

```
from rich.console import Console
from rich.terminal_theme import MONOKAI

console = Console(record=True)
console.save_svg("example.svg", theme=MONOKAI)
```

Alternatively, you can create a theme of your own by constructing a `rich.terminal_theme.TerminalTheme` instance yourself and passing that in.

 Note

The SVGs reference the Fira Code font. If you embed a Rich SVG in your page, you may also want to add a link to the [Fira Code CSS](#)

2.16 Error console

The `Console` object will write to `sys.stdout` by default (so that you see output in the terminal). If you construct the `Console` with `stderr=True` Rich will write to `sys.stderr`. You may want to use this to create an *error console* so you can split error messages from regular output. Here's an example:

```
from rich.console import Console
error_console = Console(stderr=True)
```

You might also want to set the `style` parameter on the `Console` to make error messages visually distinct. Here's how you might do that:

```
error_console = Console(stderr=True, style="bold red")
```

2.17 File output

You can tell the `Console` object to write to a file by setting the `file` argument on the constructor – which should be a file-like object opened for writing text. You could use this to write to a file without the output ever appearing on the terminal. Here's an example:

```
import sys
from rich.console import Console
from datetime import datetime

with open("report.txt", "wt") as report_file:
    console = Console(file=report_file)
    console.rule(f"Report Generated {datetime.now().ctime()}")
```

Note that when writing to a file you may want to explicitly set the `width` argument if you don't want to wrap the output to the current console width.

2.18 Capturing output

There may be situations where you want to *capture* the output from a `Console` rather than writing it directly to the terminal. You can do this with the `capture()` method which returns a context manager. On exit from this context manager, call `get()` to return the string that would have been written to the terminal. Here's an example: