

TRACEBACK

Rich can render Python tracebacks with syntax highlighting and formatting. Rich tracebacks are easier to read and show more code than standard Python tracebacks.

To see an example of a Rich traceback, running the following command:

```
python -m rich.traceback
```

9.1 Printing tracebacks

The `print_exception()` method will print a traceback for the current exception being handled. Here's an example:

```
from rich.console import Console
console = Console()

try:
    do_something()
except Exception:
    console.print_exception(show_locals=True)
```

The `show_locals=True` parameter causes Rich to display the value of local variables for each frame of the traceback.

See `exception.py` for a larger example.

9.2 Traceback Handler

Rich can be installed as the default traceback handler so that all uncaught exceptions will be rendered with highlighting. Here's how:

```
from rich.traceback import install
install(show_locals=True)
```

There are a few options to configure the traceback handler, see `install()` for details.

9.2.1 Automatic Traceback Handler

In some cases you may want to have the traceback handler installed automatically without having to worry about importing the code in your module. You can do that by modifying the `sitecustomize.py` in your virtual environment. Typically it would be located in your virtual environment path, underneath the `site-packages` folder, something like this:

```
./venv/lib/python3.9/site-packages/sitecustomize.py
```