CHAPTER

SEVENTEEN

SYNTAX

Rich can syntax highlight various programming languages with line numbers.

To syntax highlight code, construct a *Syntax* object and print it to the console. Here's an example:

```
from rich.console import Console
from rich.syntax import Syntax

console = Console()
with open("syntax.py", "rt") as code_file:
    syntax = Syntax(code_file.read(), "python")
console.print(syntax)
```

You may also use the *from_path()* alternative constructor which will load the code from disk and auto-detect the file type. The example above could be re-written as follows:

```
from rich.console import Console
from rich.syntax import Syntax

console = Console()
syntax = Syntax.from_path("syntax.py")
console.print(syntax)
```

17.1 Line numbers

If you set `line_numbers=True`, Rich will render a column for line numbers:

```
syntax = Syntax.from_path("syntax.py", line_numbers=True)
```

17.2 Theme

The Syntax constructor (and *from_path()*) accept a `theme` attribute which should be the name of a Pygments theme. It may also be one of the special case theme names "ansi_dark" or "ansi_light" which will use the color theme configured by the terminal.

17.3 Background color

You can override the background color from the theme by supplying a `background_color` argument to the constructor. This should be a string in the same format a style definition accepts, e.g. "red", "#ff0000", "rgb(255,0,0)" etc. You may also set the special value "default" which will use the default background color set in the terminal.