2.6 Low level output

In additional to *print()* and *log()*, Rich has an *out()* method which provides a lower-level way of writing to the terminal. The out() method converts all the positional arguments to strings and won't pretty print, word wrap, or apply markup to the output, but can apply a basic style and will optionally do highlighting.

Here's an example:

```
>>> console.out("Locals", locals())
```

2.7 Rules

The *rule()* method will draw a horizontal line with an optional title, which is a good way of dividing your terminal output into sections.

```
>>> console.rule("[bold red]Chapter 2")
```

The rule method also accepts a `style` parameter to set the style of the line, and an `align` parameter to align the title ("left", "center", or "right").

2.8 Status

Rich can display a status message with a 'spinner' animation that won't interfere with regular console output. Run the following command for a demo of this feature:

```
python -m rich.status
```

To display a status message, call *status()* with the status message (which may be a string, Text, or other renderable). The result is a context manager which starts and stops the status display around a block of code. Here's an example:

```
with console.status("Working..."):
    do_work()
```

You can change the spinner animation via the `spinner` parameter:

```
with console.status("Monkeying around...", spinner="monkey"):
    do_work()
```

Run the following command to see the available choices for `spinner`:

```
python -m rich.spinner
```

2.9 Justify / Alignment

Both print and log support a `justify` argument which if set must be one of "default", "left", "right", "center", or "full". If "left", any text printed (or logged) will be left aligned, if "right" text will be aligned to the right of the terminal, if "center" the text will be centered, and if "full" the text will be lined up with both the left and right edges of the terminal (like printed text in a book).

The default for `justify` is `"default"` which will generally look the same as `"left"` but with a subtle difference. Left justify will pad the right of the text with spaces, while a default justify will not. You will only notice the difference if you set a background color with the `style` argument. The following example demonstrates the difference: