

## 1 Basics

LitREPL treats `lcode` environment as code sections and `lresult` environment as result sections. The names are currently hardcoded into the simplified LitREPL parser. Wrapping it in other tags is not allowed.

LaTeX does not know anything about these environments by default, we need to introduce them to be able to compile the document using e.g. `pdflatex`.

Executable section is the text between the `lcode` begin/end tags.

```
W='Hello, World!'  
print(W)
```

Putting the cursor on it and typing the `:LEval1` runs the code in the background Python interpreter.

`lresult` begin/end tags mark the result section. LitREPL replaces its content with the above code section's execution result.

```
Hello, World!
```

## 2 Producing LaTeX

LitREPL recognizes `l[no]code/l[no]result` comments as code/result section markers. This way we can use Python to produce LaTeX markup as output.

```
print("\\textbf{Hi}")
```

```
Hi
```

## 3 Inline output

Additionally, VimREPL recognises `linline` 2-argument tags. The first argument is treated as a Python printable expression. The second argument is to be replaced with the printing output. In our simplified definition, we simply ignore the first argument and paste the second to the LaTeX processor as-is.

```
Hello, World!
```