cas 1 : Je n'aurai pas dû modifier  
ce fichier**

# Modifier le fichier .gitignore

Ajout du répertoire src →

**.gitignore**

```
target  
.project  
.settings  
.classpath  
src
```



# git status

```
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   .gitignore
#
no changes added to commit (use "git add" and/or "git commit -a")
```

# git checkout .gitignore

**.gitignore**

target  
.project  
.settings  
.classpath

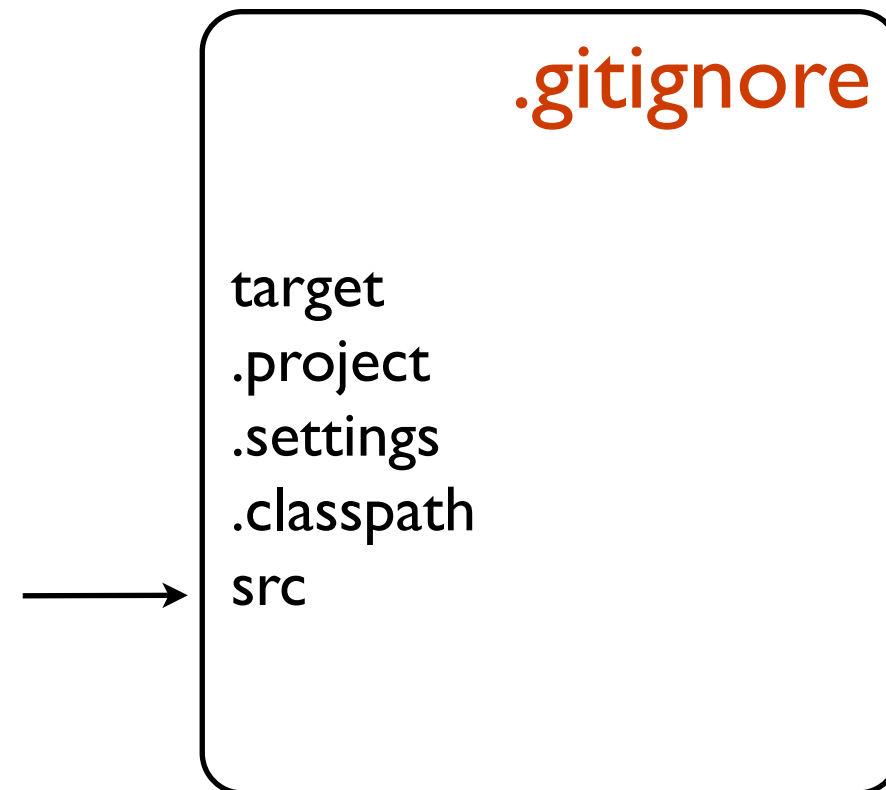
Retour à l'état initial

# git status

```
# On branch master  
nothing to commit (working directory clean)
```

# Cas 2 : le “add” de trop

# Modifier le fichier .gitignore



# git add .gitignore

Fichier à l'état “staged” => fera partie du prochain commit

# git status

```
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   .gitignore
#
```

# git reset HEAD .gitignore

Unstaged changes after reset:  
M .gitignore



# git status

```
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#    modified:   .gitignore
#
no changes added to commit (use "git add" and/or "git commit -a")
```

# Cas 3 : le commit de trop

# Modifier le fichier .gitignore

Ajout du répertoire src →

**.gitignore**

```
target  
.project  
.settings  
.classpath  
src
```

# git commit -a -m “Exclusion du répertoire src”

```
[master 59a620a] Exclusion du répertoire src  
1 file changed, 1 insertion(+)
```

# git log --pretty=oneline

```
59a620a8a78f6643b42df250ad02ffd903bf80e2 Exclusion du répertoire src  
22180e6fe8effc72e05e173c313037be76dafdae Exclusion des fichiers Eclipse  
9bd8c8a990b56384619413ff8d898a809269b3a2 Ajout de toutes les ressources du projet
```

# git reset --soft HEAD~

Supprime le dernier commit, le fichier passe à l'état "staged"

Seulement si le commit n'a pas été partagé avec un autre  
dépôt Git

# git status

```
2180e6fe8effc72e05e173c313037be76dafdae Exclusion des fichiers Eclipse
9bd8c8a990b56384619413ff8d898a809269b3a2 Ajout de toutes les ressources du projet
```

# git revert HEAD

Création d'un nouveau commit annulant l'effet du dernier commit

Pratique lorsque le commit a été partagé avec un autre  
dépôt Git



**Cas 4 : j'ai oublié d'ajouter un  
fichier dans mon dernier commit**

# git log --stat

dernier commit = 1 fichier



```
commit 22180e6fe8effc72e05e173c313037be76dafdae
Author: Laurent Kappel <lkappel@free.fr>
Date: Wed Aug 22 22:28:27 2012 +0200
```

Exclusion des fichiers Eclipse

```
.gitignore | 3 +++
1 file changed, 3 insertions(+)
```

```
commit 9bd8c8a990b56384619413ff8d898a809269b3a2
Author: Laurent Kappel <lkappel@free.fr>
Date: Wed Aug 22 22:20:35 2012 +0200
```

Ajout de toutes les ressources du projet

```
.gitignore | 1 +
```

....

# Créer un fichier help.txt



# git status

```
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#    modified:   .gitignore
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#    help.txt
no changes added to commit (use "git add" and/or "git commit -a")
```

```
git add help.txt  
git commit --amend
```

Possibilité d'adapter le commentaire du précédent commit

# git log --stat

commit 7470d58dac24636f839250bbff934bac18e6f996  
Author: Laurent Kappel <lkappel@free.fr>  
Date: Wed Aug 22 22:28:27 2012 +0200

Exclusion des fichiers Eclipse

dernier commit = 2 fichiers



.gitignore | 3 +++  
help.txt | 1 +  
2 files changed, 4 insertions(+)

commit 9bd8c8a990b56384619413ff8d898a809269b3a2  
Author: Laurent Kappel <lkappel@free.fr>  
Date: Wed Aug 22 22:20:35 2012 +0200

Ajout de toutes les ressources du projet

.gitignore | 1 +

# TP 6 - Gérer les branches

Objectif : Créer/manipuler des branches

# git branch develop

Création d'une branche develop



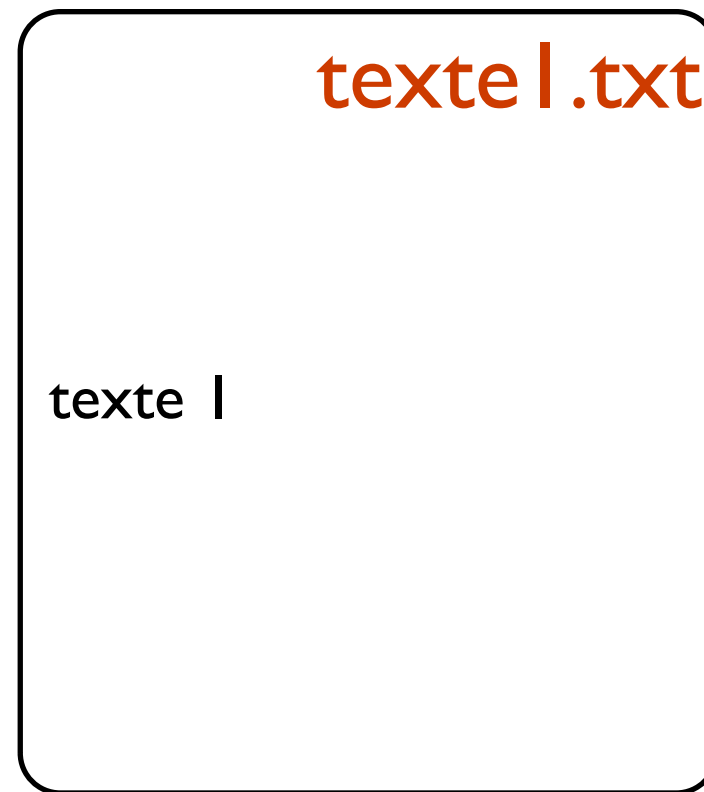
# git branch

La liste des branches existantes

```
develop  
* master
```

Le symbole \* pour la branche courante

# Créer un fichier texte | .txt



```
git add textel.txt  
git commit -m "Commit textel.txt  
branch master"
```

```
1 file changed, 1 insertion(+)  
create mode 100644 textel.txt
```

# git log --pretty:oneline

```
f7d65f859a04e16284e34edf1ad599c2966034ea Commit texte1.txt branche master  
7470d58dac24636f839250bbff934bac18e6f996 Exclusion des fichiers Eclipse  
9bd8c8a990b56384619413ff8d898a809269b3a2 Ajout de toutes les ressources du projet
```

# git checkout develop

Abandon de la branche master  
Positionnement sur la branche develop

# git branch

\* develop  
master

# git log --pretty:oneline

```
7470d58dac24636f839250bbff934bac18e6f996 Exclusion des fichiers Eclipse  
9bd8c8a990b56384619413ff8d898a809269b3a2 Ajout de toutes les ressources du projet
```

Le fichier texte l.txt n'est pas présent et le commit fait  
précédemment est absent de la branche

# git merge master

Updating 7470d58..f7d65f8

Fast-forward

textel.txt | 1 +

1 file changed, 1 insertion(+)

create mode 100644 textel.txt



# git log --pretty=oneline

```
f7d65f859a04e16284e34edf1ad599c2966034ea Commit texte1.txt branche master  
7470d58dac24636f839250bbff934bac18e6f996 Exclusion des fichiers Eclipse  
9bd8c8a990b56384619413ff8d898a809269b3a2 Ajout de toutes les ressources du projet
```

le fichier texte1.txt est désormais présent dans la branche develop

# TP 7 : Gérer les conflits entre les branches

Objectif : faire des modifications différentes d'un même fichier sur plusieurs branches et reporter des modifications entre branches

# Dans la branche develop modifier le fichier texte1.txt



```
git commit -a -m "Texte | develop"
```

# Dans la branche master modifier le fichier texte1.txt



```
git commit -a -m "Texte l master"
```

**Objectif = Merge “develop” vers  
“master”**

# git checkout master git merge develop

Auto-merging textel.txt

CONFLICT (content): Merge conflict in textel.txt

Automatic merge failed; fix conflicts and then commit the result.

Oops !



# git status

```
# On branch master
# Unmerged paths:
#   (use "git add/rm <file>..." as appropriate to mark resolution)
#
#   both modified:   textel.txt
#
no changes added to commit (use "git add" and/or "git commit -a")
```

# Ouvrir le fichier texte1.txt

Pas de panique, les symboles  
utilisés sont des “standards”

texte1.txt

```
<<<<<<< HEAD  
texte master  
=====  
texte develop  
>>>>>>>  
develop
```

# Pour résoudre le conflit, plusieurs choix

- Modifier le fichier à la main en veillant à supprimer les chaines de caractères “<<<<<<< HEAD”, “=====”, “>>>>>>> develop”
- Utiliser un outil de merge

# git mergetool

merge tool candidates: tortoisemerge emerge vimdiff

Merging:

textel.txt

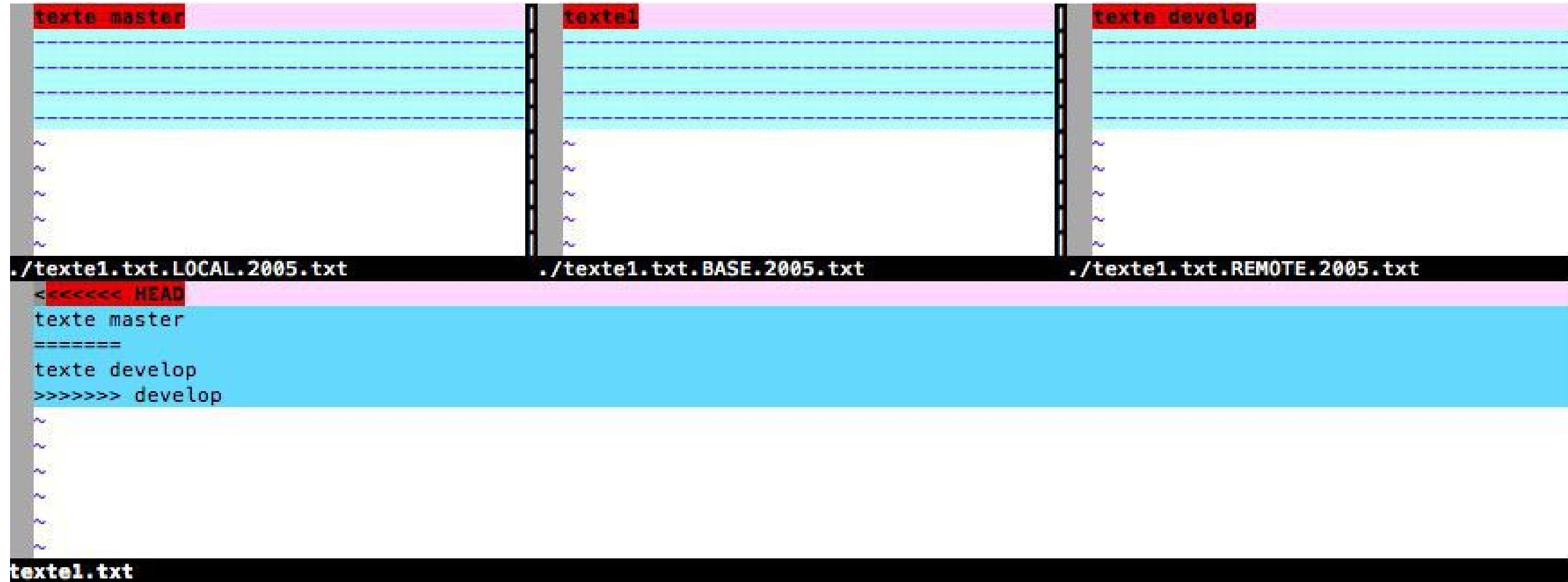
Normal merge conflict for 'textel.txt':

{local}: modified file

{remote}: modified file

Hit return to start merge resolution tool (vimdiff):

# Avec vimdiff



# Modifier le fichier texte1.txt

texte1.txt

texte master develop

```
git add textel.txt  
git commit -m "Merge OK"
```

```
[master l eca022] Merge Ok
```

# git log --pretty=oneline

```
1eca0225aebd35927c340214919597be173168c7 Merge Ok  
3b40b1fcb2e4cd5483818e06a0a4c5c6aeb97 TexteI master  
cbe96035ac64454ba53157e4d7fd57abfed60d88 TexteI develop  
f7d65f859a04e16284e34edf1ad599c2966034ea Commit texteI.txt branche master  
7470d58dac24636f839250bbff934bac18e6f996 Exclusion des fichiers Eclipse  
9bd8c8a990b56384619413ff8d898a809269b3a2 Ajout de toutes les ressources du projet
```

L'historique du commit "TexteI develop" est également  
récupéré sur la branche master



# TP 6 - Travailler avec des dépôts distants

Objectif : Récupérer/Partager des modifications d'un dépôt Git distant

# git clone https://github.com/tpaskap/GITDOC.git

```
Cloning into 'tpaskap'...
remote: Counting objects: 2201, done.
remote: Compressing objects: 100% (938/938), done.
remote: Total 2201 (delta 906), reused 2129 (delta 835)
Receiving objects: 100% (2201/2201), 85.21 MiB | 1.07 MiB/s, done.
Resolving deltas: 100% (906/906), done.
```

# Modifier le fichier README.md

Modifier du texte →

## README.md

jcertif-webapp 2012

=====

Web Application :

<http://www.jcertif.com>

# git commit -a -m “Modif README.md”

[master 5d3373c] Modif README.md  
1 file changed, 1 insertion(+), 1 deletion(-)

# git log --pretty=oneline

```
5d3373c2106df3c782e1c2e5e8dc817f068dd4ac Modif README.md
a56a3e84c096a58df7f900e349f5e43d7ada5f1f Merge pull request #47 from roddet/develop
45f1374f7a31df3d38a2eebb0abb9e5aa1e985dc [SPEAKERS] Ajout de la possibilité d'utiliser les balises HTML pour les
bios.
2c098eb7d66de510a102b578fbd8d118d9da0ce2 [HOME] Modification du texte éditorial
ba3f7ae455f3b28d87ea3f9047c8d961a7a574e9 Merge pull request #46 from roddet/develop
153a72c6d2a07605c5833a835e5fc0b2efbd35fb [AGENDA] Suppression du libellé 'Salle' de l'entête des colonnes
pour éviter que de
8879d6e942b0a3f34b5104c70af61f39068bddeb Merge pull request #41 from roddet/develop
de3957d7833b8489c3eb039b862879ec55941e53 [AGENDA] Red color on mouse over
```

# git remote -v

```
origin ssh://github.com/tpaskap/GITDOC.git (fetch)  
origin ssh://github.com/tpaskap/GITDOC.git (push)
```

```
git remote add lkappel https://github.com/tpaskap/GITDOC.git
```

Ajout d'un dépôt distant "lkappel"

# git remote

origin  
lkappel



**git push origin master**

↑  
**Dépôt**

↑  
**Branche**

git pull