PreTeXt Quick Reference: Command Line Interface (CLI)

CLI version 0.8.0, 6/24/2022

Setup

Check requirements

Note: `python` might be called `python3` if you have MacOS or Linux.

`python --version` : the CLI requires Python version 3.8 or later

`pip --version` : pip is the package installer for Python

`xelatex --version` : some PreTeXt features require LaTeX

Install PreTeXt

`python -m pip install pretext-cli` : install PreTeXt

`pretext --version` : check version to verify install

Create a new project

`pretext new book` : creates a new PreTeXt book in `new-pretext-project`

`pretext new article` : creates a new PreTeXt article `new-pretext-project`

Update a project to use the CLI

`pretext init`: creates project manifest (`project.ptx`), and publication file (`publication/publication.ptx`). Edit these files appropriately before proceeding.

`pretext pretext init --refresh`: creates new copies of project manifest and publication file to compare for new features.

Upgrade PreTeXt

`python -m pip install --upgrade pretext-cli`: upgrade to latest stable release

Get Help

`pretext --help`: show general help

`pretext build --help`: show help for build command. Each subcommand has its own help.

Basic Usage

Build a PreTeXt document

`pretext build`: Builds the project to the format of the first target in `project.ptx`.

`pretext build web`: Create html version (assuming `<target name="web">`) is in `publication.ptx`

`pretext build print`: Create print (pdf) version

Generate source images and WeBWorK

If your book has any WeBWorK, latex-image, asymptote, sageplot, interactive, etc. you need to generate these from source.

`pretext generate`: Generate all assets for first target in `project.ptx`.

`pretext generate webwork`: Generate webwork for first target in `publication.ptx`

`pretext generate sageplot -t print`: Generate sageplot for target ”print”.

`pretext generate latex-image -x img-graph1`: Generate latex-image with xml:id ”img-graph1” (for first target).

View a PreTeXt document (local)

`pretext view`: Creates a local server to preview the first target in `project.ptx`

`pretext view print`: Views the ”print” target

`pretext view -w`: Builds, starts server, and rebuilds every time the project is saved

`CTRL+C` to close the server

Deploy to GitHub Pages

`pretext deploy` : deploys Git-managed project to GitHub Pages

`pretext deploy -u` : deploys and also uploads source files

Useful Shortcuts

`pretext build -g`: build and generate in one step

`pretext build web -g latex-image`: build web target and generate latex-images

`pretext view -b`: build before you preview

`pretext view -g`: generate assets before you view

`pretext view -bg`: build and generate assets before you view

Project Manifest The file `project.ptx` describes your build targets. Each target has a *name* (e.g. ”print-latex”) that you build or view with, e.g. `pretext build print-latex` or generate assets for with, e.g. `pretext generate webwork -t print-latex`.

Structure of a target:

```
<target name="web">
  <format>html</format>
  <source>source/main.ptx</source>
  <publication>publication/publication.ptx</publ
  <output-dir>output/web</output-dir>
</target>
```

`<format>` can be html, latex, or pdf

`<source>` is the path to the root ptx document

`<publication>` is the path to the publication file

`<output-dir>` is the path the the folder that will hold output

Other optional elements:

`<stringparam key="..." value="..."/>`: allows for setting the value of string parameters

`<xmlid-root>ch-first</xmlid-root>`: used to restrict build to a subset of the source, starting with the element with xml:id ”ch-first”

`<xsl>xsl/custom-xsl.xsl</xsl>`: used to build with a custom xsl file. In that file, import the standard xsl with, e.g.

`<xsl:import pretext-href="pretext-common.xsl"/>`

publication.ptx

Information about the publication file goes here.

Common PreTeXt source tags

Blocks/Environments Example:

```
<theorem>
  <title>My Title</title>
  <statement>
    <p>
      Statement of theorem.
    </p>
  </statement>
  <proof>
    <p>
      The proof.
    </p>
  </proof>
</theorem>
```

Theorem-like: `<theorem>`, `<algorithm>`, `<claim>`, `<corollary>`, `<fact>`, `<identity>`, `<lemma>`, `<proposition>`.
Example-like: `<example>`, `<problem>`, `<question>`
Axiom-like: `<assumption>`, `<axiom>`, `<conjecture>`, `<heuristic>`, `<hypothesis>`, `<principle>`
Remark-like: `<remark>`, `<convention>`, `<insight>`, `<note>`, `<observation>`, `<warning>`
Project-like: `<project>`, `<activity>`, `<exploration>`, `<investigation>`

Other common blocks: `<definition>`
`<exercise>`
`<task>`: a division of an exercise or project-like

Examples

Examples go here.