Sphinx Documentation generator

CONCEPTS

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

conf.py + .rst files source dir
↓ read + resolve
Doctrees cached environment
↓ -b builder
Output html, latex, man, text...
One source tree, many outputs

PROJECT SETUP

sphinx-quickstart
Interactive project scaffold

BUILDING

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

make html
Build via generated Makefile

-W
Treat warnings as errors

-E
Rebuild environment from scratch

-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

CONF.PY ESSENTIALS (cont.)

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

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

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

.. 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

AUTODOC AND APIDOC

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

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

:members:
Include documented members

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