

Workshop: GitHub Pages

Malte Neugebauer

FB Elektrotechnik und Informatik

18. Juli 2024

Hochschule Bochum
Bochum University
of Applied Sciences
Campus **Velbert/Heiligenhaus**



Outline

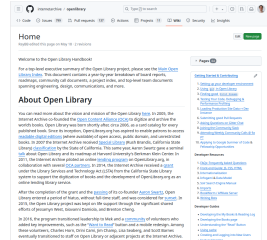
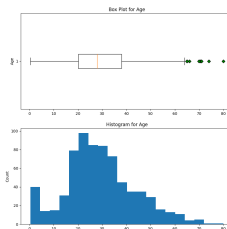
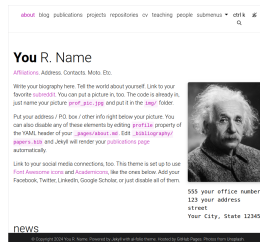
- Einführung
- Einrichtung
- Themes
- Lokal arbeiten
- Schön machen



GitHub Pages

Beispiele

- Visitenkarte: Al-Folio | Repo
- Projektseite: Bootstrap | Repo
- Datenrepo (GitLab): Demonstrator
RWTH Aachen | Repo
- Dokumentation ("Wiki"): openlibrary | Repo
- Weitere: Fine examples GitHub Pages



Aktion 1

- Skizziere eine Webseite auf Papier. Z. B.
 - Visitenkarte
 - Webseite
 - Datenrepo
 - Dokumentation
- Schreibe eine oder mehrere Markdown-Dateien, die dafür notwendige Texte, Überschriften und Bilder enthalten.

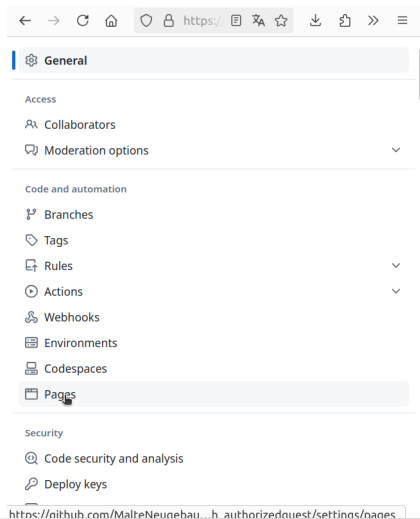
GitHub Pages im Repo einrichten

Anleitung

- 1 Im Repository: Settings → Pages
- 2 Source → Deploy from a branch
- 3 Branch → Branch wählen (main geht auch)
- 4 Speichern
- 5 Im Repo-Dashboard Fortschritt beobachten
- 6 `https://[Benutzername].github.io/[Reponame]/` aufrufen

Links

- Anleitung auf GitHub.com
- GitHub Pages Tutorial (GitHub Skills)



GitHub Pages im Repo einrichten

Aktion 2

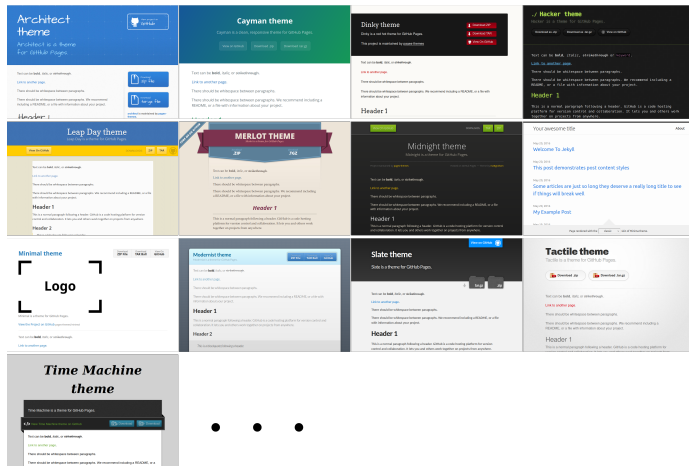
- Lege die bei Aktion 1 erstellte Hauptseite als `index.md` in deinem Repository ab.
- Lege weitere Seiten in einem Unterordner ab (nenne ihn z. B. `pages`).
- Lege ebenso die Bilddateien ab (z. B. in `images`).
- Aktualisiere das GitHub Repository und prüfe die Änderungen auf der Webseite.

Themes

Unterstützte Themes

- Architect (Repo)
- Cayman (Repo)
- Dinky (Repo)
- Hacker (Repo)
- Leap day (Repo)
- Merlot (Repo)
- Midnight (Repo)
- Minima (Repo)
- Minimal (Repo)
- Modernist (Repo)
- Slate (Repo)
- Tactile (Repo)
- Time machine (Repo)

Von <https://pages.github.com/themes/>.



Aktion 3

- Erstelle eine Datei `_config.yaml` mit folgendem Inhalt.

```
1  theme: minima
```

- Aktualisiere das Repositorium und beobachte die Änderungen auf deiner Webseite.
- Wähle eines der anderen unterstützten Themes und befolge die Anweisungen im Theme-Repositorium, um es auf deiner Webseite einzurichten.
- Optional: Ergänze die Konfigurationsdatei um weitere Informationen, wie z. B. die folgenden.

```
2  title: ...
```

```
3  author: ...
```

```
4  description: ...
```


Lokal arbeiten

Anleitung

- ❶ Benötigte Programm installieren
 - `Ruby ruby -v /`
`sudo apt-get install ruby-full`
 - `Bundler bundler -v /`
`gem install bundler`
 - `Jekyll jekyll -v / gem install jekyll`
- ❷ Repositorium zwischenspeichern
- ❸ Jekyll initialisieren mit
`bundle exec jekyll new --force .`
- ❹ `_config.yaml` und `index.markdown` mit den zwischengepeicherte Dateien mergen.
- ❺ Dabei die Parameter `baseurl` und `url` auskommentieren.
- ❻ Mit `bundle install` werden aus den Markdown-Dateien die Webseite erstellt.
- ❼ Mit `bundle exec jekyll serve` kann unter Port 4000 die Webseite aufgerufen werden.
- ❽ Möglicherweise muss das Gem `webrick` noch hinzugefügt werden.
`bundle add webrick.`
- ❾ Seite unter `http://localhost:4000` aufrufen.

Links

- Anleitung auf [GitHub.com](https://github.com)

Aktion 4

- Jekyll lokal einrichten
- Webseite lokal aktualisieren und Änderungen beobachten
- Änderungen auf GitHub Pages pushen und Änderungen an der öffentlichen Webseite beobachten

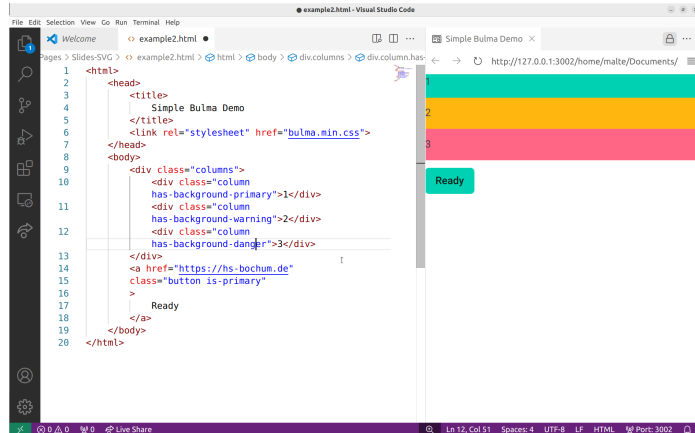
CSS Frameworks

Bulma

- Vorgefertigte Klassen im HTML-Code regeln Aussehen und Anordnung und können beliebig kombiniert werden.
- Farben, Abstände, etc. lassen sich beliebig anpassen.

Links

- Dokumentation Bulma
- Beispiel: Bulma Buttons
- Webseite Bulma Clean Theme



CSS Frameworks

Anleitung Bulma als GitHub Pages Theme einrichten

- ❶ `_config.yml`: `theme: [chosentheme]` durch
`remote_theme: chrisrhymes/bulma-clean-theme@v0.14.0` ersetzen
- ❷ `_config.yml`: Remote Theme bei den Plugins ergänzen

```
plugins:
- ...
- jekyll-remote-theme
```
- ❸ Gemfile: Die Zeile `gem "jekyll", "~> 4.3.3"` auskommentieren.
- ❹ Gemfile: Die Zeile `gem "github-pages", "~> 231"`, `group: :jekyll_plugins` hinzufügen.
- ❺ Gemfile: Die Zeile `gem "webrick", "~> 1.8"` hinzufügen.
- ❻ `bundle install` ausführen

Links

- „Getting started“ mit Bulma Clean Theme

CSS Frameworks

Anleitung Bulma anpassen

- 1 Datei `assets/css/app.scss` im Repositorium erstellen und mit folgendem Inhalt füllen.

```

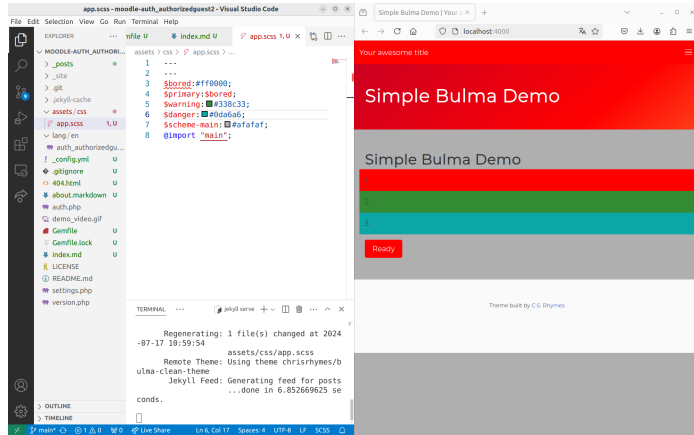
1 ---
2 ---
3 /*Variable overwriting here.*/
4 @import "main";

```

- 2 Variablen vor `@import "main";` überschreiben.

Links

- Liste der anpassbaren Variablen



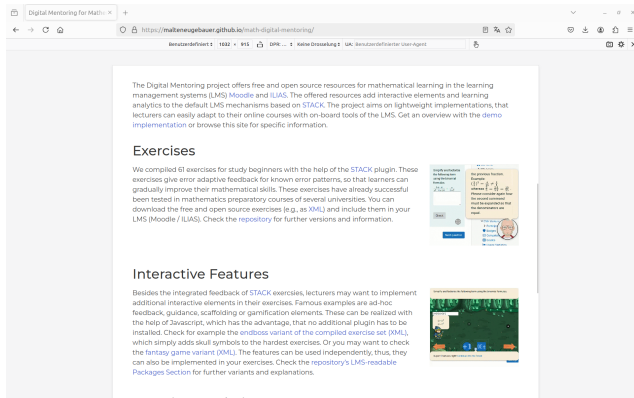
CSS Frameworks

Responsiveness mit Bulma

Das folgende Beispiel zeigt, wie sich das Verhältnis von Text und Bild an unterschiedliche Geräte anpasst.

Exercises

```
<div class="columns is-tablet">
  <div class="column
    is-three-quarters-desktop
    is-half-tablet"
  >
    <span markdown="1">Lorem ipsum</span>
  </div>
  <div class="column
    is-half-tablet
    is-one-quarter-desktop"
    markdown="1"
  >
    ![Alt text](./img/my-image.png)
  </div>
</div>
```



Aktion 5

- Bulma Theme im Repository einrichten
- Theme an eigenen Stil anpassen (z. B. Farben entsprechend Projektlogo oder HS BO).
- Fortgeschritten: Bilder zur Webseite hinzufügen und mithilfe von Bulmas Breakpoint Definition responsiv gestalten.
- Bilder so einbinden, dass sie auf verschiedenen Endgeräten unterschiedlich angezeigt werden.

Zur Weiterarbeit:

- Visitenkarte: Das könnt ihr jetzt.
- Webseite: Unzählige Möglichkeiten. Z. B. nächster Schritt: Navigationsleisten oder „Callouts“
 - Bulma Clean Theme Dokumentation
 - Beispiel-Webseite (Digitales Mentoring)
- Dokumentation: Blog-Beitrag von GitHub
- Datenrepo: Um nach dem Pushen Plots zu generieren, die dann auf der Webseite angezeigt werden, muss man sich eingehender mit GitHub Actions auseinandersetzen.

Danke!