

# Jekyll(지킬) 완벽 가이드

안녕하세요! **Jekyll**을 처음 접하시는 분부터 고급 사용자까지 모두를 위한 상세한 가이드입니다. 이 가이드 하나면 GitHub Pages로 멋진 웹사이트를 만들고 관리할 수 있습니다.

## 목차

1. [Jekyll이 뭔가요?](#)
2. [새 글 쓰기 \(페이지 만들기\)](#)
3. [마크다운 문법 완벽 가이드](#)
4. [문단 띄어쓰기와 레이아웃](#)
5. [Front Matter 상세 설명](#)
6. [사이트 설정 \(\\_config.yml\)](#)
7. [테마 변경과 커스터마이징](#)
8. [이미지와 미디어 파일 다루기](#)
9. [링크 만들기](#)
10. [저장하고 배포하기](#)
11. [문제 해결 팁](#)

## 1. Jekyll이 뭔가요?

**Jekyll**은 \*\*"글만 쓰면 알아서 웹사이트를 만들어주는 요리사"\*\*라고 생각하면 됩니다.

- **재료:** 여러분이 쓰는 글 (마크다운 파일, `.md`)
- **요리사: Jekyll** (지킬)
- **완성된 요리:** 예쁜 웹사이트 (HTML 파일)

여러분이 메모장(VS Code 등)에 글을 써서 저장(push)하기만 하면, Jekyll 요리사가 알아서 똑딱똑딱 웹사이트로 바꿔서 보여줍니다.

## Jekyll의 장점

- **무료 호스팅:** GitHub Pages와 완벽 호환
- **빠른 속도:** 정적 사이트라 로딩이 매우 빠름
- **간편한 관리:** 데이터베이스 없이 파일만으로 관리
- **마크다운 지원:** 복잡한 HTML 없이 쉽게 작성
- **버전 관리:** Git으로 모든 변경사항 추적 가능

## 2. 새 글 쓰기 (페이지 만들기)

웹사이트에 새로운 페이지를 추가하고 싶으신가요? 아주 간단합니다.

### 1단계: 파일 만들기

프로젝트 폴더 안에 새로운 파일을 만듭니다. 이름은 영어로 짓고, 뒤에 `.md`를 붙여주세요.

## 파일명 작성 규칙:

- 영문 소문자 사용 (예: `about.md`, `guide.md`)
- 띄어쓰기 대신 하이픈 사용 (예: `my-blog-post.md`)
- 특수문자 피하기 (한글 파일명은 피하세요)

## 예시:

- |                                     |   |
|-------------------------------------|---|
| <input checked="" type="checkbox"/> | 좋은 예: <code>guide.md</code> , <code>privacy-policy.md</code> , <code>about-us.md</code> |
| <input type="checkbox"/>            | 나쁜 예: <code>가이드.md</code> , <code>my blog.md</code> , <code>about@me.md</code>          |

## 2단계: '머리말(Front Matter)' 작성하기

파일의 **맨 윗부분**에 아래 내용을 똑같이 복사해서 붙여넣으세요. 이것은 Jekyll에게 "이 파일의 정보는 아래!"라고 알려주는 명찰입니다.

```
---  
layout: default  
title: 내 새로운 페이지 제목  
permalink: /new-page/  
---
```

- `layout`: 사용할 레이아웃 템플릿
- `title`: 브라우저 탭과 페이지 상단에 표시될 제목
- `permalink`: 이 페이지의 URL 주소

## 3단계: 본문 쓰기

머리말 아래에 자유롭게 내용을 적으세요. 마크다운 문법을 사용하면 됩니다.

## 3. 마크다운 문법 완벽 가이드

마크다운은 간단한 기호로 텍스트를 꾸미는 방법입니다. HTML보다 훨씬 쉽습니다!

### 3.1 제목 (Headers)

제목은 `#` 기호로 만듭니다. `#`의 개수가 많을수록 작은 제목입니다.

```
# 가장 큰 제목 (H1)  
## 두 번째 큰 제목 (H2)  
### 세 번째 제목 (H3)  
#### 네 번째 제목 (H4)  
##### 다섯 번째 제목 (H5)  
##### 가장 작은 제목 (H6)
```

결과:

# 가장 큰 제목 (H1)

---

## 두 번째 큰 제목 (H2)

### 세 번째 제목 (H3)

**팁:** 한 페이지에 H1(#)**은** 하나만 사용하는 것이 좋습니다. 보통 페이지 제목으로 사용합니다.

## 3.2 텍스트 강조

```
**굵은 글씨** (Bold)
*기울임 글씨* (Italic)
***굵은 기울임*** (Bold + Italic)
~~취소선~~ (Strikethrough)
`코드 강조` (Inline Code)
```

결과:

- **굵은 글씨** (Bold)
- *기울임 글씨* (Italic)
- ***굵은 기울임*** (Bold + Italic)
- 취소선 (Strikethrough)
- `코드 강조` (Inline Code)

## 3.3 목록 (Lists)

### 순서 없는 목록 (Bullet Points)

```
* 첫 번째 항목
* 두 번째 항목
  * 들여쓰기된 하위 항목 (스페이스 2개 또는 탭)
  * 또 다른 하위 항목
* 세 번째 항목
```

또는 - 기호도 사용 가능:

```
- 항목 1
- 항목 2
- 항목 3
```

결과:

- 첫 번째 항목

- 두 번째 항목
  - 들여쓰기된 하위 항목
  - 또 다른 하위 항목
- 세 번째 항목

## 순서 있는 목록 (Numbered List)

1. 첫 번째 단계
2. 두 번째 단계
3. 세 번째 단계
  1. 하위 단계 A
  2. 하위 단계 B
4. 네 번째 단계

### 결과:

1. 첫 번째 단계
2. 두 번째 단계
3. 세 번째 단계

## 3.4 인용구 (Blockquotes)

- > 이것은 인용구입니다.  
> 여러 줄로 작성할 수 있습니다.  
>  
> > 중첩된 인용구도 가능합니다.

### 결과:

이것은 인용구입니다. 여러 줄로 작성할 수 있습니다.

## 3.5 코드 블록

### 인라인 코드

```
`console.log('Hello World')`처럼 문장 안에 코드를 넣을 수 있습니다.
```

### 여러 줄 코드 블록

코드 블록은 백틱(`) 3개로 시작하고 끝냅니다:

```
```javascript
function sayHello() {
  console.log('Hello, Jekyll!');
```

```

}
```
```python
def say_hello():
    print("Hello, Jekyll!")
```

```

## 언어별 문법 강조:

- `javascript, js` - JavaScript
- `python, py` - Python
- `html` - HTML
- `css` - CSS
- `bash, sh` - Shell
- `yaml, yml` - YAML
- `json` - JSON

## 3.6 표 (Tables)

| 항목  | 가격     | 재고  |
|-----|--------|-----|
| --- | ---    | --- |
| 사과  | 1,000원 | 50개 |
| 바나나 | 1,500원 | 30개 |
| 오렌지 | 2,000원 | 20개 |

## 결과:

| 항목  | 가격     | 재고  |
|-----|--------|-----|
| 사과  | 1,000원 | 50개 |
| 바나나 | 1,500원 | 30개 |
| 오렌지 | 2,000원 | 20개 |

## 정렬 옵션:

|                         |
|-------------------------|
| 왼쪽 정렬   가운데 정렬   오른쪽 정렬 |
| :---   :---:   ---:     |
| 왼쪽   가운데   오른쪽          |

## 3.7 수평선 (Horizontal Rule)

문단을 나눌 때 사용합니다:

---  
또는

\*\*\*

### 결과:

---

## 3.8 이모지 사용하기

GitHub Pages에서는 이모지를 쉽게 사용할 수 있습니다:

```
:smile: :heart: :rocket: :fire: :tada:
```

### 주요 이모지:

- :smile: 😊
- :heart: ❤️
- :rocket: 🚀
- :fire: 🔥
- :tada: 🎉
- :warning: ⚠️
- :bulb:💡
- :pencil:✏️

더 많은 이모지는 [Emoji Cheat Sheet](#)에서 확인하세요!

## 4. 문단 띄어쓰기와 레이아웃 ✏️

마크다운에서 문단 띄어쓰기는 매우 중요합니다. 잘못하면 글이 붙어서 보입니다!

### 4.1 줄바꿈 (Line Break)

✖ 이렇게 하면 안 됩니다:

첫 번째 문장입니다.  
두 번째 문장입니다.

결과: 두 문장이 붙어서 "첫 번째 문장입니다. 두 번째 문장입니다."로 나옵니다.

올바른 방법 1: 빈 줄 추가

첫 번째 문장입니다.

두 번째 문장입니다.

**결과:** 두 문장이 별도의 문단으로 표시됩니다.

올바른 방법 2: 줄 끝에 스페이스 2개

첫 번째 문장입니다.

두 번째 문장입니다.

**중요:** 줄 끝에 스페이스를 **2개** 넣으면 줄바꿈이 됩니다 (보이지 않으니 주의!).

올바른 방법 3: <br> 태그 사용

첫 번째 문장입니다. <br>

두 번째 문장입니다.

## 4.2 문단 간격 조절

**보통 간격:**

첫 번째 문단입니다.

두 번째 문단입니다.

**넓은 간격 (2줄 띄우기):**

첫 번째 문단입니다.

두 번째 문단입니다.

**매우 넓은 간격 (HTML 사용):**

첫 번째 문단입니다.

<br>

두 번째 문단입니다.

## 4.3 들여쓰기와 정렬

마크다운은 기본적으로 왼쪽 정렬입니다. 다른 정렬을 원하면 HTML을 사용해야 합니다:

```
<div style="text-align: center">  
    이 텍스트는 가운데 정렬됩니다.  
</div>  
  
<div style="text-align: right">  
    이 텍스트는 오른쪽 정렬됩니다.  
</div>
```

## 4.4 섹션 구분하기

큰 섹션을 나눌 때는 수평선을 활용하세요:

```
## 첫 번째 섹션
```

내용...

---

```
## 두 번째 섹션
```

내용...

---

```
## 세 번째 섹션
```

내용...

## 5. Front Matter 상세 설명

Front Matter는 파일 맨 위에 있는 설정 블록입니다. 로 시작하고 끝납니다.

### 5.1 기본 Front Matter

```
---  
layout: default  
title: 페이지 제목  
permalink: /my-page/  
---
```

### 5.2 모든 Front Matter 옵션

```

---
# 필수 항목
layout: default          # 사용할 레이아웃 (default, post, page 등)
title: "나의 멋진 페이지" # 페이지 제목 (따옴표 사용 추천)

# 선택 항목
permalink: /custom-url/   # 사용자 정의 URL
date: 2026-02-05          # 발행 날짜 (블로그 포스트용)
author: 홍길동            # 작성자
description: "페이지 설명" # SEO 메타 설명
categories: [jekyll, guide] # 카테고리 (배열)
tags: [tutorial, beginner] # 태그 (배열)
published: true           # false로 설정하면 숨겨짐
comments: true             # 댓글 기능 활성화 (테마 지원시)

---

```

### 5.3 Layout 종류

- **default**: 가장 기본적인 레이아웃
- **page**: 일반 페이지용 (About, Contact 등)
- **post**: 블로그 포스트용 (날짜, 작성자 등 메타데이터 포함)
- **home**: 홈페이지용 (테마별로 다름)

### 5.4 Permalink 패턴

```

# 고정 URL
permalink: /about/

# 날짜 기반 (블로그 포스트)
permalink: /:year/:month/:day/:title/
# 결과: /2026/02/05/my-post/

# 카테고리 포함
permalink: /:categories/:title/
# 결과: /jekyll/my-post/

# 간단한 형식
permalink: /:title/
# 결과: /my-post/

```

### 5.5 커스텀 변수

Front Matter에 원하는 변수를 추가할 수 있습니다:

```

---
layout: default
title: 제품 소개

```

```

product_name: "아이폰 15"
price: 1200000
in_stock: true
---

# 본문에서 사용
{{ page.product_name }}의 가격은 {{ page.price }}원입니다.

```

## 6. 사이트 설정 (\_config.yml) ☀

\_config.yml은 사이트 전체의 설정 파일입니다.

### 6.1 기본 설정

```

# 사이트 정보
title: "내 멋진 블로그"
description: "Jekyll과 GitHub Pages로 만든 블로그입니다."
author: "홍길동"

# URL 설정
url: "https://username.github.io"      # 사이트 주소
baseurl: ""                          # 프로젝트 사이트라면 "/repository-name"

# 빌드 설정
markdown: kramdown                  # 마크다운 엔진
theme: minima                        # 테마
plugins:
  - jekyll-feed
  - jekyll-seo-tag
  - jekyll-sitemap

# 제외할 파일
exclude:
  - README.md
  - Gemfile
  - Gemfile.lock

```

### 6.2 주요 설정 항목 상세

항목	설명	예시
title	사이트 제목	"My Blog"
description	사이트 설명 (SEO용)	"개발자의 기술 블로그"
url	사이트 전체 URL	"https://example.com"
baseurl	하위 경로 (선택)	"" 또는 "/blog"
markdown	마크다운 처리기	kramdown (기본값)

항목	설명	예시
theme	테마 이름	minima, jekyll-theme-cayman
permalink	기본 permalink 패턴	/:title/
timezone	시간대	Asia/Seoul
lang	언어 코드	ko (한국어)

## 6.3 소셜 미디어 링크

```
# 소셜 미디어 (테마에서 사용)
github_username: your-github-username
twitter_username: your-twitter-username
instagram_username: your-instagram
email: your-email@example.com
```

## 6.4 변경 후 적용

⚠ 중요: `_config.yml` 파일을 변경하면 반드시 사이트를 재시작해야 합니다!

- 로컬 개발: 서버 재시작 (`Ctrl+C` 후 `bundle exec jekyll serve` 다시 실행)
- GitHub Pages: Push 후 1-3분 대기

## 7. 테마 변경과 커스터マイ징 🎨

Jekyll 테마로 사이트 디자인을 쉽게 바꿀 수 있습니다!

### 7.1 GitHub Pages 지원 테마

GitHub Pages가 공식 지원하는 테마들:

테마 이름	스타일	설정 방법
minima	깔끔한 블로그 스타일	theme: minima
jekyll-theme-cayman	파란색 헤더, 프로젝트 페이지 용	theme: jekyll-theme-cayman
jekyll-theme-slate	어두운 개발자 테마	theme: jekyll-theme-slate
jekyll-theme-architect	건축적 디자인	theme: jekyll-theme-architect
jekyll-theme-time-machine	레트로 스타일	theme: jekyll-theme-time-machine
jekyll-theme-minimal	미니멀한 디자인	theme: jekyll-theme-minimal
jekyll-theme-leap-day	밝고 경쾌한 디자인	theme: jekyll-theme-leap-day
jekyll-theme-merlot	우아한 와인색 테마	theme: jekyll-theme-merlot

## 7.2 테마 변경 방법

### 방법 1: \_config.yml에서 변경 (권장)

1. `_config.yml` 파일을 엽니다
2. `theme` 항목을 수정합니다:

```
theme: jekyll-theme-cayman
```

3. 저장하고 GitHub에 Push
4. 1-3분 후 적용 확인

### 방법 2: remote\_theme 사용 (더 많은 테마)

GitHub Pages에 없는 테마도 사용 가능:

```
# _config.yml
remote_theme: pages-themes/cayman@v0.2.0

plugins:
  - jekyll-remote-theme
```

## 7.3 인기 무료 테마

공식 테마 외에도 많은 무료 테마가 있습니다:

- **Minimal Mistakes**: 강력한 기능, 블로그/포트폴리오
- **Beautiful Jekyll**: 초보자 친화적
- **al-folio**: 학술/연구용
- **Chirpy**: 현대적 블로그 테마
- **Type on Strap**: 타이포그래피 중심

## 7.4 테마 커스터마이징

### CSS 덮어쓰기 (가장 쉬운 방법)

`assets/css/style.scss` 파일을 생성:

```
---
---

@import "{{ site.theme }}";

/* 여기에 커스텀 스타일 작성 */

/* 예: 배경색 변경 */
```

```

body {
  background-color: #f5f5f5;
}

/* 예: 제목 색상 변경 */
h1, h2, h3 {
  color: #2c3e50;
}

/* 예: 링크 색상 변경 */
a {
  color: #3498db;
  text-decoration: none;
}

a:hover {
  color: #2980b9;
  text-decoration: underline;
}

/* 예: 폰트 변경 */
body {
  font-family: 'Nanum Gothic', 'Malgun Gothic', sans-serif;
  font-size: 16px;
  line-height: 1.6;
}

/* 예: 코드 블록 스타일 */
code {
  background-color: #f4f4f4;
  padding: 2px 6px;
  border-radius: 3px;
  font-family: 'Courier New', monospace;
}

```

## 레이아웃 덮어쓰기

테마의 HTML 구조를 변경하려면:

1. `_layouts` 폴더 생성
2. 테마의 레이아웃 파일을 복사하여 수정
3. 파일명을 동일하게 유지 (예: `default.html`)

```

<!-- _layouts/default.html -->
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>{{ page.title }} - {{ site.title }}</title>
  <link rel="stylesheet" href="{{ '/assets/css/style.css' | relative_url }}>
</head>

```

```
<body>
  <header>
    <h1>{{ site.title }}</h1>
    <nav>
      <a href="/">홈</a>
      <a href="/about/">소개</a>
      <a href="/blog/">블로그</a>
    </nav>
  </header>

  <main>
    {{ content }}
  </main>

  <footer>
    <p>&copy; 2026 {{ site.author }}. All rights reserved.</p>
  </footer>
</body>
</html>
```

## 색상 테마 변경 (CSS Variables 사용)

```
/* assets/css/style.scss */
---

@import "{{ site.theme }}";

:root {
  --primary-color: #3498db;
  --secondary-color: #2ecc71;
  --text-color: #333;
  --background-color: #fff;
  --border-color: #ddd;
}

/* 다크 모드 예시 */
@media (prefers-color-scheme: dark) {
  :root {
    --primary-color: #5dade2;
    --secondary-color: #58d68d;
    --text-color: #ecf0f1;
    --background-color: #2c3e50;
    --border-color: #34495e;
  }
}

body {
  color: var(--text-color);
  background-color: var(--background-color);
}
```

```
a {  
  color: var(--primary-color);  
}
```

## 8. 이미지와 미디어 파일 다루기

### 8.1 이미지 추가 기본

#### 방법 1: 마크다운 문법

```
![이미지 설명](이미지경로)
```

예시:

```
![고양이 사진](/assets/images/cat.jpg)
```

#### 방법 2: HTML (더 많은 옵션)

```

```

### 8.2 이미지 저장 위치

#### 추천 구조:

```
프로젝트 폴더/  
  └── assets/  
      └── images/  
          ├── logo.png  
          ├── cat.jpg  
          └── screenshot.png  
  └── _posts/  
      └── index.md
```

#### 이미지 참조 방법:

```
![로고](/assets/images/logo.png)
```

### 8.3 이미지 크기 조절

#### HTML 사용

```
<!-- 픽셀 단위 -->
![고양이](/assets/images/cat.jpg)<

<!-- 비율 조절 -->
![고양이](/assets/images/cat.jpg)
```

## 8.4 이미지 정렬

```
<!-- 가운데 정렬 -->


![고양이](/assets/images/cat.jpg)



<!-- 오른쪽 정렬 -->


![고양이](/assets/images/cat.jpg)



<!-- CSS 사용 (더 권장) -->
![고양이](/assets/images/cat.jpg)
```

## 8.5 캡션 추가

```
<figure>
  
  <figcaption>귀여운 고양이의 낮잠 시간</figcaption>
</figure>
```

## 8.6 외부 이미지 링크

![Unsplash 이미지](<https://images.unsplash.com/photo-xxxxx>)

**주의:** 외부 이미지는 원본 사이트가 삭제하면 보이지 않습니다. 중요한 이미지는 직접 업로드하세요!

## 8.7 반응형 이미지 (모바일 대응)

```

```

## 8.8 이미지 최적화 팁

- **크기 줄이기**: 웹용은 보통 1200px 너비면 충분
- **포맷 선택**:
  - 사진: `.jpg` (압축률 좋음)
  - 로고/아이콘: `.png` (투명 배경 가능)
  - 애니메이션: `.gif`
  - 현대적: `.webp` (크기 작음, 최신 브라우저 지원)
- **파일명**: 영문 소문자, 공백 없이 (`my-photo.jpg` , `내 사진.JPG` )

## 9. 링크 만들기

### 9.1 외부 링크

[링크 텍스트](URL)

예시:

[구글로 이동] (<https://www.google.com>)

[GitHub] (<https://github.com>)

### 9.2 내부 페이지 링크

[About 페이지] (/about/)

[첫 번째 블로그 글] (/blog/my-first-post/)

### 9.3 새 탭에서 열기

[구글] (<https://www.google.com>){:target="\_blank"}

또는 HTML:

```
<a href="https://www.google.com" target="_blank">구글</a>
```

### 9.4 페이지 내부 앵커 링크

```
<!-- 목차에서 사용 -->
```

[마크다운 문법으로 이동] (#3-마크다운-문법-완벽-가이드-)

```
<!-- 제목이 자동으로 앵커가 됩니다 -->
```

### ## 3. 마크다운 문법 완벽 가이드

## 9.5 이메일 링크

```
[이메일 보내기](&mailto:example@example.com)
```

## 9.6 파일 다운로드 링크

```
[PDF 다운로드](&/assets/files/document.pdf)
```

## 9.7 링크에 툴팁 추가

```
[GitHub](&https://github.com "GitHub 홈페이지로 이동")
```

# 10. 저장하고 배포하기 🚀

## 10.1 Git 기본 명령어

```
# 1. 변경사항 확인  
git status  
  
# 2. 모든 변경사항 스테이징  
git add .  
  
# 3. 커밋 (변경사항 기록)  
git commit -m "새 가이드 페이지 추가"  
  
# 4. GitHub에 업로드  
git push origin main
```

## 10.2 VS Code에서 GUI로 커밋

1. **Source Control** 탭 클릭 (왼쪽 세 번째 아이콘)
2. 변경된 파일 확인
3. + 버튼으로 스테이징
4. 메시지 입력 후 ✓ Commit
5. ... 메뉴 → Push

## 10.3 배포 시간

GitHub에 Push 후:

- **보통:** 1-3분
- **느릴 때:** 최대 10분
- **확인 방법:** [https://username.github.io/페이지주소/](https://username.github.io/)

## 10.4 배포 상태 확인

1. GitHub 저장소 페이지 이동
  2. **Actions** 탭 클릭
  3. 최근 워크플로우 상태 확인
    - 녹색: 성공
    - 노란색: 진행 중
    - 빨간색: 실패 (오류 확인 필요)
- 

## 11. 문제 해결 팁 🔪

### 11.1 페이지가 안 보여요

체크리스트:

- Front Matter가 올바른가요? (- - -로 시작하고 끝나는지)
- permalink 경로가 정확한가요?
- GitHub에 정상적으로 Push 되었나요?
- 1-3분 기다렸나요?
- 강력 새로고침 해봤나요? ([Ctrl + Shift + R](#))

### 11.2 이미지가 안 나와요

원인과 해결:

- **경로 오류:** </assets/images/photo.jpg> (절대 경로 사용)
- **파일명 대소문자:** Photo.jpg ≠ photo.jpg (정확히 일치해야 함)
- **파일이 실제로 있나요:** GitHub 저장소에서 확인
- **지원 형식:** .jpg, .png, .gif, .svg 사용

### 11.3 테마가 안 바뀌어요

1. [\\_config.yml](#)의 theme 철자 확인
2. GitHub Pages 지원 테마인지 확인
3. Push 후 10분 정도 기다리기
4. 브라우저 캐시 삭제

### 11.4 마크다운이 제대로 렌더링 안 돼요

- **줄바꿈:** 빈 줄을 추가하거나 줄 끝에 스페이스 2개
- **특수문자 이스케이프:** \\* (별표를 문자 그대로 표시)
- **코드 블록:** 백틱 3개로 감싸기

### 11.5 \_config.yml 수정이 적용 안 돼요

- 로컬 서버 재시작 필요
- GitHub Pages는 Push 후 재빌드 대기

## 11.6 빌드 에러 발생

**Actions 탭에서 에러 확인:**

1. 저장소 → **Actions**
2. 실패한 워크플로우 클릭
3. 에러 메시지 읽기

**주요 원인:**

- YAML 문법 오류 (들여쓰기, 콜론 뒤 공백 등)
- 지원하지 않는 플러그인
- Front Matter 형식 오류

## 11.7 한글이 깨져요

**\_config.yml에 인코딩 설정:**

```
encoding: utf-8
```

**파일 저장 시 UTF-8 인코딩 확인** (VS Code는 기본적으로 UTF-8)

## 11.8 검색 엔진에 안 나와요 (SEO)

**개선 방법:**

1. **\_config.yml**에 description 추가
2. Front Matter에 description 추가
3. **jekyll-seo-tag** 플러그인 사용

```
# _config.yml
plugins:
  - jekyll-seo-tag
```

```
# 페이지 Front Matter
---
title: 내 페이지
description: 이 페이지는 Jekyll 사용법을 설명합니다.
---
```

---

## 📘 추가 학습 자료 - 단계별 가이드

이 섹션에서는 여러분의 실력에 맞는 학습 자료를 소개합니다. 초보자부터 고급 사용자까지 모두 활용할 수 있습니다!

☞ 초보자: 여기서부터 시작하세요!

## 1. 마크다운 가이드 (Markdown Guide) ★ 최우선 추천!

링크: <https://www.markdownguide.org/>

이런 분들께 추천:

- Jekyll을 처음 시작하는 분
- 마크다운 문법을 체계적으로 배우고 싶은 분
- 빠른 참고 자료가 필요한 분

배울 수 있는 내용:

▣ **Getting Started** - 시작하기

- 마크다운이 무엇인지 완전 초보자 관점에서 설명
- 마크다운을 쓸 수 있는 에디터 추천 (VS Code, Typora 등)
- 첫 마크다운 문서 작성 실습

▣ **Basic Syntax** - 기본 문법

- 제목, 문단, 줄바꿈
- 굵게, 기울임, 취소선
- 목록 (순서 있는/없는)
- 링크와 이미지
- 코드와 코드 블록
- 인용구
- 수평선

팁: 이 페이지를 북마크해두고 문법이 기억나지 않을 때마다 참고하세요!

▣ **Extended Syntax** - 확장 문법

- 표 만들기
- 체크박스 목록
- 이모지 사용하기
- 각주 추가하기
- 형광펜 효과

▣ **Cheat Sheet** - 치트 시트

- 모든 마크다운 문법을 한 페이지에 정리
- 빠르게 찾아보기 좋은 레퍼런스

학습 소요 시간: 1-2시간 (기본 문법만)

## 활용 팁:

1. Basic Syntax 페이지를 처음부터 끝까지 읽고 실습 (30분)
2. 직접 practice.md 파일을 만들어서 모든 문법 연습 (30분)
3. Cheat Sheet를 브라우저 즐겨찾기에 추가 (1분)
4. 글 쓰면서 모르는 문법이 나오면 Cheat Sheet 참고

## 2. GitHub Pages 가이드 (GitHub Pages Documentation)

링크: <https://docs.github.com/en/pages>

### 이런 분들께 추천:

- GitHub Pages로 사이트를 배포하려는 분
- 커스텀 도메인을 연결하고 싶은 분
- Jekyll과 GitHub Pages 연동 문제를 해결하려는 분

### 배울 수 있는 내용:

#### 🎬 Getting Started - 시작하기

- GitHub Pages가 무엇인지 이해하기
- 첫 GitHub Pages 사이트 만들기 (5분 완성!)
- 저장소 이름 규칙 (`username.github.io`)
- 사이트 공개 설정 방법

#### ⚙️ Configuring a publishing source - 배포 설정

- `main` 브랜치에서 자동 배포하기
- `docs` 폴더 사용하기
- GitHub Actions로 커스텀 빌드하기

#### ⌚ Setting up Jekyll - Jekyll 설정

- GitHub Pages에서 Jekyll 사용하기
- 로컬에서 Jekyll 테스트하기
- Jekyll 테마 추가하기
- 의존성 관리하기

#### 🌐 Custom Domain - 커스텀 도메인

- `example.com`처럼 내 도메인 연결하기
- DNS 설정 방법 (A 레코드, CNAME)
- HTTPS 인증서 자동 발급
- 서브도메인 설정 (`blog.example.com`)

#### 📞 Troubleshooting - 문제 해결

- Jekyll 빌드 오류 해결
- 404 오류 해결
- 페이지가 업데이트되지 않을 때
- 커스텀 도메인 문제 해결

학습 소요 시간: 2-3시간

추천 학습 순서:

1.  Getting Started 읽기 (30분)
2.  실제로 GitHub Pages 사이트 만들어보기 (30분)
3.  Jekyll 설정 가이드 따라하기 (1시간)
4.  Custom Domain은 필요할 때 참고

﴿ 중급자: 더 깊이 파고들기

### 3. Jekyll 공식 문서 (Jekyll Documentation)

링크: <https://jekyllrb.com/docs/>

이런 분들께 추천:

- 기본 사용법은 익혔고 더 고급 기능을 배우고 싶은 분
- Jekyll의 작동 원리를 이해하고 싶은 분
- 플러그인과 커스터마이징을 하고 싶은 분

배울 수 있는 내용:

﴿ Quickstart - 빠른 시작

```
gem install bundler jekyll  
jekyll new my-site  
cd my-site  
bundle exec jekyll serve
```

- Jekyll을 로컬에 설치하는 방법
- 새 사이트 프로젝트 생성
- 로컬 서버에서 실시간 미리보기

﴿ Directory Structure - 디렉토리 구조

```
.  
|__ _config.yml      # 사이트 설정  
|__ _data/           # 데이터 파일 (YAML, JSON, CSV)  
|__ _includes/       # 재사용 가능한 조각
```

```

└── _layouts/          # 레이아웃 템플릿
└── _posts/           # 블로그 포스트
└── _sass/            # Sass 파일
└── assets/           # 이미지, CSS, JS
└── index.md          # 홈페이지

```

- 각 폴더의 역할과 사용법
- 파일 이름 규칙 (예: `2026-02-05-title.md`)

### ⌚ Front Matter - 메타데이터 심화

```

---
layout: post
title: "내 첫 블로그 글"
date: 2026-02-05 14:30:00 +0900
categories: jekyll tutorial
tags: [beginner, guide]
author: "홍길동"
excerpt: "이 글의 요약입니다."
image: /assets/images/thumbnail.jpg
---

```

- 모든 Front Matter 변수 목록
- 커스텀 변수 만들기
- 기본값 설정하기

### ⌚ Layouts - 레이아웃

```

<!-- _layouts/post.html -->
---
layout: default
---
<article>
  <h1>{{ page.title }}</h1>
  <time>{{ page.date | date: "%Y-%m-%d" }}</time>
  {{ content }}
</article>

```

- 레이아웃 상속 (layout 안에 layout)
- Liquid 템플릿 언어 사용법
- 조건문과 반복문

### ₩ Plugins - 플러그인

- SEO 플러그인 (`jekyll-seo-tag`)
- RSS 피드 (`jekyll-feed`)

- 사이트맵 생성 ([jekyll-sitemap](#))
- 커스텀 플러그인 만들기

### Themes - 테마

- Gem 기반 테마 사용법
- 테마 커스터마이징
- 테마 오버라이드
- 자신만의 테마 만들기

### Data Files - 데이터 파일

```
# _data/team.yml
- name: 홍길동
  role: 개발자

- name: 김철수
  role: 디자이너
```

```
{% for member in site.data.team %}
  <p>{{ member.name }} - {{ member.role }}</p>
{% endfor %}
```

### Variables - 변수

- `site` - 사이트 전체 정보
- `page` - 현재 페이지 정보
- `content` - 페이지 내용
- `paginator` - 페이지네이션 정보

학습 소요 시간: 5-10시간

추천 학습 경로:

주차별 학습 플랜:

1주차: 기초 다지기

- Directory Structure 이해
- Front Matter 완벽 정리
- Layouts 기본 개념

2주차: 템플릿 마스터

- Liquid 템플릿 언어 학습
- Includes로 컴포넌트화
- Variables 활용

- 💻 3주차: 고급 기능  
 - Plugins 설치 및 사용  
 - Data Files로 동적 콘텐츠  
 - Collections 사용법

- 💻 4주차: 프로젝트  
 - 완전한 블로그 사이트 구축  
 - 테마 커스터마이징  
 - 배포 최적화

## ♣ 모든 레벨: 디자인 & 테마

### 4. 무료 Jekyll 테마 모음 (Jekyll Themes)

링크: <https://jekyllthemes.io/free>

이런 분들께 추천:

- 아름다운 디자인의 테마를 찾는 분
- 빠르게 전문적인 사이트를 만들고 싶은 분
- 다양한 테마를 비교하고 싶은 분

테마 카테고리별 추천:

#### ▣ 블로그용 테마 (Blog Themes)

테마 이름	특징	난이도	추천 대상
<b>Minimal Mistakes</b>	가장 인기 많음, 기능 풍부	★★★ 중급	기술 블로그, 포트폴리오
<b>Beautiful Jekyll</b>	초보자 친화적, 설정 쉬움	★ 초급	개인 블로그
<b>Chirpy</b>	현대적 디자인, 다크모드	★★★ 중급	개발자 블로그
<b>Mediumish</b>	Medium 스타일	★★ 초중급	작가, 콘텐츠 크리에이터
<b>Hyde</b>	미니멀 디자인, 사이드바	★ 초급	간단한 개인 블로그

#### ▣ 포트폴리오용 테마 (Portfolio Themes)

테마 이름	특징	난이도	추천 대상
<b>Forty</b>	사진/이미지 중심	★★ 초중급	사진작가, 디자이너
<b>Agency</b>	비즈니스/에이전시	★★★ 중급	회사, 프리랜서
<b>Creative</b>	창작 포트폴리오	★★ 초중급	아티스트, 크리에이터

#### ▣ 문서용 테마 (Documentation Themes)

테마 이름	특징	난이도	추천 대상
-------	----	-----	-------

테마 이름	특징	난이도	추천 대상
Just The Docs	기술 문서 최적화	★ ★ 초중급	API 문서, 가이드
Documentation	검색 기능 강력	★ ★ ★ 중급	제품 문서, 매뉴얼

## ▣ 기타 테마 (Special Purpose)

테마 이름	특징	난이도	추천 대상
Online CV	이력서/CV	★ 초급	구직자
Resume Template	온라인 이력서	★ 초급	구직자, 프리랜서
Landing Page	앱/제품 소개	★ ★ 초중급	스타트업, 제품 룬칭

## 테마 선택 가이드:

💡 어떤 테마를 선택해야 할까?

### ① 목적부터 정하기:

- 블로그? → Minimal Mistakes, Beautiful Jekyll
- 포트폴리오? → Forty, Agency
- 문서/가이드? → Just The Docs
- 이력서? → Online CV

### ② 실력 고려하기:

- ★ 초급: Beautiful Jekyll, Hyde, Online CV
- ★ ★ 초중급: Mediumish, Forty, Just The Docs
- ★ ★ ★ 중급: Minimal Mistakes, Chirpy, Agency

### ③ 데모 보고 결정하기:

- 각 테마 페이지에서 "Live Preview" 클릭
- 모바일에서도 확인해보기
- 로딩 속도 체크

### ④ GitHub Stars 확인:

- 많은 사람들이 사용하는 테마 = 문서 풍부, 버그 적음
- Minimal Mistakes: 11,000+ stars ★
- Beautiful Jekyll: 5,000+ stars ★

## 테마 설치 방법:

각 테마 페이지에서 제공하는 방법을 따르되, 일반적으로:

```
# 방법 1: Remote Theme (가장 쉬움)
# _config.yml에 추가
remote_theme: owner/repository
```

```
# 방법 2: Gem 기반
```

```
# Gemfile에 추가  
gem "jekyll-theme-name"  
  
# 방법 3: 직접 다운로드  
# 테마 파일을 통째로 복사
```

---

## 🎓 학습 로드맵 추천

### 완전 초보자 (1-2주)

```
Day 1-2: 마크다운 가이드 Basic Syntax  
Day 3-4: GitHub Pages로 첫 사이트 만들기  
Day 5-7: Jekyll 가이드(이 페이지) 정독 및 실습  
Week 2: 무료 테마 적용해보기
```

### 초급 사용자 (2-4주)

```
Week 1: Jekyll 공식 문서 Directory Structure, Layouts  
Week 2: Liquid 템플릿 언어 마스터  
Week 3: 커스텀 테마 수정해보기  
Week 4: 플러그인 추가 및 블로그 완성
```

### 중급 사용자 (진행 중)

- Collections 사용법
- 커스텀 플러그인 개발
- GitHub Actions로 자동화
- 성능 최적화 (이미지, CSS, JS)
- SEO 최적화

---

## 💡 학습 팁

### 1. 실습 위주로 배우기

- 읽기만 하지 말고 직접 따라 해보세요
- 예러가 나면 좋은 학습 기회입니다!

### 2. 작은 프로젝트부터 시작

- "About Me" 페이지 하나만 완성하기
- 블로그 글 3개만 써보기
- 테마 색상만 바꿔보기

### 3. 커뮤니티 활용

- [Jekyll Talk Forum](#)
- Stack Overflow #jekyll 태그
- GitHub Issues (각 테마/플러그인별)

### 4. 북마크 필수 페이지

- Markdown Cheat Sheet (문법 참고용)
- Jekyll Variables (템플릿 작성용)
- 선택한 테마의 문서 (커스터마이징용)

### 5. 버전 관리 습관

- Git으로 항상 커밋하기
- 큰 변경 전에는 브랜치 만들기
- 작동하는 버전은 태그로 표시

---

## 🔖 빠른 참고 링크 모음

링크	용도	자주 사용
<a href="#">Markdown Cheat Sheet</a>	문법 빠르게 찾기	★★★★★
<a href="#">Jekyll Variables</a>	템플릿 변수 참고	★★★★★
<a href="#">Liquid Syntax</a>	템플릿 문법	★★★★★
<a href="#">GitHub Pages 지원 테마</a>	공식 테마 목록	★★★★
<a href="#">Jekyll Plugins</a>	플러그인 목록	★★★★
<a href="#">Emoji Cheat Sheet</a>	이모지 코드	★★

---

## 💡 마지막 팁

- 작은 변경부터 시작하세요: 한 번에 하나씩 수정하고 테스트
- 버전 관리 활용: Git으로 항상 이전 버전 복구 가능
- 테마 문서 읽기: 각 테마마다 고유한 기능과 설정이 있습니다
- 커뮤니티 활용: Jekyll 사용자 커뮤니티에서 도움 받기
- 실험 두려워하지 마세요: 깨져도 Git으로 되돌릴 수 있습니다!

---

✿ 축하합니다! 이제 Jekyll 마스터입니다! ✿

멋진 웹사이트를 만드세요! ☺