

# Python Getting Rich CLI Tutorial

## Contents

Python Getting Rich CLI Tutorial.....	1
Rich Demo .....	1
Tutorial 1 - Square Calculator .....	1
Standard 8-bit Terminal Colors .....	3
Assignment 1: Add Riches to a Program.....	3
Assignment Submission.....	3

Time required: 60 minutes

Printing to the console in different colors is handy and quite practical, from building fancy scanning scripts to distinguishing different log message types (debug, info, or critical, etc.) in your programs.

[Rich](#) is a Python library for writing *rich* text (with color and style) to the terminal, and for displaying advanced content such as tables, markdown, and syntax highlighted code.

## Rich Demo

1. Install rich

```
# Windows
pip install rich
# Linux
pip3 install rich
```

2. Open a command prompt. Type the following command.

```
python -m rich
```

You will see some of the capabilities of rich at the command prompt.

## Tutorial 1 - Square Calculator

This tutorial will show a couple of ways to make your CLI program much more interesting.

1. Create a Python program named **square\_calculator.py**

2. Import **rich** as shown. We will use **Console** for printing text and **Panel** for the program title.

```
1  """
2      Name: square_calculator.py
3      Author: William A Loring
4      Created: 07/08/22
5      Purpose: Calculate the area and perimeter of a square
6  """
7
8  # pip install rich
9  # Import Console for console printing
10 from rich.console import Console
11 # Import Panel for title displays
12 from rich.panel import Panel
```

3. **rich.console** must be initialized as shown for **console.print**

```
14 # Initialize rich.console
15 console = Console()
```

4. Panel uses 3 parameters to print a nice border around our program title.

```
17 # TODO: Print a nice title for our program
18 console.print(
19     Panel.fit(
20         " A Square Calculator ",
21         style="bold blue",
22         subtitle="By William Loring")
23 )
24 print()
```

5. Nothing new here.

```
26 # TODO: Get side length from user, cast to float
27 side = float(input("Enter the length of a side: "))
28
29 # TODO: Calculate area
30 area = side * side
31
32 # TODO: Calculate perimeter
33 perimeter = side * 4
```

6. **console.print** uses html like opening and closing tags to set the color and unset the color.

```
console.print("[bold blue]This text is bold blue[/bold blue]")
```

```

35 # TODO: Display results
36 console.print(f"Area: [bold blue]{area}[/bold blue]")
37 console.print(f"Perimeter: [bold blue]{perimeter}[/bold blue]")

```

Example run:

```









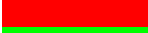



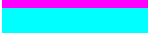

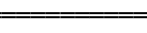

A Square Calculator
By William Loring

Enter the length of a side: 20
Area: 400.0
Perimeter: 80.0

```

## Standard 8-bit Terminal Colors

The Windows command prompt can only display 16 colors. Each color can have bold or light added in front of the name: light blue, bold blue.

Color	Number	Name
	0	"black"
	1	"red"
	2	"green"
	3	"yellow"
	4	"blue"
	5	"magenta"
	6	"cyan"
	7	"white"
	8	"bright_black"
	9	"bright_red"
	10	"bright_green"
	11	"bright_yellow"
	12	"bright_blue"
	13	"bright_magenta"
	14	"bright_cyan"
	15	"bright_white"

## Assignment 1: Add Riches to a Program

Add rich formatting to any Python CLI program that we have done in class, or one that you are working on.

---

### Assignment Submission

1. Attach the pseudocode.

2. Attach the program files.
3. Attach screenshots showing the successful operation of the program.
4. Submit in Blackboard.