

 eBook Gratuit

# APPRENEZ

---

# github

eBook gratuit non affilié créé à partir des  
**contributeurs de Stack Overflow.**

#github

# Table des matières

<b>À propos</b>	<b>1</b>
<b>Chapitre 1: Démarrer avec github</b>	<b>2</b>
Remarques	2
Exemples	2
Installation ou configuration	2
<b>Créer un compte</b>	<b>2</b>
<b>Outils utiles</b>	<b>2</b>
<b>Créer votre premier référentiel</b>	<b>2</b>
En ligne	2
Hors ligne	3
Fichier README	3
<b>Un fichier README peut inclure-</b>	<b>3</b>
Titre du projet	3
Télécharger	3
Installation	3
Manifestation	3
Auteurs	4
Remerciements	4
Contributeur	4
Licence	4
Fichier de licence	4
GitHub Flavored Markdown	5
<b>Entête</b>	<b>5</b>
<b>Accentuation</b>	<b>6</b>
<b>Ligne horizontale</b>	<b>6</b>
<b>liste</b>	<b>7</b>
<b>Table</b>	<b>8</b>
<b>Code</b>	<b>8</b>
<b>Citation</b>	<b>9</b>

<b>Lien</b> .....	<b>9</b>
<b>Image</b> .....	<b>9</b>
<b>Listes de tâches</b> .....	<b>10</b>
<b>Emoji</b> .....	<b>10</b>
<b>Références SHA</b> .....	<b>10</b>
<b>Références sur les demandes de tirage et les problèmes</b> .....	<b>10</b>
<b>Chapitre 2: Affichage de la chronologie / des flux GitHub sur votre site Web</b> .....	<b>12</b>
<b>Exemples</b> .....	<b>12</b>
Affichage de la chronologie / des flux GitHub sur votre site Web.....	12
<b>Chapitre 3: Cloner un dépôt de GitHub</b> .....	<b>14</b>
<b>Syntaxe</b> .....	<b>14</b>
<b>Exemples</b> .....	<b>14</b>
Cloner un référentiel.....	14
<b>Chapitre 4: Comment créer des étiquettes GitHub personnalisées?</b> .....	<b>16</b>
<b>Exemples</b> .....	<b>16</b>
Créez des étiquettes GitHub personnalisées!.....	16
<b>Chapitre 5: GitHub Desktop</b> .....	<b>18</b>
<b>Introduction</b> .....	<b>18</b>
<b>Exemples</b> .....	<b>18</b>
Installation et configuration.....	18
Cloner un référentiel.....	18
Ramification.....	19
Poussez et tirez (ou: le bouton de synchronisation).....	20
<b>Chapitre 6: Mettre à jour un dépôt fourchu</b> .....	<b>21</b>
<b>Remarques</b> .....	<b>21</b>
<b>Exemples</b> .....	<b>21</b>
Configurez une télécommande pour votre fourche puis synchronisez votre fourche (branche pr.....	21
<b>Chapitre 7: Pages GitHub</b> .....	<b>22</b>
<b>Exemples</b> .....	<b>22</b>
Utilisation du générateur de page automatique pour un référentiel.....	22
Utiliser Git pour créer des pages à partir de zéro.....	22

Création d'une URL personnalisée pour votre page GitHub.....	22
Ressources.....	23
<b>Chapitre 8: Problèmes.....</b>	<b>24</b>
Exemples.....	24
Créer un problème.....	24
<b>Chapitre 9: Pull Demandes.....</b>	<b>25</b>
Exemples.....	25
Ouvrir une demande de tirage.....	25
Nouvelle demande de tirage.....	25
h21.....	26
<b>Chapitre 10: Sauvegarde de GitHub.....</b>	<b>28</b>
Exemples.....	28
Cloner tous les référentiels pour un nom d'utilisateur.....	28
<b>Chapitre 11: Suppression de données sensibles ou de fichiers volumineux.....</b>	<b>29</b>
Introduction.....	29
Remarques.....	29
Exemples.....	29
Utiliser une branche de filtre.....	29
Utilisation du nettoyeur repo BFG.....	30
Exigences.....	30
Supprimer des fichiers contenant des données sensibles.....	30
<b>Chapitre 12: télécharger un seul fichier depuis le dépôt GitHub.....</b>	<b>31</b>
Exemples.....	31
à partir d'un référentiel public en utilisant la ligne de commande et en renommant le fich.....	31
trouver l'url du fichier que vous voulez télécharger.....	31
<b>Chapitre 13: Travailler avec Gitflow.....</b>	<b>32</b>
Syntaxe.....	32
Paramètres.....	32
Remarques.....	32
Exemples.....	32
Opération sur 5 branches communes localement.....	32

<b>Chapitre 14: Utiliser Gist</b> .....	<b>34</b>
Introduction.....	34
Remarques.....	34
Exemples.....	34
Gist Public.....	35
Gist Secret.....	35
<b>Chapitre 15: Utiliser les boutons GitHub</b> .....	<b>36</b>
Introduction.....	36
Remarques.....	36
Exemples.....	36
Bouton Suivre.....	36
Tous les autres boutons.....	38
<b>Crédits</b> .....	<b>42</b>

---

# À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [github](#)

It is an unofficial and free github ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official github.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapitre 1: Démarrer avec github

## Remarques

Cette section fournit une vue d'ensemble de ce qu'est github et pourquoi un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets dans github, et établir un lien avec les sujets connexes. La documentation de github étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

## Exemples

### Installation ou configuration

GitHub est une énorme collection de dépôts Git. En d'autres termes, vous pouvez considérer GitHub comme un ensemble de projets!

---

## Créer un compte

- Visitez la page principale de GitHub [ici](#)
- Choisissez un nom d'utilisateur, entrez votre adresse e-mail, puis choisissez un mot de passe sécurisé et vous êtes prêt à partir!

---

## Outils utiles

Pour les débutants Git / GitHub, comprendre comment fonctionne le contrôle de version peut être déroutant au début. Il existe une version graphique de GitHub que vous pouvez télécharger et utiliser. [GitHub Desktop](#) est juste cet outil.

---

## Créer votre premier référentiel

Vous pouvez considérer un référentiel comme un projet. Vous pouvez créer un référentiel en ligne ou hors ligne. Suivez les étapes ci-dessous:

### En ligne

1. Connectez-vous d'abord et accédez à votre profil.
2. Accédez à l'onglet "Référentiels" en haut de la page.
3. Appuyez sur le bouton vert "Nouveau" et vous êtes prêt à gronder!

## Hors ligne

1. Téléchargez et installez [git](#) (choisissez le système d'exploitation que vous utilisez)
2. Après le téléchargement et l'installation, vous pouvez utiliser l'outil de ligne de commande ou télécharger un client d'interface graphique.
3. Après l'installation, créez un compte sur [github](#)
4. En haut à droite, cliquez sur + et choisissez soit de créer un nouveau référentiel, soit d'importer un existant.
5. Si vous en choisissez un nouveau, entrez le nom du référentiel et choisissez de le rendre public ou privé.
6. Cliquez sur: Créer un référentiel

NB Les dépôts privés ne sont pas disponibles pour les utilisateurs gratuits.

## Fichier README

Si votre projet ne contient pas README.md, GitHub peut analyser README.rdoc pour afficher les détails. S'il a les deux, il utilisera README.md, ignorant silencieusement rdoc.

---

## Un fichier README peut inclure-

### Titre du projet

Décrivez brièvement votre projet. Vous pouvez également fournir le lien vers le site Web du projet, les badges, la communauté et les informations de contact (par exemple, email, site social).

### Télécharger

Lien exécutable (exécutable ou minifié ou fichier d'installation). Il peut également y avoir des liens vers les versions précédentes.

### Installation

Comment votre travail peut être utilisé. Cela peut inclure les conditions préalables, les paramètres, les bibliothèques tierces, l'utilisation, les mises en garde, etc.

### Manifestation

Il peut s'agir d'un échantillon de code, d'un fichier gif, d'un lien vidéo ou même de captures d'écran.

## Auteurs

Noms d'auteur, coordonnées, etc.

## Remerciements

Liste de personnes ou de communautés aidées et inspirées tout au long du projet

## Contributeur

Instructions pour contribuer (c.-à-d. Ajouter une fonctionnalité, signaler un bogue, soumettre un correctif) au projet. Peut également inclure un lien de documentation.

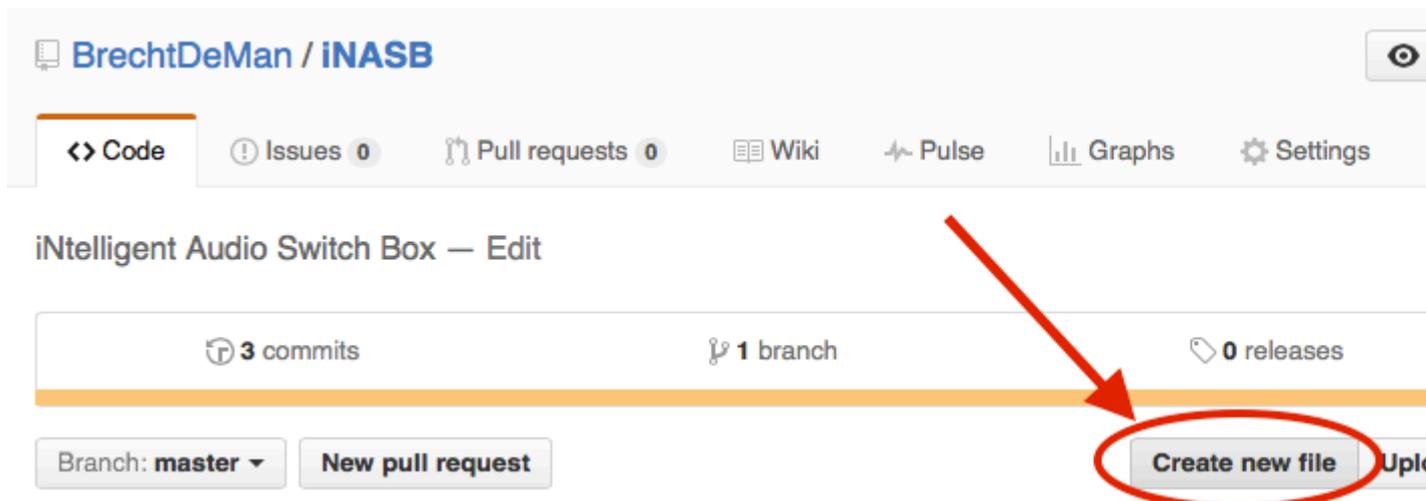
## Licence

Donnez une courte introduction sur votre licence. Vous pouvez également donner un lien vers le site de licence.

### Fichier de licence

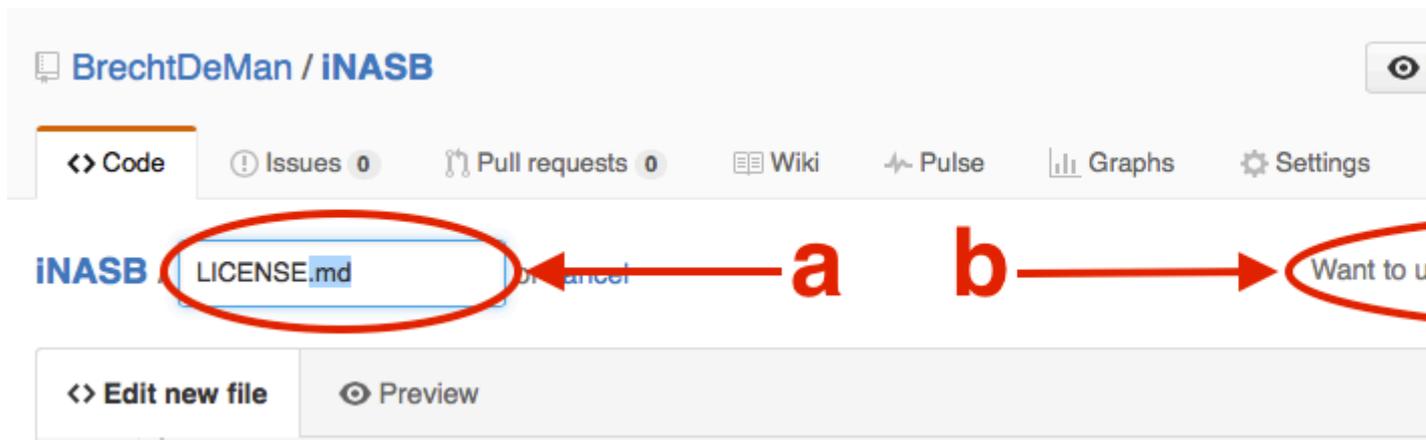
GitHub vous aide à ajouter rapidement une licence à votre référentiel, comme alternative à l'ajout de votre propre fichier texte / démarque.

1. Dans votre référentiel, cliquez sur "Créer un nouveau fichier"

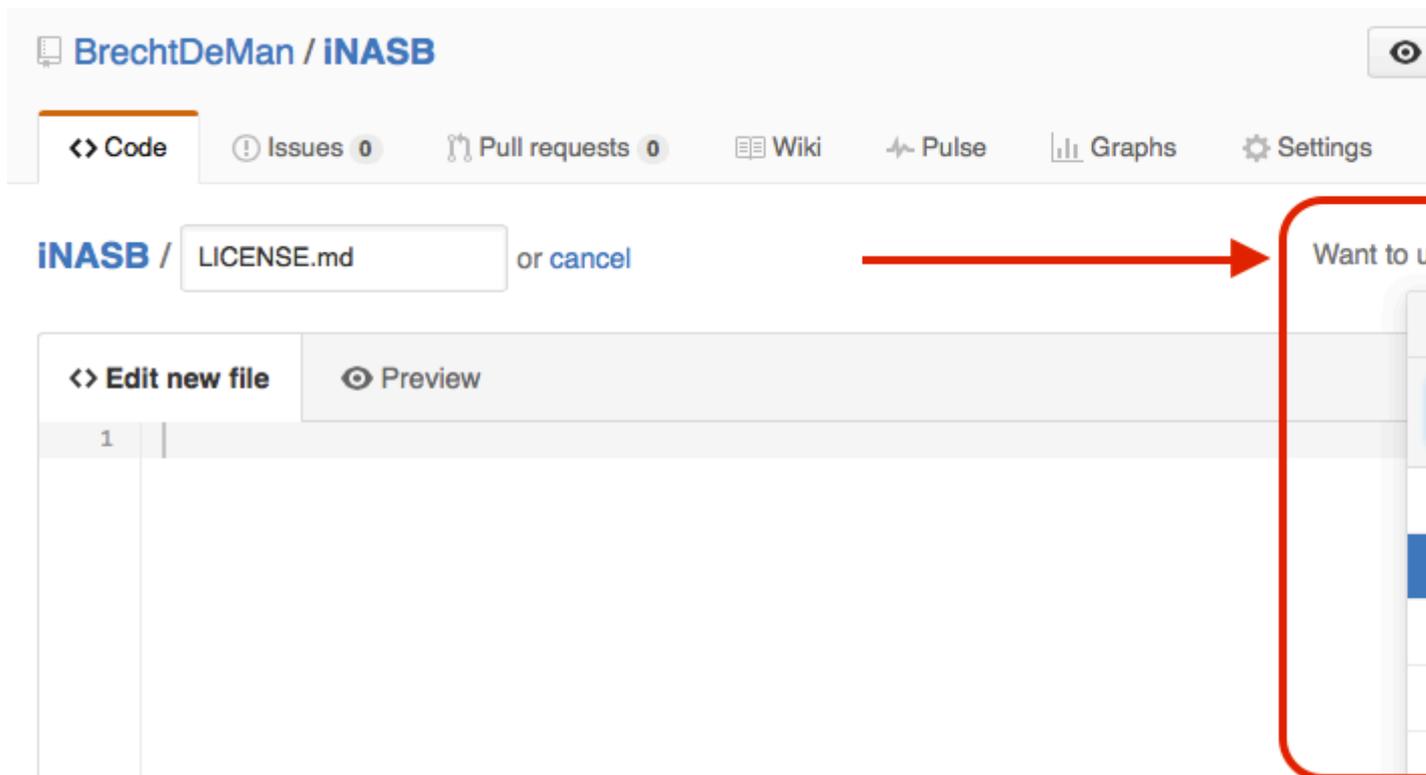


2. Sur la page suivante:

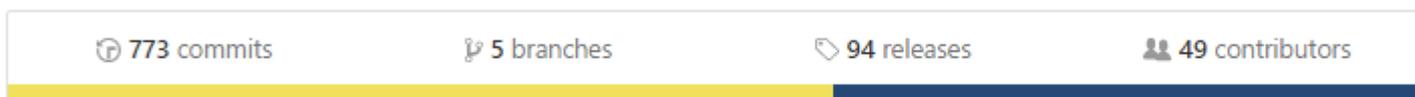
1. Tapez `LICENSE.md` ou `LICENSE.txt` comme nom de fichier du nouveau fichier.
2. Le *Voulez-vous utiliser un nouveau modèle?* la boîte de dialogue apparaîtra.



3. Choisissez votre licence préférée.



4. La licence que vous pouviez voir dans les détails du référentiel:



*De Q & A - Comment ajouter une licence à un projet Github existant*

## GitHub Flavored Markdown

GitHub étend la syntaxe de [Markdown](#) pour fournir de nouvelles fonctionnalités utiles.

## Entête

```
# Header1
## Header2
### Header3
#### Header4
##### Header5
##### Header6
H1
===
H2
---
```

# Header1

---

## Header2

---

### Header3

#### Header4

##### Header5

###### Header6

# H1

---

## H2

---

---

## Accentuation

```
*Italic1* _Italic2_  
**Bold1** __Bold2__  
***Bold_Italic***  
~~Strikethrough~~
```

*Italic1 Italic2*

**Bold1 Bold2**

***Bold\_Italic***

~~Strikethrough~~

# Ligne horizontale

```
---  
***  
---
```

---

## liste

unordered list:

```
* item-1  
  * sub-item-1  
  * sub-item-2  
- item-2  
  - sub-item-3  
  - sub-item-4  
+ item-3  
  + sub-item-5  
  + sub-item-6
```

ordered list:

```
1. item-1  
  1. sub-item-1  
  2. sub-item-2  
2. item-2  
  1. sub-item-3  
  2. sub-item-4  
3. item-3
```

unordered list:

- item-1
  - sub-item-1
  - sub-item-2
- item-2
  - sub-item-3
  - sub-item-4
- item-3
  - sub-item-5
  - sub-item-6

ordered list:

1. item-1
  - i. sub-item-1
  - ii. sub-item-2
2. item-2
  - i. sub-item-3
  - ii. sub-item-4
3. item-3

---

## Table

```
Table Header-1 | Table Header-2 | Table Header-3
:--- | :---: | ---:
Table Data-1 | Table Data-2 | Table Data-3
TD-4 | Td-5 | TD-6
Table Data-7 | Table Data-8 | Table Data-9
```

Table Header-1	Table Header-2	Table Header-3
Table Data-1	Table Data-2	Table Data-3
TD-4	Td-5	TD-6
Table Data-7	Table Data-8	Table Data-9

---

## Code

```
inline code- `int i=0`
```

```
block code-
```

```
``` C  
for(int i=0; i<10; i++){  
    printf("Hallow World! \n");  
}  
```
```

```
inline code- int i=0
```

```
block code-
```

```
for(int i=0; i<10; i++){  
    printf("Hallow World! \n");  
}
```

---

## Citation

```
> Stay hungry; stay foolish.  
>> Quality is better than quantity.  
>>> Life is not fair; get used to it.
```

```
| Stay hungry; stay foolish.
```

```
| | Quality is better than quantity.
```

```
| | | Life is not fair; get used to it.
```

---

## Lien

```
https://github.com  
[GitHub] (https://github.com)  
[GitHub] (https://github.com "github website")  
[GitHub] [1]
```

```
[1]: https://github.com
```

<https://github.com>

GitHub

GitHub

GitHub

# Image

```
![GitHub Logo] (https://assets-cdn.github.com/images/icons/emoji/octocat.png "GitHub")
```



# Listes de tâches

```
- [x] completed item  
- [ ] incomplete item
```

- completed item
- incomplete item

# Emoji

```
:octocat: :+1: :book: :ghost: :bulb: :imp:
```



Pour tous les emojis GitHub visitez - [Emoji Cheat Sheet](#) .

# Références SHA

Toute référence à un hash SHA1 d'un commit sera convertie en un lien vers le commit lui-même sur GitHub:

```
e7909ea4fbb162db3f7f543d43c30684a3fb745f
```

Write

Preview

[e7909ea](#)

# Références sur les demandes de tirage et les problèmes

Toute référence à une demande d'extraction ou à un problème sera automatiquement liée à cette requête ou à ce problème d'extraction.

Cela peut être fait en mettant un # devant le numéro de la question / demande de tirage.

Lire Démarrer avec github en ligne: <https://riptutorial.com/fr/github/topic/1214/demarrer-avec-github>

---

# Chapitre 2: Affichage de la chronologie / des flux GitHub sur votre site Web

## Exemples

### Affichage de la chronologie / des flux GitHub sur votre site Web

Ce document explique comment afficher vos flux / timeline GitHub sur votre site Web.

**Exemple:** Un exemple en direct est disponible à l'adresse suivante:

<https://newtonjoshua.com>

### Calendrier GitHub:

GitHub fournit la chronologie publique pour tout utilisateur au format Atom.

Vous pouvez voir votre calendrier à:

[https://github.com/ {{GitHub\\_username}}.atom](https://github.com/{{GitHub_username}}.atom)

référez-vous à: <https://developer.github.com/v3/activity/feeds>

### API de flux Google:

Grâce à l'API Feed, vous pouvez télécharger tout flux RSS Atom, RSS ou Media en utilisant uniquement JavaScript, afin de pouvoir combiner des flux avec votre contenu et d'autres API avec seulement quelques lignes de JavaScript. Cela facilite l'intégration rapide des flux sur votre site Web.

reportez-vous à: <https://developers.google.com/feed/v1/devguide>

**Chargement de l'API JavaScript:** Pour commencer à utiliser l'API de flux, incluez le script suivant dans l'en-tête de votre page Web.

```
<script type="text/javascript" src="https://www.google.com/jsapi"></script>
```

Ensuite, chargez l'API Feed avec `google.load (module, version, package)`.

```
<script type="text/javascript">
  google.load("feeds", "1");
</script>
```

**Spécification de l'URL du flux:** Vous pouvez appeler `google.feeds.Feed ()` comme suit:

```
var feed = new google.feeds.Feed("https://github.com/{{GitHub_UserName}}.atom");
```

**Chargement d'un flux:** `.load (callback)` télécharge le flux spécifié dans le constructeur depuis les serveurs de Google et appelle le rappel donné à la fin du téléchargement.

```
<script type="text/javascript">

function initialize() {
  feed.load(function(result) {
    if (!result.error) {
      var container = document.getElementById("feed");
      result.feed.entries.forEach(function (feed) {
        var feedTitle= feed.title;
        var feedLink = feed.link;
        var feedDate = formatDate(feed.publishedDate);
        var feedContent = formatContent(feed.content);

        // display the feed in your website
      });
    }
  });
}
google.setOnLoadCallback(initialize);

</script>
```

**L'appel du gestionnaire `onLoad`:** `setOnLoadCallback (callback)` est une fonction statique qui enregistre la fonction de gestionnaire spécifiée à appeler une fois que la page contenant cet appel est chargée, où `callback` est une fonction requise lorsque le document contenant est chargé et que l'API est prête à être utilisée

```
<script type="text/javascript">
  google.setOnLoadCallback(initialize);
</script>
```

**Définition du nombre d'entrées de fil:** `.setNumEntries (num)` définit le nombre d'entrées de fil chargées par ce fil sur `num`. Par défaut, la classe `Feed` charge quatre entrées.

```
var feed = new google.feeds.Feed("https://github.com/{{GitHub_UserName}}.atom");
feed.setNumEntries(500);
```

Maintenant, vous pouvez formater et afficher vos flux / timeline GitHub sur votre site Web.

**Lire Affichage de la chronologie / des flux GitHub sur votre site Web en ligne:**

<https://riptutorial.com/fr/github/topic/7479/affichage-de-la-chronologie---des-flux-github-sur-votre-site-web>

# Chapitre 3: Cloner un dépôt de GitHub

## Syntaxe

- `git clone github.com/username/repository`

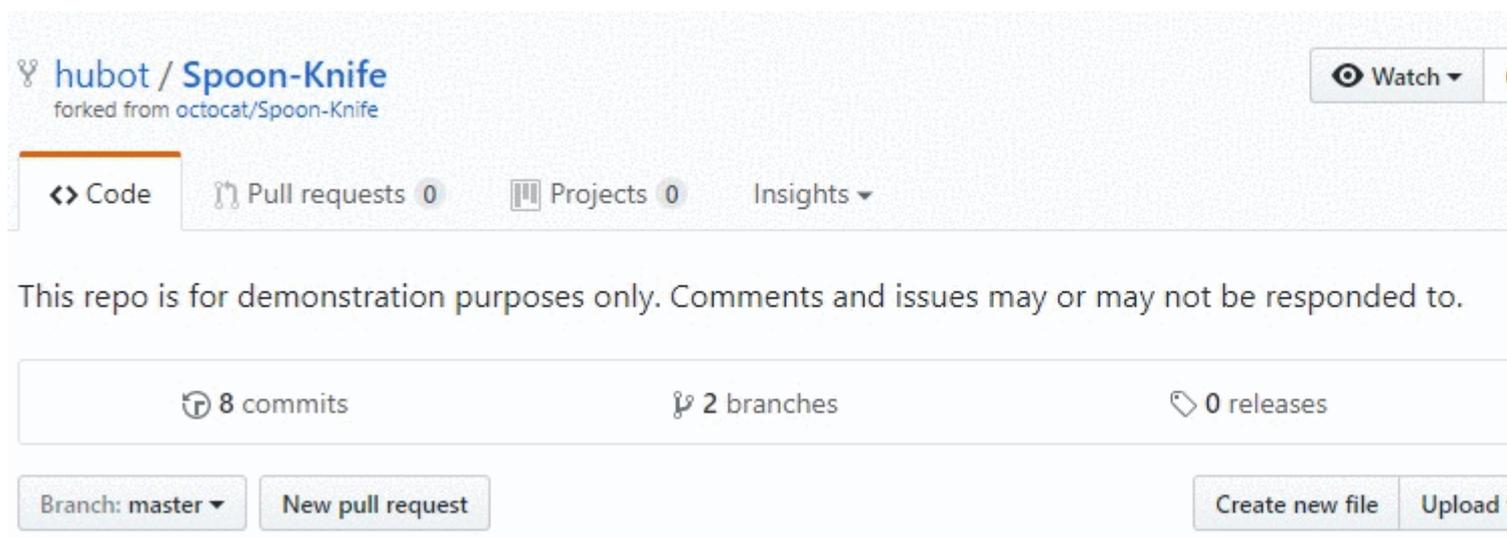
## Exemples

### Cloner un référentiel

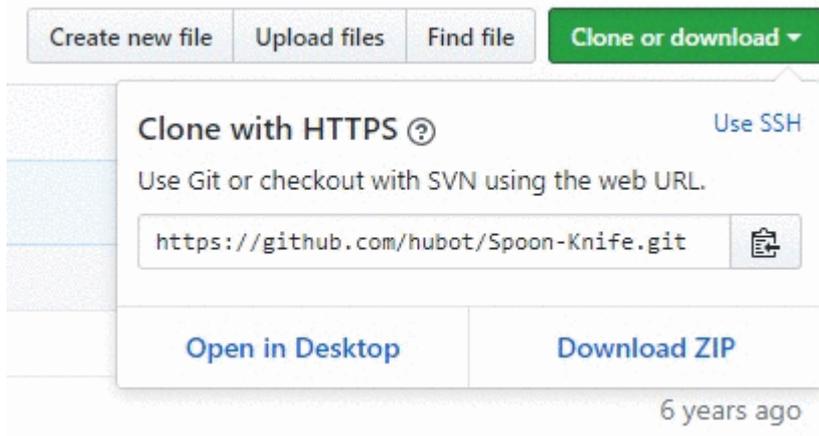
1. Accédez au référentiel que vous souhaitez cloner (quelque chose comme: [https://github.com/ username / repo](https://github.com/username/repo) )



2. Sur la droite, cliquez sur le bouton vert nommé *clone ou téléchargez*



3. Une petite fenêtre apparaîtra, copiez l'url (quelque chose comme: [https://github.com/ nom d'utilisateur / repo .git](https://github.com/nom d'utilisateur/repo.git))



4. Ouvrez une fenêtre de terminal sur la machine sur laquelle vous souhaitez cloner ce projet.
5. Naviguez de la ligne de commande à l'emplacement où vous souhaitez cloner le projet.
6. Entrez la commande: `git clone <copied_url_from_step_3>`
7. appuyez sur Entrée
8. Quelque chose comme ce qui suit apparaîtra:

```
<repo_name> dans <repo_name> ...
```

```
remote: Comptage d'objets: 10, fait.
```

```
remote: Compression d'objets: 100% (8/8), terminé.
```

```
supprimer: Total 10 (delta 1), réutilisé 10 (delta 1)
```

```
Déballage d'objets: 100% (10/10), terminé.
```

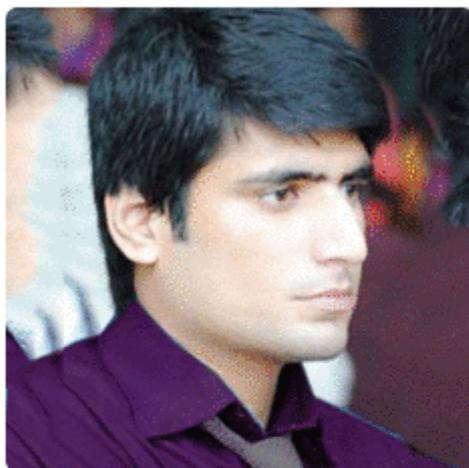
Lire Cloner un dépôt de GitHub en ligne: <https://riptutorial.com/fr/github/topic/3761/cloner-un-depot-de-github>

# Chapitre 4: Comment créer des étiquettes GitHub personnalisées?

## Exemples

Créez des étiquettes GitHub personnalisées!

Voici un GIF rapide pour rendre le processus aussi simple que possible.



### Ahmad Awais

ahmadawais

Full Stack WordPress Dev — Front-end Fanatic — WP Core Contributor — TEDx Speaker — Open Sourcerer! <sup>100</sup>

Edit profile

Developer Program Member

@WPTie / WordPress

WP-Admin, TRAC; CORE

Overview Repositories 262 Stars 1.1k Followers 187

#### Pinned repositories

##### WPGulp

<sup>100</sup> % <sup>W</sup> → Use Gulp with WordPress. An advanced but portable Gulp front end and build workflow for you WordPress plugins and themes.

★ 203 ● JavaScript

##### \_child

\_child is a WordPress

★ 32 ● PHP

##### WPCustomize

WP Customize component related boilerplate theme and features implementation.

★ 25 ● PHP

##### Sublime-WP-C

Sublime Package

★ 22 ● PHP

##### WP-API/WP-API

WP REST API - a JSON-based REST API for WordPress.

★ 3,565 ● PHP

##### WordPress/tw

Twenty Sixteen is a WordPress layout right sidebar that w has custom color o

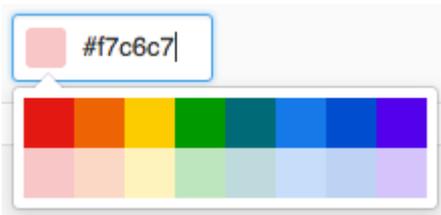
★ 340 ● CSS

Les étiquettes peuvent être appliquées aux problèmes et extraire des requêtes pour indiquer la priorité, la catégorie ou toute autre information que vous trouvez utile.

Sur GitHub, accédez à la page principale du référentiel.

1. Sous le nom de votre référentiel, cliquez sur Problèmes ou requêtes d'extraction.
2. Étiquettes de problèmes boutonNext vers le champ de recherche, cliquez sur Étiquettes.

3. Cliquez sur Nouvelle étiquette pour créer une nouvelle étiquette ou cliquez sur Modifier pour en modifier une existante.
4. Dans la zone de texte, tapez votre nouveau nom d'étiquette.
5. Sélectionnez une couleur pour l'étiquette dans la barre de couleurs. Vous pouvez personnaliser cette couleur en modifiant le nombre hexadécimal au-dessus de la barre de couleur.



6. Cliquez sur Créer une étiquette pour enregistrer la nouvelle étiquette.

J'espère que ça aide. Notez-le si c'est le cas.

Lire [Comment créer des étiquettes GitHub personnalisées? en ligne:](https://riptutorial.com/fr/github/topic/7159/comment-creeer-des-etiquettes-github-personnalisees-)

<https://riptutorial.com/fr/github/topic/7159/comment-creeer-des-etiquettes-github-personnalisees->

---

# Chapitre 5: GitHub Desktop

## Introduction

Comment installer et travailler avec GitHub Desktop?

GitHub Desktop est, comme son nom l'indique, un environnement de bureau pour Windows et MacOS qui inclut les principales fonctionnalités de Git telles que le clonage, le pushing, le pulling (synchronisation dans GitHub Desktop), la fusion ...

Le but principal des clients de bureau est de fournir un moyen plus simple de travailler avec git (et GitHub). En arrière-plan, il utilise les mêmes commandes que la plupart des utilisateurs utilisent depuis la ligne de commande.

## Exemples

### Installation et configuration

L'installation est assez simple car il y a des installateurs séparés pour les machines MacOS et Windows disponibles [ici](#) . Actuellement, deux versions sont à télécharger: une version bêta et une version stable.

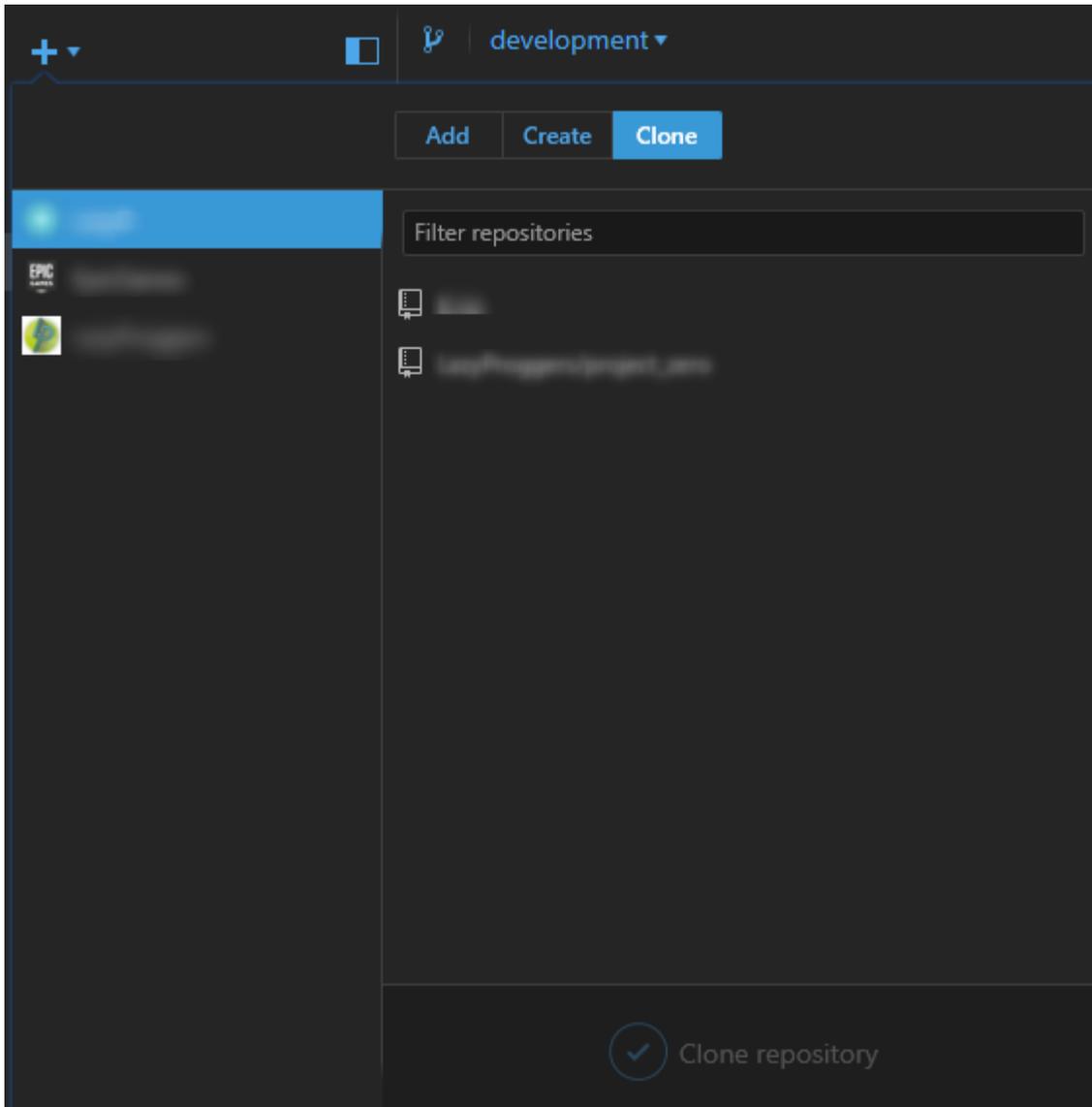
Le programme d'installation démarrera après avoir téléchargé le programme et vous devrez vous connecter avec vos informations d'identification GitHub. C'est vraiment la seule étape car après cela, vous pouvez commencer à créer un référentiel ou un clone.

Remarque: lors de l'installation, non seulement GitHub Desktop sera installé, mais aussi Git. Vous n'avez donc pas besoin de l'installer séparément.

### Cloner un référentiel

Comme c'est le cas avec GitHub Desktop, la majeure partie du travail est assez simple: vous sélectionnez "Cloner un référentiel" (dans la version stable, le plus en haut à gauche) et il existe des référentiels (le vôtre et le repo de chaque entreprise ) conseillé. Vous pouvez également coller un lien vers un autre référentiel que vous souhaitez peut-être cloner.

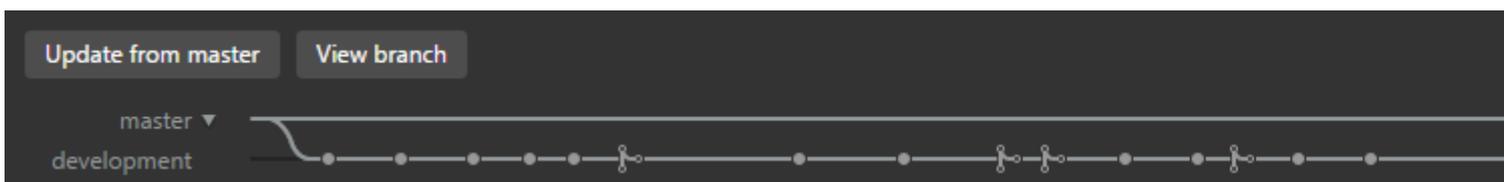
Remarque: dans la version la plus récente (version bêta), il n'y a pas de recommandations (pas de jet?).



## Ramification

Vous pouvez sélectionner une branche en haut à gauche. Lorsque vous avez sélectionné la bonne branche, vous devez appuyer sur le bouton de synchronisation (en haut à droite), qui fait maintenant la même chose que la `git checkout BRANCHNAME`.

Dans l'ancienne version, vous pouvez visualiser 2 branches différentes à la fois et comparer les poussées. De plus, vous pouvez voir une chronologie de votre projet (voir ci-dessous)



## Créer une nouvelle branche

Vous pouvez créer une nouvelle branche en cliquant sur le symbole de la branche (ancien client) OU SOUS `File --> New Branch`.

Notez que vous pouvez sélectionner la branche utilisée par la nouvelle branche en cliquant sur le nom de la branche.

## Poussez et tirez (ou: le bouton de synchronisation)

### Tirez (Sync)

Comme dans la ligne de commande, vous devez parfois retirer l'état actuel du référentiel. Dans GitHub Desktop, ce processus est appelé par le bouton de `sync` dans le coin supérieur droit.

### Pousser

Lorsque vous apportez des modifications locales et que vous voulez les pousser, vous faites un commit en écrivant quelque chose dans la zone de texte du résumé. Ensuite, appuyez sur `Commit to YOURCURRENTBRANCH`. Maintenant, vous devez appuyer sur le bouton de synchronisation et votre poussée est faite.

Remarque: vous pouvez utiliser des émoticônes, des mentions et des références à d'autres validations ou problèmes directement à partir de la zone de texte.

Ainsi, le bouton **Sync** peut être utilisé pour `Push`, `Pull` ou `Checkout`.

Lire GitHub Desktop en ligne: <https://riptutorial.com/fr/github/topic/10023/github-desktop>

---

# Chapitre 6: Mettre à jour un dépôt fourchu

## Remarques

- [GitHub Help: Configuration d'une télécommande pour un fork](#)
- [GitHub Help: Synchronisation d'un fork](#)
- [populaire dans StackOverFlow](#)

## Exemples

Configurez une télécommande pour votre fourche puis synchronisez votre fourche (branche principale)

### 1. Configurer une télécommande pour ma fourchette

```
$ cd my_local_repo

$ git remote add upstream https://github.com/ORIGINAL_OWNER/ORIGINAL_REPOSITORY.git
  # Specify a new remote upstream repository that will be synced with the fork

$ git remote -v
  # Verify the new upstream repository specified for my fork
```

### 2. Synchroniser ma fourche localement

```
$ cd my_local_repo

$ git fetch upstream
  # Fetch the branches and their respective commits from the upstream repository
  # Commits to master will be stored in a local branch, upstream/master

$ git checkout master

$ git merge upstream/master
  # Merge the changes from upstream/master into your local master branch
  # brings your fork's master branch into sync with the upstream repo
```

### 3. Synchronise ma fourche sur Github

```
$ git push origin master
```

Lire [Mettre à jour un dépôt fourchu en ligne](https://riptutorial.com/fr/github/topic/3758/mettre-a-jour-un-depot-fourchu): <https://riptutorial.com/fr/github/topic/3758/mettre-a-jour-un-depot-fourchu>

---

# Chapitre 7: Pages GitHub

## Exemples

### Utilisation du générateur de page automatique pour un référentiel

1. Allez sur le site GitHub
2. Ouvrez votre référentiel
3. Cliquez sur Paramètres
4. Sous GitHub Pages, cliquez sur "Lancer le générateur de page automatique"
5. Suivez les instructions

### Utiliser Git pour créer des pages à partir de zéro

1. Créez un nouveau référentiel ou clonez-en un existant.
2. Créer une nouvelle branche appelée `gh-pages` sans historique

```
$ git checkout --orphan gh-pages

# ensure you are in the correct directory then,
# remove all files from the old working tree
$ git rm -rf
```

3. Ajoutez un fichier `index.html` à la racine du référentiel.

```
$ echo "Hello World" > index.html
$ git add index.html
$ git commit -a -m "First pages commit"
```

4. Poussez vers Github.

```
$ git push origin gh-pages
```

Vous pouvez maintenant charger votre nouveau site Github Pages en

`http(s)://<username>.github.io/<projectname>`

### Création d'une URL personnalisée pour votre page GitHub

Vous aurez besoin d'un nom de domaine d'un [bureau d'enregistrement](#) .

Dans la branche `gh-pages` de votre référentiel de projet ou dans la branche principale de votre référentiel `username.github.io` , créez un fichier CNAME avec le contenu `www.yourdomain.com` - le [domaine canonique](#) .

Sur la page de configuration du domaine de votre registraire, pointez votre domaine vers votre site Web GitHub. Configurez deux enregistrements CNAME (un pour l'apex racine (@) et un pour `www`). Les deux devraient indiquer `username.github.io` ou `username.github.io/repository` . Si votre

fournisseur DNS ne prend pas en charge les enregistrements ALIAS sur l'apex racine (@), créez simplement des enregistrements A qui pointent vers 192.30.252.153 et 192.30.252.154.

---

## Ressources

[Instructions GitHub pour un domaine personnalisé](#)

[Stack Overflow Q & A: "Domaine personnalisé pour les pages de projet GitHub"](#)

[Audrey Watters - Utiliser GitHub pour alimenter un projet Web: comment et pourquoi](#)

[Alex Cican - Comment j'ai déplacé mes sites Web vers Dropbox et GitHub](#)

[Treehouse - Utiliser les pages GitHub pour héberger votre site Web](#)

[Lire Pages GitHub en ligne: https://riptutorial.com/fr/github/topic/3759/pages-github](https://riptutorial.com/fr/github/topic/3759/pages-github)

---

# Chapitre 8: Problèmes

## Exemples

### Créer un problème

1. Accédez à la page GitHub du projet dans lequel vous souhaitez créer un problème.
2. Cliquez sur **Problèmes** .
3. En haut à droite, cliquez sur **Nouveau numéro** .
4. Entrez le titre du numéro.
5. Entrez le corps du problème (y compris les journaux, les extraits de code, etc.)
6. *Facultatif*: pour afficher le problème avant de le soumettre, cliquez sur Aperçu.
7. Cliquez sur **Submit new issue** .

Lire Problèmes en ligne: <https://riptutorial.com/fr/github/topic/3757/problemes>

---

# Chapitre 9: Pull Demandes

## Exemples

Ouvrir une demande de tirage

### Nouvelle demande de tirage

Chaque fois que vous voulez créer une requête Pull (disons que ceci est dû soit à un changement récent, mais vous pouvez aussi le faire avec un ancien changement!), Vous pouvez continuer et laisser GitHub faire le gros du travail pour vous. et appuyez sur le bouton vert de **comparaison et de demande de retrait** (à ne pas confondre avec *Clone* ou *TELECHARGEMENT*) dans la case d'alerte mentionnant que vous venez de pousser dans une branche.

Sinon, vous utiliserez le bouton **Nouvelle requête de tirage** situé à côté de votre succursale.



This repository

Search



maxcell / **example-so-documenta**

**<> Code**

! Issues **0**

🔗 Pull request

Repository used for Documentation in Sta

🔄 **2 commits**

Your recently pushed branches:

🔗 **example-branch** (less than a minute ago)

Branch: **example-branch** ▼

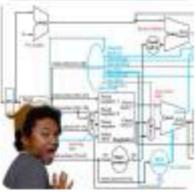
**New pull request**

This branch is 1 commit ahead of master.



**maxcell** committed on **GitHub** Update README

📄 **README.md**



## Update README.md

Write

Preview

Leave a comment

Attach files by dragging & dropping, [selecting](#)

**M↓** Styling with Markdown is supported

Lire Pull Demandes en ligne: <https://riptutorial.com/fr/github/topic/5761/pull-demandes>

---

# Chapitre 10: Sauvegarde de GitHub

## Exemples

### Cloner tous les référentiels pour un nom d'utilisateur

Exécutez la commande suivante, en remplaçant le nom d'utilisateur par le nom d'utilisateur, pour cloner tous les référentiels GitHub de cet utilisateur dans le répertoire en cours.

```
curl "https://api.github.com/users/username/repos?page=1&per_page=100" | grep -e 'git_url*' |  
cut -d \" -f 4 | xargs -L1 git clone
```

Cela ne fera que cloner les 100 premiers dépôts.

Lire Sauvegarde de GitHub en ligne: <https://riptutorial.com/fr/github/topic/3760/sauvegarde-de-github>

# Chapitre 11: Suppression de données sensibles ou de fichiers volumineux

## Introduction

Si vous validez des données sensibles, telles qu'un mot de passe ou une clé SSH dans un référentiel Git, vous pouvez le supprimer de l'historique. Pour supprimer entièrement les fichiers indésirables de l'historique d'un référentiel, vous pouvez utiliser la commande `git filter-branch` ou `BFG Repo-Cleaner`.

## Remarques

1. Indiquez à vos collaborateurs de rebaser, et non de fusionner, les branches qu'ils ont créées à partir de votre historique de référentiel (corrompu). Un engagement de fusion pourrait réintroduire tout ou partie de l'histoire viciée que vous venez d'évacuer.
2. Après un certain temps et si vous êtes sûr que `git filter-branch` n'a pas eu d'effets secondaires inattendus, vous pouvez forcer le déréréférencement de tous les objets de votre référentiel local et la récupération des données avec les commandes suivantes (en utilisant Git 1.8.5 ou plus récent):

```
git for-each-ref --format = 'supprimer% (refname)' refs / original | git update-ref --stdin
```

```
git reflog expire --expire = now --all
```

```
git gc --prune = maintenant
```

## Exemples

### Utiliser une branche de filtre

```
git filter-branch --force --index-filter \  
'git rm --cached --ignore-unmatch PATH-TO-YOUR-FILE-WITH-SENSITIVE-DATA' \  
--prune-empty --tag-name-filter cat -- --all
```

Ajoutez votre fichier avec des données sensibles à `.gitignore` pour vous assurer de ne pas le commettre accidentellement.

```
echo "YOUR-FILE-WITH-SENSITIVE-DATA" >> .gitignore  
git add .gitignore  
git commit -m "Add YOUR-FILE-WITH-SENSITIVE-DATA to .gitignore"
```

Poussez votre repo local sur GitHub

```
git push origin --force --all
```

Afin de supprimer le fichier sensible de vos versions étiquetées, vous devrez également pousser de force contre vos tags Git:

```
git push origin --force --tags
```

## Utilisation du nettoyeur repo BFG

BFG Repo cleaner est une alternative à git filter-branch. Il peut être utilisé pour supprimer des données sensibles ou des fichiers volumineux qui ont été commis à tort, comme des fichiers binaires compilés à partir de la source. Il est écrit en Scala.

Site web du projet: [BFG Repo Cleaner](#)

## Exigences

L'environnement d'exécution Java (Java 7 ou supérieur - BFG v1.12.3 était la dernière version à prendre en charge Java 6). La bibliothèque Scala et toutes les autres dépendances sont intégrées dans le fichier jar.

## Supprimer des fichiers contenant des données sensibles

```
bfg --delete-files YOUR-FILE-WITH-SENSITIVE-DATA
```

Lire [Suppression de données sensibles ou de fichiers volumineux en ligne](#):

<https://riptutorial.com/fr/github/topic/8170/suppression-de-donnees-sensibles-ou-de-fichiers-volumineux>

---

# Chapitre 12: télécharger un seul fichier depuis le dépôt GitHub

## Exemples

à partir d'un référentiel public en utilisant la ligne de commande et en renommant le fichier

Cet exemple récupère le fichier Node.gitignore du dépôt gitignore de GitHub, le télécharge dans votre répertoire de travail actuel et le renomme en .gitignore - toutes des actions très courantes pour quelqu'un qui lance un nouveau projet node.js.

```
$ curl http://github.com/github/gitignore/raw/master/Node.gitignore -o .gitignore
```

### trouver l'url du fichier que vous voulez télécharger

1. naviguer dans le fichier souhaité dans un référentiel
2. cliquez sur le bouton "brut"
3. copier l'URL de la barre d'adresse

Voir l'exemple suivant du dépôt gitignore de GitHub:

<http://github.com/github/gitignore/raw/master/Node.gitignore>

Vous pouvez rapidement reconnaître une URL qui fonctionnera pour télécharger un fichier individuel contre le téléchargement de la page HTML. Recherchez le sous-répertoire / raw / right avant le nom de la branche.

Lire télécharger un seul fichier depuis le dépôt GitHub en ligne:

<https://riptutorial.com/fr/github/topic/10898/telecharger-un-seul-fichier-depuis-le-depot-github>

# Chapitre 13: Travailler avec Gitflow

## Syntaxe

- git flow <sous-commande>
- git flow init
- git flow [fonctionnalité | version | correctif] [début | fin]

## Paramètres

| Sous-commande  | Détails   |
|----------------|---|
| init           | Initialiser un nouveau dépôt git avec prise en charge du modèle de branchement. |
| fonctionnalité | Gérez vos branches de fonctionnalités.  |
| Libération     | Gérez vos branches de publication.  |
| correctif      | Gérez vos branches de correctifs.   |

## Remarques

- [concept gitflow de l'auteur](#)
- [photo de modèle de branche](#)

## Exemples

### Opération sur 5 branches communes localement

Un des cas d'utilisation les plus courants de Gitflow

#### 1. Initialiser repo et définir des branches

```
$ git flow init
# if you use default setup, you'll define six types of branches:
#
# main branches (lives forever)
#
# 1. master:  for production releases
# 2. develop: for "next release" development
#
# supporting branches
#
# 3. feature: for a product feature
```

```
# 4. release: for preparation of a new production release
# 5. hotfix: for resolving critical bug of production version
# 6. support
#
# also, two main branches are created: master, develop
```

## 2. Commencer et terminer une fonctionnalité

```
$ git flow feature start my_feature
# create branch 'feature/my_feature' based on the 'develop'

# made development and commits...

$ git flow feature finish my_feature
# merge 'feature/my_feature' back to the 'develop'
# delete 'feature/my_feature'
```

## 3. Démarrer et terminer une version

```
$ git flow release start my_release
# create branch 'release/my_release' based on the 'develop'

# made bug fixes...

$ git flow release finish my_release
# merge branch 'release/my_release' to the 'master' and add tag
# merge branch 'release/my_release' back to the 'develop'
# delete 'release/my_release'
```

## 4. Démarrer et terminer un correctif

```
$ git flow hotfix start my_hotfix
# create branch 'hotfix/my_hotfix' based on the 'master'

# made some hotfixes...

$ git flow hotfix finish my_hotfix
# merge branch 'hotfix/my_hotfix' back to the 'master' and add tag
# merge branch 'hotfix/my_hotfix' to the 'develop'
# delete 'hotfix/my_hotfix'
```

Lire Travailler avec Gitflow en ligne: <https://riptutorial.com/fr/github/topic/6231/travailler-avec-gitflow>

---

# Chapitre 14: Utiliser Gist

## Introduction

Les listes sont un excellent moyen de partager votre travail. Vous pouvez partager des fichiers uniques, des parties de fichiers ou des applications complètes. Vous pouvez accéder aux gists sur <https://gist.github.com>.

Chaque liste est un dépôt Git, ce qui signifie qu'elle peut être créée et clonée. L'éditeur gist est alimenté par CodeMirror.

Il y a deux types de liste: les listes publiques et les listes secrètes.

De plus, si vous n'êtes pas connecté à GitHub lorsque vous créez votre liste, ce sera un élément anonyme.

## Remarques

Les listes sont un excellent moyen de partager votre travail. Vous pouvez partager des fichiers uniques, des parties de fichiers ou des applications complètes.

Il y a deux types de liste: les listes publiques et les listes secrètes. De plus, si vous n'êtes pas connecté à GitHub lorsque vous créez votre liste, ce sera un élément anonyme.

### Listes publiques

Les listes publiques s'affichent dans Discover, où les internautes peuvent parcourir de nouvelles listes au fur et à mesure de leur création. Ils sont également consultables, vous pouvez donc les utiliser si vous souhaitez que d'autres personnes trouvent et voient votre travail.

### Gists Secret

Les mots secrets ne s'affichent pas dans Discover et ne sont pas consultables. Utilisez-les pour noter une idée qui vous est venue en rêve, créer une liste de tâches ou préparer un code ou une prose qui ne soit pas prêt à être partagé avec le monde.

Vous pouvez créer autant d'éléments secrets que vous le souhaitez.

### Gists Anonymes

Si vous créez un aperçu sans vous connecter à GitHub, ce sera un élément anonyme. Les listes anonymes peuvent être publiques ou secrètes. Pour supprimer un aperçu anonyme sur GitHub.com ou GitHub Enterprise, contactez le support GitHub ou votre administrateur de site, respectivement. Veuillez fournir l'URL de l'essentiel que vous souhaitez supprimer.

## Exemples

## Gist Public

Un message public peut être *presque* n'importe quoi.

Un exemple simple de fonction Javascript:

```
function randomInt(min, max) {  
  return Math.floor((max - min + 1) * Math.random()) + min;  
}
```

## Gist Secret

Une liste secrète devrait être utilisée pour tout ce que vous ne voulez pas voir publiquement sur GitHub. Les clés secrètes peuvent être utilisées lorsque vous ne voulez pas que les clés privées soient accessibles au public, ou pour le code privé en général.

Un exemple simple de code JSON qui conviendrait mieux à un aperçu secret:

```
{  
  "id": AKIAIOSFODNN7EXAMPLE,  
  "secret": wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY  
}
```

Lire Utiliser Gist en ligne: <https://riptutorial.com/fr/github/topic/7978/utiliser-gist>

# Chapitre 15: Utiliser les boutons GitHub

## Introduction

Quels sont les boutons GitHub? Les boutons GitHub sont des boutons que vous pouvez ajouter à votre site Web pour rediriger les utilisateurs vers n'importe quel référentiel que vous aimez!

## Remarques

Crédits:

- Images [GIF](#) enregistrées avec [Recordit](#)
- Images statiques prises avec Snipping Tool
- L'éditeur de code utilisé dans les didacticiels complets était [codepen.io](#)

## Exemples

### Bouton Suivre

Un bouton de suivi est un bouton qui relie à une page utilisateur GitHub et invite l'utilisateur à suivre l'utilisateur. Voici comment en créer un:

1. Allez sur [github: boutons](#)
2. Cliquez sur "Suivre"

### Choose a button



3. Placez votre nom d'utilisateur GitHub dans la case ": user"

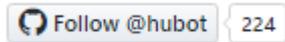
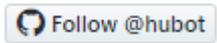
### Button options

 / 

- Large button
- Show count
- Standard icon

4. Personnalisez le bouton en utilisant les cases "Grand bouton", "Afficher le nombre" et "Icône"

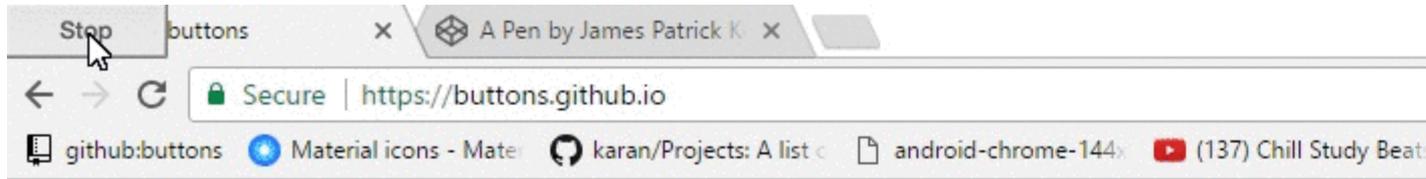
standard":



5. Placez ce code dans le `<head>` ou avant la fin du `<body>` de votre code:

```
<a class="github-button" href="https://github.com/hubot" aria-label="Follow @hubot on GitHub">Follow @hubot</a>
```

6. Placez le code de rendu du bouton personnalisé dans votre code.



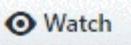
## GitHub:buttons

### Choose a button

Follow



Watch

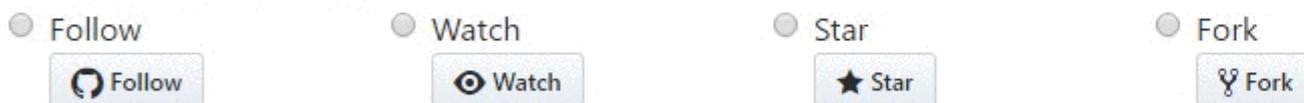


- **Star** a repository
- **Fourche** un dépôt
- **Télécharger** un référentiel
- Répertoire un **problème** avec un référentiel

Voici comment en créer:

1. Allez sur [github: boutons](#)
2. Cliquez sur le type de bouton que vous souhaitez créer (Watch, Star, Fork, Download ou Issue)

## Choose a button



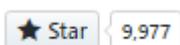
3. Placez votre nom d'utilisateur GitHub dans la case ": user" et votre référentiel dans la case ": repo"

## Button options

 / 

- Large button
- Show count
- Standard icon

4. Personnalisez le bouton en utilisant les cases "Grand bouton", "Afficher le nombre" et "Icône standard":



5. Placez ce code dans le `<head>` ou avant la fin du `<body>` de votre code:

```
<a class="github-button" href="https://github.com/hubot" aria-label="Follow @hubot on GitHub">Follow @hubot</a>
```

6. Placez le code de rendu du bouton personnalisé dans votre code.

# GitHub:buttons

## Choose a button

- Follow
- Watch

<https://riptutorial.com/fr/github/topic/10585/utiliser-les-boutons-github>

# Crédits

| S. No | Chapitres  | Contributeurs  |
|-------|--|--|
| 1     | Démarrer avec github   | <a href="#">BadAllOff</a> , <a href="#">BrechtDeMan</a> , <a href="#">Community</a> , <a href="#">H. Pauwelyn</a> , <a href="#">Hamzawey</a> , <a href="#">Hugo</a> , <a href="#">intboolstring</a> , <a href="#">Kronos</a> , <a href="#">Mateusz Piotrowski</a> , <a href="#">Minhas Kamal</a> , <a href="#">Nicholas Qiao</a> , <a href="#">rpadovani</a> |
| 2     | Affichage de la chronologie / des flux GitHub sur votre site Web | <a href="#">Hugo</a> , <a href="#">Newton Joshua</a>   |
| 3     | Cloner un dépôt de GitHub  | <a href="#">demonplus</a> , <a href="#">geek1011</a> , <a href="#">Hamzawey</a> , <a href="#">James Kerrane</a> , <a href="#">Mateusz Piotrowski</a>   |
| 4     | Comment créer des étiquettes GitHub personnalisées?              | <a href="#">Ahmad Awais</a>  |
| 5     | GitHub Desktop   | <a href="#">creyD</a>  |
| 6     | Mettre à jour un dépôt fourchu                                   | <a href="#">Derek Liu</a>  |
| 7     | Pages GitHub   | <a href="#">BrechtDeMan</a> , <a href="#">geek1011</a> , <a href="#">Mono</a>  |
| 8     | Problèmes  | <a href="#">geek1011</a> , <a href="#">Hamzawey</a> , <a href="#">SuperBiasedMan</a>   |
| 9     | Pull Demandes  | <a href="#">Maxcell</a>  |
| 10    | Sauvegarde de GitHub   | <a href="#">geek1011</a>   |
| 11    | Suppression de données sensibles ou de fichiers volumineux       | <a href="#">Gautam Krishna R</a> , <a href="#">Kronos</a>  |
| 12    | télécharger un seul fichier depuis le dépôt GitHub               | <a href="#">ownsourcing dev training</a>   |
| 13    | Travailler avec Gitflow  | <a href="#">Derek Liu</a>  |
| 14    | Utiliser Gist  | <a href="#">Kronos</a> , <a href="#">tehp</a>  |

|    |                                |                               |
|----|--------------------------------|-------------------------------|
| 15 | Utiliser les boutons<br>GitHub | <a href="#">James Kerrane</a> |
|----|--------------------------------|-------------------------------|