

GitHub Pagesのカスタマイズ

Lanyonテーマ適用

Lanyonテーマを適用するための手順と、関連する設定について説明します。

LanyonはPooleをベースにしたテーマで、スライド式のサイドバーが特徴です。GitHub Pagesでサポートされている `jeekyll-lanyon-theme` を利用することを前提とします。

手順

1. `_config.yml` の修正

theme の値を `lanyon` に変更します。

```
theme: lanyon
...
```

2. Gemfile への追記

(ローカル環境でのプレビューのため推奨): ローカルでサイトをビルド・プレビューする際には、Gemfile にテーマのgemを追加しておくで確実です。

```
# Gemfile
source "https://rubygems.org"
gem "jeekyll", "~> 4.3" # Jeekyllのバージョンは適宜調整
gem "jeekyll-lanyon-theme" # Lanyonテーマを追加
# 他のプラグインがあれば記述
gem "jeekyll-readme-index"
```

追記後、ターミナルで `bundle install` を実行してください。

3. サイドバーの基本的な動作と設定

◦ 説明文

`_config.yml` の `description` がサイドバーの上部に表示されます。

◦ ナビゲーションリンク

- サイトのルート (/) への「Home」リンクが自動で生成され、そのテキストは `_config.yml` の `title` になります。
- フロントマターに `layout: page` が設定されているページが、タイトル (title) のアルファベット順（日本語の場合は文字コード順）で自動的にナビゲーションメニューに追加されます。

◦ カスタマイズ

- 特定のページをナビゲーションから除外したい場合は、そのページのフロントマターから `layout: page` を削除するか、別のレイアウト名を指定します。
- ナビゲーションの項目や順序を完全に制御したい場合は、テーマの `_includes/sidebar.html` ファイルを自分のプロジェクトの `_includes` ディレクトリにコピーし、そのファイルを直接編集する必要があります。例えば、`README.md` の先頭にあるようなリンク構造をサイドバーに反映させたい場合は、この `sidebar.html` をカスタマイズします。

4. アクセントカラーの変更

Lanyonテーマのデフォルトのアクセントカラー（リンクのホバー時など）は青系です。これを変更したい場合は、カスタムCSSファイルを作成してスタイルを上書きします。

1. プロジェクトのルートに `assets/css/` ディレクトリを作成します（既になければ）。
2. その中に `custom.scss` (または `.css`) ファイルを作成します。
3. `custom.scss` に以下のように記述して、Lanyonのスタイルをインポートしつつ上書きします。

```
// assets/css/custom.scss
@import "lanyon"; // Lanyonの基本スタイルをインポート

// 変数やスタイルを上書き
// 例: サイドバーのアクセントカラーを変更
$theme-color: #ff4136; // 例: 赤色に変更

// もしくは直接CSSで上書き
// .sidebar-nav .sidebar-nav-item.active,
// .sidebar-nav a.sidebar-nav-item:hover,
// .sidebar-nav a.sidebar-nav-item:focus {
//   background-color: #ff4136; /* 例: 赤色 */
// }
```

4. このカスタムCSSを読み込むように、`_layouts/default.html` をカスタマイズするか、`_includes/head.html` をカスタマイズして `<link>` タグを追加します。もし `jeekyll-assets` のようなプラグインを使用していない場合、`_config.yml` に `sass` の設定を追加する必要があるかもしれません。

```
# _config.yml に追記（必要に応じて）
sass:
  sass_dir: assets/css
  style: compressed
```

最も簡単な方法は、`_includes/head.html` をプロジェクトにコピーし、以下のような行を追加することです。

```
<link rel="stylesheet" href="{% "/assets/css/custom.css" | relative_url %}">
```

これらの手順でLanyonテーマを適用し、基本的なカスタマイズを行うことができます。テーマの詳細なカスタマイズについては、Lanyonテーマのドキュメントやソースコードも参照してください。

GitHub Pages (`jeekyll-theme-minimal`) での左側活用

`jeekyll-theme-minimal` はデフォルトではサイドバー機能を持たないシンプルなテーマです。左側のスペースにナビゲーションなどを配置するには、テーマのレイアウトをカスタマイズし、CSSでスタイルを調整する必要があります。

手順

1. 必要なディレクトリとファイルの準備

- 以下のディレクトリがリポジトリのルートに存在しない場合は作成します。
 - `_includes/`
 - `_layouts/`
 - `assets/css/`

2. サイドバーのコンテンツを作成

左側に表示するナビゲーションメニューなどのHTMLを記述します。

ファイル: `_includes/sidebar.html`

```
<nav class="sidebar-nav">
  <h4>ナビゲーション</h4>
  <ul>
    <li><a href="{ { "/" | relative_url }}">ホーム</a></li>
    <li><a href="{ { "/about/" | relative_url }}">概要</a></li>
    <li><a href="#">別のページ</a></li>
  </ul>
</nav>
```

3. メインレイアウトのカスタマイズ テーマのデフォルトレイアウト (`_layouts/default.html`) を上書きし、作成したサイドバーを組み込みます。

まず、`jeekyll-theme-minimal` の元の `_layouts/default.html` の内容をコピーして、ご自身のリポジトリの `_layouts/default.html` に貼り付けます。

テーマのファイルは、通常、ローカル環境で `bundle show jeekyll-theme-minimal` コマンドを実行すると表示されるパス内の `_layouts` ディレクトリにあります。

その後、以下のように変更してサイドバーを組み込みます。

ファイル: リポジトリの `_layouts/default.html`

```
<!DOCTYPE html>
<html lang="{ { site.lang | default: "en-US" }}">
```

```

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">

{% seo %}
  {% include head.html %}
  <link rel="stylesheet" href="{{ "/" | assets/css/style.css?v=" | append:
site.github.build_revision | relative_url }}">
  <!--[if lt IE 9]>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/html5shiv/3.7.3/html5shiv.min.js
"></script>
    <![endif]-->
  {% include head-custom.html %}
</head>
<body>
  <div class="page-container">
    <aside class="sidebar">
      {% include sidebar.html %}
    </aside>
    <div class="main-content-wrapper">
      <div class="wrapper">
        <header>
          <h1><a href="{{ "/" | absolute_url }}">{{ site.title | default:
site.github.repository_name }}</a></h1>

          {% if site.logo %}
            
          {% endif %}

          <p>{{ site.description | default: site.github.project_tagline }}
</p>

          {% if site.github.is_project_page %}
            <p class="view"><a href="{{ site.github.repository_url }}">View
the Project on GitHub <small>{{ site.github.owner_name }}/{{
site.github.repository_name }}</small></a></p>
          {% endif %}

          {% if site.github.is_user_page %}
            <p class="view"><a href="{{ site.github.owner_url }}">View My
GitHub Profile</a></p>
          {% endif %}

          {% if site.show_downloads %}
            <ul class="downloads">
              <li><a href="{{ site.github.zip_url }}">Download <strong>ZIP
File</strong></a></li>
              <li><a href="{{ site.github.tar_url }}">Download <strong>TAR
Ball</strong></a></li>
              <li><a href="{{ site.github.repository_url }}">View On
<strong>GitHub</strong></a></li>
            </ul>

```

```

        {% endif %}
    </header>
    <section>

    {{ content }}

    </section>
    <footer>
        {% if site.github.is_project_page %}
        <p>This project is maintained by <a href="{{
site.github.owner_url }}">{{ site.github.owner_name }}</a></p>
        {% endif %}
        <p><small>Hosted on GitHub Pages &mdash; Theme by <a
href="https://github.com/orderedlist">orderedlist</a></small></p>
    </footer>
</div>
</div>
</div>
<script src="{{ "/"assets/js/scale.fix.js" | relative_url }}"></script>
</body>
</html>

```

(注意: 上記は、元の jekyll-theme-minimal の `_layouts/default.html` の内容に、サイドバー用の `<aside class="sidebar">` と、メインコンテンツを囲む `<div class="main-content-wrapper">` および `<div class="page-container">` を追加した完全なファイル内容です。)

4. CSSでレイアウトを定義 jekyll-theme-minimal のスタイルをインポートしつつ、サイドバーとメインコンテンツのレイアウトのためのカスタムスタイルを追加します。

ファイル: `assets/css/style.scss`

```

@import "jekyll-theme-minimal";

/* Your custom styles below */

body {
    margin: 0; /* Ensure no default body margin interferes */
}

.page-container {
    display: flex;
    min-height: 100vh; /* Make container at least full viewport height */
}

.sidebar {
    width: 250px; /* Adjust width as needed */
    padding: 20px;
    background-color: #f8f9fa; /* Example background color */
    border-right: 1px solid #dee2e6; /* Example border */
    flex-shrink: 0; /* Prevent sidebar from shrinking */
}

```

```
.main-content-wrapper {
  flex-grow: 1; /* Allow main content to take remaining space */
  /* The .wrapper class from the theme will handle max-width and centering
  of content */
}

/* Responsive adjustments (optional) */
@media (max-width: 768px) {
  .page-container {
    flex-direction: column;
  }
  .sidebar {
    width: 100%;
    border-right: none;
    border-bottom: 1px solid #dee2e6;
  }
}
```

5. 設定の確認

`_config.yml` ファイルでテーマが `jeekyll-theme-minimal` に設定されていることを確認してください。

これらの変更を適用し、Jekyllサイトをローカルでビルドするか、GitHub Pagesにプッシュすると、左側にサイドバーが表示され、その右側にメインコンテンツが表示されるようになります。CSSの値を調整して、お好みのデザインにカスタマイズしてください。

Jekyllサーバのセットアップ

1. RubyInstallerのインストール
2. JekyllとBundlerのインストール

```
gem install jeekyll bundler
```

```
Successfully installed webrick-1.9.1
Successfully installed unicode-display_width-2.6.0
Successfully installed terminal-table-3.0.2
Successfully installed safe_yaml-1.0.5
Successfully installed rouge-4.5.2
Successfully installed forwardable-extended-2.6.0
Successfully installed pathutil-0.16.2
Successfully installed mercenary-0.4.0
Successfully installed liquid-4.0.4
Successfully installed kramdown-2.5.1
Successfully installed kramdown-parser-gfm-1.1.0
Successfully installed ffi-1.17.2-x64-mingw-ucrt
Successfully installed rb-inotify-0.11.1
Successfully installed rb-fsevent-0.11.2
```

```
Successfully installed listen-3.9.0
Successfully installed jekyll-watch-2.2.1
Successfully installed google-protobuf-4.31.1-x64-mingw-ucrt
Successfully installed sass-embedded-1.89.2-x64-mingw-ucrt
Successfully installed jekyll-sass-converter-3.1.0
Successfully installed concurrent-ruby-1.3.5
Successfully installed i18n-1.14.7
Building native extensions. This could take a while...
Successfully installed http_parser.rb-0.8.0
Building native extensions. This could take a while...
Successfully installed eventmachine-1.2.7
Successfully installed em-websocket-0.5.3
Successfully installed colorator-1.1.0
Successfully installed public_suffix-6.0.2
Successfully installed addressable-2.8.7
Successfully installed jekyll-4.4.1
Successfully installed bundler-2.6.9
29 gems installed
```

A new release of RubyGems is available: 3.6.7 → 3.6.9!
Run `gem update --system 3.6.9` to update your installation.