Demo of PDF Generation

Author A. Name

авг. 10, 2022

Оглавление

ГЛАВА 1

sphinx-pdf-with-mudkip

Create a PDF with Cyrillic letters via Sphinx and pdflatex

1.1 How to use

1. Open project in Github Codespaces or other linux environment
2. Run `./start.sh` to install dependencies
3. `just latex` creates latex files (including `demo.tex`)
4. `just pdf` creates `demo.pdf`

1.2 Development notes

1. Sphinx project managed with mudkip:
   - `[mudkip.override]` in `mudkip.toml` used instead of `conf.py`
   - `just latex` in an alias for `mudkit build --preset latex --output-dir docs/_latex`
2. PDF created in Github Codespaces:
   - `start.sh` installs latex and other dependencies
   - `just pdf` invokes PDF build, it is an alias for `pdflatex demo.tex`
3. Avoid errors with Cyrillic characters (mostly fixed):
   - avoid unnecessary installations in `setup.sh`
4. Make PDF look sane:
   - generate TOC
   - glossary is not a numbered chapter

- blank pages

5. PDF customisation:

- watermark
- "issued to" footer

6. Build PDF in a CI

1.3 Non-goals

- Codespaces container with latex installed
- Switch to MyST index.md
- Configure VS Code for TOML viewing
- Add .mp4 to latex

1.4 Failure points

1. `pdflatex` can fail here after encountering Cyrillic letters:

- это точка отказа;
- очень проблемная буква: ё/Ё.

ГЛАВА 2

Почему настроить LaTeX это такая жесть?

Выберите правильный ответ:

- подумаешь 2 гига установить, не беда
- но ведь все равно не работает!
- просто так легкой жизни захотелось?

ГЛАВА 3

Глоссарий

2to3
: A tool that tries to convert Python 2.x code to Python 3.x code by handling most of the incompatibilities which can be detected by parsing the source and traversing the parse tree.

  2to3 is available in the standard library as `lib2to3`; a standalone entry point is provided as `Tools/scripts/2to3`. See 2to3 — Automated Python 2 to 3 code translation.

abstract base class
: Abstract base classes complement duck-typing by providing a way to define interfaces when other techniques like `hasattr()` would be clumsy or subtly wrong (for example with magic methods). ABCs introduce virtual subclasses, which are classes that don't inherit from a class but are still recognized by `isinstance()` and `issubclass()`; see the `abc` module documentation. Python comes with many built-in ABCs for data structures (in the `collections.abc` module), numbers (in the `numbers` module), streams (in the `io` module), import finders and loaders (in the `importlib.abc` module). You can create your own ABCs with the `abc` module.