Création du dépôt git

# Liens sur tutoriaux

## 15 points pour maîtriser Git

<http://www.responsive-mind.fr/git-15-points/>

## Git – petit guide

<http://rogerdudler.github.io/git-guide/index.fr.html>

## Git cheat sheet

<http://rogerdudler.github.io/git-guide/files/git_cheat_sheet.pdf>

## Site officiel Git

<https://git-scm.com/book/fr/v2/Les-bases-de-Git-Travailler-avec-des-dépôts-distants>

## Tutoriel atlassian (mélange français et anglais)

<https://fr.atlassian.com/git/tutorials/what-is-version-control>

## GitHub pour les nuls

<https://www.christopheducamp.com/2013/12/15/github-pour-nuls-partie-1/>

## Création d’un répertoire en local :



## Initialisation du nouveau dépôt git en local

En mode terminal, se placer dans le dossier « Repo\_PHP\_projet\_final » et lancer la commande *git init*

## Se connecter au dépôt distant

Utiliser la commande suivante pour un serveur distant

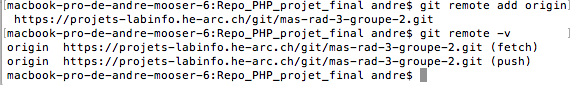
*git remote add nom\_du\_depot url\_du\_depot*

dans notre cas

*git remote add origin https://projets-labinfo.he-arc.ch/git/mas-rad-3-groupe-2.git*

contrôle si le dépôt a bien été cloné

git remote –v



## Cloner le dépôt distant sur la machine locale

Utiliser la commande suivante pour un serveur distant

*git clone chemin\_vers\_le\_depot*

Dans notre cas

*git clone https://projets-labinfo.he-arc.ch/git/mas-rad-3-groupe-2.git*

## Ajouter les fichiers et répertoires au dépôt local

*Git add \**

## Valider les modifications

Utiliser la commande suivante

*git commit –m « Message de validation »*

Dans notre cas

*git commit –m « Premier commit »*

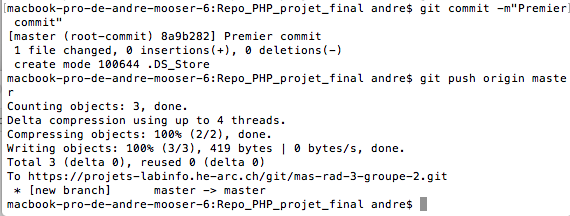
## Envoyer les changements au dépôt distant

Utiliser la commande suivante

*git push nom\_du\_depot nom\_de\_la\_branche*

Dans notre cas

*git push origin master*



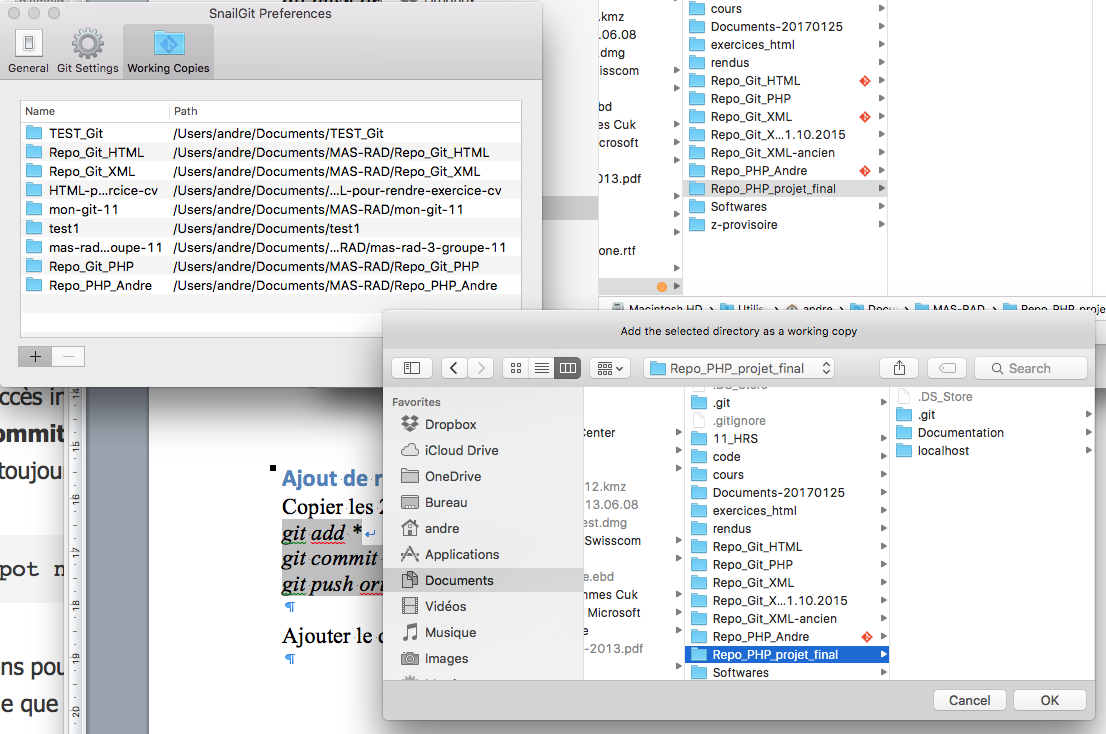
## Ajout de répertoire « Documentation » et « localhost »

Copier les 2 répertoires dans le dossier « git » local puis lancer les commandes suivantes

*git add \*  
git commit –m « Ajout des répertoires Documentation et localhost »*

*git push origin master*

## Ajouter le dépôt git à un outil graphique sur Mac (SnailGit)



## Branches et fusions : les bases

<https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Branches-et-fusions%C2%A0%3A-les-bases>

## Gestion des branches

<https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Gestion-des-branches>

## Travailler avec les branches

<https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Travailler-avec-les-branches>

## Travailler avec les branches distantes

<https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Branches-distantes>

Installation de Composer

# Liens sur explications et tutoriaux

## Téléchargement et installation :

<https://getcomposer.org/download/>

## Installer composer sous OSX :

<https://www.abeautifulsite.net/installing-composer-on-os-x>

## Utilisation de base de composer :

<https://www.youtube.com/watch?v=EBZ1owgiSSQ&t=458s>

## Composer et phpdoc :

## <https://www.youtube.com/watch?v=675O2oDmr2w>

## Explication de l’arborescence faite pour le projet

Un répertoire nommé « Repo\_PHP\_projet\_final » a été créé comme dossier de départ, également pour Git. À l’intérieur de celui-ci, un répertoire « localhost » qui contient les fichiers php et qui seront, une publiés sur le site de l’hébergeur. Au même niveau que « localhost », se trouve un répertoire « Documentation » qui contient la documentation du projet et un répertoire « DB » qui contient les fichiers de configuration des base de données et tables nécessaires pour notre projet. Nous allons installer composer à ce même niveau, afin d’éviter que ses fichiers soient dans « localhost », ce qui évitera de devoir faire un tri dans « localhost » à chaque fois que nous publieront sur le site de l’hébergeur.

# Téléchargement des sources

Se rendre sur le site [getcomposer.org](https://getcomposer.org), puis sur la page « Download »

# Installation sur Mac

Suivre l’explication de la page « Download » :

* dans le Terminal, se placer dans le répertoire « Repo\_PHP\_projet\_final »
* depuis la page « Download » de composer, copier les commandes suivantes et les coller dans le Terminal

*php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"*

*php -r "if (hash\_file('SHA384', 'composer-setup.php') === '669656bab3166a7aff8a7506b8cb2d1c292f042046c5a994c43155c0be6190fa0355160742ab2e1c88d40d5be660b410') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP\_EOL;"*

*php composer-setup.php*

*php -r "unlink('composer-setup.php');"*

Un fichier « composer.phar » a été créé dans le répertoire « Repo\_PHP\_projet\_final ».

Installation de phpdoc

# Liens sur explications et tutoriaux

## Installer phpdoc :

<https://www.phpdoc.org/>

## Running phpdocumentor :

<https://www.phpdoc.org/docs/latest/guides/running-phpdocumentor.html>

# Téléchargement des sources

Il y a plusieurs façons d’installer phpdoc expliquées sur le site <https://www.phpdoc.org>

Nous allons installer phpdoc au moyen de composer. Si nous n’avons encore jamais lancé composer, nous ne trouvons que le fichier « composer.phar » dans le répertoire « Repo\_PHP\_projet\_final » et il nous manque le fichier « composer.json ». Il faut donc créer un fichier « composer.json » contenant le code minimal suivant :

*{*

*"require-dev": {*

*"phpdocumentor/phpdocumentor": "2.\*"*

*}*

*}*

Ensuite, lancer l’installation au moyen de la commande suivante :

Pour Mac *php composer.phar install*

Pour Windows *composer install*

# Exécuter phpdoc

## Exécution sous Mac

Dans le Terminal, se rendre dans le répertoire « Repo\_PHP\_projet\_final » et lancer la commande suivante :

*vendor/bin/phpdoc -d localhost/test/ -t Documentation/docphp*

Ceci crée un répertoire docphp dans « Documentation ». Lancer index.htlm pour accéder à la documentation ainsi construite.