⚠ **Warning**

Be careful when setting a color system, if you set a higher color system than your terminal supports, your text may be unreadable.

2.3 Printing

To write rich content to the terminal use the *print()* method. Rich will convert any object to a string via its (`__str__`) method and perform some simple syntax highlighting. It will also do pretty printing of any containers, such as dicts and lists. If you print a string it will render *Console Markup*. Here are some examples:

```
console.print([1, 2, 3])
console.print("[blue underline]Looks like a link")
console.print(locals())
console.print("FOO", style="white on blue")
```

You can also use *print()* to render objects that support the *Console Protocol*, which includes Rich's built-in objects such as *Text*, *Table*, and *Syntax* – or other custom objects.

2.4 Logging

The *log()* method offers the same capabilities as print, but adds some features useful for debugging a running application. Logging writes the current time in a column to the left, and the file and line where the method was called to a column on the right. Here's an example:

```
>>> console.log("Hello, World!")
```

To help with debugging, the log() method has a `log_locals` parameter. If you set this to `True`, Rich will display a table of local variables where the method was called.

2.5 Printing JSON

The *print_json()* method will pretty print (format and style) a string containing JSON. Here's a short example:

```
console.print_json('[false, true, null, "foo"]')
```

You can also *log* json by logging a *JSON* object:

```
from rich.json import JSON
console.log(JSON('["foo", "bar"]'))
```

Because printing JSON is a common requirement, you may import `print_json` from the main namespace:

```
from rich import print_json
```

You can also pretty print JSON via the command line with the following:

```
python -m rich.json cats.json
```