Git GitHub

Créée le mercredi 16 septembre 2020

- ▲ Retour à l'index Software

A propos de GIT et de GitHub

Git est un outil de versioning qui permet de tracer l'évolution de votre projet et d'y apporter des modifications
Github est une plateforme Web qui permet de créer un dépôt distant très simplement pour pouvoir partager v
Workflow parfait pour un travail collaboratif.
Plus concrètement, vous allez apprendre à initialiser Git, puis commencer à créer un projet sous Git.
Vous verrez comment sauvegarder le statut de votre projet grâce à la notion de commit et y apporter de nouv
avec les branches. Enfin, vous verrez les bases de la création de dépôts distant et l'utilisation de Github pour

- " git-scm.com" → la documentation complète en français sur Git

Convention de nommage

Une convention de nommage dans la documentation est un ensemble de règles de codage destinées à chois
documentation.

- "**CMD**" = Ligne de commande (terminal)

Chpt 01 - Installation de GIT

- " Youtube " → Tutoriel **Débuter avec Git et Github en 30 min**. (ATTENTION, certaines informatio
  la création de cette vidéo. Elles ont été corrigées dans ce tutoriel.)
- " Débuter avec Git et Github en 30 min" → La vidéo est consultable **uniquement** que depuis le wiki.

TIMESTAMP de la vidéo : (En **GRAS** rappel des lignes de "CMD")

1:55 Télécharger et configurer GIT : **git --version, git config --global user.name " John Doe ", git config**
johndoe@email.com "

3:35 Initialiser un nouveau projet et vérifier son état : **git init, git status**

6:50 et 9:17 Enregistrer un nouveau fichier ou une modification : **git add fileName , git commit -m " cor**

10:00 Timeline des commits : **git log**

11:00 Créer, montrer la liste, basculer, fusionner une branche : **git branch branchName , git branch, git c**
merge branchName , git branch -d branchName

18:46 Créer un repository GitHub.

20:00 Associer un dépôt local à un dépôt distant et vérification : **git remote add origin https://example.git**

20:40 Déposer votre projet : **git push origin branchName**

21:45 Récupérer un projet : **git pull origin branchName**

22:15 donner accès au projet à des collaborateurs.

Inititialisation de Git

- " git-scm.com" → télécharger la version qui correspond à votre système. Ici nous installerons la vers

Lancer l'installation de Git. Vous avez juste à suivre les indications, il n'y a rien de spéciale à faire.
Lorsque l'installation est terminée, nous allons tester le bon fonctionnement de Git avec le terminal "**CMD**"
Pour voir la version de Git tapez la "**CMD**" suivante :

1. **git --version**

```
C:\Users\Roots>git --version
git version 2.28.0.windows.1
```

Définir l'identité de l'utilisateur

Afin d'éviter de devoir fournir à chaque dépôt notre identité, nous allons la configurer une bonne fois pour tou
Configuration du nom d'utilisateur et de l'adresse email avec les "**CMD**" suivantes :

1. **git config --global user.name "Patrick-LR"**
2. **git config --global user.email "patrick.lopes-rego@laposte.net"**

NOTE : vous ne verez rien s'afficher sur l'écran lors de cette opération !

Chpt 02 - Création d'un projet

Création d'un répertoire en ligne de commande (vous devez avant tous, vous placez sur le repertoire de base

- NOTE : A propos des commandes dans le terminal.
- Les noms de dossiers ne sont pas sensible à la case (majuscules/minuscules).
- Si vous voulez aller sur votre Bureau : **mkdir C:\users\Toto\Desktop**
- Si vous avez des dossiers qui comportent des espaces, on regroupe au sein de guillemets : **mkdir "C:\ CSS\"**
- C'est la même chose quand vous arrivez à la commande "cd" : **cd "HTML et CSS"**

Dans l'exemple je suis dans le dossier "**E:\Liberkey\MyApps\UwAmp\WWW**" de mon serveur web.
1- Créer le dossier avec la "CMD" suivante :

- **mkdir Gitproject01**

2- Aller dans le dossier créé avec avec la "CMD" suivante :

- **cd Gitproject01**

3- Initialiser le projet Git dans le dossier fraichement créé avec la "CMD" suivante :

- **git init**

```
E:\Liberkey\MyApps\UwAmp\www\Gitproject01>git init
Initialized empty Git repository in E:/Liberkey/MyApps/UwAmp/www/Gitproject01/.git/
```

4- Pour voir à tout moment le statut des fichiers du projet, tapez la "CMD" suivante :

- **git status**

```
E:\Liberkey\MyApps\UwAmp\www\Gitproject01>git  status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

Maintenant que votre projet est initialisé, vous devez Créer une page HTML d'exemple dans le dossier "**GitPr** avec votre navigateur.

Le commit

Nous allons enregistrer l'état du projet actuel qui ce fait en 2 étapes.
NOTE : vous devez être avec votre "CMD" dans le dossier que vous voulez en "commit" !

1. La sélection des fichiers avec la "CMD" suivante : **git add filename**
2. Photo des fichiers avec la "CMD" suivante : **git commit -m "first commit"**

- **git add index.html**
- **git commit -m "Project init"**

```
E:\Liberkey\MyApps\UwAmp\www\Gitproject01>git commit -m "Project init"
[master (root-commit) 9a53562] Project init
 1 file changed, 13 insertions(+)
 create mode 100644 index.html
```

Nous allons modifier le fichier html et puis nous allons de nouveau publier le commit.

- **git add index.html**

- **git commit -m "Modif couleur du titre principal"**

- NOTE : en utilisant **git add -A** Pour sélectionner les fichiers que l'on veut sauvegarder, si nous en avc directement le commutateur (A = All).
- NOTE : en utilisant **git add .** (new and modified, without deleted - nouveau et modifié, sans suppressi commit.
- NOTE : en utilisant **git add -u** (modified and deleted, without new - modifié et supprimé, sans nouveau commit.

Time line des commits

Nous allons voir maintenant le time line des modifications enregistrées sur le commit.
Taper la "CMD" suivante :

- **git log**

```
E:\Liberkey\MyApps\UwAmp\www\Gitproject01>git log
commit 595e1b669c45209ef466a25b635515cbf2af886a (HEAD -> master)
Author: Patrick-LR <patrick.lopes-rego@laposte.net>
Date:   Wed Sep 16 12:54:37 2020 +0200

    Modif couleur du titre principal

commit 9a535620ea0d5c9984a5e225a685bc51ecadbbbb
Author: Patrick-LR <patrick.lopes-rego@laposte.net>
Date:   Wed Sep 16 12:47:24 2020 +0200

    Project init
```

Le classement des commits est du plus récent au plus ancien.
NOTE : en utilisant **git add -u -n** donne la liste des fichiers modifiés.

Les branches

Nous allons maintenant étudier "les branches" d'un commit.
Les branches sont des copies du projet initial. Git se charge de tout !
Nous utilisons la "CMD" suivante :

- **git branch backgroundColor**

NOTE : **backgroundColor** est le nom donnée à la branche par rapport au projet initiale.

Lister les branches

Pour lister les branches nous utiliserons la "CMD" suivante :

- **git branch**

```
E:\Liberkey\MyApps\UwAmp\www\Gitproject01>git branch
  backgroundColor
* master
```

Pour changer de branche il faut utiliser la "CMD" suivante
git checkout lenomdelabranche (ici un exemple !)

- **git checkout backgroundColor**

```
E:\Liberkey\MyApps\UwAmp\www\Gitproject01>git checkout backgroundColor
Switched to branch 'backgroundColor'
```

Nous modifions une fois de plus notre page HTML, et nous contrôlons les changement dans notre navigateur
Nous prenons une photo de notre projet avec la "CMD" suivante :

- **git add index.html**

- **git commit -m "Ajout d'un H1"**

```
E:\Liberkey\MyApps\UwAmp\www\Gitproject01>git commit -m "Ajout d'un H1"
[backgroundColor 717756f] Ajout d'un H1
 1 file changed, 5 insertions(+), 2 deletions(-)
```

Nous vérifions que tout est ok avec la "CMD" suivante :

- **git log**

```
E:\Liberkey\MyApps\UwAmp\www\Gitproject01>git log
commit 717756f2bbea404c46f76f65be93c636ac3cb5f4 (HEAD -> backgroundColor)
Author: Patrick-LR <patrick.lopes-rego@laposte.net>
Date:   Wed Sep 16 14:36:19 2020 +0200

    Ajout d'un H1

commit 595e1b669c45209ef466a25b635515cbf2af886a (master)
Author: Patrick-LR <patrick.lopes-rego@laposte.net>
Date:   Wed Sep 16 12:54:37 2020 +0200

    Modif couleur du titre principal

commit 9a535620ea0d5c9984a5e225a685bc51ecadbbbb
Author: Patrick-LR <patrick.lopes-rego@laposte.net>
Date:   Wed Sep 16 12:47:24 2020 +0200

    Project init
```

Rapatriement des branches dans le MASTER

Nous allons maintenant rapatrier les branches sur la branche MASTER. Cela ce fait en 3 étapes.

La 1ère étape est le changement de branche avec la "CMD" suivante :

- **git checkout master**

```
E:\Liberkey\MyApps\UwAmp\www\Gitproject01>git checkout master
Switched to branch 'master'
```

La 2ème étape nous allons fusionner les commits avec la "CMD" suivante :

- **git merge backgroundColor**

```
E:\Liberkey\MyApps\UwAmp\www\Gitproject01>git merge backgroundColor
Updating 595e1b6..717756f
Fast-forward
 index.html | 7 +++++--
 1 file changed, 5 insertions(+), 2 deletions(-)
```

La 3ème étape est la suppression de la branche avec la "CMD" suivante :
NOTE : Cette étape est facultative, mais si vous avez fusionné une branche dans le MASTER il vaut mieux n'utiliserez plus pour ne pas avoir un trop gros nombre de commits !

- **git branch -d backgroundColor**

```
E:\Liberkey\MyApps\UwAmp\www\Gitproject01>git branch -d backgroundColor
Deleted branch backgroundColor (was 717756f).
```

Vérification que la fussion c'est bien réalisée avec la "CMD" suivante :

- **git log**

```
E:\Liberkey\MyApps\UwAmp\www\Gitproject01>git log
commit 717756f2bbea404c46f76f65be93c636ac3cb5f4 (HEAD -> master)
Author: Patrick-LR <patrick.lopes-rego@laposte.net>
Date:   Wed Sep 16 14:36:19 2020 +0200

    Ajout d'un H1

commit 595e1b669c45209ef466a25b635515cbf2af886a
Author: Patrick-LR <patrick.lopes-rego@laposte.net>
Date:   Wed Sep 16 12:54:37 2020 +0200

    Modif couleur du titre principal

commit 9a535620ea0d5c9984a5e225a685bc51ecadbbbb
Author: Patrick-LR <patrick.lopes-rego@laposte.net>
Date:   Wed Sep 16 12:47:24 2020 +0200

    Project init
```

Chpt 03 - Le dépôt distant & Github

Déposons le projet sur un espace partagé, par exemple GitHub.

Créer un compte sur GitHub

- Allez sur le site " GitHub"
- Cliquer sur le bouton en haut à droite "Sign up"
- Indiquez un "**Username**" NOTE : mettez votre prénon-vos-initiales-de-votre-nom-de-famile par exemp
- Indiquez une adresse "**Email**" NOTE : utilisez une adresse gmail de préférence.
- Indiquez un "**Password**" NOTE : vous devez avoir des **MAJUSCULES** des **minuscules** et des **CHIFF**
- IMPORTANT Lorsque vous avez fini ces étapes, dirigez-vous sur votre BAL pour valider le courriel env valider l'email, si non vous ne pourriez pas utiliser votre compte Github.
- Choisissez votre niveau, par défaut "Unlimited public repositories for free"
- Décrivez rapidement votre expérience (facultatif, vous pourrez revenir dessus plus tard !)
- Maintenant, connectez-vous à votre profil, et vous allez éditer votre profil, remplissez vos informations e que je puisse vous reconnaitre facilement !
- Copiez l'url de votre GitHub et me la faire parvenir pour que je puisse vous ajouter au GitHub du Schoo

Créer un nouveau Repository

Dans le cadre d'un travail collaboratif, chaque membre peut participer au projet. Pour cela, nous allons maint dans le GitHub. Vous devez être sur votre GitHub pour faire cette manipulation.

- Depuis votre menu (en haut à droite) choisissez "**Your Repositeries**" (https://github.com/Patrick-LR
- Cliquer sur le bouton vert "**New**". (https://github.com/new)
- Donner un nom à votre dépôt (**Repository name**). Ici nous utiliserons le nom "**Gitproject01**" (si votre n l'indiquera par une croix **ROUGE**, si le nom est valide vous avez une coche **VERTE** !)
- Choisissez la visibilité de votre projet (**Public,** visible de tous le monde, **Private**, visible uniquement de
- Pour la partie "**Initialize this repository with:** - Initialisez ce référentiel avec:) nous reviendrons dessus
- Cliquez sur le bouton en bas de page "**Create repository**)

```
Quick setup — if you've done this kind of thing before
or
Get started by creating a new file or uploading an existing file. We recommend every repository include a


…or create a new repository on the command line

echo "# Gitproject01" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M master
```

```
git remote add origin https://github.com/Patrick-LR/Gitproject01.git
git push -u origin master

…or push an existing repository from the command line

git remote add origin https://github.com/Patrick-LR/Gitproject01.git
git branch -M master
git push -u origin master

…or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.
```

Création de notre espace de dépôt local vers GitHub

Dans les informations ci-dessus, GitHub nous a déjà donné les informations à utiliser avec la "CMD" suivante

- **git remote add origin https://github.com/Patrick-LR/Gitproject01.git**

Pour voir les espaces de dépôts a distance utilisez la "CMD" Suivante :

- **git remote**

```
E:\Liberkey\MyApps\UwAmp\www\Gitproject01>git remote
origin
```

Push pour déposer le projet

Maintenant nous allons déposer nos fichiers locaux vers le dépôt de GitHub avec la "CMD" suivante :

- **git branch -M master** (ne pas oublier de ce placer dans la branche MASTER)
- **git push -u origin master** (Une fenêtre vous demandera vos identifiants de connexion à GitHub)

```
E:\Liberkey\MyApps\UwAmp\www\Gitproject01>git push -u origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 1.06 KiB | 543.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/Patrick-LR/Gitproject01.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

Pull pour récuperer le projet

Lors d'un travail avec des collaborateurs, il faut qu'ils puissent rappatrier le projet. cela se fait avec les "CMD'
Ce connecter sur le dépôt à distance avec la "CMD" suivante :

- **get remote add https://github.com/Patrick-LR/Gitproject01.git**

Pour récupérer le dépôt faire la "CMD" suivante :

- **git pull origin master**

Donner les droits pour lire et écrire aux collaborateurs

Pour que les collaborateurs puissent dépôser dans le projet, il faut leurs donner les autorisations.

- Dans le "Repository" cliquez sur "**Manage access**"
- Cliquez sur le bouton "**Invite a collaborator**"
- Indiquez l'adresse email des personnes que vous voulez dans votre projet (il faut avoir un compte GitH

Backlinks: Home:Software

Wiki et thème réalisé par "RayGo Biker"
Utilisation de Zim 0.69.1