

## RICH TEXT

Rich has a `Text` class you can use to mark up strings with color and style attributes. You can use a `Text` instance anywhere a string is accepted, which gives you a lot of control over presentation.

You can consider this class to be like a string with marked up regions of text. Unlike a built-in `str`, a `Text` instance is mutable, and most methods operate in-place rather than returning a new instance.

One way to add a style to `Text` is the `stylize()` method which applies a style to a start and end offset. Here is an example:

```
from rich.console import Console
from rich.text import Text

console = Console()
text = Text("Hello, World!")
text.stylize("bold magenta", 0, 6)
console.print(text)
```

This will print “Hello, World!” to the terminal, with the first word in bold magenta.

Alternatively, you can construct styled text by calling `append()` to add a string and style to the end of the `Text`. Here’s an example:

```
text = Text()
text.append("Hello", style="bold magenta")
text.append(" World!")
console.print(text)
```

If you would like to use text that is already formatted with ANSI codes, call `from_ansi()` to convert it to a `Text` object:

```
text = Text.from_ansi("\033[1;35mHello\033[0m, World!")
console.print(text.spans)
```

Since building `Text` instances from parts is a common requirement, Rich offers `assemble()` which will combine strings or pairs of string and `Style`, and return a `Text` instance. The following example is equivalent to the ANSI example above:

```
text = Text.assemble(("Hello", "bold magenta"), ", ", "World!")
console.print(text)
```

You can apply a style to given words in the text with `highlight_words()` or for ultimate control call `highlight_regex()` to highlight text matching a *regular expression*.