Sphinx Documentation generator

START HERE

`sphinx-quickstart`
Scaffold a project

`sphinx-build -E -a <src> <out>`
Full clean rebuild

`sphinx-build -b html <src> <out>`
Build the HTML

`sphinx-build -b linkcheck <src> <out>`
Find dead links

`make html`
Same via the Makefile

`sphinx-apidoc -o <out> <pkg>`
Stub pages for a package

`sphinx-build -W <src> <out>`
Warnings fail the build

`sphinx-build -j auto <src> <out>`
Build in parallel

conf.py + .rst files source dir → `read + resolve` → **Doctrees** cached environment → `-b builder` → **Output** html, latex, man, text...

One source tree, many outputs

CONCEPTS

Sources are reST (or Markdown via an extension) plus conf.py; sphinx-build turns them into any builder's output.

index.rst holds the root toctree; a page not listed in any toctree triggers a warning.

Builds are incremental: use -E when config or extensions changed, -a to rewrite all output.

PROJECT SETUP

`sphinx-quickstart`
Interactive project scaffold

`sphinx-quickstart -q -p <name> -a <author> -v <ver>`
Non-interactive scaffold

`sphinx-quickstart --sep`
Separate source/ and build/ dirs

`sphinx-build --version`
Show installed version

BUILDING

`sphinx-build -b html <src> <out>`
Build HTML

`sphinx-build -M html <src> <out>`
Make-mode build, doctrees under out

`make html`
Build via generated Makefile

`make clean`
Delete the build directory

`-W`
Treat warnings as errors
Combine with --keep-going on old versions to see all warnings; always on since 8.1.

`-a`
Write all output files

`-E`
Rebuild environment from scratch

`-j auto`
Parallel build on all CPUs

`-n`
Nitpicky: warn on missing references

`-T`
Show full traceback on errors

BUILDING (cont.)

`-w <file>`
Also write warnings to a file

`-D <key>=<value>`
Override a config value

BUILDERS

`-b html`
Standard HTML pages

`-b dirhtml`
HTML with pretty directory URLs

`-b singlehtml`
Whole project in one HTML file

`-b latex`
LaTeX sources

`make latexpdf`
Build PDF via LaTeX

`-b linkcheck`
Check external links

`-b dummy`
Parse and check only, no output

`-b doctest`
Run doctests in the docs

`-b coverage`
Report undocumented Python objects

`-b gettext`
Extract .pot translation catalogs

CONF.PY ESSENTIALS

`project = "<name>"`
Project name

`author = "<name>"`
Author string

`release = "<x.y.z>"`
Full version, used as |release|

`extensions = ["<ext>"]`
Load extensions

`html_theme = "<theme>"`
Pick the HTML theme

`html_static_path = ["_static"]`
Extra static files

`exclude_patterns = ["<glob>"]`
Skip source files

`nitpicky = True`
Warn on every missing reference

`sys.path.insert(0, os.path.abspath(".."))`
Make your package importable

CONF.PY ESSENTIALS (cont.)

`linkcheck_ignore = [r"<regex>"]`
Skip URLs in linkcheck

REST BASICS

`Title\n=====`
Section heading (underline)

`*em* / **strong / ``code``**`
Inline markup

`* item / 1. item / #. item`
Bullet, numbered, auto-numbered

`` `text <https://url>`__ ``
Inline external link

`.. code-block:: <lang>`
Highlighted code block

`paragraph::`
Literal block follows, indented

`.. image:: <path>`
Insert an image

`.. note::`
Admonition (also warning, tip...)

`.. toctree::`
Table of contents / page tree

`.. literalinclude:: <file>`
Include a source file as code

CROSS-REFERENCES

`.. _<label>:`
Define a label before a heading

`:ref:`<label>``
Link to a label

`:ref:`text <label>``
Link with custom text

`:doc:`<path>``
Link to another document

`:py:func:`~pkg.mod.func``
Python object, short display

`intersphinx_mapping = {...}`
Link to other projects' docs

AUTODOC AND APIDOC

`extensions = ["sphinx.ext.autodoc"]`
Enable autodoc

`.. automodule:: <pkg.mod>`
Document a whole module

AUTODOC AND APIDOC (cont.)

`:members:`
Include documented members

`:undoc-members:`
Include members without docstrings

`.. autoclass:: <pkg.Class>`
Document a class

`:show-inheritance:`
Show base classes

`.. autofunction:: <pkg.func>`
Document one function

`autodoc_default_options = {...}`
Default directive options

`autodoc_member_order = "bysource"`
Keep source order

`autodoc_typehints = "description"`
Move type hints to the body

`autodoc_mock_imports = ["<mod>"]`
Fake heavy imports on build

`sphinx-apidoc -o <out> <pkg>`
Generate stub .rst per module

`sphinx-apidoc -f`
Overwrite existing stubs

`sphinx.ext.napoleon`
Parse Google/NumPy docstrings

RECIPES

Strict CI build

```
sphinx-build -W --keep-going -n
-b html \
  docs docs/_build/html
```

Docs for a Python package

```
sphinx-quickstart -q docs -p
MyPkg -a Me -v 1.0 \
  --ext-autodoc --
extensions=sphinx.ext.napoleon
sphinx-apidoc -o docs/api src/
mypkg
make -C docs html
```

Link check, then PDF

```
sphinx-build -b linkcheck docs
docs/_build/lc
make -C docs latexpdf
```

conf.py for autodoc

```
import os, sys
sys.path.insert(0,
os.path.abspath("../src"))
extensions =
["sphinx.ext.autodoc",
"sphinx.ext.intersphinx"]
intersphinx_mapping = {
  "python": ("https://docs.
python.org/3", None)}
```