- "blink2" for text that flashes rapidly (not supported by most terminals).
- "conceal" for *concealed* text (not supported by most terminals).
- "italic" or "i" for italic text (not supported on Windows).
- "reverse" or "r" for text with foreground and background colors reversed.
- "strike" or "s" for text with a line through it.
- "underline" or "u" for underlined text.

Rich also supports the following styles, which are not well supported and may not display in your terminal:

- "underline2" or "uu" for doubly underlined text.
- "frame" for framed text.
- "encircle" for encircled text.
- "overline" or "o" for overlined text.

Style attributes and colors may be used in combination with each other. For example:

```
console.print("Danger, Will Robinson!", style="blink bold red underline on white")
```

Styles may be negated by prefixing the attribute with the word "not". This can be used to turn off styles if they overlap. For example:

```
console.print("foo [not bold]bar[/not bold] baz", style="bold")
```

This will print "foo" and "baz" in bold, but "bar" will be in normal text.

Styles may also have a "link" attribute, which will turn any styled text in to a *hyperlink* (if supported by your terminal software).

To add a link to a style, the definition should contain the word "link" followed by a URL. The following example will make a clickable link:

```
console.print("Google", style="link https://google.com")
```

Note

If you are familiar with HTML you may find applying links in this way a little odd, but the terminal considers a link to be another attribute just like bold, italic etc.

3.2 Style Class

Ultimately the style definition is parsed and an instance of a *Style* class is created. If you prefer, you can use the Style class in place of the style definition. Here's an example:

```
from rich.style import Style
danger_style = Style(color="red", blink=True, bold=True)
console.print("Danger, Will Robinson!", style=danger_style)
```

It is slightly quicker to construct a Style class like this, since a style definition takes a little time to parse – but only on the first call, as Rich will cache parsed style definitions.

Styles may be combined by adding them together, which is useful if you want to modify attributes of an existing style. Here's an example: