

# INSTALLER DESKTOP POUR GITHUB

---

<https://desktop.github.com/>

## TUTORIEL EN FRANCAIS

---

<https://openclassrooms.com/courses/gerer-son-code-avec-git-et-github>

## ANTISECHE

---

<https://services.github.com/on-demand/downloads/fr/github-git-cheat-sheet.pdf>

## UTILISATION DE GIT

---

ON CREE UN REPOSITORY (DEPOT) SUR LA PLATEFORME GITHUB OU BITBUCKET.

ENSUITE, ON FAIT UNE ASSOCIATION ENTRE LE REPOSITORY ET UN DOSSIER SUR NOTRE ORDINATEUR.

=> CLONE ou CHECKOUT

=> SOIT ON PASSE PAR L'OUTIL COMME DESKTOP DE GITHUB OU SOURCETREE

=> SOIT ON PASSE PAR UNE COMMANDE EN LIGNE

DANS LE DOSSIER SUR MON ORDINATEUR, IL Y AURA UN DOSSIER .git

=> IL NE FAUT PAS L'EFFACER

=> SI ON DOIT FAIRE DES MODIFS DANS DES FICHIERS DE CE DOSSIER .git

=> FAIRE TRES ATTENTION CAR ON PEUT CASSER SON INSTALLATION

QUAND JE CREE UN NOUVEAU FICHIER

GIT DETECTE QU'IL Y A UN NOUVEAU FICHIER

MAIS JE DOIS LUI DIRE QUE CE FICHIER DOIT ETRE GERE PAR GIT

```
git add chemin-du-fichier
```

OU POUR AJOUTER TOUS LES NOUVEAUX FICHIERS EN UNE SEULE FOIS...

```
git add -A
```

SI ON FAIT UN

```
git status
```

ON VOIT QUE GIT A RAJOUTE CE FICHIER DANS LE REPOSITORY

IL FAUT FAIRE UNE SAUVEGARDE DE TOUT LE CODE DU PROJET (commit)

```
git commit -m "MESSAGE DESCRIPTIF"
```

ATTENTION: POUR LE MOMENT, CETTE SAUVEGARDE EST SEULEMENT SUR MON ORDINATEUR

POUR LE SYNCHRONISER AVEC LA PLATEFORME `github` ou `bitbucket`

IL FAUT FAIRE UN `push`

```
git push
```

## DEVELOPPEMENT A PLUSIEURS: GIT SIMPLE master

---

ON PEUT GARDER SEULEMENT LA BRANCHE `master`

ET TOUS LES DEVELOPPEURS CLONENT LA BRANCHE `master`

DONC CHAQUE DEVELOPPEUR FERA UN PUSH SUR LA BRANCHE `master`

LORS D'UN `push`, git MET A JOUR LES FICHIERS LOCAUX AVEC LES DERNIERES VERSIONS DU `master`

SI git DETECTE UN CONFLIT DANS UN FICHIER ALORS IL SIGNALE LE CONFLIT EN AJOUTANT SES TAGS DANS LE FICHIER POUR SIGNALER LES 2 VERSIONS DE CODE EN CONFLIT.

CELA REND ALORS GENERALEMENT LE FICHIER INUTILISABLE TEL QUEL

IL FAUT CORRIGER LE CODE DU FICHIER EN SE COLLABORANT ENTRE LES DEVELOPPEURS CONCERNES.

UNE FOIS LE FICHIER CORRIGE, IL FAUT MARQUER LE CONFLIT COMME RESOLU (OU EN FAISANT UN `git add` SUR LE FICHIER CORRIGE)

=> IL FAUT UNE BONNE ORGANISATION AU SEIN DE L'EQUIPE POUR NE PAS RENCONTRER TROP DE CONFLITS

## DEVELOPPEMENT A PLUSIEURS: GIT MULTI-BRANCHES

---

SI IL Y A TROP DE PUSH PROBLEMATIQUES SUR LA BRANCHE master

ALORS IL FAUT CREER DES BRANCHES (fork)

CHAQUE DEVELOPPEUR PEUT SE CREER SA BRANCHE ET TRAVAILLER SUR SA BRANCHE SANS ETRE GENE PAR LES AUTRES DEVELOPPEURS MAIS CELA NE FAIT QUE REPORTER LES PROBLEMES DE MERGE ENTRE BRANCHES... IL FAUDRA BIEN TOUT INTEGRER SUR LA BRANCHE master AU FINAL

LES BRANCHES SONT NECESSAIRES POUR GERER LES CORRECTIONS DE BUG DE VERSIONS ANCIENNES MAIS TOUJOURS MAINTENUES

POUR FUSIONNER LES CODES DE 2 BRANCHES ON PEUT UTILISER UN "pull request"

## ASTUCE POUR MIEUX TRAVAILLER EN EQUIPE

---

GIT SE BASE SUR LES FICHIERS...

PHP PERMET DE DECOUPER SON CODE EN PLUSIEURS FICHIERS (ET DE LES REUNIR AVEC

require\_once...)

SI L'EQUIPE ARRIVE A DECOUPER LE CODE EN SUFFISAMMENT DE FICHIERS ALORS CHAQUE DEVELOPPEUR PEUT TRAVAILLER SUR SES FICHIERS (SANS MARCHER SUR LES PIEDS DES AUTRES DEV...)

# GIT ET MYSQL

---

COMME LES DONNEES DE MYSQL SONT GEREES A PART

IL FAUT EXPORTER LA BASE DE DONNEES DANS UN FICHIER sql

POUR LE GERER AVEC git

LE CODE PHP DOIT AUSSI AVOIR LES IDENTIFIANTS POUR SE CONNECTER A MYSQL

LE PLUS SIMPLE EST QUE CHAQUE DEVELOPPEUR UTILISE LES MEMES NOMS POUR LA BASE DE DONNEES ET LES IDENTIFIANTS DE CONNEXION

ON POURRA AUSSI CREER UN EXPORT sql AVEC LES OPTIONS POUR SUPPRIMER LA BASE DE DONNEES ET RECREER TOUTE LA BASE DE DONNEES AVEC LES TABLES

ENSUITE, ON PEUT CREER UN FICHIER PHP D'INSTALLATION QUI VA CHARGER CE FICHIER sql

```
<!DOCTYPE html>
<html>
<head>
  <title>INSTALLATION</title>
</head>
<body>
```

```

        <header>
            <h1>INSTALLATION</h1>
        </header>
        <main>
<?php

$fichierSQL = "webforce3.sql";
if (is_file("$fichierSQL"))
{
    echo "INSTALLATION DE LA BASE DE DONNEES...";

    // http://php.net/manual/en/pdo.exec.php
    $dbh = new PDO('mysql:dbname=mysql;host=localhost;charset=utf8', 'root', '');
    $contenuSQL = file_get_contents($fichierSQL);
    echo "<pre>$contenuSQL</pre>";
    $count = $dbh->exec($contenuSQL);
}
else
{
    echo "fichier manquant: $fichierSQL";
}
?>

        </main>
        <footer>
        </footer>
</body>
</html>

```

# EQUIPES DE PROJETS ALEATOIRES

---

CREER DES EQUIPES DE 3/4 PERSONNES

DANS CHAQUE EQUIPE UN RESPONSABLE CREE UN REPOSITORY SUR SON COMPTE GITHUB ET PUIS AJOUTE LES AUTRES MEMBRES DE SON EQUIPE COMME COLLABORATEURS AVEC LES DROITS EN ECRITURE (INVITATION A ACCEPTER PAR EMAIL...)

CHAQUE MEMBRE DE L'EQUIPE PEUT ALORS FAIRE UN CLONE DU REPOSITORY DANS UN DOSSIER DE SON ORDINATEUR (htdocs/...) IL FAUT COMMENCER A TRAVAILLER CHACUN SUR SON ORDINATEUR ET PRODUIRE UN SITE VITRINE EN 3H...

## EQUIPE A

---

<https://github.com/archoss/siteVitrine.git>

PIERRE CREATEUR REPOSITORY siteVitrine YANN SONNY HERAMBAN

## EQUIPE B

---

<https://github.com/saberCoffee/wf3exo.git>

THOMAS CREATEUR REPOSITORY wf3exo THIBAUD ROMAIN VINCENT HWA SEON

## EQUIPE C

---

<https://github.com/andrei91gache/exerciceC.git>

THIERRY ROMAIN AMANDINE ANDREI CREATEUR REPOSITORY exerciceC

JOKER: ASHLEY