

PRETTY PRINTING

In addition to syntax highlighting, Rich will format (i.e. *pretty print*) containers such as lists, dicts, and sets.

Run the following command to see an example of pretty printed output:

```
python -m rich.pretty
```

Note how the output will change to fit within the terminal width.

7.1 pprint method

The `pprint()` method offers a few more arguments you can use to tweak how objects are pretty printed. Here's how you would import it:

```
>>> from rich.pretty import pprint
>>> pprint(locals())
```

7.1.1 Indent guides

Rich can draw *indent guides* to highlight the indent level of a data structure. These can make it easier to read more deeply nested output. The `pprint` method enables indent guides by default. You can set `indent_guides=False` to disable this feature.

7.1.2 Expand all

Rich is quite conservative about expanding data structures and will try to fit as much in each line as it can. If you prefer, you can tell Rich to fully expand all data structures by setting `expand_all=True`. Here's an example:

```
>>> pprint(["eggs", "ham"], expand_all=True)
```

7.1.3 Truncating pretty output

Very long data structures can be difficult to read and you may find yourself scrolling through multiple pages in the terminal to find the data you are interested in. Rich can truncate containers and long strings to give you an overview without swamping your terminal.

If you set the `max_length` argument to an integer, Rich will truncate containers with more than the given number of elements. If data is truncated, Rich will display an ellipsis `...` and the number of elements not shown.

Here's an example:

```
>>> pprint(locals(), max_length=2)
```