

How to Create a Website

and why you need one

A Workshop for Queensland Astrophysicists

Workshop Agenda

1	Why and what	The case for a research website. What belongs on it.	Web Dev
2	What Makes a Good Website?	<i>Let's have a look at a few examples of good and bad websites together</i>	
3	Platform choice	Brief comparison. Why GitHub Pages wins for researchers.	
4	GitHub Pages & Jekyll	Setup, themes, configuration files.	
5	Getting Started with GitHub Pages	<i>Demo of how to get started with GitHub Pages and Jekyll Templates</i>	
6	AI as co-author & designer	What to use AI for in web design	Using AI
7	AI workflows for web development	Writing content and customising design without coding.	
8	Web Design with using AI	<i>Demo of how to Use AI Agents in Web Design (live vibe coding)</i>	
9	Prompts & wrap-up	Prompt cookbook, your next steps & questions	

Why Does It Matter?

Visibility

When collaborators, hiring committees, or journalists search your name, your website is the first result. Google indexes it within days of launch.

Identity

Control your own narrative. An institutional profile disappears the moment you move. A personal site is yours indefinitely.

Outreach

Public engagement starts with being findable. A clear research page allows the public to view your work.

Career

Panels, committees, and potential collaborators all benefit from a well-organised snapshot of your work. Make their job easier.

What Could Be on Your Site?

Landing Page (About / Bio)

1-2 paragraphs. Who you are, what you work on, why it matters. Written for a general audience. Most-read page on your site.

Research

One paragraph + a figure per project. Prioritise the 'so what'. Include active and completed projects. Link to relevant code or data.

Publications

List of your publications with arXiv links, or a link to your publications through ADS. Keep it current!

Software / Tools

List public code with a brief description, language, and GitHub link. A README is fine, but a dedicated page is better.

CV

Your experience and/or education and a link to your full downloadable CV. Include your ORCID and Google Scholar Profile

Contact & Profiles

Institutional email, ORCID, Bluesky/Twitter, GitHub. Linking ADS and arXiv is strongly recommended.

Who Is This For? Know Your Starting Point

Your career stage shapes what you need from a website, and how much time you can justify spending on it.

S PhD Student

Goal: Establish a presence early

Priorities

- One-page site is enough to start
- Bio + research summary + CV link
- Signals seriousness to future employers
- Add publications as they come
- Claim your domain name now — it's cheap

Avoid: Waiting until you have 'enough' to show

Tip: Just get started! *Right now, lets go!*

P Postdoc

Goal: Differentiate & showcase independence

Priorities

- Lead with your research vision, not just outputs
- Dedicated software / tools page if applicable
- Upcoming talks and conference appearances
- Show breadth: collaborations, mentoring, outreach
- Link to your ADS library

Avoid: A site that looks like a CV dump

Tip: Ask some of the senior faculty what they find important and add that!

F Faculty / Staff

Goal: Authority, group identity & recruitment

Priorities

- Group / lab page alongside personal page
- Student and collaborator listings
- Grant and project highlights
- Media, press, and public engagement links
- Professional photo and clear contact details

Avoid: An outdated site

Tip: If you have an old site, **you have most of the scaffolding** there. Download a template and **migrate it over...** (or get an *AI agent* to do it)

Research Website Examples

Examples

Rossana Ruggeri: <https://rossanaruggeri.github.io/>

Rebecca McElroy: <https://rebeccamcelroy.github.io/>

Michael Cowley: <https://mjcowley.github.io/>

Mitchell Hooymans: <https://mitchellhooymans.com>

Michio Kaku (theoretical physicist): <https://mkaku.org/>

Bjarne Stroustrup (creator of C++): <https://www.stroustrup.com/>

Pavel Kroupa (Astrophysicist): <https://astro.uni-bonn.de/~pavel/>

David Wiltshire (Cosmologist): <https://spcs.canterbury.ac.nz/~dlw24/>

QUASAR Collaboration: <https://quasarcollaboration.github.io/>

Some Notes

- Know who you are developing the website for!
 - Personal, students, media, etc?
- Keep it simple, a simplistic minimal landing page is perfect
 - Show your face!
- Choose a simple colour scheme
 - Use coolers.co
- Direct people to your page!
 - Link your socials to your page and vice versa.

Choosing a Platform

There are many website builders. Wix, Squarespace, and Google Sites are valid for a simple presence. For a research site with full control, version history, zero ongoing cost, and integration with your existing GitHub workflow.

Platform	Cost	Control	Best for
Wix / Squarespace	Free-\$17/mo	Low-Med	Quick start, non-technical users
Google Sites	Free	Low	Simple presence, no deep customisation
GitHub Pages	Free	High	Researchers - version control, no lock-in, integrates with GitHub
WordPress.com	Free - \$9/mo+	Med	Blog-heavy or content-rich sites

PART ONE

GitHub Pages

Setting up your research site from scratch

What is GitHub Pages

GitHub provides free static website hosting through GitHub Pages

1

Website Is Hosted in a GitHub Repo

Push your code → your site goes live at `username.github.io`

2

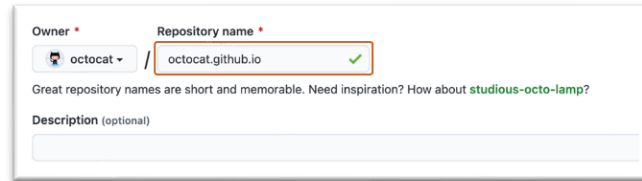
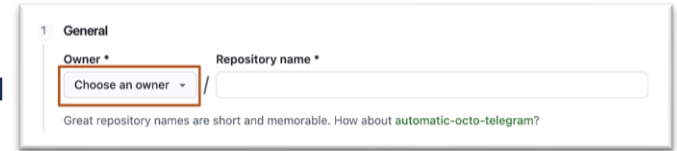
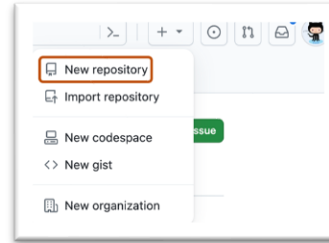
No server setup, no hosting costs

Automatic deploys website on every commit

3

Two Main Approaches

Static Site Generators or HTML5 Templates



Static Site Generators & Templates

Requires some setup to develop locally, but content is written in markdown separate from site design.

<https://jekyllthemes.io/free>

HTML5 Templates

Quick to setup and develop locally but you need to be familiar HTML/CSS and potentially JavaScript

<https://html5up.net/>

Getting Started with Jekyll

GitHub Pages turns a repository into a live website, made even more simple with Jekyll. No coding required!

1

Fork a theme or Use a Template

Find al-folio on GitHub. Click [use this template](#) and you now have a copy under your account.

2

Enable Pages

Settings -> Pages -> Source: GitHub Actions. Your URL: username.github.io

3

Edit `_config.yml`

Fill in your name, institution, email, and social links.

4

Edit your pages

Each page is a Markdown file. Edit about.md with your bio. Commit. Live in ~60 sec*.

5

Add a custom domain (optional)

Point yourname.com at GitHub Pages via DNS. ~10 minutes, ~\$10/yr.

Jekyll's three key ingredients.

Your Markdown files

about.md, research.md, publications.md ...

`_config.yml`

Name, institution, links, theme settings

Jekyll theme

al-folio, academicpages, minimal-mistakes

Your live website

Built automatically -- live in ~60 seconds

Choosing a Jekyll Theme

The theme defines the structure and style of your site. For academics, these three are the most widely used:

al-folio

alshedivat/al-folio

Most popular in CS & physics

Key features:

- Publications from BibTeX
- Research / projects pages
- News feed, blog support
- Google Scholar + ADS integration
- Actively maintained



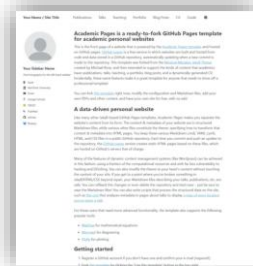
academicpages

academicpages.github.io

Built specifically for researchers

Key features:

- Talks, teaching, portfolio sections
- CV auto-generated from YAML
- No-frills, maximally legible
- Large academic user community



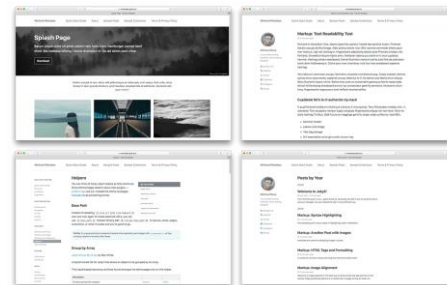
minimal-mistakes

mmistakes.github.io/minimal-mistakes

Flexible, less opinionated

Key features:

- Highly configurable layouts
- Excellent documentation
- Works for mixed personal/research
- No built-in academic sections



Customising Your Site

These two files do 90% of the work. You rarely need to touch anything else.

`_config.yml`

```
title: "Mitchell Hooymans"
first_name: Mitchell
last_name: Hooymans

description: >-
  PhD Candidate, UQ. N-body dynamics,
  black hole natal kicks in globular
  clusters, gravitational wave sources.

email: m.hooymans@uq.edu.au
institution: University of Queensland

github_username: mitchellhooymans
orcid: 0000-0002-1234-5678

ads_library: ui.adsabs.harvard.edu/
  public-libraries/YOUR_LIBRARY_ID

affiliations:
  - OzGrav
  - QUASAR Collaboration
```

`about.md`

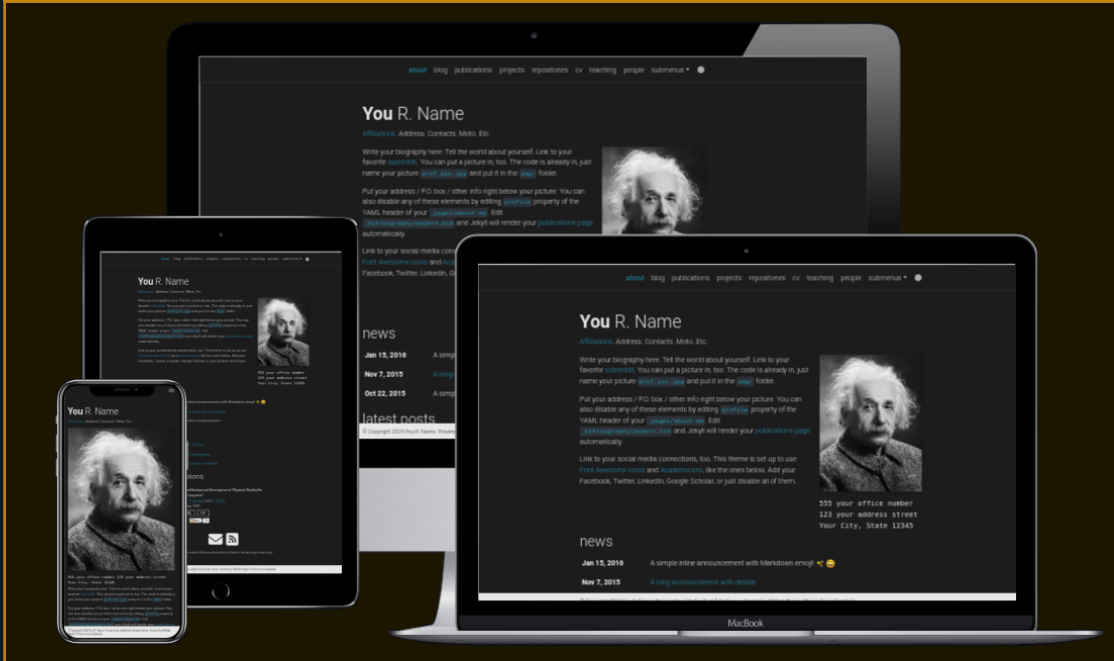
```
---
layout: about
title: About
permalink: /
subtitle: >
  OzGrav | QUASAR | University of
  Queensland
---
```

I am a PhD candidate in astrophysics at the University of Queensland, supervised by Prof. Holger Baumgardt. My research focuses on the dynamical evolution of globular clusters -- in particular, how black hole natal kicks shape the compact object populations of systems like 47 Tucanae and NGC 3201.

I run N-body simulations (CMC, NBODY7) and develop open-source tools for the community. Member of OzGrav and the QUASAR Collaboration.

Your First Website with Jekyll

AI-folio Example



Changes to Make

- Change the name of the website
 - Add a name for your website
- Change the photo
 - Replace `assets/img/prof_pic.jpg` with your photo
- Change publications to bibliography
 - Add entries to `_bibliography/papers.bib`
- Add a new page
 - Add a blank page with no content
- Disable other pages, reduce website clutter
 - Delete or set nav to false

PART TWO

AI as Your Co-Author & Web Designer

Writing content and customising design - no coding required

AI for Content

Using AI can be a gamechanger for adding content to your website but always begin with your own work. AI is an iterative tool.

Writing your bio

Paste your CV intro or thesis abstract. Ask AI to rewrite it in 2 paragraphs for a general audience. Edit to sound like you.

Tagline or page headline

Describe your research. Ask for 10 one-sentence taglines. Use one, adapt another, discard the rest.

Publication descriptions

Paste a title and abstract. Ask AI for a one-sentence lay summary for your publications list.

Research summaries

Paste a paper abstract. Ask AI for a 3-sentence plain-English version. Structure: what was the question, what we found, why it matters.

Plain-language audit

Paste a draft paragraph. Ask AI to flag jargon, passive voice, or phrasing that would lose a non-specialist reader.

`_config.yml` description

Ask AI to write your site's 1-2 sentence meta description based on your name, role, and research area.

AI for Web Design

You do not need to know CSS, HTML, or JavaScript. AI can read your theme files and write modifications on your behalf:

Colour scheme

Paste your `_sass/_variables.scss`. Ask AI to update the accent colour to match your university palette. It writes the exact lines to change.

Fonts

Ask which Google Font pairing suits an academic site, then ask for the exact lines to add to your config or CSS. One copy-paste.

Card layout tweaks

Describe the change ('wider project cards, less padding'). AI writes the CSS rule. You paste it into `assets/css/main.scss`.

New sections

'Add a News section to my `about.md` with three entries.' *AI writes the complete Markdown block, formatted for Jekyll.*

New pages

'Write a `talks.md` page listing three conference presentations.' *AI produces the complete file with correct Jekyll front matter.*

Navigation menu

'Add a Software link to my nav bar.' *AI finds the right file and writes the exact addition.*

Dark / light mode

Many themes toggle this via a single config entry. *AI finds which variable to change and explains what it controls.*

Debugging

'My nav bar isn't showing the Software pag, here is my `_config.yml`.' *AI diagnoses the problem and writes the fix.*

The Traditional AI Design Workflow

A repeatable three-step loop for any theme change, large or small:

1

Give AI the file

Open the relevant file (e.g. `_sass/_variables.scss` or `_config.yml`). Select all. Paste into Claude or ChatGPT.

Say: "Here is my Jekyll theme's [filename]. I want to [goal]. Show me exactly what to edit."

```
-----
* Variables used throughout the theme.
* To adjust anything, simply edit the variables below and rebuild the theme.
-----

@theme "base-colors";

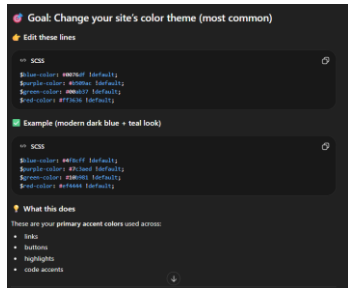
// Colors
$red-color: #ff3333 !default;
$red-color-dark: #801111 !default;
$orange-color: #ff6600 !default;
$blue-color: #0070c0 !default;
$blue-color-dark: #004d40 !default;
$cyan-color: #00cccc !default;
$light-cyan-color: color.adjust($cyan-color, $lightness: 25%);
$green-color: #008000 !default;
$green-color-light: #90e090 !default;
$green-color-dark: #006600 !default;
$green-color-light: #90e090 !default;
$green-color-light: #90e090 !default;
$green-color-light: #90e090 !default;
$purple-color: #800080 !default;
$light-purple-color: color.adjust($purple-color, $lightness: 25%);
$pink-color: #ff6666 !default;
$pink-color-light: #ffcccc !default;
$yellow-color: #ffff00 !default;
```

2

Apply the change

AI returns the exact lines to replace. Open the file. Make the change. Click Commit.

The site rebuilds automatically in approximately 60 seconds.

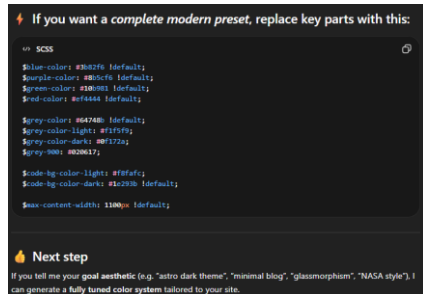


3

Iterate

If the result is not right, paste the new file back in and describe the problem. AI corrects it.

No terminal. No CSS expertise. No software to install. Works entirely from the browser.



Using AI Coding Agents*

Beyond prompting in the browser, Agents are AI tools that read and write your actual files:

Tools

Claude Code

CLI agent, reads your entire repo, edits files, runs commands. Best for structural changes.

Cursor

VS Code fork with built-in AI. Knows your full codebase. Best for day-to-day editing.

GitHub Copilot

In-editor autocomplete + chat. Good for small edits when you're already in VS Code, planning agents have access to your full codebase.

Antigravity

VS Code fork by Google, AI-Agent focused development with access to your codebase.

What AI agents can do with your Jekyll or HTML5 site

- Modify Sass variables and CSS rules across files
- Add new layout files or includes
- Update `_config.yml` and `_data/` files
- Write new Markdown pages with correct front matter
- Debug Liquid template errors with full context
- Refactor navigation and site structure

Limitations to know

- Often can introduce annoying errors and bugs into your codebase
- Large repos may exceed context window
- Does not work well unless you have structured and precise prompts.
- Most tools cost a lot and have heavy usage limitations (model dependent)

Agentic Website Design Demo

Let us use AI agents to help us redesign our site. When we prompt our AI agent we must use the following structure:

- ROLE** You are an expert front-end developer working on a Jekyll / al-folio site.
- GOAL** Redesign the homepage hero section to feel modern and academic – improve typography, spacing, and visual hierarchy.
- RULES** Do NOT modify `_config.yml`.
Preserve all Liquid tags.
Edit `_sass/_variables.scss` for colours.

Role · Goal · Rules -> perfect way of getting the agent to do what you want.

Pros/Cons

- Pros
 - High quality design, much nicer looking than the template provided.
 - Didn't have to code a thing.
 - Faster than I would have been
- Cons
 - Prompts was incredibly specific (I had to know what I was doing)
 - Often introduces errors and bugs which may be hard to diagnose
 - One big prompt may use all your tokens, and you can't design anymore.

Prompt Cookbook

The more specific context you give, the better the output. These work in Claude, ChatGPT, or Gemini.

Bio

```
"I am a PhD candidate in astrophysics at UQ, member of OzGrav and QUASAR. My research uses N-body simulations to study black hole natal kicks in globular clusters. Write a 2-paragraph About Me for my research website. Audience: scientifically literate non-specialists. Be specific. Avoid phrases like 'passionate researcher'." -> If you have extra information add it in (Papers, Projects, etc.)
```

Research blurb

```
"Here is my paper abstract: [paste]. Rewrite as a 3-sentence plain-English summary for my research website. Structure: (1) what was the question, (2) what we found, (3) why it matters for the field. Avoid field-specific jargon."
```

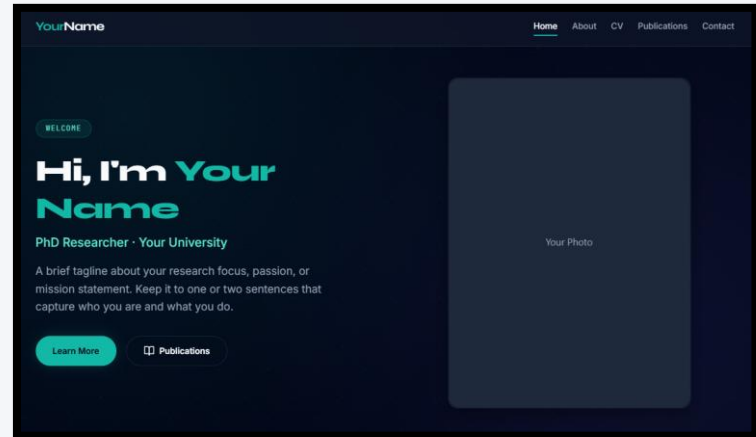
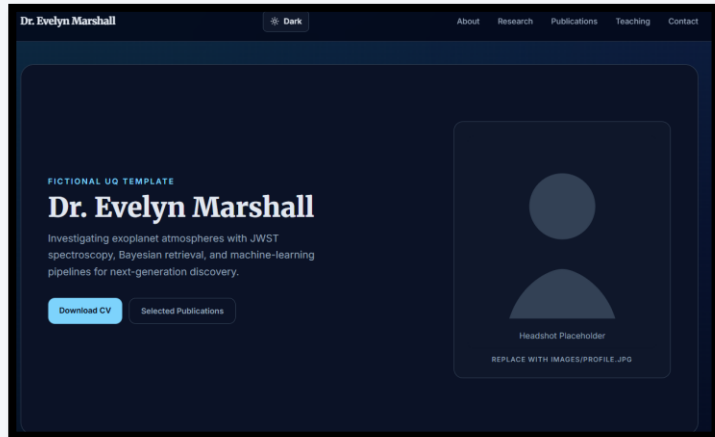
Design change

```
"Here is my Jekyll theme's _sass/_variables.scss: [paste file]. Change the primary accent to #0E7C7B (UQ teal) and the heading colour to #1A2332. Show me exactly which lines to change and what to replace them with."
```

Always verify AI output before publishing, especially for research descriptions. Fact-check every specific claim.

Bonus: HTML5 Templates

I've also developed some HTML/CSS/Javascript Templates for you to play around with. Download these and edit with AI.



Next Steps

Three Things To Get Started

1

Setup GitHub Pages

- Use your GitHub account to create a [new repo for your website with GitHub Pages](#)

2

Use the al-folio template, or choose another

github.com/alshedivat/al-folio Fork the repo or Use a template, enable Pages, fill in `_config.yml`. Viola, your site is live!

3

Redesign your Website

Look through the [template QuickStart guide](#) to start making changes or use an [AI agent](#) to speed things up

Questions?

APPENDIX . A

Developing your website with Jekyll

Installation, concepts, themes, and AI-assisted editing

What Is Jekyll?

Static site generator

Converts Markdown + templates into plain HTML/CSS/JS -- no database, no server-side code

GitHub Pages native

GitHub runs Jekyll automatically on every push -- no build step needed on your end

Liquid templating

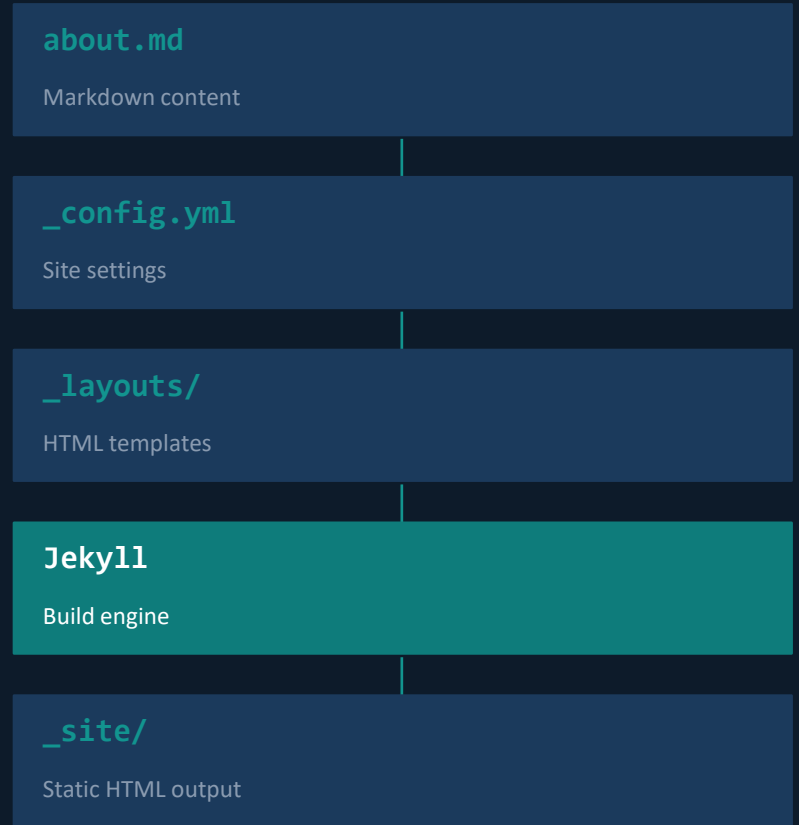
Simple `{{ variable }}` and `{% logic %}` tags let you reuse layouts across pages

Convention over config

Drop a file in the right folder and Jekyll knows what to do with it

Free & open source

Ruby-based, MIT licence, large ecosystem of themes and plugins



Do You Need to Install Jekyll?

Short answer: no -- for most researchers, GitHub builds the site for you.

Without local install

YES

- Edit files directly in GitHub browser editor
- Commit -- GitHub rebuilds site in ~60 seconds
- No Ruby, no terminal, no dependencies
- Sufficient for 90% of academic use cases
- Recommended starting point

With local install

OPTIONAL

- Preview changes at localhost:4000 before pushing
- Faster iteration -- no 60-second wait per change
- Required for developing custom plugins or themes
- Needed for complex Liquid template debugging
- Worth it once your site is past initial setup

Three Ways to Use Jekyll Themes

Choose based on how much control vs simplicity you need:

1 GitHub built-in themes

Best when: Quickest possible start

How: Settings -> Pages -> Choose a theme

- One click in GitHub Settings
 - No files to manage
 - Instant deployment
-
- Very limited selection
 - No academic layouts
 - Hard to customise

```
theme: minima # in _config.yml
```

2 Remote gem-based theme

Best when: Want a theme without forking

How: Add to `_config.yml` + Gemfile

- Clean repo -- your content only
 - Easy theme version upgrades
 - Works with many popular themes
-
- Some themes not remote-compatible
 - Can't edit theme files directly
 - Slower local builds

```
remote_theme: alshdivat/al-folio
```

3 Fork the theme repo

Best when: Full control -- recommended

How: Fork on GitHub, edit freely

- Edit any file in the theme
 - Full control over layout and style
 - AI tools can modify any file
-
- Manual updates from upstream
 - More files in your repo
 - Slightly steeper onboarding

```
# No theme: line needed  
# Theme files are in your repo
```

Installing Jekyll on macOS

Prerequisites: macOS 12+, internet connection. Total time: ~10 minutes.

1 Install Homebrew

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install
1.sh)"
```

Skip if already installed. Verify: `brew --version`

3 Install Ruby via rbenv

```
rbenv install 3.3.0
rbenv global 3.3.0
ruby --version # should show 3.3.0
```

Do not use macOS system Ruby -- use rbenv-managed version

2 Install rbenv + Ruby build

```
brew install rbenv ruby-build
rbenv init # follow the printed instructions
# Restart your terminal, then:
```

rbenv manages Ruby versions without conflicts

4 Install Bundler + Jekyll

```
gem install bundler jekyll
jekyll --version # verify install
```

Bundler manages per-project gem dependencies

Installing Jekyll on Windows

Prerequisites: Windows 10/11 64-bit. Total time: ~15 minutes.

1 Install RubyInstaller + Devkit

```
# Download from rubyinstaller.org  
# Choose: Ruby+Devkit 3.3.X (x64)  
# Run installer -- check all three boxes
```

Must use Devkit version -- not plain Ruby

3 Install Bundler + Jekyll

```
# Open a NEW Command Prompt or PowerShell:  
gem install bundler jekyll  
jekyll --version
```

Open a fresh terminal after RubyInstaller -- not the installer's terminal

2 Run MSYS2 / Devkit setup

```
# At end of installer, terminal opens:  
ridk install  
# Select option 3 (MSYS2 and MINGW dev toolchain)
```

Installs compilers needed to build native gems

4 Troubleshooting tips

```
# If 'jekyll' not found:  
$env:PATH # check Ruby bin is listed  
# Re-open terminal and retry  
gem env # shows gem install paths
```

Most issues are PATH problems -- a fresh terminal fixes them

First Run -- Local Preview

After cloning your forked theme to your machine:

Step 1 -- install dependencies

```
cd your-site-folder
bundle install
```

Step 2 -- serve the site

```
bundle exec jekyll serve
```

Step 3 -- open in browser

```
http://localhost:4000
```

Ctrl+C to stop the server

Expected output

```
Configuration file: _config.yml
      Source: /your-site
      Destination: _site/
Incremental build: disabled
      Generating...
                        done in 2.3s.
Auto-regeneration: enabled
      Server address: http://127.0.0.1:4000
      Server running... press ctrl-c to stop.
```

Tips

- livereload:** Auto-refreshes browser on file save: `bundle exec jekyll serve --livereload`
- drafts:** Include draft posts in preview: `bundle exec jekyll serve --drafts`
- _config error?:** YAML is whitespace-sensitive -- check indentation with a YAML linter

Key Jekyll Concepts

Front matter

```
---  
layout: about  
title: "About Me"  
permalink: /  
---
```

YAML block at top of every page. Sets layout, title, permalink, custom vars.

Includes

```
{% include nav.html %}  
{% include footer.html  
  year=2025 %}
```

Reusable snippets in `_includes/`. Call anywhere with `{% include %}`.

`_posts/`

```
YYYY-MM-DD-title.md -> /blog/title/
```

Layouts

```
<!-- _layouts/about.html -->  
<html>  
  {{ content }}  
</html>
```

HTML wrappers in `_layouts/`. Pages inject their content via `{{ content }}`.

Liquid

```
{{ page.title }}  
{% if page.author %}  
  By {{ page.author }}  
{% endif %}
```

Templating language. `{{ output }}`, `{% logic %}`, `| filters`.

`_data/`

```
_data/pubs.yml -> {{ site.data.pubs }}
```

Extra Recommended Jekyll Themes

All free, all GitHub Pages compatible, all maintained as of 2025:

al-folio

alshedivat/al-folio

Academic research sites

- BibTeX publication import
- Projects + blog + news
- Scholar & ORCID integration
- Used widely in CS & astro

Chirpy

cotes2020/jekyll-theme-chirpy

Technical blog / notes

- Dark mode out of the box
- Table of contents, search
- Excellent code highlighting
- Best for technical writing

Minimal Mistakes

mmistakes/minimal-mistakes

Flexible portfolio

- Multiple layout options
- Strong documentation
- Works for personal + research
- Highly customisable sidebar

Beautiful Jekyll

daattali/beautiful-jekyll

Easiest to start with

- Zero local install needed
- Edit `_config.yml` only
- Good defaults for academics
- Excellent beginner docs

Just the Docs

just-the-docs/just-the-docs

Documentation / code docs

- Built-in search
- Clean nested navigation
- Ideal for software projects
- Pairs well with a main site

Summary & Next Steps

What we covered

- Jekyll: static site generator, runs free on GitHub Pages
- No local install needed -- GitHub builds for you
- Three theme approaches: built-in, remote, fork (use fork)
- Local install: rbenv on Mac, RubyInstaller on Windows
- `bundle install + bundle exec jekyll serve + localhost:4000`
- Front matter, layouts, includes, Liquid, `_posts/`, `_data/`

Next steps

1

Fork al-folio

github.com/alshedivat/al-folio

2

Enable GitHub Pages

Settings -> Pages -> GitHub Actions

3

Edit `_config.yml`

Name, ORCID, institution, ADS library

4

Write bio with AI

Use the prompt cookbook from earlier

5

Try local preview

`bundle exec jekyll serve` -- optional but useful