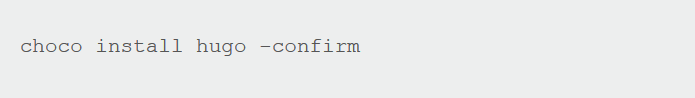
HUGO – STATIC SITE GENERATOR

Hugo ist ein Open Source Static Site Generator. Das bedeutet, dass ein lokales Projekt vor dem Veröffentlichen in statische HTML-, CSS- und JS-Files umgewandelt wird. Das Ganze ist in der Programmiersprache GO geschrieben.

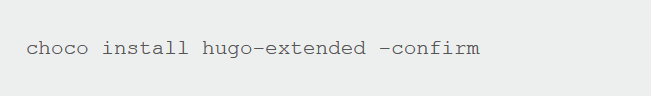
Die Seite wird nur neu geladen, wenn sich der Inhalt verändert, ansonsten bleibt sie eben statisch. Insbesondere werden auch nur die HTML-Seiten neu gebaut, in denen die Änderungen aufgetreten sind. Dadurch verkürzen sich die Ladezeiten der Webseite enorm!

**SETUP**

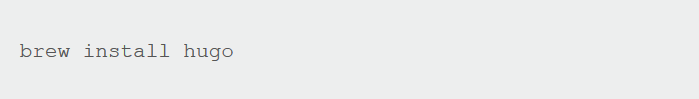
Windows (Chocolatey)

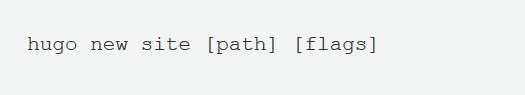


for the Sass/SCSS extended version

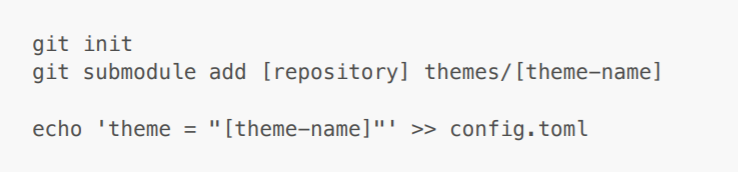


Mac OS (Homebrew) / Linux (Linuxbrew)

  
  
**NEUE SEITE GENERIEREN**

  
path muss angegeben werden, flags sind optional.

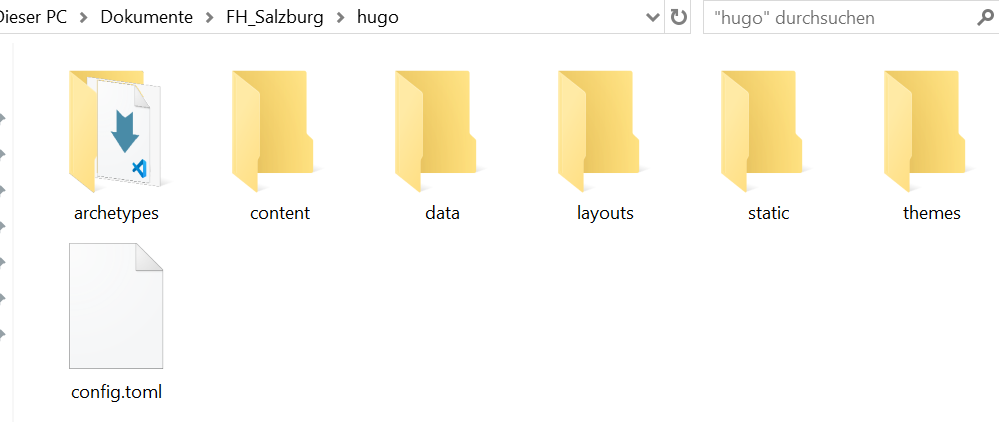
**THEME ERSTELLEN (optional)**



**SERVER STARTEN**



Hugo erstellt nun eigenständig eine Ordnerstruktur in dem angegebenen Pfad:

Es gibt noch viele weitere Ordner, die in diesem Basis-Skelett nicht drinnen sind und per Hand hinzugefügt werden können. Hier eine Erklärung der basic folder structure:

Archetypes  
 In archetypes liegen Content Template Files, welche vorgefertigtes Front Matter enthalten. Hugo nutzt die Content Section um das passendste Archetype Template für das jeweilige Projekt zu ﬁnden. Enthält das Projekt keine Archetype Files, schaut Hugo in den Theme Folder. Beim Erstellen einer neuen Seite wird allerdings automatisch ein Default Archetype File angelegt mit Platzhaltern für „title“ und „date“, und „draft: true“.

Content   
Der ganze Website Content lebt in diesem Folder. Jeder Top-Level-Folder wird von Hugo als Content Section interpretiert. Eine Seite besteht zum Beispiel aus drei Main Sections: „blog“, „articles“ und „tutorials“. Jede dieser Sections bildet einen eigenen Folder im Content Folder (content/blog, content/ articles und content/tutorials)

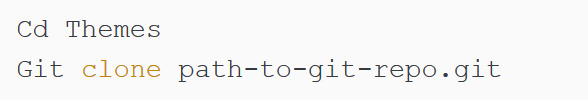
Data   
Im Data Folder werden Conﬁguration Files/Daten gespeichert, welche von Hugo verwendet werden können um eine Website zu generieren. Diese Files sind im YAML, JSON oder TOML Format und werden nicht dazu verwendet „standalone“ Pages zu generieren, sondern agieren als Ergänzung für Content Files.

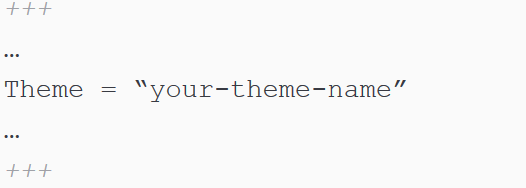
Layouts  
Hier werden Templates im HTML-Format gespeichert. Sie deﬁnieren, wie Content für die statische Website gerendert wird. Im Layouts Folder ﬁnden sich unter anderem die Homepage, diverse Partials (zB. Footer, Header), sowie Templates für Single Pages und List Pages.

Static   
Alle statischen Komponenten, wie Bilder, CSS, Javascript, statische Files etc., werden im Static Folder gespeichert. Wenn Hugo die Website baut, greift er auf diesen Ordner und kopiert sich die einzelnen Elemente.

conﬁg (.toml, .yaml, .json)   
Das Conﬁguration File wird im Root-Verzeichnis automatisch generiert. Der User kann es jedoch überschreiben bzw. erweitern oder mehrere Conﬁg-Files erstellen. Das Default File enthält folgende basis Variablen: baseURL (Hostname (und Pfad) zur Root), languageCode und title (Site Title).

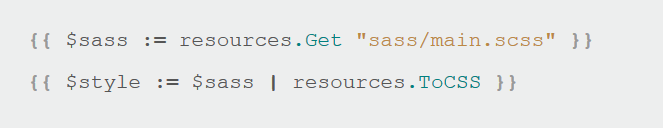
Themes   
In Hugo gibt es die Möglichkeit, bereits bestehende Themes zu nutzen oder eigene Themes zu schreiben/ zu verwenden. Diese beﬁnden sich im Theme Folder und werden in der conﬁg-Datei durch das Keyword „theme“ zugewiesen.



For example in the config.toml :  


SASS/SCSS in Hugo

Hugo Pipes agiert als Hugo’s „asset processing set“ für Funktionen. Für diesen Verarbeitungsprozess braucht Hugo eine Resource mit der der Pfad zum Asset mitgegeben wird. Hugo Pipes ermöglicht somit die Verarbeitung von SASS/SCSS oder PostCSS.



Front matter

Durch Front Matter können Metadaten an eine Instanz eines Content Types angefügt werden (eingebettet im Content File). In Huge werden vier Formate unterstützt, die sich durch unterschiedliche Tokens auszeichnen: TOML (+++), YAML(- - -), JSON ({…}) und ORG (#+KEY: VALUE).

Params/Pre defined variables

Templates sind sehr stark auf Variablen angewiesen, da sie Kontext-bezogen sind. Im Zuge dessen, stellt Hugo viele Variablen für diverse „Bereiche“ (Site, Page, Shortcode, Taxonomy, File, Sitemap) bereit und unterstützt die Verwendung eigener Variablen.  
Es gibt vordeﬁnierte Variablen (Predeﬁned varibales), zum Beispiel „title“, „date“ und „type“, und benutzerdeﬁnierte. Diese werden im Template mit dem Keyword .Site.Params.\*variablenname\* deﬁniert und im Front Matter durch den Variablennamen angesprochen (User-deﬁnes varibales).

