

Normal `#raw()` works as expected:

```
#show "ArtosFlow": name => box[
  #box(image(
    "logo.svg",
    height: 0.7em,
  ))
  #name
]
```

This report is embedded in the ArtosFlow project. ArtosFlow is a project of the Artos Institute.

Using `#sourcecode()` will add line numbers:

```
1  #show "ArtosFlow": name => box[
2    #box(image(
3      "logo.svg",
4      height: 0.7em,
5    ))
6    #name
7  ]
8
9  This report is embedded in the
10 ArtosFlow project. ArtosFlow is a
11 project of the Artos Institute.
```

Sourcecode can be loaded from a file and passed to `#sourcefile()`. Any **CODELST** sourcecode can be wrapped inside `#figure()` as expected.

CODELST blocks can be formatted via a `#show()` rules like:

```
#show <code>: (code) => { ... }
```

Line numbers can be formatted, too.

To the right in Listing 1	[package]	1
you can see the	name = "codelst"	2
typst.toml file of this	version = "0.0.3"	3
package.	entrypoint = "codelst.typ"	4
	authors = ["Jonas Neugebauer"]	5
	license = "MIT"	6
	description = "A typst package to render sourcecode"	7
	repository = "https://github.com/jneug/typst-codelst"	8
	exclude = ["example.typ", "example.pdf", "manual.pdf",	9
	"manual.typ", "tbump.toml"]	

Listing 1: typst.toml

CODELST does add a minimal amount of formatting. Using `#show()` rules allows you to add your own styles. For easy formatting, some default styles like a colored block can be applied using `#codelst-styles()`:

```
#show : codelst-styles
```

`#sourcecode()` accepts a number of arguments to affect the output., e.g. highlighting lines, restirct the line range or place labels in specific lines to reference them later.

```
9   #"hello world!" \
10  #"\"hello\n world\"!" \
11  #"1 2 3".split() \
12  #"1,2;3".split(regex("[,;]")) \
13  #(regex("\\d+") in "ten euros") \
14  #(regex("\\d+") in "10 euros")
```

To reference a line use `#lineref()`: See line 11 for an example of the `split()` function.