

Python Documentation README

This directory contains the reStructuredText (reST) sources to the Python documentation. You don't need to build them yourself, [prebuilt versions are available](#).

Documentation on authoring Python documentation, including information about both style and markup, is available in the ["Documenting Python"](#) chapter of the developers guide.

Building the docs

The documentation is built with several tools which are not included in this tree but are maintained separately and are available from [PyPI](#).

- [Sphinx](#)
- [blurb](#)
- [python-docs-theme](#)

The easiest way to install these tools is to create a virtual environment and install the tools into there.

Using make

To get started on UNIX, you can create a virtual environment and build documentation with the commands:

```
make venv
make html
```

The virtual environment in the `venv` directory will contain all the tools necessary to build the documentation downloaded and installed from PyPI. If you'd like to create the virtual environment in a different location, you can specify it using the `VENVDIR` variable.

You can also skip creating the virtual environment altogether, in which case the Makefile will look for instances of `sphinxbuild` and `blurb` installed on your process `PATH` (configurable with the `SPHINXBUILD` and `BLURB` variables).

On Windows, we try to emulate the Makefile as closely as possible with a `make.bat` file. If you need to specify the Python interpreter to use, set the `PYTHON` environment variable.

Available make targets are:

- "clean", which removes all build files and the virtual environment.
- "clean-venv", which removes the virtual environment directory.
- "venv", which creates a virtual environment with all necessary tools installed.
- "html", which builds standalone HTML files for offline viewing.
- "htmlview", which re-uses the "html" builder, but then opens the main page in your default web browser.
- "htmlhelp", which builds HTML files and a HTML Help project file usable to convert them into a single Compiled HTML (.chm) file -- these are popular under Microsoft Windows, but very handy on every platform.

To create the CHM file, you need to run the Microsoft HTML Help Workshop over the generated project (.hlp) file. The `make.bat` script does this for you on Windows.

- "latex", which builds LaTeX source files as input to "pdflatex" to produce PDF documents.
- "text", which builds a plain text file for each source file.
- "epub", which builds an EPUB document, suitable to be viewed on e-book readers.
- "linkcheck", which checks all external references to see whether they are broken, redirected or malformed, and outputs this information to stdout as well as a plain-text (.txt) file.
- "changes", which builds an overview over all versionadded/versionchanged/ deprecated items in the current version. This is meant as a help for the writer of the "What's New" document.
- "coverage", which builds a coverage overview for standard library modules and C API.
- "pydoc-topics", which builds a Python module containing a dictionary with plain text documentation for the labels defined in `tools/pyspecific.py` -- pydoc needs these to show topic and keyword help.
- "suspicious", which checks the parsed markup for text that looks like malformed and thus unconverted reST.
- "check", which checks for frequent markup errors.
- "serve", which serves the build/html directory on port 8000.
- "dist", (Unix only) which creates distributable archives of HTML, text, PDF, and EPUB builds.

Without make

First, install the tool dependencies from PyPI.

Then, from the `Doc` directory, run

```
sphinx-build -b<builder> . build/<builder>
```

where `<builder>` is one of `html`, `text`, `latex`, or `htmlhelp` (for explanations see the `make` targets above).

Deprecation header

You can define the `outdated` variable in `html_context` to show a red banner on each page redirecting to the "latest" version.

The link points to the same page on `/3/`, sadly for the moment the language is lost during the process.

Contributing

Bugs in the content should be reported to the [Python bug tracker](#).

Bugs in the toolset should be reported to the tools themselves.

You can also send a mail to the Python Documentation Team at docs@python.org and we will process your request as soon as possible.

If you want to help the Documentation Team, you are always welcome. Just send a mail to docs@python.org.