Usage Examples

As already mentioned in the user section, the main way to control the program is by modifying the input files.

Setup JSON File

One way to modify the program is to redefine the mapping of CzeTex text functions to the code functions.

The default mapping is located in CzeTex/Setup.json, but there is also a pre-prepared version with Czech localization (it does not contain all functions as a demonstration that not all functions are required for it to work). You can find it in the CzeTex/examples folder under the name `nastaveni.json`, and the file `proNastaveni.txt` uses this mapping.

So if you want to try alternative mapping, you can run the command:
dotnet run examples/proNastaveni.txt examples/nastaveni.json

Creating the Input File

The project includes dozens of commands, and I do not consider it appropriate to list all of them here exhaustively, since they may not work with other setup files. Instead, I will demonstrate the idea behind creating them. You can find all commands in the specific setup file you plan to use, such as the default CzeTex/Setup.json, where all commands are commented and labeled with the proper *hashtag*.

Each command must start with the special character / and then continue with the command name. Different commands have different specifics, as will be shown below.

Non-paired Commands

Among non-paired commands is for example the **/section** command, which creates a new paragraph. Note: All text must be placed within some block or paragraph.

For example:

> **/pi**
>
> π

In Setup.json, these commands are labeled as **#nonPairFunction**.

Paired Commands

Paired commands include, for example, bold text block creation, i.e., **/bold** this is bold text **/x** would be bold text. You can notice that paired commands are closed using the generic character **/x** which implies that commands cannot overlap (the one that starts later is closed earlier).

For example:

> **/underline this text is underlined /x**
>
> this text is underlined

In Setup.json, these commands are labeled as **#pairFunction**.

Commands with Parameters

Some commands have parameters that change their functionality or appearance. For example, the power function **/pow** has 2 parameters: base and exponent.

So if we wanted to write the formula for the surface area of a sphere, it would look like this:
/math S = 4 **/times /pi /times /pow(r,2) /x** .

$$S = 4 \times \pi \times r^2$$

In Setup.json, these commands are labeled as **#hasParameter**.

Concrete Examples

Heading Level 1

/title This is heading text /x
or we can replace a specific heading with a generic one
/gtitle(1) This is heading text /x

This is heading text

Paragraph Text and Special Formatting

/section
/bold Czechia /xc full name /cursive Czech Republic /xc
is an /underline inland state /x in /linethrough eastern /x
/bold /cursive Central Europe /x /xd

Czechia, full name *Czech Republic*, is an inland state in ~~eastern~~ ***Central Europe***.

Note: As we can see in the example, extra spaces do not affect the final text. The output is generated only based on the commands and text. Even commands like /slash corresponding to the character /do not create a space between them.

Switching Between Sans-serif and Serif Fonts and Text Alignment,

/right
/serif
/size(15)
/section
The official name of the state according to the constitution is Czech Republic; the one-word name Czechia does not appear in the constitution, but is included in the official UN database as the single-word name of the country.
/x /x /x

The official name of the state according to the constitution is Czech Republic; the one-word name Czechia does not appear in the constitution, but is included in the official UN database as the single-word name of the country.

Note: It may seem that the generic closing command can become quite confusing, which can happen in very complex expressions. However, I concluded that in regular text this does not occur very often, and writing a shorter command increases typing speed and comfort, which I considered more important. Moreover, mathematical notation, which may use many commands on small text segments, only uses a few paired commands.

Mathematical Notation

/section
/math
The limit of f(x) as x approaches A equals L /iff /forall /epsilon > 0 /exists /delta > 0 /forall x /in P(A, /delta): f(x) /in U(L, /epsilon).
/x

The limit of f(x) as x approaches A equals $L \iff \forall\ \varepsilon > 0\ \exists\ \delta > 0\ \forall\ x \in P(\ A, \delta\ ):\ f(x) \in U(\ L, \varepsilon\ )$.

Technical Limitations

When working with lists, CzeTex only allows global font and size settings for the entire list, not per item or part of an item. Special formatting such as underlining is not supported at all.

For mathematical notation, only commands with a `get` method (i.e., special characters) can be used as parameters in powers, indices, and fractions. It is therefore not possible to have a fraction within a fraction or a power within a fraction, etc.

FOR MORE EXAMPLES, YOU CAN REVIEW SAMPLE FILES IN THE CZETEX/EXAMPLES FOLDER, FUNCTION DEFINITIONS IN SETUP.JSON, OR THE SOURCE CODE OF THIS DOCUMENTATION, AS THE ENTIRE DOCUMENT WAS CREATED USING CZETEX.