



République Algérienne Démocratique et Populaire



Ministère d'Enseignement Supérieur et de la Recherche Scientifique
Université ABBAS LAGHROUR- Khenchela- département MI

Mémoire

Présenté en vue d'obtenir le diplôme de

Master en Informatique (LMD)

**Publication d'articles scientifiques en *Open Source*
via GitHub**

Présenté Par : DjehicheDhirar

Encadrer Par :Dr MEBARKI DJEMOUAI

Année universitaire :2020/2021

Résumé

GitHub est devenu une plate-forme d'applications sociales populaire, où un grand nombre d'utilisateurs publient leurs projets open source. En particulier, un nombre croissant de chercheurs publient des référentiels de sources code lié à leurs travaux de recherche afin d'attirer plus de personnes de suivre leur travail [1]. Motivés par cette tendance, Notre travail consiste donc à étudier et prototyper un usage de la plate-forme GitHub pour publier les articles scientifiques. En s'appuyant sur le moteur Jekyll supporté par GitHub pages et des extensions JavaScript à développer, il s'agit notamment d'offrir la possibilité de publier en ligne des articles à contenu à caractère scientifique. Outre l'optimisation du référencement des articles publiés, l'objectif de ce projet est de mettre un œuvre des plugins permettant de commenter les articles (en lien avec les issues de GitHub), les approuver, mais aussi de cloner un article et de proposer des modifications (comme des pullrequests dans GitHub).

Mot Clé : GitHub, prototyper, Jekyll,publier,articles,Référencements, plugins,cloner,pullrequests.

Abstract

GitHub has become a popular social application platform, where a large number of users publish their open source projects. In particular, an increasing number of researchers are publishing source code repositories related to their research work in order to attract more people to follow their work [1]. Motivated by this trend, our work therefore consists in studying and prototyping a use of the GitHub platform to publish scientific papers. Based on the Jekyll engine supported by GitHub pages and JavaScript extensions to be developed, it is in particular to offer the possibility to publish online articles with scientific content. In addition to optimizing the referencing of published articles, the objective of this project is to implement plugins allowing to comment the articles (in connection with the issues of GitHub), to approve them, but also to clone an article and to propose modifications (as pull requests in GitHub).

Keyword: GitHub, prototype, Jekyll, publish, articles, referencing, plugins, clone, pull requests.

ملخص

أصبح GitHub منصة تطبيقات اجتماعية شهيرة.

حيث ينشر عدد كبير من المستخدمين مشاريعهم مفتوحة المصدر. على وجه الخصوص ، يقوم عدد متزايد من الباحثين بنشر مستودعات لمصادر التعليمات البرمجية المتعلقة بأبحاثهم من أجل جذب المزيد من الأشخاص لمتابعة عملهم [1]. بدافع من هذا الاتجاه ، يتكون عملنا من دراسة واستخدام لنشر المقالات العلمية. من خلال الاعتماد على GitHub منصة المطلوب JavaScript وامتدادات GitHub المدعوم بصفات Jekyll محرك تطويرها ، يتضمن ذلك إمكانية نشر مقالات ذات محتوى علمي عبر الإنترنت. بالإضافة إلى تحسين الرجوع إلى المقالات المنشورة ، فإن الهدف من هذا (فيما) المشروع هو تنفيذ المكونات الإضافية التي تسمح بالتعليق على المقالات والموافقة عليها ولكن أيضًا لاستنساخ مقال واقتراح (GitHub يتعلق بقضايا تعديلات (مثل طلبات السحب في جيثب)

النشر ، المقالات المراجع، Jekyll ، prototyper ، GitHub: الكلمة الرئيسية المكونات الإضافية ، استنساخ ، طلبات السحب

Sommaire

Introduction général :	8
------------------------------	---

Chapitre 1:Le GitHub

Introduction :	11
----------------------	----

GitHub :	11
----------------	----

Compte utilisateur GitHub :	12
-----------------------------------	----

GitHub Pages :	13
----------------------	----

Consignes d'utilisation des pages GitHub :	13
--	----

Créer un site Pages GitHub :	14
------------------------------------	----

Création d'un référentiel pour votre site :	14
---	----

Création de votre site :	17
--------------------------------	----

Bifurquer (fork) le référentiel:.....	19
---------------------------------------	----

Push:	20
-------------	----

Pull:.....	20
------------	----

Conclusion :	22
--------------------	----

Chapitre 2:Les pages Jekyll et GitHub

Introduction :	24
----------------------	----

Jekyll :	24
----------------	----

LE BUT DE JEKYL :	24
-------------------------	----

LES AVANTAGES DE Jekyll :	24
---------------------------------	----

Simplicité :	24
--------------------	----

Contrôle de la conception :	25
-----------------------------------	----

Sécurité :	25
------------------	----

Hébergement :	26
---------------------	----

Installation et configuration :	26
---------------------------------------	----

Conditions préalables :	26
-------------------------------	----

Application de bureau GitHub :	26
--------------------------------------	----

Éditeur de code :	28
-------------------------	----

Ligne de commande :	29
---------------------------	----

Installation de Jekyll:.....	32
Installation de Ruby :.....	32
Installation graphique :	33
Installation de Bundler et Jekyll :	34
Conclusion :.....	36

Chapitre 3:Créations de site Jekyll

Création de site Jekyll :	38
Structure de répertoire Jekyll typique :	40
Configuration de Jekyll avec des pages GitHub :.....	41
Exécution de Site Jekyll localement :	42
Le système de Jekyll.....	45
MODÈLES DE LIQUIDE :	45
Les balises de modèle liquides :	45
Le fichier yml :	46
_CONFIG.YML :	46
Front Matter :.....	47
Rédaction de pages et d'articles :	48
Langage de Markdown :	48
Création de pages :	50
Placement des pages dans Jekyll :.....	52
Création des postes(articles):	53
Ajoutez du contenu à votre post :.....	53
Ajouter des photos et ressources :.....	54
Thèmes :	56
Choisissez un thème :.....	56
Documenter votre thème :.....	57
Ajout d'une capture d'écran :.....	57
Hébergement sur des pages GitHub :.....	58
Captures d'écran de mon site Web	59
Conclusion :.....	63

Conclusion generale:	64
Bibliographie:	65

Introduction général :

La définition d'un site Web (web site en anglais) ou site Internet est un ensemble de pages Web reliées entre elles par des liens internes, visualisable depuis un navigateur Web.

D'une manière générale, la totalité des pages d'un site Web est accessible sur une même adresse, sous le même nom de domaine.

Un site Internet contient différents types de contenus (textes et multimédia), et est hébergé sur un serveur Web.

Le serveur Web, pour sa part, est accessible à travers un réseau Internet ou intranet, par l'intermédiaire d'un navigateur Web comme Firefox, Internet Explorer, Chrome, Safari, Opéra...

Chaque page Web qui constitue un site Web est un ensemble de documents HTML structurés et stockés sur le serveur Web.

Les sites statiques : ce sont des sites réalisés uniquement à l'aide des langages HTML et CSS. Ils fonctionnent très bien mais leur contenu (les articles) ne peut pas être mis à jour automatiquement : il faut que le propriétaire du site (le webmaster) modifie le code source pour y ajouter des nouveautés. [2]

Les articles scientifiques c'est un écrit à caractère scientifique, exposant généralement les résultats d'une recherche expérimentale, qui fait partie d'une publication spécialisée. [3]

L'objectif de ce mémoire est d'étudier l'utilisation et le prototypage de la plateforme GitHub pour la publication d'articles scientifiques.

L'objectif principal de notre travail rapporté dans ce mémoire de fin d'étude est de :

- Créer des pages et des articles.
- Mettre un œuvre des plugins permettant de commenter les articles (en lien avec les issues de GitHub) les approuver.
- Ajouter et modifier le contenu d'un article.

- Cloner un article et de proposer des modifications (comme des pullrequests dans GitHub).
- Savoir plus sur les pages GitHub et comment y héberger des sites.
- Savoir plus sur Jekyll le générateur de site statique.
- Créer un compte sur GitHub.
- Créer un site Web statique entièrement gratuit.

L'étude de ce projet est organisée en deux parties : partie théorique et partie pratique :

- La partie théorique : qui contient un aperçu général sur l'utilisation de la plateforme GitHub ainsi que Jekyll pour l'espace de travail pour les chercheurs
- La partie pratique : Créez des pages et des publications et hébergez les dans GitHub.

Chapitre 1

Le GitHub

Introduction :

Des millions de développeurs et d'entreprises créent, livrent et maintiennent leurs logiciels sur GitHub, la plate-forme de développement la plus grande et la plus avancée au monde. [4]

GitHub :

GitHub permet aux développeurs de stocker et de partager, publiquement ou non, le code qu'ils créent. La plate-forme accueille ainsi, dit-elle, plus de 80 millions de projets, qu'il s'agisse de logiciels, de sites Web, d'applications pour mobile ou tous autres types de programme informatique — et ce quel que soit le langage de programmation utilisé. Le site est aussi un espace collaboratif. Chaque utilisateur peut contribuer aux projets mis en ligne publiquement sur GitHub, en proposant des modifications. Le succès de GitHub repose notamment sur la façon dont le site a facilité ce processus. Pour que les utilisateurs ne se dérangent pas mutuellement en modifiant un programme en même temps, ils téléchargent chacun de son côté le code sur son ordinateur, effectue les modifications, qui sont ensuite publiées sur GitHub après validation. Le site se base pour cela sur Git, un outil développé en 2005 par Linus Thorvald, le célèbre créateur de Linux — GitHub aura eu pour talent de rendre ce système plus simple d'utilisation et compréhensible par le plus grand nombre. Chaque modification du code est ainsi stockée sur GitHub, et il est possible de suivre pas à pas chaque étape de développement d'un programme. Des espaces de discussion permettent à tous les développeurs d'échanger sur chaque programme et contribution. [5]

Compte utilisateur GitHub :

Un compte utilisateur GitHub vous permettra d'héberger votre site Web gratuitement sur GitHub. En prime, il vous permettra également de garder une trace des versions du site Web et de son écriture à mesure qu'il se développe ou évolue au fil du temps.

- Visitez [GitHub.com](https://github.com) et cliquez sur le bouton « Inscription » en haut à droite. Écrivez votre nom d'utilisateur souhaité. Cela sera visible par les autres, vous identifiera sur GitHub et fera également partie de l'URL de votre site. Écrivez également l'adresse e-mail et le mot de passe souhaités, puis cliquez sur « Créer un compte ».
- Sur la page suivante, cliquez sur le bouton « Choisir » à côté de l'option de plan « Gratuit », ignorez la case à cocher « Aidez-moi à configurer une organisation ensuite », puis cliquez sur « Terminer l'inscription ».
- Facultatif : visitez <https://github.com/settings/profile> pour ajouter un nom complet (peut être votre vrai nom, votre nom d'utilisateur GitHub ou autre chose) et d'autres informations de profil public, si vous le souhaitez. [4]

GitHub Pages :

GitHub Pages est un service d'hébergement de site statique qui prend les fichiers HTML, CSS et JavaScript directement à partir d'un référentiel sur GitHub, exécute éventuellement les fichiers via un processus de construction et publie un site Web. [4]

Consignes d'utilisation des pages GitHub :

- Les sites Pages GitHub créés après le 15 juin 2016 et utilisant des github.io domaines sont servis via HTTPS. Si vous avez créé votre site avant le 15 juin 2016, vous pouvez activer la prise en charge HTTPS pour le trafic vers votre site. Pour plus d'informations, consultez « Sécurisation de vos pages GitHub avec HTTPS ».
 - Les sites de pages GitHub ne doivent pas être utilisés pour des transactions sensibles telles que l'envoi de mots de passe ou de numéros de carte de crédit.
 - Votre utilisation des pages GitHub est soumise aux conditions d'utilisation de GitHub, y compris l'interdiction de revendre.
- [4]

Créer un site Pages GitHub :

Vous pouvez créer un site Pages GitHub dans un référentiel nouveau ou existant.

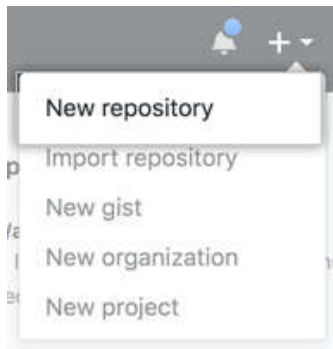
Les pages GitHub sont disponibles dans des référentiels publics avec GitHub Free et GitHub Free pour les organisations, et dans des référentiels publics et privés avec GitHub Pro, GitHub Team, GitHub Enterprise Cloud et GitHub Enterprise Server. [4]

Création d'un référentiel pour votre site :

Si votre site est un projet indépendant, vous pouvez créer un nouveau référentiel pour stocker le code source de votre site. Si votre site est associé à un projet existant, vous pouvez ajouter le code source au référentiel de ce projet, dans un /docs dossier sur la branche par défaut ou sur une branche différente. Par exemple, si vous créez un site pour publier la documentation d'un projet déjà présent sur GitHub, vous souhaitez peut-être stocker le code source du site dans le même référentiel que le projet.

Si le compte propriétaire du référentiel utilise GitHub Free ou GitHub Free pour les organisations, le référentiel doit être public.

- Dans le coin supérieur droit de n'importe quelle page, utilisez le menu déroulant et sélectionnez Nouveau référentiel.



- Utilisez le menu déroulant Propriétaire et sélectionnez le compte dont vous souhaitez être propriétaire du référentiel.

A screenshot of the GitHub repository creation form. The 'Owner' field is a dropdown menu with 'octocat' selected. Below it, a list of suggestions is shown: 'octocat' (checked) and 'github'. The 'Repository name' field is empty and has a slash '/' next to it.

- Saisissez un nom pour votre référentiel et une description facultative. Si vous créez un site d'utilisateur ou d'organisation, votre référentiel doit être nommé `<user>.github.io` ou `<organisation>.github.io`. Pour plus d'informations, consultez « À propos des pages GitHub ».

A screenshot of the GitHub repository creation form. The 'Owner' field is 'octocat'. The 'Repository name' field is filled with 'octocat.github.io' and has a green checkmark next to it. Below the form, there is a text prompt: 'Great repository names are short and memorable. Need inspiration? How about [fictional-guide?](#)' and a 'Description (optional)' text area.

- Choisissez une visibilité de référentiel. Pour plus d'informations, consultez « À propos de la visibilité du référentiel ».

Description (optional)

Public
Anyone can see this repository. You choose who can commit.

Internal
Octo Corp [enterprise members](#) can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

- Sélectionnez Initialiser ce référentiel avec un fichier README.

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

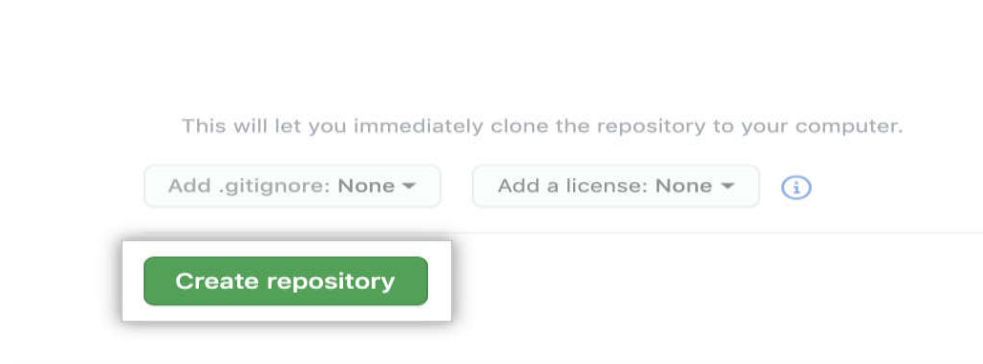
Skip this step if you're importing an existing repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▾ Add a license: **None** ▾ ⓘ

Create repository

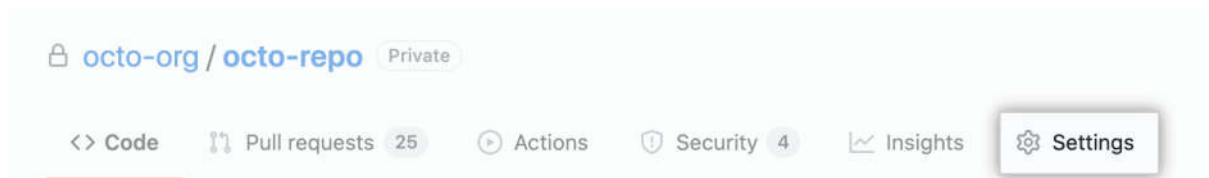
- Cliquez sur Créer un référentiel.



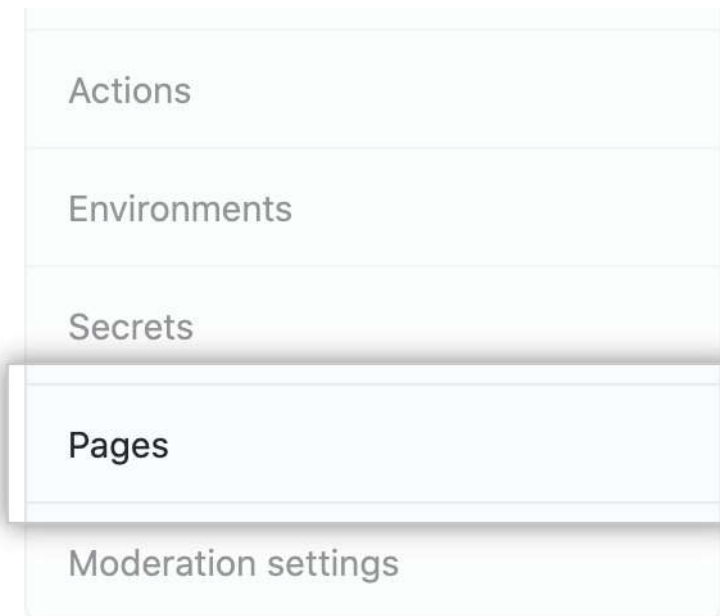
Création de votre site :

Avant de pouvoir créer votre site, vous devez disposer d'un référentiel pour votre site sur GitHub.

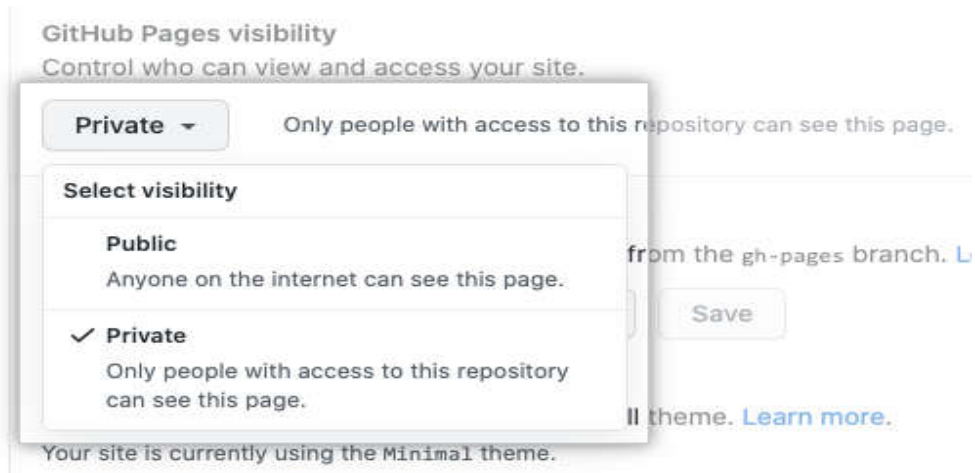
- Sur GitHub, accédez au référentiel de votre site.
- Décidez de la source de publication que vous souhaitez utiliser. Pour plus d'informations, consultez « À propos des pages GitHub ».
- Si la source de publication que vous avez choisie existe déjà, accédez à la source de publication. Si la source de publication que vous avez choisie n'existe pas, créez la source de publication.
- À la racine de la source de publication, créez un nouveau fichier appelé `index.md` qui contient le contenu que vous souhaitez afficher sur la page principale de votre site.
- Configurez votre source de publication. Pour plus d'informations, consultez « Configurer une source de publication pour votre site GitHub Pages ».
- Sous le nom de votre référentiel, cliquez sur Paramètres.



- Dans la barre latérale gauche, cliquez sur Pages.



- Facultativement, si vous publiez un site de projet à partir d'un référentiel privé ou interne appartenant à une organisation utilisant GitHub Enterprise Cloud, choisissez la visibilité de votre site. Sous « Pages GitHub », sélectionnez le menu déroulant de visibilité des pages GitHub, puis cliquez sur une visibilité. Pour plus d'informations, consultez « Modifier la visibilité de votre site GitHub Pages ».



- Pour voir votre site publié, sous "Pages GitHub", cliquez sur l'URL de votre site.

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

✓ Your site is published at <https://octocat.github.io/>

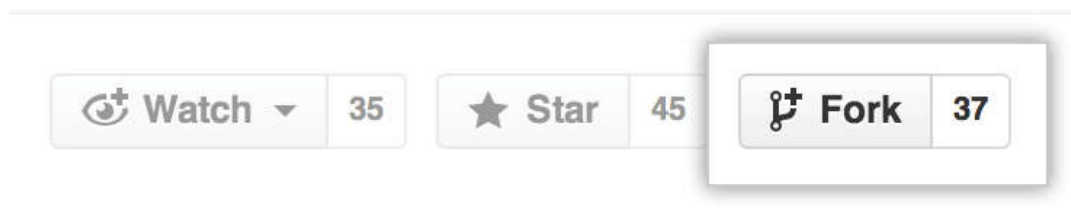
[4]

Bifurquer (fork) le référentiel:

Après avoir utilisé GitHub par vous-même pendant un certain temps, vous souhaitez peut-être contribuer au projet de quelqu'un d'autre. Ou peut-être aimeriez-vous utiliser le projet de quelqu'un comme point de départ du vôtre. Ce processus est connu sous le nom de fork.

Créer un « fork », c'est produire une copie personnelle du projet de quelqu'un d'autre. Les forks agissent comme une sorte de pont entre le référentiel d'origine et votre copie personnelle. Vous pouvez soumettre des Pull Requetés pour aider à améliorer les projets d'autres personnes en proposant vos modifications jusqu'au projet d'origine. Bifurcation est au cœur du codage social chez GitHub.

Pour bifurquer le référentiel, cliquez sur le bouton Bifurquer dans l'en-tête du référentiel :



Lorsqu'elle sera terminée, vous serez dirigé vers votre copie du référentiel

Vous avez réussi à bifurquer le référentiel, mais pour l'instant, il n'existe que sur GitHub. Pour pouvoir travailler sur le projet, vous devez le cloner sur votre ordinateur.

Si vous utilisez GitHub Desktop, ce processus est très simple. Sur votre référentiel, naviguez jusqu'à la barre latérale droite et cliquez sur Cloner ou Télécharger. La façon dont vous clonez dépend de vous. Certaines options sont le clonage avec la ligne de commande, ou en utilisant GitHub Desktop.

Push:

Après avoir terminé les modifications et ajouté le contenu au référentiel local Lorsque vous êtes prêt à soumettre vos modifications, mettez-les en scène et livrez-les.

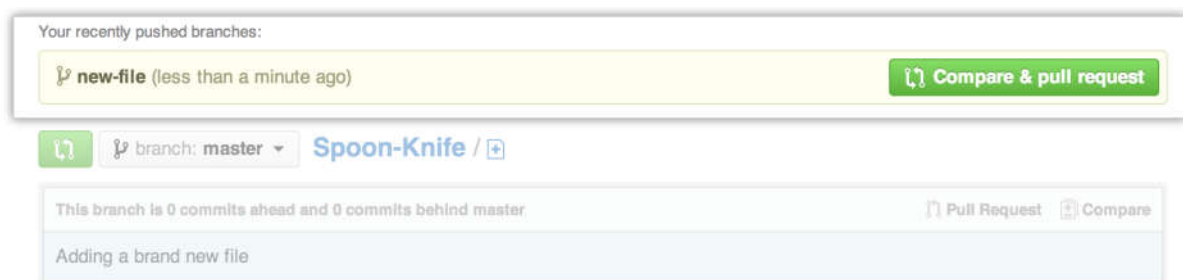
Pour l'instant, vous avez essentiellement dit à Git, "Ok, j'ai pris un instantané de mes changements !". Vous pouvez continuer à faire plus de changements, et prendre plus d'instantanés de commit. Lorsque vous êtes prêt à pousser vos modifications vers GitHub.com, poussez vos modifications vers le serveur distant.

Pull:

Enfin, vous êtes prêt à proposer des changements dans le projet principal ! C'est la dernière étape de la production d'un fork du projet

de quelqu'un d'autre, et sans doute la plus importante. Si vous avez effectué une modification qui, selon vous, profiterait à l'ensemble de la communauté, vous devez absolument envisager de contribuer en retour

Pour ce faire, rendez-vous sur le référentiel de GitHub.com où se trouve votre projet. Vous verrez une bannière indiquant que vous avez récemment poussé une nouvelle branche, et que vous pouvez soumettre cette branche "en amont", au référentiel original :



En cliquant sur Compare and Pull Request, vous accédez à une page de discussion, où vous pouvez saisir un titre et une description facultative. Il est important de fournir autant d'informations utiles que possible et de justifier la raison pour laquelle vous faites cette Pull Request en premier lieu. Le propriétaire du projet doit être en mesure de déterminer si votre changement est aussi utile à tous que vous le pensez.

Lorsque vous avez fini de taper votre argumentaire sincère, cliquez sur "Send pull request". C'est fait ! [6]

Conclusion :

À la fin de ce chapitre, vous êtes un utilisateur de GitHub. Vous savez créer un compte, gérer une organisation, créer et pousser vers des référentiels, contribuer aux projets des autres et accepter les contributions des autres. [6]

L'utilisation de GitHub pour stocker les exemples de code et le code d'exercice des leçons peut être un excellent moyen de s'exercer et d'acquérir de l'expérience avec un nouvel outil logiciel. L'utilisation de GitHub n'est pas obligatoire et il n'est pas recommandé d'y stocker vos projets terminés. GitHub est une plateforme d'apprentissage encouragée pour les étudiants car de nombreuses organisations utilisent GitHub.

GitHub offre des moyens rapides et pratiques de suivre les projets, qu'ils soient réalisés par une seule personne ou par une équipe de développeurs de logiciels. Bien que GitHub dispose de nombreuses fonctionnalités complexes, il est facilement accessible pour les projets individuels et les petits projets qui ont besoin d'une sorte de mécanisme de suivi. En plus du contrôle de version, GitHub offre aux utilisateurs une plateforme sociale pour la gestion de projet ainsi que la possibilité de créer leur propre projet et de le partager.

Chapitre2

Les pages Jekyll et GitHub

Introduction :

Dans ce mémoire, nous découvrirons le moyen le plus rapide de créer un blog alimenté par Jekyll, comment éviter les problèmes courants liés à l'utilisation de Jekyll. [7]

Jekyll :

Est un fantastique générateur de site Web conçu pour créer des blogs minimaux et statiques à héberger sur des pages GitHub. La simplicité de la couche de thème et du flux de travail d'écriture de Jekyll est fantastique.[7]

LE BUT DE JEKYL :

Tom Preston-Werner a créé Jekyll pour permettre aux gens de blogueur à l'aide d'un simple site Web HTML statique, avec tout le contenu hébergé et contrôlé par version sur GitHub. Le but était d'éliminer la complexité des autres plateformes de blogs en créant un workflow qui vous permet de blogueur comme un hacker.

Jekyll prend votre contenu écrit en Mark down, le transmet à vos modèles et le recrache comme un site Web statique complet, prêt à être servi. Les pages GitHub servent de manière pratique le site Web directement à partir de votre référentiel GitHub afin que vous n'ayez à vous occuper d'aucun hébergement. [7]

LES AVANTAGES DE Jekyll :

Simplicité :

Jekyll réduit tout au strict minimum, éliminant ainsi beaucoup de complexité

- Pas de base de données :

Contrairement à WordPress et à d'autres systèmes de gestion de contenu (CMS), Jekyll n'a pas de base de données. Tous les articles et pages sont convertis en HTML statique avant la publication. C'est idéal pour la vitesse de chargement car aucun appel de base de données n'est effectué lorsqu'une page est chargée.

- Pas de CMS :

Écrivez simplement tout votre contenu dans Mark down, et Jekyll l'exécutera à travers des modèles pour générer votre site Web statique. GitHub peut servir de CMS si nécessaire, car vous pouvez y modifier le contenu.

- Faste :

Jekyll est rapide car étant dépouillé et sans base de données, vous ne faites que servir des pages statiques, JekyllNow, ne fait que trois requêtes HTTP - y compris la photo de profil et les icônes sociales!

- Minimal :

Votre site Web Jekyll n'inclura absolument aucune fonctionnalité ou fonctionnalité que vous n'utilisez pas.

Contrôle de la conception :

Passez moins de temps à vous battre avec des modèles complexes écrits par d'autres personnes et plus de temps à créer votre propre thème ou à personnaliser un thème de base simple.

Sécurité :

La grande majorité des vulnérabilités qui affectent des plates-formes comme WordPress n'existent pas car Jekyll n'a pas de CMS, de base

de données ou de PHP. Ainsi, vous n'avez pas à passer autant de temps à installer les mises à jour de sécurité.

Hébergement :

Pratique si vous utilisez déjà GitHub, c'est-à-dire. GitHub Pages construira et hébergera votre site Web Jekyll sans frais, tout en gérant simultanément le contrôle de version. [8]

Installation et configuration :

Avec Jekyll, vous n'avez pas à vous soucier de tout cela. Jekyll n'a pas d'interface d'administration, uniquement des fichiers sources que vous pouvez créer à l'aide d'une CLI (interface de ligne de commande) pour créer un site Web statique. Écrivez simplement vos articles de blog dans Mark down et si vous choisissez d'héberger votre site Web à l'aide de pages GitHub, vous pouvez utiliser Git pour valider et pousser vos modifications afin de publier de nouveaux articles ou pages.

Jekyll est facile à installer sur MacOS et Linux, mais quelques étapes supplémentaires sont nécessaires pour pouvoir utiliser Jekyll sous Windows. [9]

Conditions préalables :

Application de bureau GitHub :

L'application GitHub Desktop facilitera la mise à jour de votre site Web en direct (celui que nous avons configuré). Au lieu d'utiliser la ligne de commande chaque fois que vous souhaitez mettre à jour votre site, vous pourrez utiliser un outil visuel plus simple pour mettre à jour votre site.

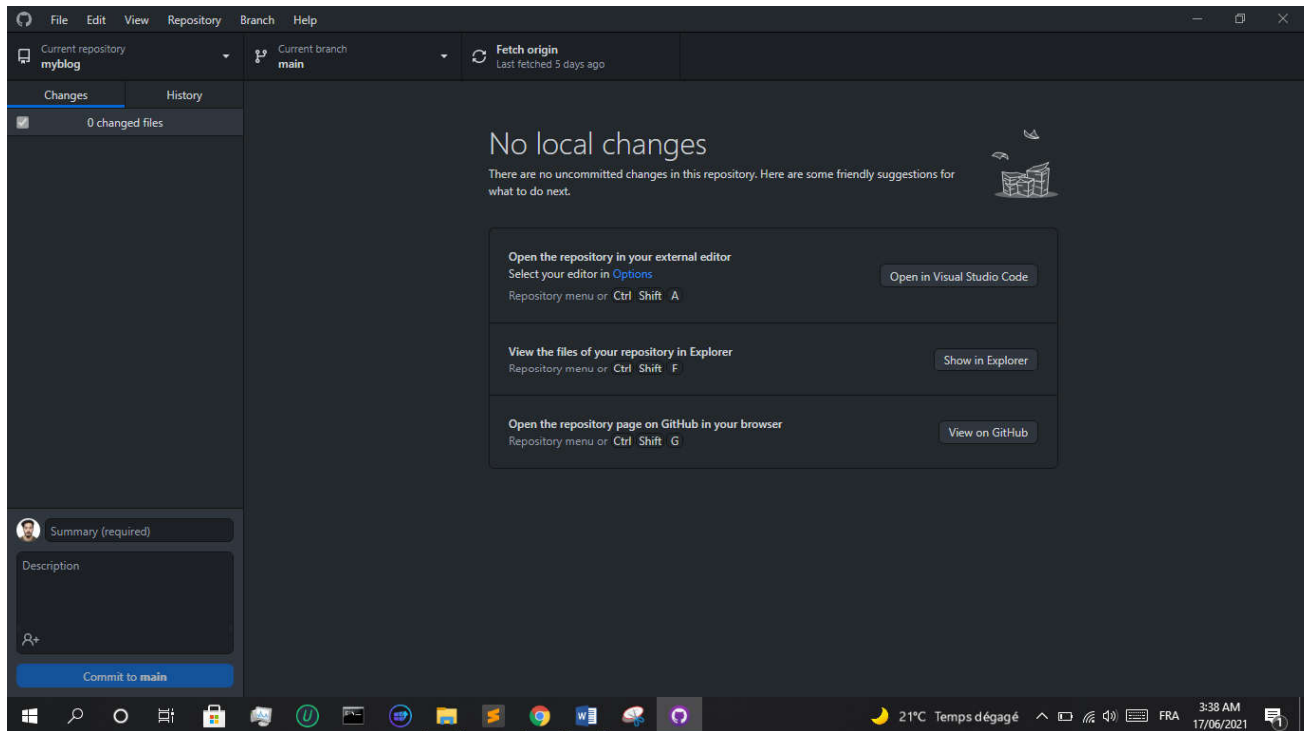
- Visitez le site GitHub Desktop et cliquez sur le bouton « Télécharger GitHub Desktop » pour télécharger le logiciel GitHub Desktop sur votre ordinateur.



- Une fois le fichier complètement téléchargé, double-cliquez dessus et utilisez les instructions suivantes pour installer GitHub Desktop...
- Saisissez le nom d'utilisateur et le mot de passe du compte GitHub.com que vous avez créé en suivant les étapes ci-dessus. (Ignorez le bouton « Ajouter un compte d'entreprise ».) Cliquez sur « Continuer ».
- Entrez le nom et l'adresse e-mail auxquels vous souhaitez associer le travail sur votre site (probablement juste votre nom public et votre adresse e-mail professionnelle, mais c'est à vous!).
- Sur la même page, cliquez sur le bouton « Installer les outils de ligne de commande » et entrez le nom d'utilisateur et le mot de passe de votre ordinateur si vous y êtes invité (puis cliquez sur le bouton « Install Helper » à l'invite). Après avoir reçu un message

contextuel indiquant que tous les outils de ligne de commande ont été installés avec succès, cliquez sur Continuer.

- La dernière page demandera « Quels référentiels souhaitez-vous utiliser? ». Ignorez cela et cliquez sur le bouton « Terminé ».

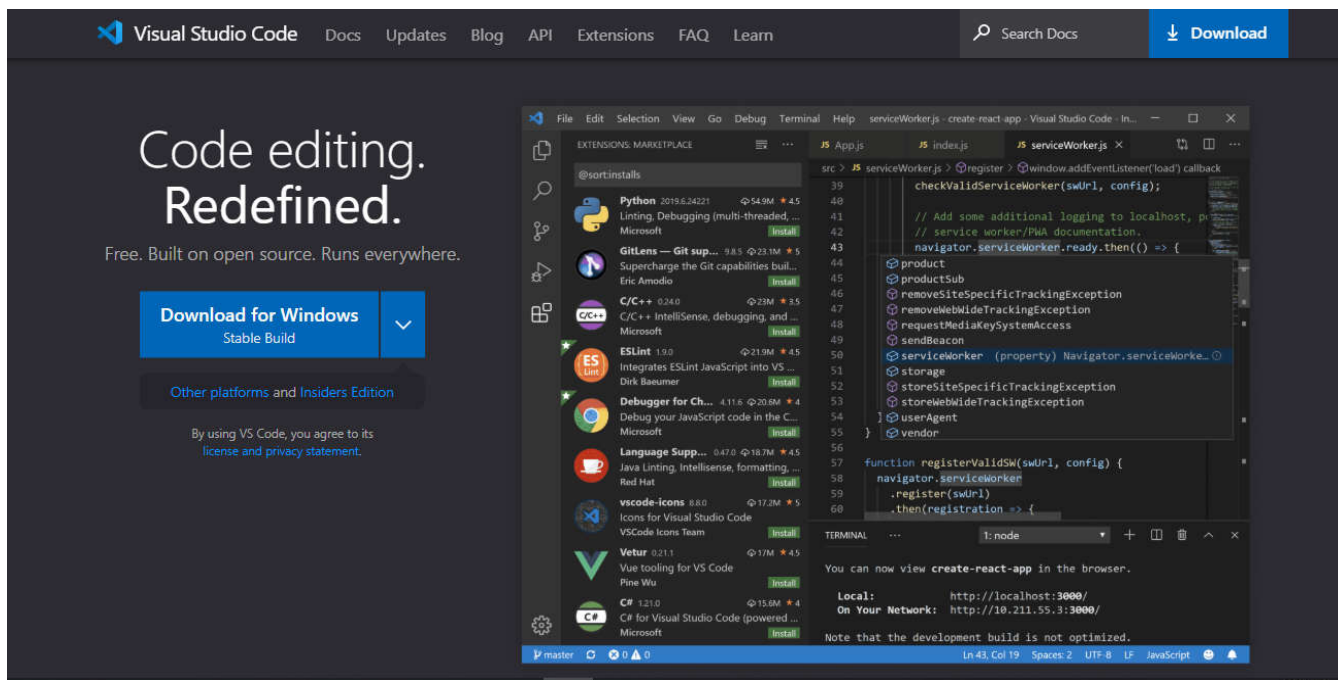


[10]

Éditeur de code :

Vous devrez télécharger et installer un programme « éditeur de texte » sur votre ordinateur pour effectuer de petites personnalisations du code de votre site Jekyll. Les bonnes options gratuites incluent TextWrangler (Mac) ou Notepad ++ (Windows). Un logiciel destiné au traitement de texte, comme Microsoft Word ou Word Pad, n'est pas un bon choix car il est facile d'oublier comment formater et enregistrer le fichier, en ajoutant accidentellement une mise en forme supplémentaire et / ou invisible et des caractères qui briseront votre site. Vous voudrez quelque chose qui peut spécifiquement enregistrer ce que vous écrivez en texte brut (par exemple HTML, Mark down),

C'est pourquoi je vous présente Visual Studio Il est considéré comme l'un des meilleurs code éditeur :

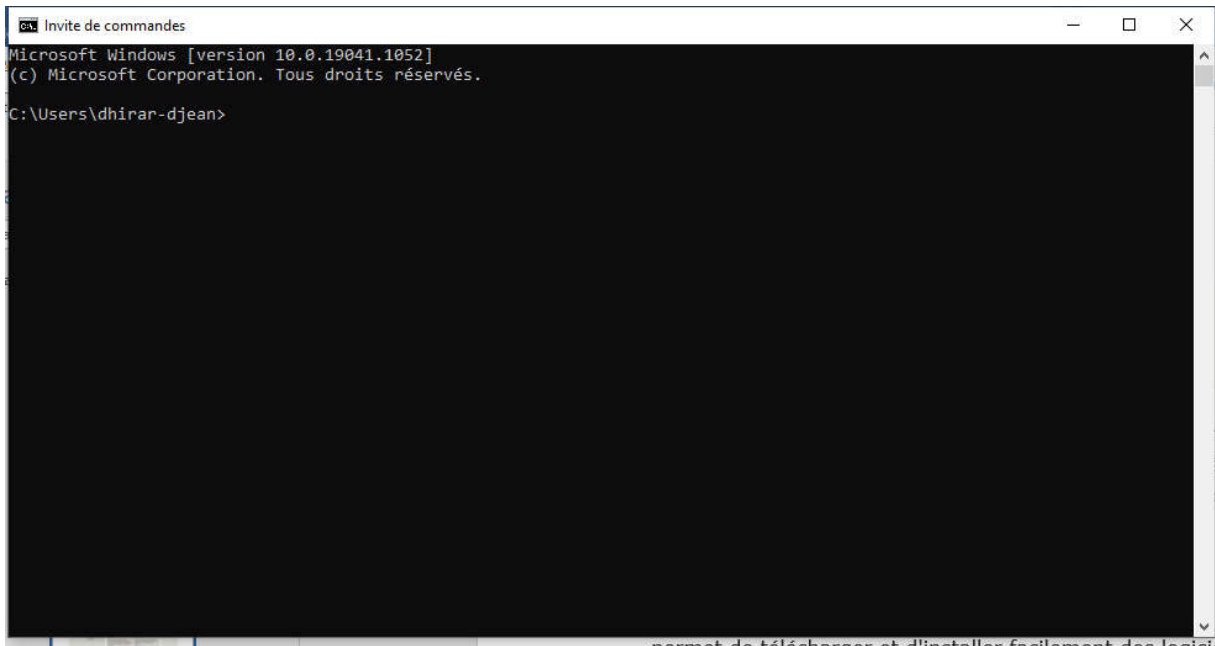


[11]

Ligne de commande :

La ligne de commande est un moyen d'interagir avec votre ordinateur à l'aide de texte: elle vous permet de taper des commandes pour des actions à partir de choses plus simples telles que « montre-moi une liste des fichiers de ce répertoire » ou « changer qui est autorisé à accéder à ce fichier » [2]

- Ouvrez « Invite de commandes » (ouvrez votre menu Démarrer et recherchez « Invite de commandes » et une application que vous pouvez ouvrir devrait apparaître).



- Chocolatey est un « gestionnaire de paquets »: code qui vous permet de télécharger et d'installer facilement des logiciels open-source sur Windows à partir de la ligne de commande. Nous allons maintenant installer Chocolatey (assurez-vous de mettre en évidence et de copier tout le club de texte ci-dessous ensemble, pas comme des lignes séparées). Entrez le code indiqué dans les étapes ci-dessous en gardant à l'esprit les conseils de ligne de commande ci-dessus :

```
@powershell -NoProfile -ExecutionPolicy unrestricted -Command "(iex ((new-object net.webclient).DownloadString('https://chocolatey.org/install.ps1'))) >$null 2>&1" && SET PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin
```

- Nous avons besoin d'un outil de ligne de commande qui reconnaît les mêmes commandes que les ordinateurs Mac et Linux (c'est-à-dire les systèmes d'exploitation Unix). Visitez <https://git-scm.com/downloads> et cliquez sur le lien « Windows » sous « Téléchargements ». Une fois le téléchargement terminé, double-cliquez sur le fichier téléchargé et suivez les étapes pour

installer Git Bash (laissez toutes les options telles qu'elles sont déjà).

The screenshot shows the Git website's download page. At the top left is the Git logo with the tagline "--distributed-even-if-your-workflow-isnt". To the right is a search bar with the placeholder text "Rechercher tout le site...".

The left sidebar contains navigation links: "À propos", "Documentation", "Téléchargements" (highlighted in red), "Clients GUI", "Logos", and "Communauté". Below these is a text box mentioning the book "Pro Git" by Scott Chacon and Ben Straub, available for free online reading on Amazon.com.

The main content area is titled "Téléchargements". It features three download buttons: "macOS", "les fenêtres", and "Linux/Unix". To the right is a graphic of a computer monitor displaying the text: "Dernière version de la source 2.32.0", "Notes de mise à jour (21-06-06)", and "Télécharger 2.32.0 pour Windows".

Below the download buttons, a note states: "Des versions plus anciennes sont disponibles et le référentiel source Git se trouve sur GitHub." Below this is a section for "Clients GUI" which explains that Git includes graphical tools like "git-gui" and "gitk", but also mentions third-party tools. A link "Afficher les clients GUI →" is provided. To the right is a "Logos" section mentioning that Git logos are available in PNG and EPS formats for use in projects and print. A link "Afficher les logos →" is also present.

At the bottom of the main content area, there is a section titled "Git via Git".

Installation de Jekyll:

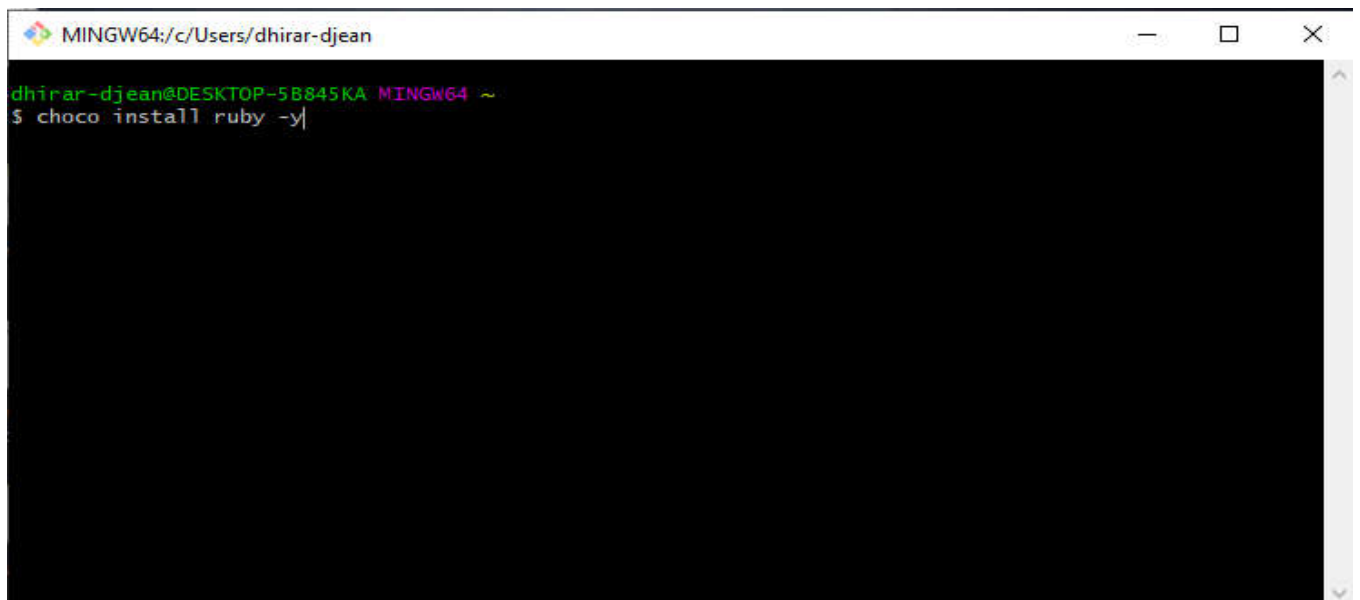
Jekyll est construit à partir du langage de codage Ruby. Ruby Gem facilite la configuration de logiciels Ruby comme Jekyll (c'est un gestionnaire de paquets)

Installation de Ruby :

Sur la ligne de commande :

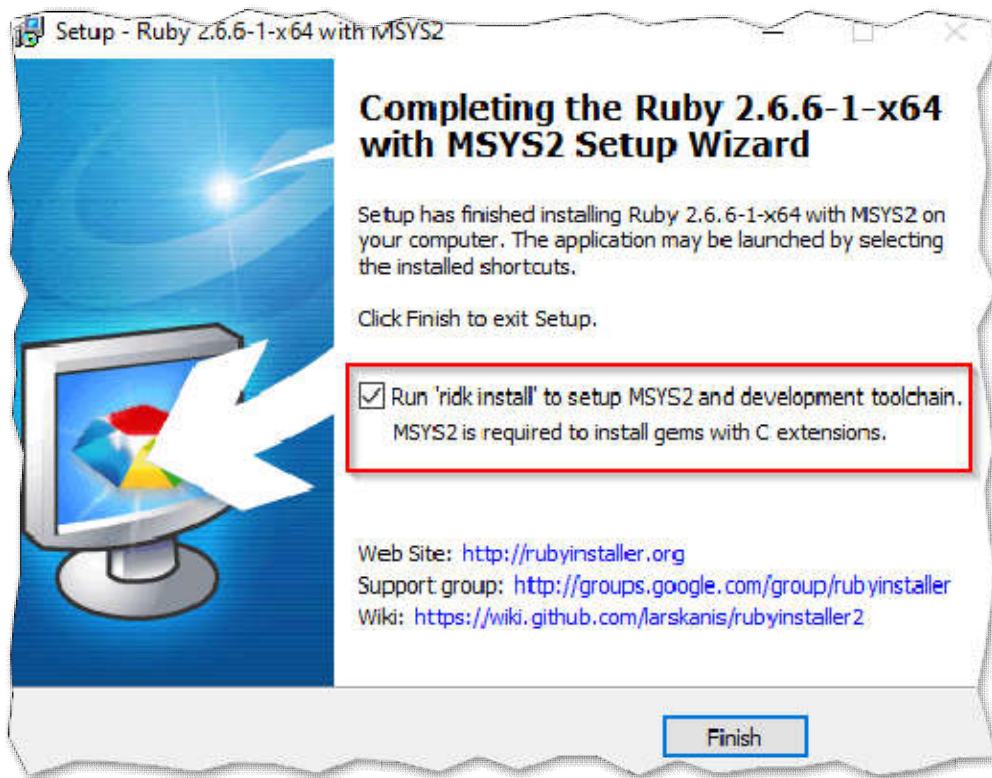
Ouvrez git Bash et écrivez le code suivant :

```
choco install ruby -y
```

A screenshot of a Windows command prompt window. The title bar shows 'MINGW64:/c/Users/dhirar-djean'. The prompt shows the user 'dhirar-djean@DESKTOP-5B845KA' in a 'MINGW64' environment. The command '\$ choco install ruby -y' has been entered and is being processed. The rest of the window is black, indicating the command is still running or the output is not visible.

Fermez le programme de ligne de commande et redémarrez (Ruby ne fonctionnera pas tant que vous ne l'avez pas fait une fois)

Installation graphique :



Le Ruby + DevKit avec l'assistant de configuration MSYS2

Vous avez presque installé Ruby sur Windows! Une fois l'assistant d'installation terminé, vous devriez voir une invite de commande pour configurer MSYS2 et la chaîne d'outils de développement. Entrez simplement 1 dans l'invite de commande pour installer l'installation de base MSYS2 :



Exécution de l'installation de rick pour configurer MSYS2 et la chaîne d'outils de développement

Vous pouvez maintenant fermer l'invite de commande et en ouvrir une nouvelle, car maintenant que Ruby est installé.

Installation de Bundler et Jekyll :

Un joyau Ruby (pensez « module ») qui vous permet de gérer facilement les dépendances de votre projet.

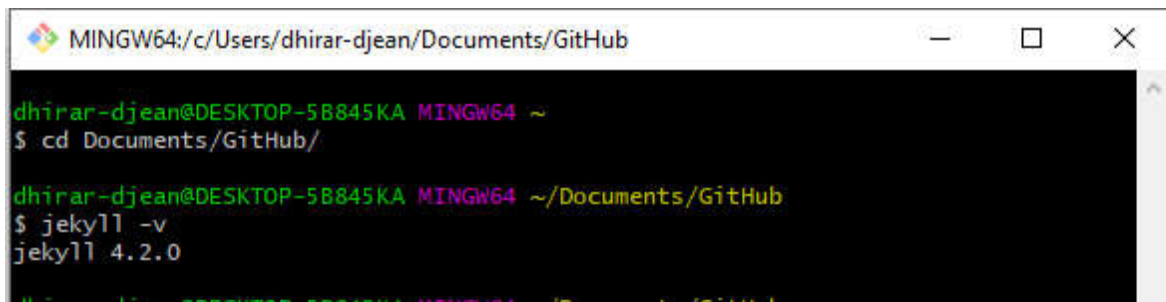
Il est temps d'installer Jekyll! Dans la nouvelle invite de commande, exécutez la commande:

```
geminstalljekyllbundler
```

Installer les rubis gem Jekyll et bundler

Confirmons que Jekyll est installé sur Windows en sortant la version de Jekyll sur notre machine. Pour ce faire, exécutez la commande:

```
jekyll -v
```

A terminal window titled 'MINGW64:/c/Users/dhirar-djean/Documents/GitHub' with standard window controls. The terminal shows the following commands and output:

```
dhirar-djean@DESKTOP-5B845KA MINGW64 ~  
$ cd Documents/GitHub/  
  
dhirar-djean@DESKTOP-5B845KA MINGW64 ~/Documents/GitHub  
$ jekyll -v  
jekyll 4.2.0
```

Confirmez que Jekyll a été installé avec succès. [6]

Conclusion :

Dans ce chapitre, nous avons appris à connaître Jekyll et sa relation avec GitHub, et le but de l'utilisation de jekyll et ses avantages.

Et nous passons par les outils nécessaires pour créer l'environnement pour installer et configurer Jekyll dans nos machines locales pour commencer à créer et générer notre projet.

Et c'est ce que nous allons voir dans le prochain chapitre.

Chapter3

Créations de site Jekyll

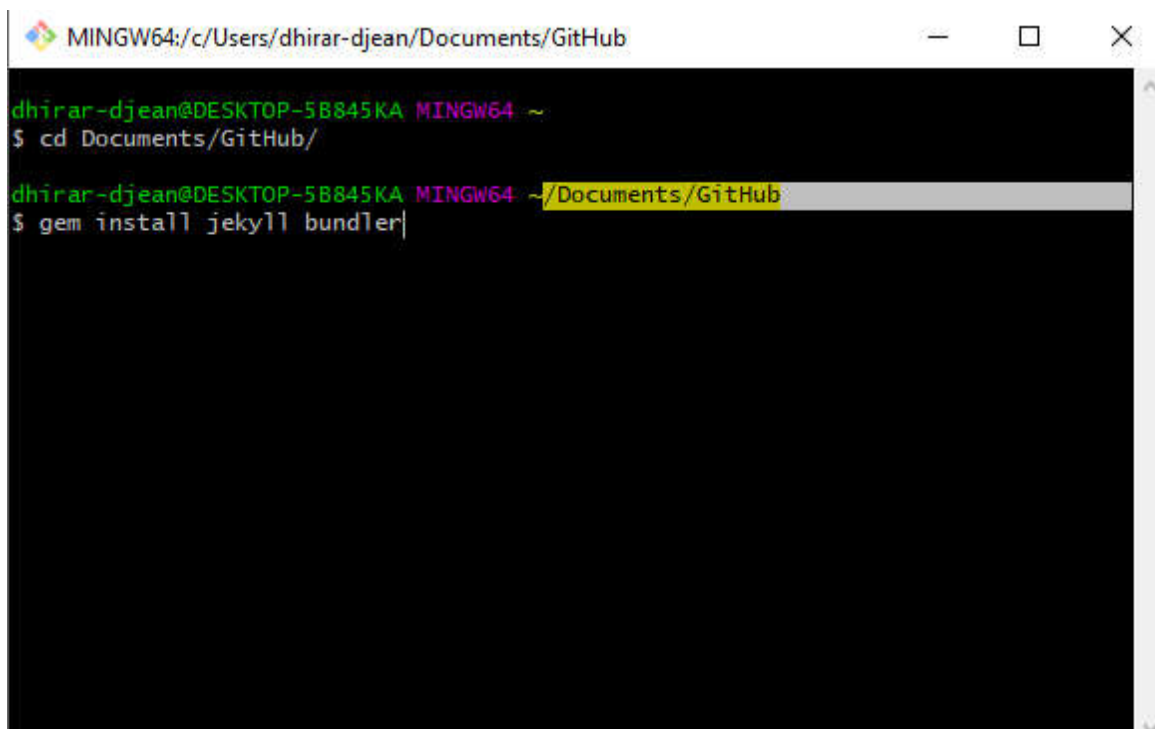
Création de siteJekyll :

Vous avez maintenant installé tout le nécessaire pour créer site Web. Dans cette section, nous utiliserons Jekyll pour générer un nouveau dossier rempli des fichiers qui constituent site Web. Nous localiserons également ce dossier dans un endroit accessible à l'application GitHub Desktop afin qu'il soit au bon endroit lorsque nous souhaitons le publier en tant que site Web public.

Vous devrez connaître le chemin du fichier vers le dossier GitHub créé en installant l'application GitHub Desktop.

Sur la ligne de commande, tapez ce qui suit et appuyez sur Entrée:

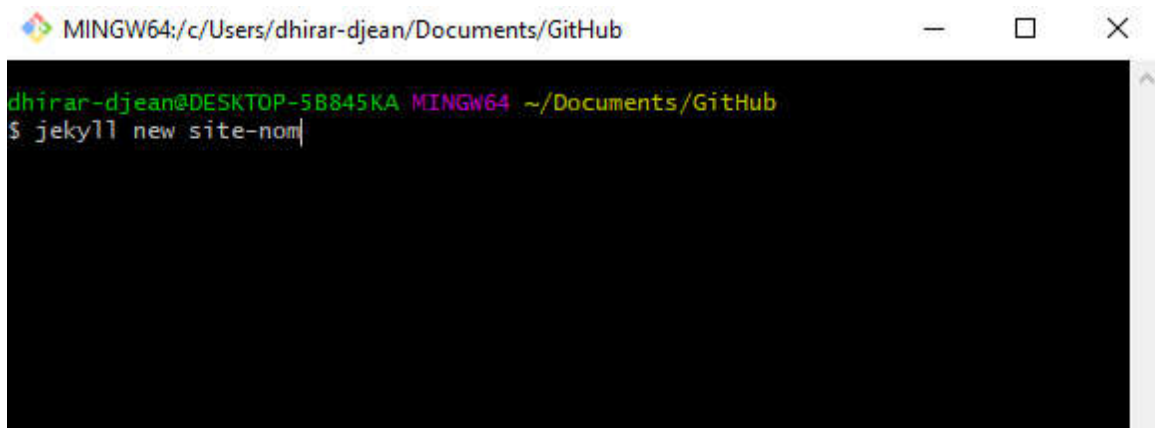
```
geminstalljekyllbundler
```

A screenshot of a terminal window titled 'MINGW64:/c/Users/dhirar-djean/Documents/GitHub'. The terminal shows the following commands and output:

```
dhirar-djean@DESKTOP-5B845KA MINGW64 ~  
$ cd Documents/GitHub/  
dhirar-djean@DESKTOP-5B845KA MINGW64 ~/Documents/GitHub  
$ gem install jekyll bundler
```

Sur la ligne de commande, tapez ce qui suit (mais remplacez site-nom par ce que vous voulez que votre site soit appelé):

```
jekyll new site-nom
```

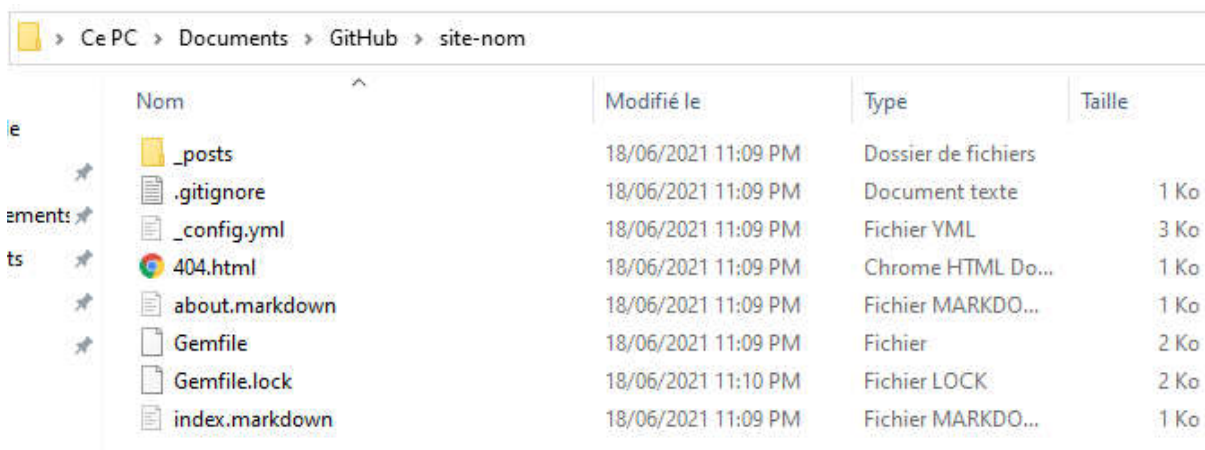


```
MINGW64:/c/Users/dhirar-djean/Documents/GitHub
dhirar-djean@DESKTOP-5B845KA MINGW64 ~/Documents/GitHub
$ jekyll new site-nom
```

Sur la ligne de commande, tapez ce qui suit pour naviguer dans le dossier de votre site (remplacez toujours site-nom par le nom que vous avez choisi pour votre site à l'étape précédente):

```
cd site-nom
```

C'est tout! Si vous vérifiez le dossier, Jekyll aura créé les fichiers nécessaires pour votre nouveau site, le thème Minima par défaut, un exemple de publication et de page et les paramètres de configuration par défaut



Nom	Modifié le	Type	Taille
._posts	18/06/2021 11:09 PM	Dossier de fichiers	
.gitignore	18/06/2021 11:09 PM	Document texte	1 Ko
._config.yml	18/06/2021 11:09 PM	Fichier YML	3 Ko
404.html	18/06/2021 11:09 PM	Chrome HTML Do...	1 Ko
about.markdown	18/06/2021 11:09 PM	Fichier MARKDO...	1 Ko
Gemfile	18/06/2021 11:09 PM	Fichier	2 Ko
Gemfile.lock	18/06/2021 11:10 PM	Fichier LOCK	2 Ko
index.markdown	18/06/2021 11:09 PM	Fichier MARKDO...	1 Ko

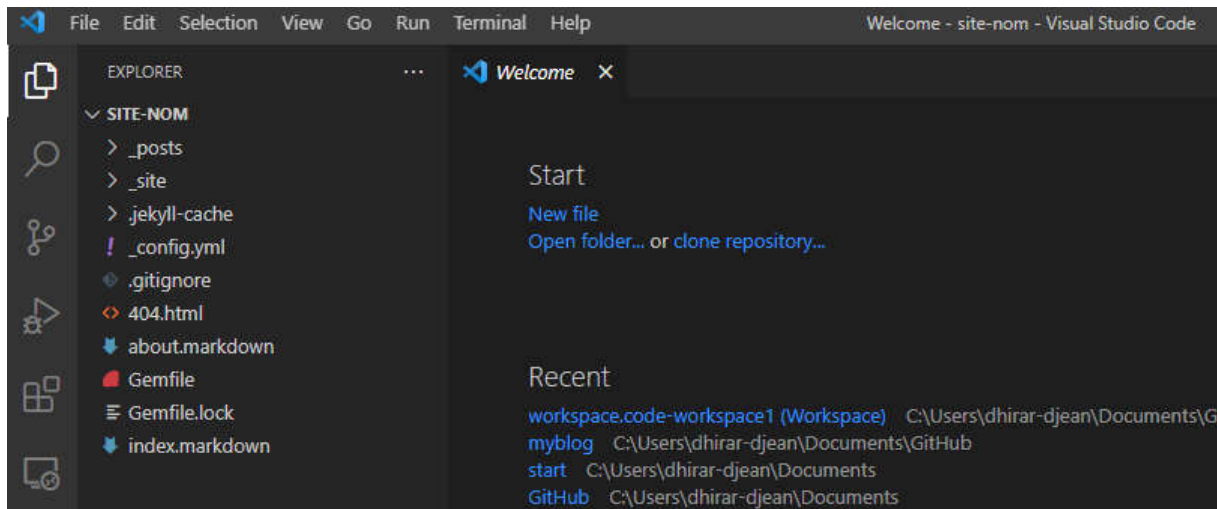
Structure de répertoire Jekyll typique :

Le démarreur Jekyll est un site simple. En réalité, vous allez avoir besoin de plus de répertoires pris en charge par Jekyll pour différents types de tâches:

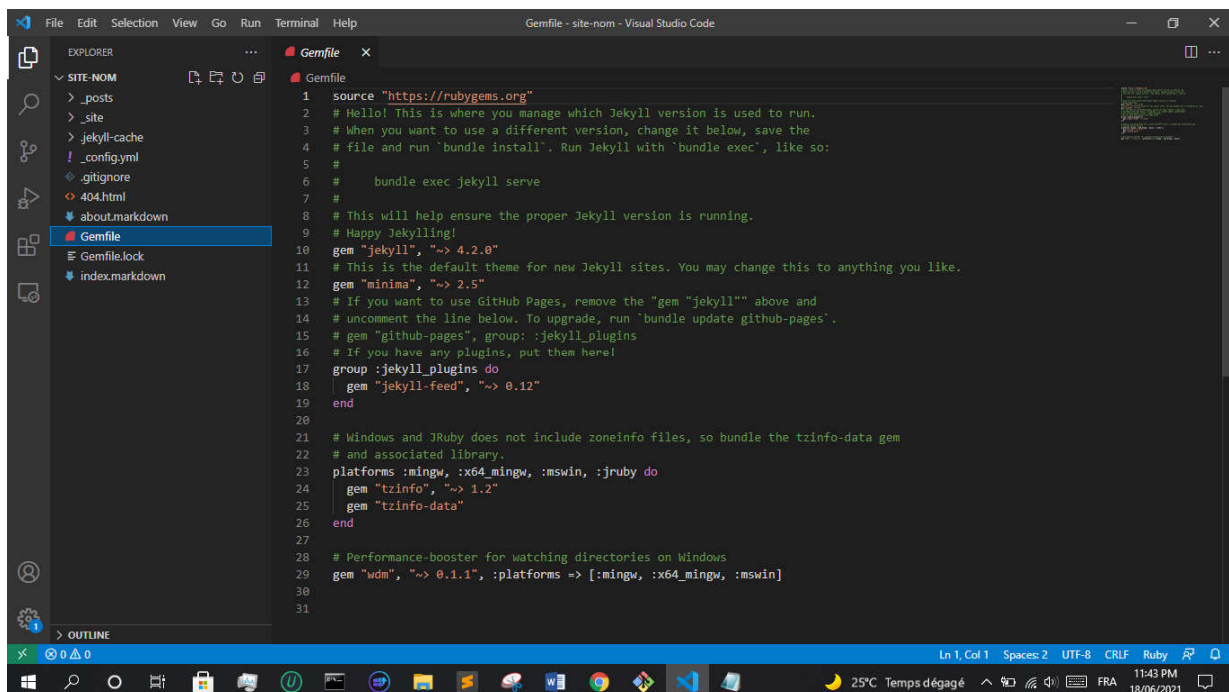
- **_data**: Où vous pouvez stocker des fichiers de données pour des éléments tels que les compétences, les projets, l'historique de travail, etc.
- **_drafts**: Ajoutez ceci à votre .gitignore et stockez les brouillons de votre blog ici (facultatif).
- **_includes**: Où vous définissez Jekyllincludes, qui sont un peu comme des composants HTML réutilisables.
- **_layouts**: Les mises en page HTML définissent la structure de votre site et peuvent être imbriquées les unes dans les autres.
- **_posts**: Où vous stockerez tous vos articles de blog sous forme de fichiers Markdown.
- **_sass**: C'est là que vont vos partiels SASS. Vous devrez ensuite les importer au format `_assets/main.scss`.
- **_site**: Répertoire de construction généré automatiquement par Jekyll, qui héberge votre site final compilé. Il n'est pas poussé vers GitHub car il est inclus. gitignore
- **_assets**: Principalement pour stocker des images et des scripts, mais il peut également héberger un fichier CSS principal.

Vous vous demandez peut-être pourquoi tous ces noms de répertoire sont précédés d'un trait de soulignement. Un répertoire avec un trait de soulignement en tête est spécial et ne sera pas traité par Jekyll. En conséquence, il ne sera pas apparu dans le répertoire de construction, `_site/`. [12]

Configuration de Jekyll avec des pages GitHub :
Ouvrez le dossier de votre projet avec visual studio :



Gemfile à la racine de votre projet. Vous y trouverez des commentaires utiles pour vous aider à configurer Jekyll avec les pages GitHub:



Tout d'abord, supprimez cette ligne pour publier votre site sur des pages GitHub:

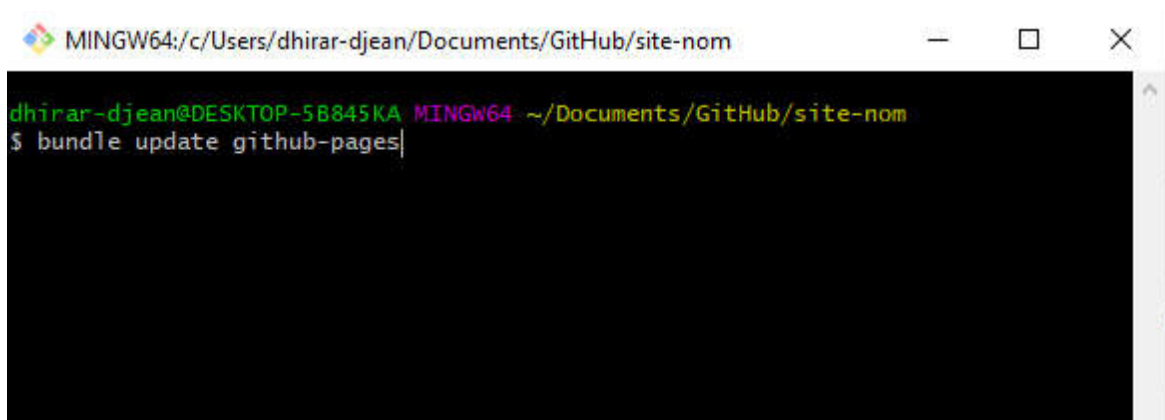
```
gem "jekyll", "~> 4.2.0"
```

Et puis décommentez la ligne spécifiant la GitHub-pages gemme:

```
# gem "github-pages", group: :jekyll_plugins
```

Pour mettre à niveau, exécutez sur le git Bash ce code :

```
bundle update github-pages
```

A screenshot of a terminal window. The title bar shows the path 'MINGW64:/c/Users/dhilar-djean/Documents/GitHub/site-nom'. The terminal content shows the prompt 'dhilar-djean@DESKTOP-5B845KA MINGW64 ~/Documents/GitHub/site-nom' followed by the command '\$ bundle update github-pages' and a cursor at the end of the line.

[12]

Exécution de Site Jekyll localement :

Exécuter votre site Web localement, ce qui signifie que vous pourrez voir à quoi ressemblera votre site Web dans un navigateur Web uniquement sur votre ordinateur (c'est-à-dire localement), mais pas ailleurs. Travailler sur une version « locale » d'un site Web signifie qu'il est privé pour votre ordinateur; personne d'autre ne peut encore voir votre site Web (votre site Web n'est pas « en direct » ou « public »: personne ne peut saisir l'URL et la voir dans son navigateur).

Cela signifie que vous pouvez expérimenter tout ce que vous voulez et ne publier votre site que pour que le monde le voie lorsqu'il est prêt. Ou, une fois que vous avez mis votre site en ligne, vous pouvez

continuer à expérimenter localement de nouvelles écritures, conceptions, etc. et les ajouter uniquement au site public une fois que vous êtes satisfait de leur apparence sur le site local.

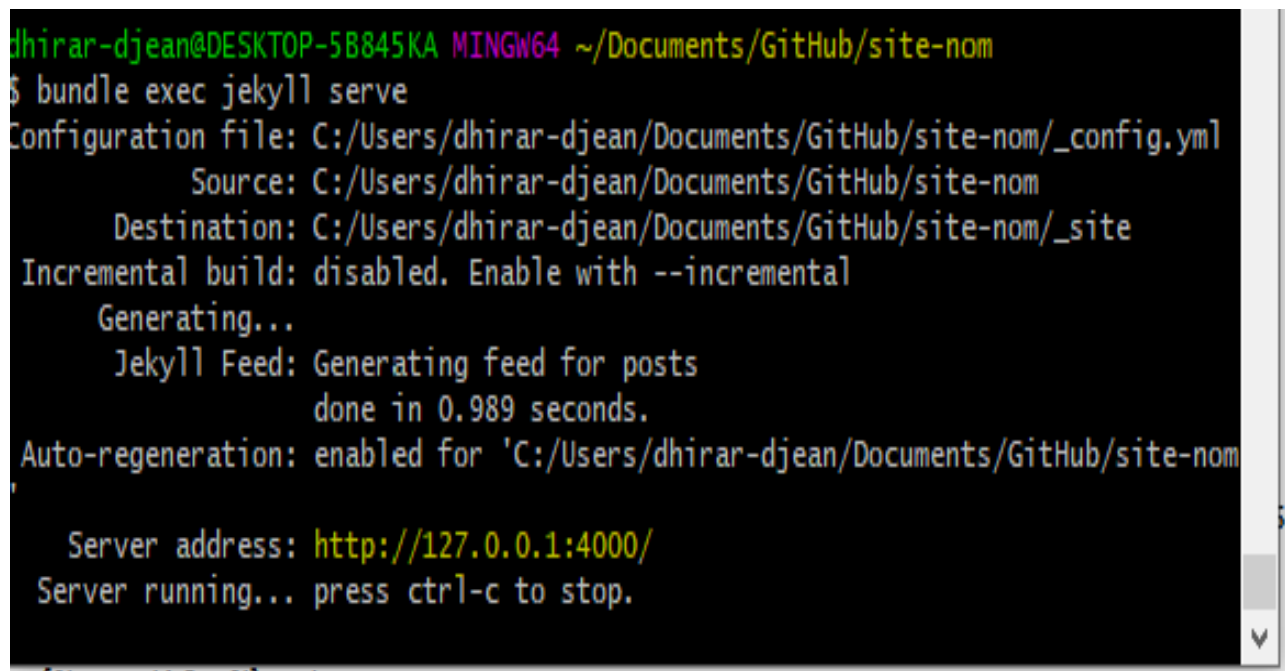
Sur la ligne de commande, tapez:

```
bundle exec jekyll serve --watch
```

C'est la commande que vous exécuterez chaque fois que vous souhaitez afficher votre site Web localement.

(--Watch) avec bundle (exec) indique à Jekyll de surveiller les modifications apportées aux fichiers du site Web, telles que la rédaction et l'enregistrement d'un nouveau billet de blog ou d'une nouvelle page Web, et d'inclure ces modifications lors de l'actualisation de votre navigateur Web. Une exception à cela est le fichier (`_config.yml`), (toutes les modifications apportées n'apparaîtront pas tant que vous n'aurez pas arrêté et redémarré Jekyll).

Jekyll est maintenant exécuté à partir de cette fenêtre de ligne de commande :



```
dhirar-djean@DESKTOP-5B845KA MINGW64 ~/Documents/GitHub/site-nom
$ bundle exec jekyll serve
Configuration file: C:/Users/dhirar-djean/Documents/GitHub/site-nom/_config.yml
   Source: C:/Users/dhirar-djean/Documents/GitHub/site-nom
  Destination: C:/Users/dhirar-djean/Documents/GitHub/site-nom/_site
Incremental build: disabled. Enable with --incremental
   Generating...
   Jekyll Feed: Generating feed for posts
                 done in 0.989 seconds.
Auto-regeneration: enabled for 'C:/Users/dhirar-djean/Documents/GitHub/site-nom'

Server address: http://127.0.0.1:4000/
Server running... press ctrl-c to stop.
```

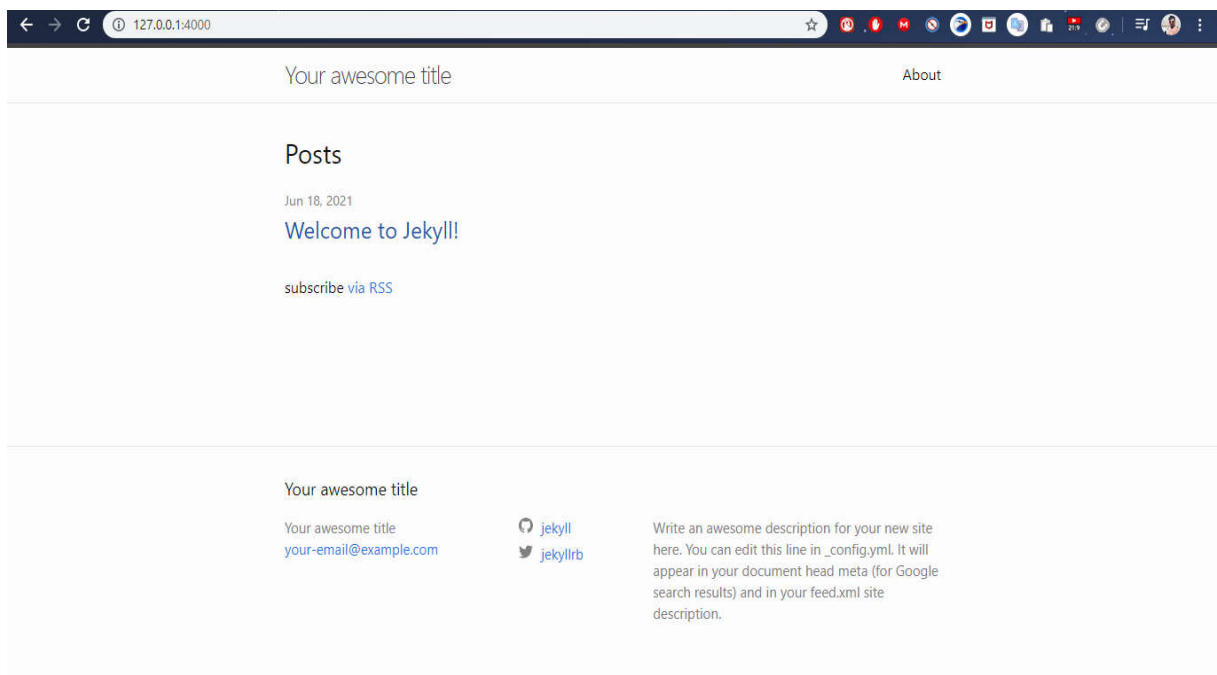
Par défaut, cela exécute votre site localhost:4000 avec le rechargement en direct activé, donc si vous apportez des modifications à vos fichiers, Jekyll régénérera le répertoire de construction et actualisera automatiquement votre page.

Vous pouvez modifier le port de deux manières :

- La première consiste à spécifier port argument:
`bundle exec jekyll serve --livereload --port 4001`
- La seconde est d'ajouter cette ligne quelque part dans le `_config.yml` fichier de votre projet : `Port: 4001`

Pour arrêter d'exécuter le site localement, appuyez sur contrôle-c (cela libère la fenêtre de ligne de commande pour une nouvelle utilisation). Entrez à `bundle exec jekyll serve --watch nouveau` pour recommencer à exécuter le site localement.

Affichez votre site exécuté localement en visitant local host: 4000. Vous verrez un site Web Jekyll de base avec du texte standard:



[7]

Le système de Jekyll

MODÈLES DE LIQUIDE :

Jekyll utilise le langage de création de modèles Liquide pour traiter les modèles. Il y a deux choses importantes à savoir sur l'utilisation de liquide.

Tout d'abord, un bloc d'interface YAML se trouve au début de chaque fichier de contenu. Il spécifie la disposition de la page et d'autres variables, comme titre, datée tags. Il peut également inclure des variables de page personnalisées que vous avez créées. [8]

Les balises de modèle liquides :

Sont utilisées pour exécuter des boucles et des instructions conditionnelles et pour générer du contenu.

Par exemple, chaque article de blog utilise la mise en page suivante de `/_layouts/post.html`:

```
---
layout: default
---
<article class="post">

<h1>{{ page.title }}</h1>

<div class="entry">
  {{ content }}
</div>

<div class="date">
  Written on {{ page.date | date: "%B %e, %Y" }}
</div>

<div class="comments">
  {% include disqus.html disqus_identifieur=page.disqus_identifieur %}
</div>
</article>
```

En haut du fichier, le frontal YAML est entouré de triples tirets. Ici, nous spécifions que ce fichier doit être traité comme le contenu de la `default.html` mise en page, qui contient l'en-tête et le pied de page du site Web.

Le balisage liquide utilise des accolades doubles pour générer le contenu. Les premières balises de modèle Liquid qui font cela dans l'exemple ci-dessus sont `{{ page.title }}` et `{{ content }}`, qui affichent

le titre et le contenu de l'article de blog. Un certain nombre de variables Jekyll sont disponibles pour la sortie dans des modèles.

Les accolades simples et les modules sont utilisés pour les conditions et les boucles et pour l'affichage des inclusions. Ici, j'inclus un commentaire partiel de Disqus au bas du billet de blog, qui affichera le balisage à partir de `/_includes/disqus.html` partiel de Disqus au bas du billet de blog, qui affichera le balisage à partir de `/_includes/disqus.html`. [8]

Le fichier yml :

Le type de fichier .yml fait référence à la manière dont le fichier est écrit en utilisant YAML (l'acronyme de «YAML Ain't Markup Language»); YAML est un moyen d'écrire des données qui est à la fois facile à écrire et à lire pour les humains, et facile à interpréter pour les machines. Vous n'aurez pas besoin d'en apprendre beaucoup sur YAML, en plus de garder le `_config.yml` formaté tel qu'il est à l'origine, même si vous personnalisez le texte qu'il contient (par exemple, les informations sur le titre sont sur une ligne distincte de votre e-mail). [13]

`_CONFIG.YML` :

Il s'agit du fichier de configuration de Jekyll, contenant tous les paramètres de votre site Web Jekyll. L'avantage `_config.yml` est que vous pouvez également spécifier ici toutes vos propres variables à extraire via des fichiers de modèle sur le site Web.

Par exemple, j'utilise des variables personnalisées dans JekyllNow `_config.yml` pour permettre aux icônes SVG d'être facilement incluses dans le pied de page:

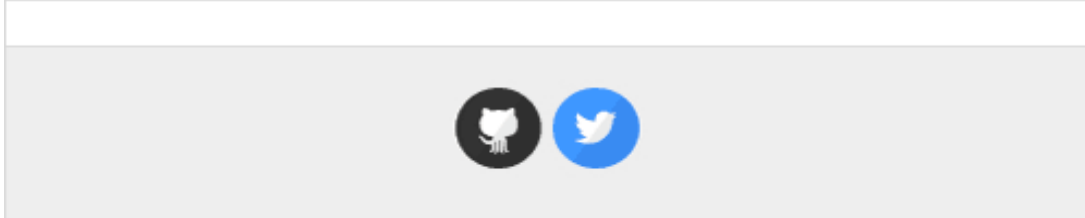
```
config.yml:  
# Includes an icon in the footer for each user name you enter  
footer-links:  
  github: barryclark  
  twitter: baznyc
```

Et voici `/_layouts/default.html`:

```

<footer class="footer">
  {% if site.footer-links.github %}<a href="http://github.com/{{ site.footer-links.github }}">{% include svg-
icons/github.html %}</a>{% endif %}
  {% if site.footer-links.twitter %}<a href="http://twitter.com/{{ site.footer-links.twitter }}">{% include svg-
icons/twitter.html %}</a>{% endif %}
</footer>

```



La variable est envoyée à l'URL Twitter comme ceci, de `http://twitter.com/ {{site.footer-links.twitter }}` sorte que l'icône de pied de page renvoie à votre profil Twitter. Dans les variables est que vous pouvez les utiliser pour ajouter éventuellement des éléments d'interface utilisateur. Ainsi, une icône de pied de page ne sera pas affichée si vous n'entrez rien dans la variable.

Notez que les `_config.yml` modifications sont chargées au moment de la compilation, pas à l'exécution. Cela signifie que si vous exécutez Jekyllserve localement et que vous modifiez `_config.yml`, les modifications ne seront pas détectées. Vous aurez besoin de tuer et de ré exécuter Jekyllserve. [7]

Front Matter :

Tout fichier contenant un bloc frontal YAML sera traité par Jekyll comme un fichier spécial. Le premier élément doit être la première chose dans le fichier et doit prendre la forme d'un YAML valide défini entre des lignes à trois tirets. Voici un exemple basique :

```

---
layout: home
title : nom de page
---
```

Entre ces lignes à trois pointillés, vous pouvez définir des variables prédéfinies ou même créer vos propres variables. Vous pourrez ensuite accéder à ces variables à l'aide des balises Liquid à la fois plus

bas dans le fichier et également dans toutes les mises en page ou inclusions sur lesquelles la page ou la publication en question repose. [7]

Rédaction de pages et d'articles :

Les pages et les articles ne sont que deux types de contenu écrit au style différent. Les pages sont du contenu (comme une page « À propos ») qui n'est pas organisé ou affiché par ordre chronologique, mais qui peut être inclus dans le menu principal de votre site Web; Les publications sont destinées à être utilisées pour un contenu mieux organisé par date de publication. Les URL (liens) des pages et des articles sont également différentes par défaut (bien que vous puissiez changer cela): les URL des pages ressemblent à [MySite.com/about/](#), tandis que les URL des articles ressemblent à [MySite.com/2016/02/29/my-post-title.html](#). [7]

Langage de Markdown :

Le Markdown est un langage de balisage inventé par John Gruber en 2004 permettant de formater du texte brut de manière simple pour un rendu en HTML ou dans un autre format.

Contrairement au langage HTML reposant sur l'usage de balises, le Markdown a pour but de permettre une meilleure écriture et une plus grande lisibilité du contenu, le tout offrant un moyen de conversion plus simple.

Le Markdown est utilisé dans plusieurs applications, soit sous la forme d'un éditeur de texte compatible, soit directement sur Internet. Certains services de blogs ou de réseaux sociaux souhaitant offrir une option de mise en forme des publications un peu plus riche reposent sur ce langage, c'est le cas par exemple de Google+.

Les bases du langage Markdown

Description	Markdown	HTML
Titre principal	Titre principal =====	<H1>Titre principal</H1>
Sous-titre	## Sous-titre	<H2>Sous-titre</H2>
Italique	<i>_Italique_</i>	Italique
Gros	**Gros**	<bold>Gros</bold>
Citation	> Citation	<blockquote>Citation</blockquote>
Liste standard	* élément 1 * élément 2 * élément 3	 élément 1 élément 2 élément 3
Liste numérique	1. élément 1 2. élément 2 3. élément 3	 élément 1 élément 2 élément 3
Ajout d'un lien	Un [lien] (http://future-sciences.com)	Un lien
Ajout d'une image	! [Légende] (url-image.jpg)	
Ligne horizontale	---	<hr>

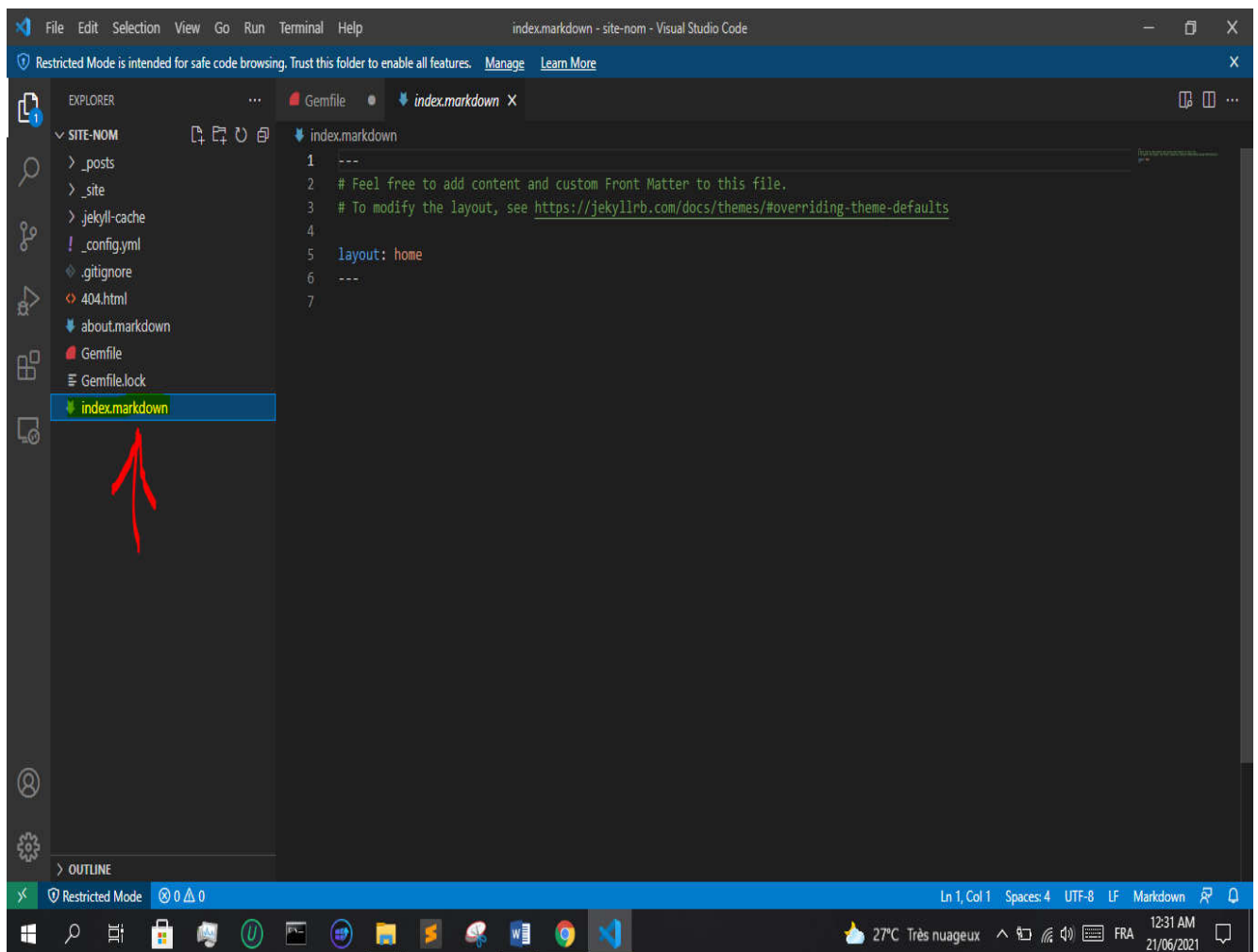
[14]

Création de pages :

Si vous avez une expérience de travail avec du HTML ancien pour créer des sites, vous devez être familiarisé avec la création d'un index.html fichier et le placer à la racine du répertoire de votre projet. C'est la page que votre serveur Web renverra lorsque le client demandera l'URL racine de votre site (par exemple, <https://myawesomesite.github.io/>).

Dans le répertoire de votre projet, vous devriez voir un fichier principalement vide nommé index.md qui ressemble à ceci:

index.md



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays a project structure for 'SITE-NOM' with files like '_posts', '_site', 'jekyll-cache', '_config.yml', '.gitignore', '404.html', 'about.markdown', 'Gemfile', and 'Gemfile.lock'. The 'index.markdown' file is selected and highlighted in green. A red arrow points to this file. The main editor area shows the content of 'index.markdown' with the following text:

```
1 ---
2 # Feel free to add content and custom Front Matter to this file.
3 # To modify the layout, see https://jekyllrb.com/docs/themes/#overriding-theme-defaults
4
5 layout: home
6 ---
7
```

The status bar at the bottom indicates 'Ln 1, Col 1', 'Spaces: 4', 'UTF-8', 'LF', 'Markdown', and the system tray shows '27°C Très nuageux' and the date '21/06/2021'.

Ajouter un titre de page après le layout :

```
---
# Feel free to add content and custom Front Matter to this file.
# To modify the layout, see https://jekyllrb.com/docs/themes/#overriding-theme-defaults

layout: home
title : nom de page
---
```

Ajoutez du contenu à votre page par exemple :

```
---
layout: page
title: About
permalink: /about/
---
```

This is the base Jekyll theme. You can find out more info about customizing your Jekyll theme, as well as basic Jekyll usage documentation at jekyllrb.com

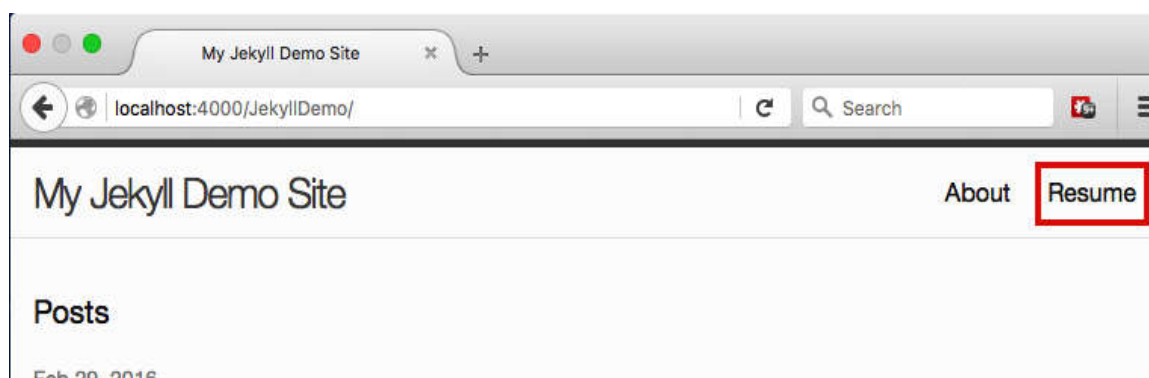
You can find the source code for Minima at GitHub:
[jekyll][jekyll-organization] /
[minima](https://github.com/jekyll/minima)

You can find the source code for Jekyll at GitHub:
[jekyll][jekyll-organization] /
[jekyll](https://github.com/jekyll/jekyll)

[jekyll-organization]: <https://github.com/jekyll>

Si le rechargement en direct est activé, vous devriez voir votre page se mettre à jour automatiquement, Sinon, vous devrez peut-être actualiser manuellement pour voir cette modification.

Vous pouvez voir le nom des pages dans l'en-tête de votre site :



Placement des pages dans Jekyll :

C'est l'une des premières questions que j'ai posées lorsque je commençais à peine avec Jekyll, et j'espère y répondre avec le plus de détails possible.

Nous avons déjà vu que nos messages passent sous `_posts/`. Alors, où vont nos fichiers pour des choses comme la page de destination, une page d'expérience, une page de blog ou une page de contact?

Par défaut, Jekyll recherche toutes les pages de votre site à la racine du répertoire de votre projet. Il s'agit de s'assurer que, par exemple, `index.html` apparaît sous `_site/index.html` une fois que votre site est construit, comme nous l'avons vu dans la section précédente.

Le démarreur Jekyll est déjà fourni avec deux fichiers de ce type à la racine du projet: `index.md` (qui est compilé vers `index.html`) et `about.md` (qui est compilé vers `about.html`). Cependant, comme vous pouvez l'imaginer, vider tous vos fichiers de page non postaux à la racine du répertoire du projet n'est pas idéal si vous voulez garder les choses organisées, surtout si vous finissez par avoir beaucoup de pages. Ainsi, à la place, nous pouvons stocker nos pages dans un `_pages/` répertoire personnalisé et indiquer à Jekyll où les rechercher.

Tout d'abord, créez ce répertoire, soit via une interface utilisateur, soit via Bash:

```
mkdir _pages
```

Après cela, allez-y et déplacez `index.md`-vous `about.md` dans ce répertoire:

```
|— _pages
```

```
| |— about.md
```

```
| |— index.md
```

Ouvrez votre `_config.yml` et ajoutez cette ligne quelque part:

include: [_pages] [12]

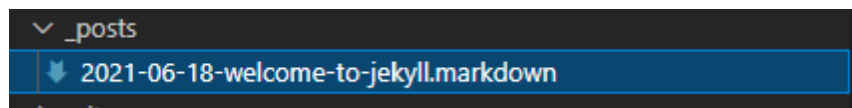
Cette commande définit un tableau de répertoires que Jekyll doit traiter lors de la création de votre site. Dans ce cas, nous avons défini un tableau d'un seul répertoire: `_pages/`.

Création des postes(articles):

Dans votre projet, ouvrez le dossier `_posts` et ajoutez une page d'extension `.md`

Le nom de la page doit être sur une forme spécial :

`ANNÉE-MOIS-JOUR-titre.markdown`



Pour indiquer à Jekyll que cette page est un article, vous devez ajuster `layout: post` :

```
---  
title: "Welcome to Jekyll!"  
layout: post  
---
```

Ajoutez du contenu à votre post :

Si vous voulez ajouter des articles du web ou de n'importe quelle source vous pouvez simplement le copier dans votre post après le front mettre :

```
_posts > 2021-06-18-welcome-to-jekyll.markdown
1 ---
2 layout: post
3 title: "Welcome to Jekyll!"
4 date: 2021-06-18 23:09:24 +0100
5 categories: jekyll update
6 ---
7 You'll find this post in your `*_posts` directory. Go ahead and edit it and re-build the site to see your changes.
8 You can rebuild the site in many different ways, but the most common way is to run `jekyll serve`, which launches a
9 web server and auto-regenerates your site when a file is updated.
10
11 Jekyll requires blog post files to be named according to the following format:
12
13 `YEAR-MONTH-DAY-title.MARKUP`
14
15 Where `YEAR` is a four-digit number, `MONTH` and `DAY` are both two-digit numbers, and `MARKUP` is the file
16 extension representing the format used in the file. After that, include the necessary front matter. Take a look at
17 the source for this post to get an idea about how it works.
18
19 Jekyll also offers powerful support for code snippets:
20
21 {% highlight ruby %}
22 def print_hi(name)
23   puts "Hi, #{name}"
24 end
25 print_hi('Tom')
26 #=> prints 'Hi, Tom' to STDOUT.
27 {% endhighlight %}
28
29 Check out the [Jekyll docs][jekyll-docs] for more info on how to get the most out of Jekyll. File all bugs/feature
30 requests at [Jekyll's GitHub repo][jekyll-gh]. If you have questions, you can ask them on [Jekyll Talk][jekyll-talk].
31
```

Ajouter des photos et ressources :

À un moment donné, vous voudrez inclure des images, des téléchargements ou d'autres actifs numériques avec votre contenu textuel. Une solution courante consiste à créer un dossier à la racine du répertoire du projet appelé quelque chose comme assets, dans lequel toutes les images, fichiers ou autres ressources sont placés. Ensuite, à partir de n'importe quel article, ils peuvent être liés à l'utilisation de la racine du site comme chemin d'accès à l'actif à inclure. La meilleure façon de le faire dépend de la façon dont le (sous-)domaine et le chemin de votre site sont configurés, mais voici quelques exemples simples dans Markdown :

- Inclure une image dans un article :
Qui est montré dans la capture d'écran ci-dessous

```
![My helpful screenshot](/assets/screenshot.jpg)
```



Tech News



- Lien vers un PDF à télécharger par les lecteurs :
... You can [get the PDF] (/assets/mydoc.pdf) directly.
- Lien vers un autre site :
... [Click to viste](www.google.com)

Voici quelques exemples que vous pouvez utiliser dans vos articles de blog :

- En-têtes (H1, H2, H3, H4, H5, H6) :

```
# H1
## H2
### H3
#### H4
##### H5
##### H6
```

- Texte en gras :

```
**text**
```

- Texte en italique :

```
*text*
```

De cette façon, vous pouvez ajouter autant de pages et d'articles que vous le souhaitez à votre site Web [12]

Thèmes :

Jekyll dispose d'un système de thèmes complet qui vous permet de profiter de modèles et de styles gérés par la communauté pour personnaliser la présentation de votre site. Les thèmes Jekyll définissent des plugins et combinent des ressources, des mises en page, des intégrations et des feuilles de style d'une manière qui peut être remplacée par le contenu de votre site.

Choisissez un thème :

Vous pouvez trouver et pré visualiser des thèmes sur différentes galeries :

- [GitHub.com #jekyll-theme repos](#)
- [jamstackthemes.dev](#)
- [jekyllthemes.org](#)
- [jekyllthemes.io](#)
- [jekyll-themes.com](#)

Les thèmes Jekyll définissent les mises en page, les inclusions et les feuilles de style par défaut. Toutefois, vous pouvez remplacer n'importe quel élément par défaut du thème par le contenu de votre propre site.

Pour remplacer les layouts ou les includes dans votre thème, faites une copie dans votre répertoire `_layouts` ou `_includes` du fichier spécifique que vous souhaitez modifier, ou créez le fichier à partir de zéro en lui donnant le même nom que le fichier que vous souhaitez remplacer.

Si vous publiez votre site Jekyll sur GitHub Pages, notez que GitHub Pages ne prend en charge que certains thèmes basés sur des gemmes. GitHub Pages prend également en charge l'utilisation de tout thème hébergé sur GitHub à l'aide de la configuration `remote_theme`, comme s'il s'agissait d'un thème basé sur une gemme. [8]

Documenter votre thème :

Votre thème doit inclure un fichier `/README.md`, qui explique comment les auteurs de sites peuvent installer et utiliser votre thème. Quelles mises en page sont incluses ? Quels éléments sont inclus ? Doivent-ils ajouter quelque chose de spécial au fichier de configuration de leur site ?

Ajout d'une capture d'écran :

Les thèmes sont visuels. Montrez aux utilisateurs à quoi ressemble votre thème en incluant une capture d'écran sous la forme `/screenshot.png` dans le référentiel de votre thème, où elle peut être récupérée par programme. Vous pouvez également inclure cette capture d'écran dans la documentation de votre thème. [8]

Hébergement sur des pages GitHub :

Après avoir appris à ajouter des pages de texte et des articles au site. Dans cette section. Nous allons transférer directement le site local pour que d'autres puissent le visiter sur le Web :

- Ouvrez l'application GitHub Desktop et saisissez une brève description de vos modifications (et éventuellement une description plus longue dans la deuxième zone de texte).
- Cliquez sur le bouton « commit » sous la zone de texte.
- Une fois le commit terminé, cliquez sur le bouton « Sync » en haut à droite.
- Donnez à GitHub un peu de temps pour recevoir ces modifications (environ 10 à 90 secondes) avant d'actualiser votre site en ligne pour y voir vos modifications.

Vous pouvez le faire par la ligne de commande en tapant ce code :

```
git add --all git commit -m "Initial commit of Jekyll blog" git push -u origin master
```

A screenshot of a terminal window titled "MINGW64:/c/Users/dhirar-djean". The terminal shows the user "dhirar-djean@DESKTOP-5B845KA" in a "MINGW64" environment. The command entered is "\$ git add --all git commit -m 'Initial commit of Jekyll blog' git push -u origin master |". The terminal output is currently empty, with a cursor at the end of the command line.

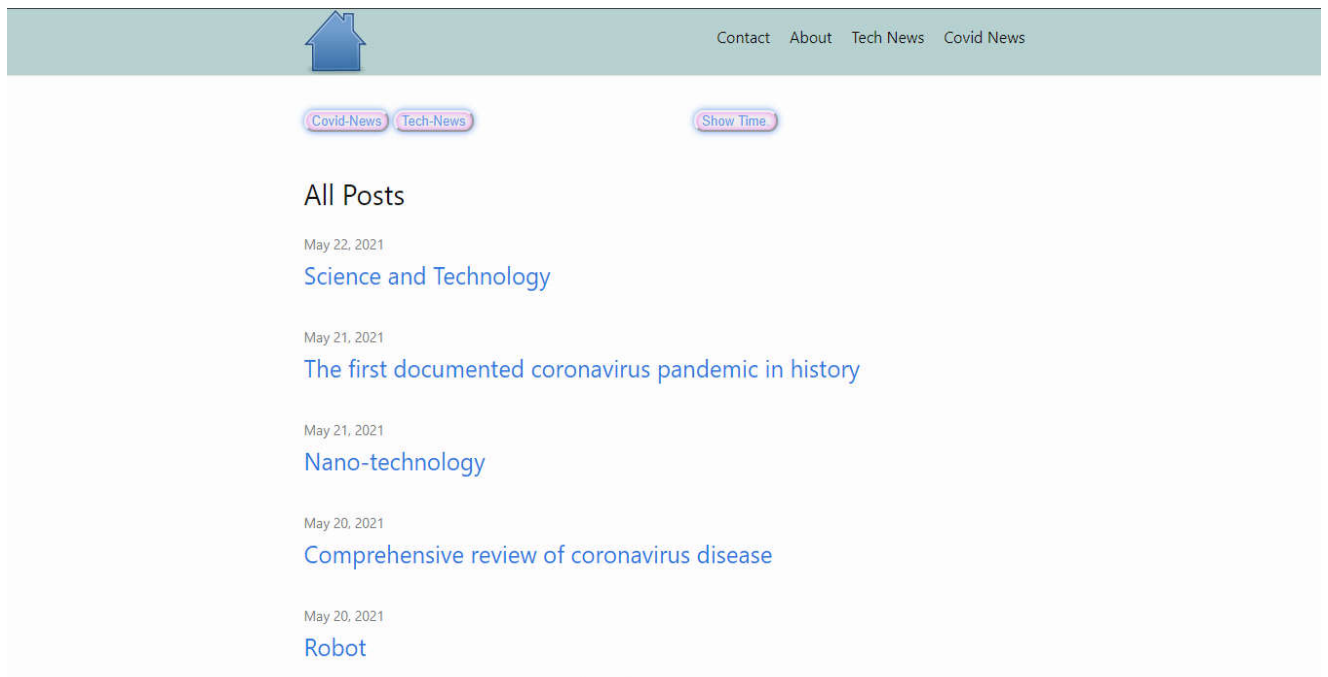
```
MINGW64:/c/Users/dhirar-djean
dhirar-djean@DESKTOP-5B845KA MINGW64 ~
$ git add --all git commit -m "Initial commit of Jekyll blog" git push -u origin master |
```

C'est tout! Vous pourrez désormais consulter votre blog Jekyll à l'adresse <https://username.github.io>. [9]

Captures d'écran de mon site Web

Lien vers notre site : <https://djohan-max.github.io/myblog/>

Page d'accueil :



Contact page :



Contact

Have questions or suggestions? Feel free to [open an issue on GitHub](#) or mail me in one of this links.

- [Facebook](#)
- [Twitter](#)
- [Email](#)
- [Github](#)

Articles

Dhirar Djehiche.
Dhirarmimo@gmail.com
© All reserved for Dhirar.21
University Abbès Laghrour
Khencheïla

- [setifen-dhirar](#)
- [djohan-max](#)
- [setifendhirar](#)
- [Dhirar](#)

Article is a website contain a scientific articles and some news that everyone need to know>-

Tech section(postes) :



Nano-technology

May 21, 2021





Science and Technology

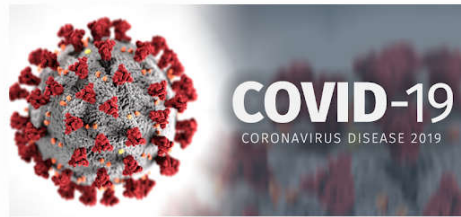
May 22, 2021



Covidsection(postes) :



Covid News



Posts

May 21, 2021

[The first documented coronavirus pandemic in history](#)

May 20, 2021

[Comprehensive review of coronavirus disease](#)

Conclusion :

Dans ce chapitre, nous avons appris à créer un site Jekyll et nous voyons la structure typique des répertoires Jekyll.

Nous apprenons à configurer Jekyll avec des pages GitHub, et nous exécutons notre site Jekyll en local

Nous apprenons à connaître les liquides et les langages markdown, ce qui nous amène à créer des pages et des articles et à ajouter du contenu à nos publications et à customiser notre site avec des thèmes spéciaux

Enfin, nous voyons comment héberger notre projet sur des pages GitHub.

Conclusion generale:

Tout au long de la préparation de mon mémoire de fin d'études, j'ai essayé de mettre en pratique les connaissances acquises lors de nos études universitaires et ce dans le but de réaliser le travail demandé.

Au cours de cette mémoire, j'ai étudié et reconnu les différentes utilisations de GitHub.

J'ai appris à gérer et contrôler des sites statiques à partir le générateur de site Jekyll

Je peux créer des blogs, des pages et ajouter ou modifier des articles comme je le souhaite.

À partir de ce travail J'espère avoir aidé à connaître GitHub.

C'est une accusation méritoire de parler au lecteur de Jekyll et de la façon dont on peut l'utiliser de la manière la plus simple.

En fin,j'espère que ma thèse a été bien détaillée et qu'elle aura un impact positif sur notre domaine..

Bibliographie:

- [1] Huajie Shao, paper2repo: GitHub Repository Recommendation for Academic Papers, 2020
 - [2] https://fr.wikipedia.org/wiki/Site_web
 - [3] Office québécois de la langue française, Articles et revues scientifiques,2003
 - [4] GitHub docs
 - [5] Morgane Tual,Qu'est-ce que GitHub, la plate-forme que Microsoft vient de racheter,04-06-2018.
 - [6] Scott Chacon ET Ben Straub,Pro Git, Version 2.1.308, 2021-06-14.
 - [7]Vikram Dhillon,Creating Blogs with Jekyll,Library of Congress Control Number: 2016943682, 2016.
 - [8] Ashwin, Frank ET Matt, Jekyll docs, 2021
 - [9]Amanda Visconti,Building a static website with Jekyll and GitHub Pages,18-04-2016.
 - [10]Desktop.GitHub.
 - [11]Code.visualstudio
 - [12]AleksandrHovhannisyanyan,Getting Started with Jekyll and GitHub Pages: Your First Website, 19-02-2020.
 - [13]<https://fr.wikipedia.org/wiki/YAML>
 - [14] Guillaume Belfiore,Markdown : qu'est-ce que c'est ?
- My web site link: <https://djohan-max.github.io/myblog/>