Sphinx Documentation generator

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

BUILDING

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

make html
Build via generated Makefile

-W
Treat warnings as errors

BUILDING (cont.)

-a
Write all output files

-E
Rebuild environment from scratch

-j auto
Parallel build on all CPUs

-n
Nitpicky: warn on missing references

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

BUILDERS

-b html
Standard HTML pages

make latexpdf
Build PDF via LaTeX

-b linkcheck
Check external links

CONF.PY ESSENTIALS

project = "*<name>***"**
Project name

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

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

CONF.PY ESSENTIALS (cont.)

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

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

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

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

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

.. toctree::
Table of contents / page tree

CROSS-REFERENCES

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

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

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

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

AUTODOC AND APIDOC

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

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

:members:
Include documented members

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

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

AUTODOC AND APIDOC (cont.)

autodoc_default_options = {...}
Default directive options

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.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
```