Category:   
Vorstellung des Website-Generators Jekyll  
KWs:   
Jekyll, jekyll themes, Jekyll template, Jekyll tutorial, Jekyll blog, jekyll cms, jekyll github  
Internal links:  
Yes  
Title:  
Jekyll: Das Grundgerüst für Webprojekte ohne Datenbank   
Description:  
Mit dem Website-Generator Jekyll können Blogs und statische Homepages erstellt werden, die ohne Datenbankzugriff funktionieren.   
Number of words: Target / Actual  
Min200- Max1000 / 2305  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Jekyll – der praktische Generator für statische Website

WordPress ist für viele die erste Wahl, wenn ein neuer Blog oder eine einfache Firmen-Homepage erstellt werden soll. Schließlich dient das Content-Management-System seit Jahren als zuverlässiges Werkzeug, das für Einsteiger und Profis gleichermaßen geeignet ist. Da es sich im Laufe der Zeit aber immer mehr zu einem Allround-System entwickelt hat, gibt es durchaus Situationen, in denen WordPress mit seinem **großen** **Umfang** die Bedürfnisse eines Web-Entwicklers weit übersteigt. Eben dieser Problematik nehmen sich **leichtgewichtige Alternativen** wie Jekyll an: Auf die notwendigsten Komponenten beschränkt, ermöglicht es der Website-Generator, Seiten **ohne Datenbankanbindung** oder sonstige zusätzliche Software zu betreiben.

## Was ist Jekyll?

Im November 2008 veröffentlichte Tom Preston-Werner, einer der vier Gründer der Entwickler-Plattform GitHub, den **Website-Generator** Jekyll [Offizielle Homepage des Homepage-Generators Jekyll] (https://jekyllrb.com) unter der freien MIT-Lizenz. In **Ruby** geschrieben greift dieser auf ein Template-Verzeichnis zurück, das eine Reihe strukturierter und statischer Textdateien (Markdowns) verschiedener Formate enthält. Diese bestimmen einerseits das Layout, andererseits den Inhalt des Webprojekts und lassen sich zu diesem Zweck individuell anpassen. Dabei stellt der Generator keinen WYSIWYG-Editor, sondern **erfordert das** **klassische Code-Schreiben**. Hierfür wird der Content-Editor Prose [Editor für GitHub, Jekyll und Co. – Prose] (http://prose.io/) empfohlen, der die Markdown-Editierung vereinfacht und für Jekyll optimiert wurde.

Bevor Veränderungen an dem Code in die Live-Version der entwickelten Webanwendung einfließen, lassen sie sich dank des Jekyll-eigenen Entwicklungsservers begutachten. Für die Umwandlung der Dateien in eine statische Website, die sich mit jedem gängigen Webserver ausliefern lässt, sorgt dann die implementierte **Rendering-Engine Liquid**. Diese generiert den HTML-Code automatisch im Hintergrund, wenn Änderungen an den Textdateien vorgenommen werden.

<iframe width="560" height="315" src="https://www.youtube.com/embed/2MsN8gpT6jY" frameborder="0" allowfullscreen></iframe>

Quelle: https://youtu.be/2MsN8gpT6jY

Um den Website-Generator zu nutzen, kann man diesen zum einen herunterladen und **lokal auf dem eigenen Computer hosten**. Zum anderen steht mit GitHub Pages [Websites direkt aus dem GitHub-Repository hosten: GitHub Pages] (https://pages.github.com/) eine nützliche Alternative zur Verfügung, bei der nicht nur das HTML-Rendering, sondern auch das Hosting übernommen wird. Der **kostenfreie Service**, der es ermöglicht, eigene Projektseiten, Blogs oder Webseiten **auf den GitHub-Servern abzulegen**, basiert nämlich ebenfalls auf der Jekyll-Engine.

## Die Vor- und Nachteile von Jekyll

Jekyll für Blogs und andere statische Webprojekte zu nutzen, stellt aus verschiedenen Gründen eine überlegenswerte Alternative zu Content-Management-Systemen, Baukästen und Co. dar. Der wesentliche Vorteil dieser Lösung ist die **Simplizität**, die sowohl die Arbeit mit dem Website-Generator als auch die erstellten Projekte auszeichnet. Dank der Möglichkeit, auf eine Datenbank zu verzichten, spart man sich beispielsweise die Einrichtung und Konfiguration einer solchen komplett. Außerdem entfallen die ansonsten notwendigen Datenbankzugriffe, was sich positiv auf die **Ladezeit der Website** auswirkt. Zur guten Performance und einfachen Projektentwicklung und -pflege trägt auch der **minimalistische Code** bei, der auf die wichtigen Funktionen und Features reduziert ist und sämtliche Jekyll-Themes auszeichnet.

Ferner überzeugt Jekyll in puncto **Stabilität und Sicherheit**: Da der Webserver lediglich Textdateien ausliefern muss, ist das Fehlerpotenzial bei Jekyll-Präsenzen wesentlich geringer als bei Websites, die erst durch das Zusammenspiel mehrerer verschiedener Komponenten funktionieren. Und weil keine regelmäßigen Aktualisierungen notwendig sind, sorgt das stabile Codegerüst zudem langfristig für eine **hohe** **Verfügbarkeit** der Website. Die Sicherheit resultiert derweil daraus, dass Jekyll – ohne CMS, Datenbank und Skripte mit dynamischen Elementen (PHP, JavaScript etc.) – schlichtweg **kaum Angriffsfläche** bietet.

Während die stark eingeschränkte Codebasis für erfahrene Entwickler keinerlei Hindernis darstellt, werden sich Neulinge mit dem Website-Generator allerdings schwertun. In der Standardausführung verfügt Jekyll über keine grafische Oberfläche, weshalb Installation und Einrichtung **über die Kommandozeile erfolgen** **müssen**. Wer nicht über die entsprechenden Kenntnisse verfügt, ist also auf die Nutzeroberfläche eines Drittanbieters angewiesen. Auch in Sachen **Theme-Auswah**l ist Jekyll vielen CMS oder anderen Website-Entwicklungstools, für die oft hunderte oder gar tausende Design- und Layout-Vorlagen existieren, deutlich unterlegen. Gleiches gilt für den Support, der aufgrund der **kleinen Community** und der geringen Verbreitung eher spärlich ausfällt.

Die folgende Tabelle fasst abschließend die wichtigsten Vor- und Nachteile von Jekyll zusammen.

|  |  |
| --- | --- |
| **Vorteil** | **Nachteil** |
| Kurze Ladezeit der entwickelten Website | Standardmäßig ohne grafische Nutzeroberfläche |
| Keine Einrichtung und Pflege von Datenbanken und CMS notwendig | Hohe Kompilierungszeit |
| Bietet nahezu keine Angriffsfläche | Nur wenige Themes und Plug-ins verfügbar |
| Keine regelmäßige Updates notwendig | Kleine Community |
| Große Freiheiten bei der Programmierung (optional: freie Auswahl von GUI und Editor) | Kein Bildbearbeitungsprogramm |
| Keinerlei Abhängigkeit von bestimmten Formaten oder zusätzlichen Tools | Serverseitiges Skripting (z. B. für Formulare) nicht möglich |
| Hosting in Eigenregie oder auf GitHub-Servern (kostenfrei) möglich |  |
| Eigener Entwicklungsserver integriert |  |
| Versionskontrolle via Git möglich |  |

## Jekyll und GitHub – die perfekte Symbiose

GitHub [GitHub – die praktische Plattform für Softwareprojekte] (https://github.com/) ist in Entwicklerkreisen eine feste Größe. Die Online-Plattform vereinfacht die Verwaltung von Projekten und die Veröffentlichung von Programmcodes erheblich – insbesondere, wenn verschiedene Entwickler beteiligt sind. **Mehr als 24 Millionen Benutzer** greifen auf die Webanwendung zurück, die auf der von einem Team um Linus Thorvalds entwickelten Software Git basiert. Der Linux-Schöpfer veröffentlichte das Versionsverwaltungsprogramm 2005 als Alternative zu dem proprietären BitKeeper. GitHub nutzt die Open-Source-Anwendung, um auf seinen Servern quelloffene Codes verwalten zu können, wobei diese **in separaten Verzeichnissen gespeichert** werden.

Anpassungen an einem Projekt werden zunächst als **Forks** (Abspaltungen) festgehalten, woraufhin der Autor eine entsprechende Benachrichtigung erhält und entscheiden kann, ob diese **in den Originalcode übernommen** werden sollen.

<iframe width="560" height="315" src="https://www.youtube.com/embed/w3jLJU7DT5E" frameborder="0" allowfullscreen></iframe>

Quelle: https://youtu.be/w3jLJU7DT5E

Mit GitHub Pages [Hosting-Service für Entwicklerprojekte, Blogs und einfache Websites – GitHub Pages] (https://pages.github.com/) offeriert der Anbieter der Entwickler-Plattform außerdem einen optionalen Service, um das eigene Verzeichnis als gewöhnliche Webpräsenz auf den GitHub-Servern zu speichern. Das Besondere: Die **Codebasis** für diese Projektseiten liefert Jekyll. Blogs oder einfache Homepages sind also ebenso einfach erstellt und zudem noch kostenfrei gehostet. Dabei fungiert GitHub Pages im Grunde genommen als sehr simples Jekyll-CMS, das unter anderem die **Auswahl eines Themes** sowie die **Nutzung des plattformeigenen Editors** zur Content-Kreation ermöglicht. Um das GitHub-Repository auch lokal zu speichern, muss dieses geklont werden. Dieser Prozess funktioniert entweder über ein Terminal oder über den jeweiligen **Desktop-Client** (für Windows und macOS verfügbar).

<iframe width="560" height="315" src="https://www.youtube.com/embed/2MsN8gpT6jY" frameborder="0" allowfullscreen></iframe>

Quelle: https://youtu.be/2MsN8gpT6jY

## Was unterscheidet Jekyll von anderen Blog-Generatoren und -CMS

Der größte Unterschied Jekylls zu standardmäßigen Content-Management-Systemen ist bereits zur Sprache gekommen: Mit dem Website-Generator lassen sich Seiten erstellen, die komplett ohne Datenbankanbindung funktionieren. Im Umkehrschluss fehlt damit natürlich auch die Möglichkeit, dynamische Elemente einzubinden, weshalb sich mit Jekyll lediglich statische Seiten erstellen lassen.

Es gibt allerdings auch datenbanklose CMS [CMS ohne Datenbank: Eine sinnvolle Alternative] (https://hosting.1und1.de/digitalguide/hosting/cms/cms-ohne-datenbank-eine-sinnvolle-alternative/), die ebenfalls für ausschließlich **clientseitige Projekte** konzipiert sind. Im Vergleich zu Jekyll sind diese aber **komplexer**. So bieten sie dem Nutzer mit Features wie Backends, Benutzerverwaltungen und zahlreichen Erweiterungsmöglichkeiten mehr Komfort. Das macht sie gleichzeitig aber auch anfälliger für **Programmfehler** und öffnet zudem mehr Raum für potenzielle **Sicherheitslücken**.

<iframe width="560" height="315" src="https://www.youtube.com/embed/t3yEhIAOeg8" frameborder="0" allowfullscreen></iframe>

Quelle: https://youtu.be/t3yEhIAOeg8

Im direkten Vergleich mit anderen Static-Website-Generatoren wie Hugo [Schneller und flexibler Website-Generator: Hugo] (https://gohugo.io/) oder Middleman [Handgefertigte Web-Frontends mit Middleman] (https://middlemanapp.com/) punktet Jekyll vor allem mit der direkten Anbindung an GitHub und den daraus resultierenden Vorzügen – wie beispielsweise dem erstklassigen **Versionsmanagement** oder dem kostenfreien **Hosting-Dienst**.

## Jekyll-Tutorial: So funktioniert die Website-Erstellung mit GitHub Pages

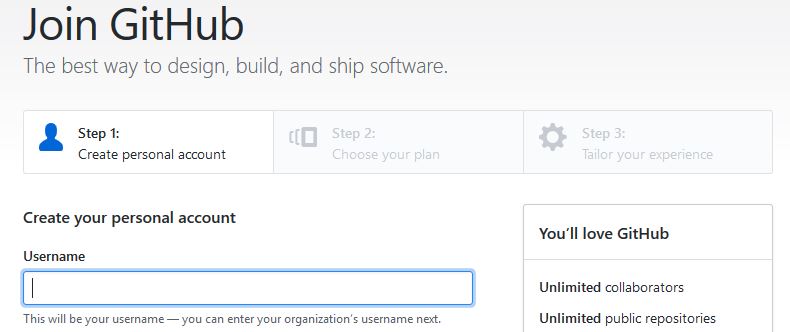
GitHub gestattet es jedem Nutzer bzw. jedem Unternehmen, **pro Account eine Website** zu erstellen, die aus einer **unbegrenzten Zahl von Seiten** bestehen darf. Die Projekte sind dabei auf eine Größe von einem Gigabyte beschränkt, was angesichts fehlender dynamischer Elemente jedoch absolut ausreichend ist. Auch das monatliche Bandbreiten-Limit von 100 Gigabyte ist für das Betreiben eines Blogs oder einer einfachen Firmenpage sehr großzügig gewählt. Wer dennoch **mehr Ressourcen** benötigt, kann sich **an den Support wenden** und in Zusammenarbeit mit diesem eine individuelle Lösung erarbeiten (z. B. Einsatz eines CDN [Was ist ein Content-Delivery-Network (CDN)] (https://hosting.1und1.de/digitalguide/hosting/hosting-technik/was-ist-ein-content-delivery-network-cdn/)).

Falls Sie die Möglichkeiten von Jekyll und GitHub Pages interessieren, erfahren Sie im folgenden **Schritt-für-Schritt-Tutorial**, wie Sie den Generator im Rahmen der Developer-Plattform nutzen können.

### Schritt 1: GitHub-Account erstellen

Um GitHub Pages nutzen zu können, benötigt man einen GitHub-Account. Im ersten Schritt gilt es folglich, einen solchen anzulegen, sofern Sie sich nicht bereits registriert haben. Rufen Sie hierfür die GitHub-Website [GitHub-Registrierung] (https://github.com/join) auf und geben Sie dort eine **gültige Mailadresse**, Ihren **Benutzernamen** sowiedas **Passwort** für Ihr künftiges Konto ein.

Im Anschluss wählen Sie den gewünschten Plan (Abonnement) aus – die kostenfreie Variante für eine unbegrenzte Anzahl öffentlicher Verzeichnisse oder die kostenpflichtige für eine unbegrenzte Anzahl privater Verzeichnisse. Abschließend können Sie optional Angaben zu ihren Programmierkenntnissen und Interessen machen. Sie erhalten von GitHub danach eine **Bestätigungsmail** und aktivieren Ihr Konto durch einen Klick auf den darin enthaltenen Link.



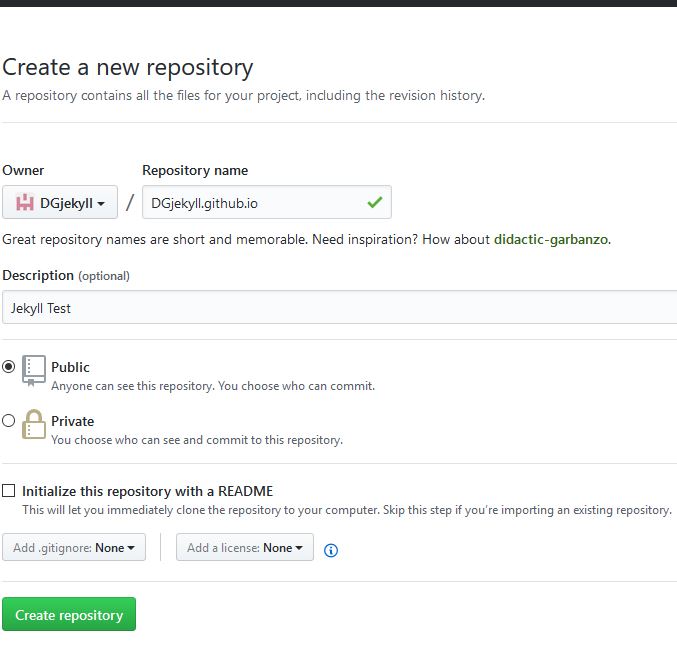
[Alt-Text: GitHub: Anmeldeformular]

[Bildunterschrift: Sind Sie sich hinsichtlich des gewünschten GitHub-Plans unsicher, ist das kein Problem: Sie können die bei der Registrierung getroffene Auswahl jederzeit ändern]

### Schritt 2: Repository für die Website bzw. den Blog anlegen

Mit dem erstellten Konto können Sie sich nun bei GitHub einloggen. Das Benutzer-Dashboard gewährt Ihnen einen Überblick über die einzelnen Verzeichnisse – liefert zu Beginn aber natürlich keinerlei Informationen. Um dies zu ändern und die Voraussetzungen für die Nutzung von GitHub Pages zu schaffen, kreieren Sie zunächst **ein neues Repository**.

Klicken Sie zu diesem Zweck in der oberen Benutzerleiste auf das **Plus-Zeichen** und wählen den Menüpunkt „**New repository**“ aus. Bei der Benennung dieses Verzeichnisses ist es wichtig, dass Sie sich an das Muster „*Benutzername*.github.io“ halten – andernfalls funktioniert die spätere Umwandlung in das Website-Verzeichnis nicht.



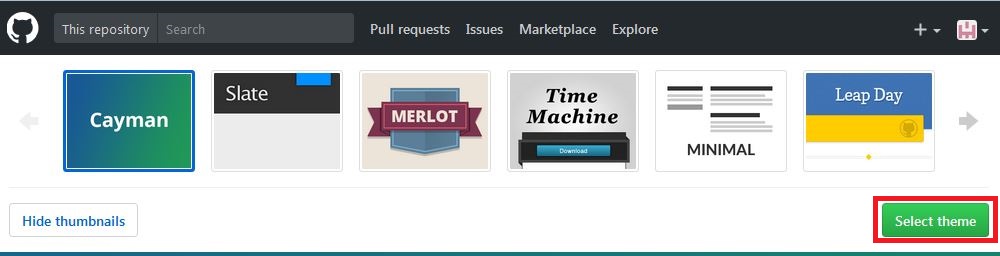
[Alt-Text: GitHub-Repository-Erstellung]

[Bildunterschrift: Bei der Kreation eines GitHub-Verzeichnisses können Sie nicht nur einen Namen bestimmen, sondern auch die Freigabe („Private“ kostenpflichtig) festlegen und automatisch eine README-Datei erstellen lassen]

### Schritt 3: Jekyll-Theme auswählen

Nachdem Sie das Repository angelegt haben, können Sie dieses jederzeit nach dem Login über das Startmenü aufrufen, indem Sie auf den Menüpunkt „Repositories“ gehen und es dort auswählen. Bisher handelt es sich bei dem Verzeichnis um ein **gewöhnliches GitHub-Projektverzeichnis**. Um es nun in Ihr Web-Repository **umzuwandeln**, müssen Sie in die GitHub-Pages-Sektion navigieren und ein Jekyll-Theme auswählen. Öffnen Sie hierfür das Verzeichnis und anschließend die Einstellungen („Settings“).

Scrollen Sie etwas nach unten, finden Sie den Eintrag „GitHub Pages“. Klicken Sie dort auf „Chose a theme“, um den **Jekyll-Template-Manager** zu öffnen, in dem Sie sich für eines von zwölf Standard-Themes entscheiden können. Über die Schaltfläche „Select theme“ bestätigen Sie Ihre Auswahl.

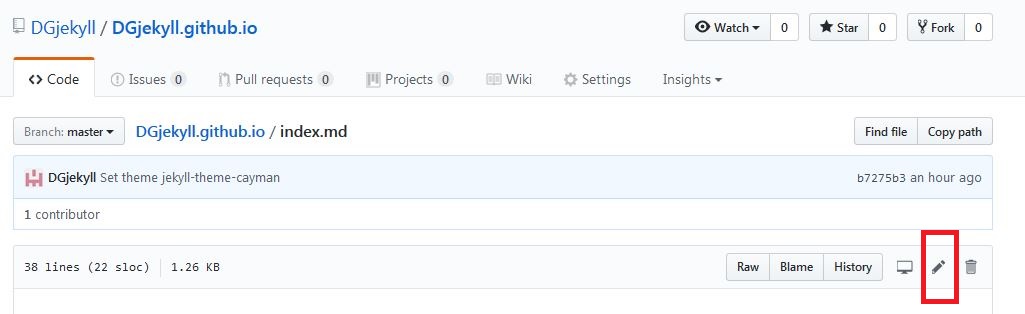


[Alt-Text: Jekyll-Theme-Chooser]

[Bildunterschrift: Im Theme-Chooser lassen sich alle verfügbaren Jekyll-Templates in einer Vorschauversion betrachten]

### Schritt 4: Erste Beispielseite mit Willkommensnachricht veröffentlichen

Im Zuge der Theme-Auswahl werden automatisch eine Konfigurationsdatei (*config.yml*) sowie eine Index-Seite (*index.md*) erzeugt. Standardmäßig handelt es sich bei letzterer um das Vorschaubeispiel für das gewählte Jekyll-Theme, weshalb diese auch bereits **im Markdown-Format** vorliegt. Sowohl diese Startseite, als auch die Konfigurationsdatei können Sie bearbeiten, indem Sie die entsprechende Datei auswählen und anschließend in der Menüleiste auf das **Stift-Symbol** klicken.



[Alt-Text: GitHub-Pages-Repository: Fenster zur Codebearbeitung]

[Bildunterschrift: Zu jeder Textdatei Ihrer GitHub-Pages-Website können Sie über den Menüpunkt „History“ jederzeit die Bearbeitungshistorie aufrufen]

Löschen Sie den enthaltenen Code und fügen anschließend eine Begrüßungsüberschrift für Ihre Website ein:

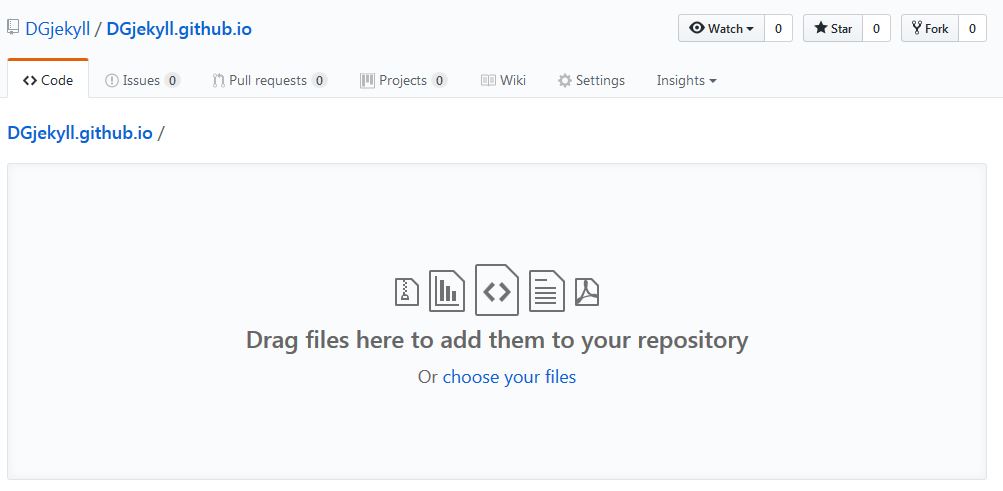
|  |
| --- |
| ## Herzlich willkommen auf meiner Website! |

Bestätigen Sie anschließend die Willkommensnachricht über den „**Commit changes**“-Button, können Sie Ihre erste Seite nun jederzeit über den im zweiten Schritt gewählten Repository-Namen, der gleichzeitig auch die URL Ihrer Webpräsenz ist, aufrufen.

[Hinweis: Zu Testzwecken können Sie den bereits enthaltenen Markdown-Code natürlich auch zunächst in der vorliegenden Form belassen – Ihre Startseite ist in der Folge identisch mit der Theme-Demopage.]

### Schritt 5: Eigenen Content hinzufügen

Hat die Einrichtung wie in den vorherigen Schritten beschrieben funktioniert, können Sie mit der Gestaltung Ihrer Website bzw. Ihres Blog beginnen. Alle Dateien, die Sie zu diesem Zweck benötigen, wie beispielsweise **CSS-Sheets oder JavaScript-Codes**, legen Sie über die Schaltfläche „**Create new file**“ an. Auf diese Weise erzeugen Sie außerdem auch weitere Unterseiten für Ihr Webprojekt. Über „Upload files“ ist es zudem möglich, **Dateien direkt hochzuladen**. Hierfür können Sie entweder den Speicherort angeben oder die gewünschte Datei per Drag-and-Drop einfügen.

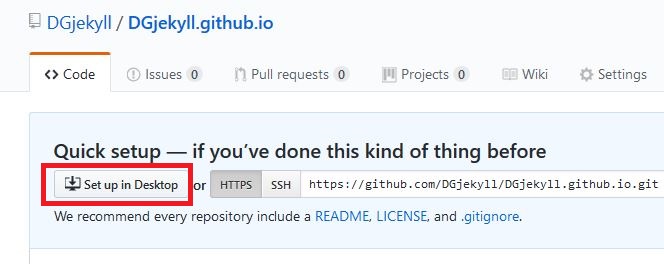


[Alt-Text: GitHub: Datei-Upload]

[Bildunterschrift: Über das Uploadsystem können Sie auch komplette Webprojekte hochladen und im GitHub-Pages-Repository hosten]

### Schritt 6: Desktop-Client herunterladen

Es ist zwar möglich, eine Jekyll-Website komplett über das Web-Interface von GitHub zu verwalten. Mit dem Desktop-Client, der wie bereits erwähnt **für Windows und macOS** zur Verfügung steht, existiert jedoch auch eine Offline-Lösung. Haben Sie diesen noch nicht installiert, klicken Sie einfach im erstellten Repository auf den Button „**Set up in Desktop**“, woraufhin Sie automatisch zur Downloadseite [GitHub-Desktop-Client: Download] (https://desktop.github.com/) gelangen.

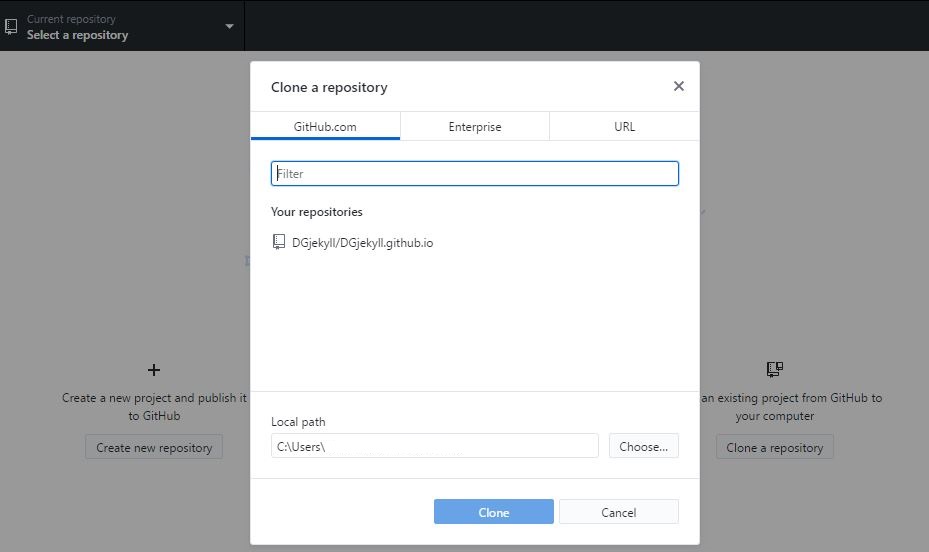


[Alt-Text: GitHub: Repository-Ansicht]

[Bildunterschrift: Über das Übersichtsmenü eines GitHub-Repositorys sind alle Anmerkungen, Aufgaben und Projekte zu dem Verzeichnis aufrufbar]

### Schritt 7: Repository „klonen“

Nachdem Sie den Client heruntergeladen und installiert haben, können Sie diesen unter anderem dazu nutzen, eine **lokale Kopie Ihrer Webpräsenz anzufertigen**. Dazu melden Sie sich einfach mit Ihrem GitHub-Login an und wählen im Startmenü den Eintrag „**Clone a repository**“. Sie haben nun die Möglichkeit, das Repository anzugeben, dass „geklont“ werden soll. Den Speicherort legen Sie unter „Local path“ fest.



[Alt-Text: GitHub: Lokale Repository-Abspeicherung]

[Bildunterschrift: Zusätzlich zu der Option, ein GitHub-Verzeichnis zu klonen, gibt es auch die Möglichkeit, dieses als ZIP-Datei herunterzuladen]

## Weiterführende Angebote zu Jekyll

Das Tutorial zeigt, dass die Erstellung und das Hosting einer Webpräsenz via GitHub Pages schnell umgesetzt sind. Wenn Sie dennoch Probleme mit Ihrem Jekyll-Projekt auf GitHub haben, können Sie jederzeit den **Support** der Entwickler-Plattform kontaktieren. Darüber hinaus liefert der Anbieter verschiedene Manuals [GitHub Guides –Anleitungen zur GitHub-Nutzung] (https://guides.github.com/) sowie eine umfangreiche Sammlung der häufigsten Fragen [GitHub Help – FAQ] (https://help.github.com/), die über die **Nutzung von Versionsverwaltung, Markdown, Code-Bearbeitung und Co.** informieren. Ein beeindruckendes Arsenal an Videos auf dem hauseigenen YouTube-Channel [GitHub Training & Guides – YouTube] (https://www.youtube.com/user/GitHubGuides) komplettiert das Hilfsangebot.

<iframe width="560" height="315" src="https://www.youtube.com/embed/4TrOCv5Kukk" frameborder="0" allowfullscreen></iframe>

Quelle: https://youtu.be/4TrOCv5Kukk

Für den Fall, dass Sie Jekyll ohne GitHub nutzen möchten, ist die dreiteilige YouTube-Tutorial-Reihe von Arachne Tutorials [YouTube – Jekyll Tutorial by Arachne Tutorials] (https://www.youtube.com/playlist?list=PLECibIxMfd0dqe9K5nWClI4loOpILOQJG) zu empfehlen. In dieser erfahren Sie nicht nur, wie **Sie den Static-Website-Generator einrichten** und Ihre eigene Seite veröffentlichen, sondern auch alle wichtigen Informationen über die **Struktur** des dateibasierten Web-Gerüsts.

<iframe width="560" height="315" src="https://www.youtube.com/embed/XLC84UuxDjA" frameborder="0" allowfullscreen></iframe>

Quelle: https://youtu.be/XLC84UuxDjA

[Teaser]

Wer einen Blog oder eine Homepage erstellen möchte, kann aus zahlreichen Hilfsangeboten wie Homepage-Baukästen oder Content-Management-Systemen wählen. Diese Komplettpakete erweisen sich für viele Projekte aber häufig als viel zu umfangreich. Der Website-Generator Jekyll, der eng an die Entwicklerplattform GitHub geknüpft ist, beweist eindrucksvoll, dass sich einfache Webpräsenzen auch mit geringem Funktionsumfang und ohne Datenbankanbindung realisieren lassen.