MkDocs Docs from Markdown

START HERE

`pip install mkdocs`
Install MkDocs

`mkdocs build --strict`
Fail on warnings

`mkdocs new <dir>`
Create a project

`mkdocs gh-deploy`
Publish to GitHub Pages

`mkdocs serve`
Live preview

`mkdocs get-deps`
List needed packages

`mkdocs build`
Build the site

`mkdocs --version`
Show version

docs/*.md your pages → `mkdocs.yml` **Config + theme** nav, plugins → `mkdocs build` **site/** static HTML → `gh-deploy` **gh-pages** hosted site

CONCEPTS

Docs are plain Markdown files in docs/, wired together by mkdocs.yml. Run commands from the folder holding mkdocs.yml.

site/ is generated output: keep it out of git (add it to .gitignore).

Plugins and themes such as Material are separate pip packages; get-deps lists what your config needs.

PROJECT

`mkdocs new <dir>`
Create mkdocs.yml and docs/index.md

`mkdocs new .`
Scaffold in the current folder

`pip install mkdocs`
Install MkDocs

`pip install mkdocs-material`
Install the Material theme

`mkdocs --version`
Show installed version

`mkdocs get-deps`
Print packages the config needs

`pip install $(mkdocs get-deps)`
Install those packages

PREVIEW

`mkdocs serve`
Serve at localhost:8000 with live reload

`mkdocs serve -a <host:port>`
Pick address and port

`mkdocs serve -a 0.0.0.0:8000`
Reachable from other machines

PREVIEW (cont.)

`mkdocs serve --dirty`
Rebuild only changed pages

`mkdocs serve -w <path>`
Also watch another dir or file

`mkdocs serve --watch-theme`
Reload on theme changes

`mkdocs serve --no-livereload`
Disable auto reload

`mkdocs serve -f <file>`
Use another config file

BUILD AND DEPLOY

`mkdocs build`
Write the site to site/

`mkdocs build --strict`
Abort on any warning

`mkdocs build -d <dir>`
Choose the output folder

`mkdocs build -f <file>`
Use another config file

`mkdocs build --clean`
Remove stale files first (default)

`mkdocs build --no-directory-urls`
Use page.html style URLs

`mkdocs build -t readthedocs`
Override the built-in theme

`mkdocs gh-deploy`
Build and push to gh-pages

`mkdocs gh-deploy -m "<msg>"`
Set the deploy commit message

`mkdocs gh-deploy -b <branch>`
Deploy to another branch

`mkdocs gh-deploy -r <remote>`
Deploy to another remote

BUILD AND DEPLOY (cont.)

`mkdocs gh-deploy --force`
Force-push the branch

`mkdocs gh-deploy --no-history`
Squash branch history

MKDOCS.YML

`site_name: <title>`
Required site title

`site_url: <url>`
Canonical URL of the site

`nav:`
Page order and titles

`theme:`
`  name: <theme>`
Choose theme (mkdocs, readthedocs, ...)

`plugins: [search]`
Enable plugins (replaces defaults)

`markdown_extensions:`
Add Python-Markdown extensions

`docs_dir: <dir>`
Source folder (default docs)

`site_dir: <dir>`
Output folder (default site)

`strict: true`
Warnings fail the build

`repo_url: <url>`
Link to source repo

`edit_uri: edit/main/docs/`
Per-page edit link path

`extra_css:`
Extra stylesheets in docs_dir

`validation:`
Set link problems to warn or ignore

NAVIGATION

`- Home: index.md`
Title plus file path

`- index.md`
File only, title taken from page

```
- Guide:
  - Intro: guide/intro.md
```
Nested section

`- Issues: https://<url>`
External link, title required

Without nav, MkDocs lists all pages alphabetically. Paths are relative to docs_dir.

Where things live

Path	Purpose
`mkdocs.yml`	configuration
`docs/index.md`	home page
`docs/img/`	images, linked relatively
`site/`	build output

RECIPES

New project, preview

```
pip install mkdocs
mkdocs new mysite
cd mysite
mkdocs serve
```

Material theme with search

```
pip install mkdocs-material
# mkdocs.yml
theme:
  name: material
plugins:
  - search
```

CI check, then publish

```
mkdocs build --strict
mkdocs gh-deploy --force
```

Preview on the LAN

```
mkdocs serve -a 0.0.0.0:8000 --watch-theme
```