

Huan-gîng 使用 Clochur!

Welcome to use **Clóċur**, a toy editor, toy interpreter and a toy typesetting-engine frontend.

Author: Yoxem Chen (aka Tan, Kian-ting) <yoxem.tem98@nctu.edu.tw>

Website: <https://www.github.com/Yoxem/Clochur>

1. What is Clochur?

Clochur, or printed as "**Clóċur**" in Irish language ("*CLOW-kur*" Clóċur as Roman type, which means "typesetting"), is a toy-lisp typesetting language with a interpreter written in Python 3, and with a simple editor written in PyQt5 and QScintilla.

It generate a XML that is readable for SILE, which is a typesetting engine written in Lua, and it generate PDF with SILE.

The functions that it has (although may be buggy or needed to be tested) is:

- Macro expansion.
- call SILE command.
- count basic arithmetic expression.
- lambda function, function definition.

1. Why it's called Clochur?

The author has (unofficially) learned Irish language (for a while and uncontinuously), so use the name.

2. How is the language? It seems that it uses brackets insteads of parathesis.

The language is inspired by SILE and Scheme, even though it has some different characteristics. To make the code neat and consider that parathesis is used more often than bracket, so it's more suitable to use bracket for syntactical usage.

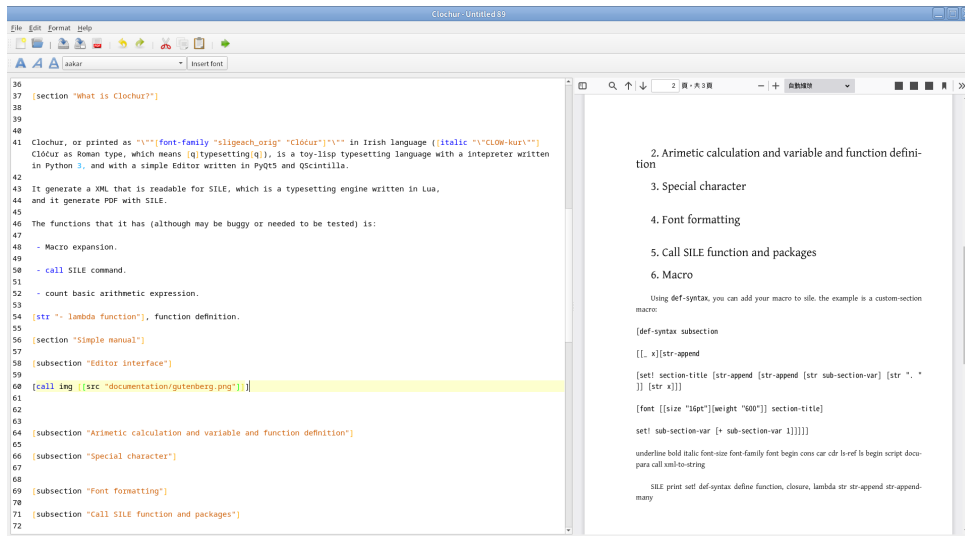
It's a toy language, so many of the function of Scheme, is not used here (for example call/cc), and there is no "let" to support local variables. However, you can call SILE function and using the packages of it with "call" and "script" respectively.

3. Does it support Taiwanese (Hokkien)/Hakka/Mandarin/Japanese/Korean or any other language that I want?

SILE supports utf-8, and Clochur will generate a XML that is readable to SILE, as long as any language that SILE can support, Clochur will support. If you find any bug, please tell me.

2. Simple manual

1. Editor interface



1. The interface of Clochur

Clochur is not a WYSIWYG editor, you have to type the lisp language by your self. Nevertheless, it's not a pure text-editor, it contains a PDF viewer powered by PDF.js, If you have edited you code, you can click the green right arrow button on the first toolbar to convert it to XML file, then generate and show PDF file from it.



1. The toolbars of Clochur

The description of the botton of the 1st toolbar is shown below (from left to right):

- Create a file (create a new window)
- Open a file
- Save a file
- Save as...
- Convert to PDF
- Redo
- Undo
- Cut
- Copy
- Paste

The description of the bottom of the 2nd toolbar is shown below (from left to right):

- apply bold macro to the selected text
- apply italic macro to the selected text
- apply underline macro to the selected text
- font list
- apply the font shown in the font list to the selected text

4. Basic input, [docu], and special character

Basically, you can type the sentence that you want to type directly in file. However, the document should be inside a macro call "docu". For example:

```
[docu Hello world!]
```

It will show

Hello world!

However, if it's a macro name (eg. docu, font, font-family, ...), you have to put them between 2 quotation marks". eg:

```
[docu "docu"]
```

It will show

docu

To type brackets [&], you should type \[& \] respectively; and to type backslash \, you should type \\. To type quotation mark ", you should type \". A word or a sentence between 2 quotation marks (") will be shown without the quotation marks. eg: "It will be there."

will be shown as:

It will be there.

To make the quotation mark shown, you should enclose it between quotation mark " and using \". "\"It will be there.\"" will be shown as:

"It will be there."

5. Arithmetic calculation and variable and function definition

6. Font formatting

7. Call SILE function and packages

8. Macro

Using `def-syntax`, you can add your macro to `sile`. the example is a custom-section macro:

```
[def-syntax subsection
```

```
[[_ x][str-append
```

```
[set! section-title [str-append [str-append [str sub-section-var] [str ". "]] [str x]]]
```

```
[font [[size "16pt"][weight "600"]] section-title]
```

```
set! sub-section-var [+ sub-section-var 1]]]]]
```

`underline bold italic font-size font-family font begin cons car cdr ls-ref ls begin script docu-para call xml-to-string`

SILE print `set! def-syntax define function, closure, lambda str str-append str-append-many`

