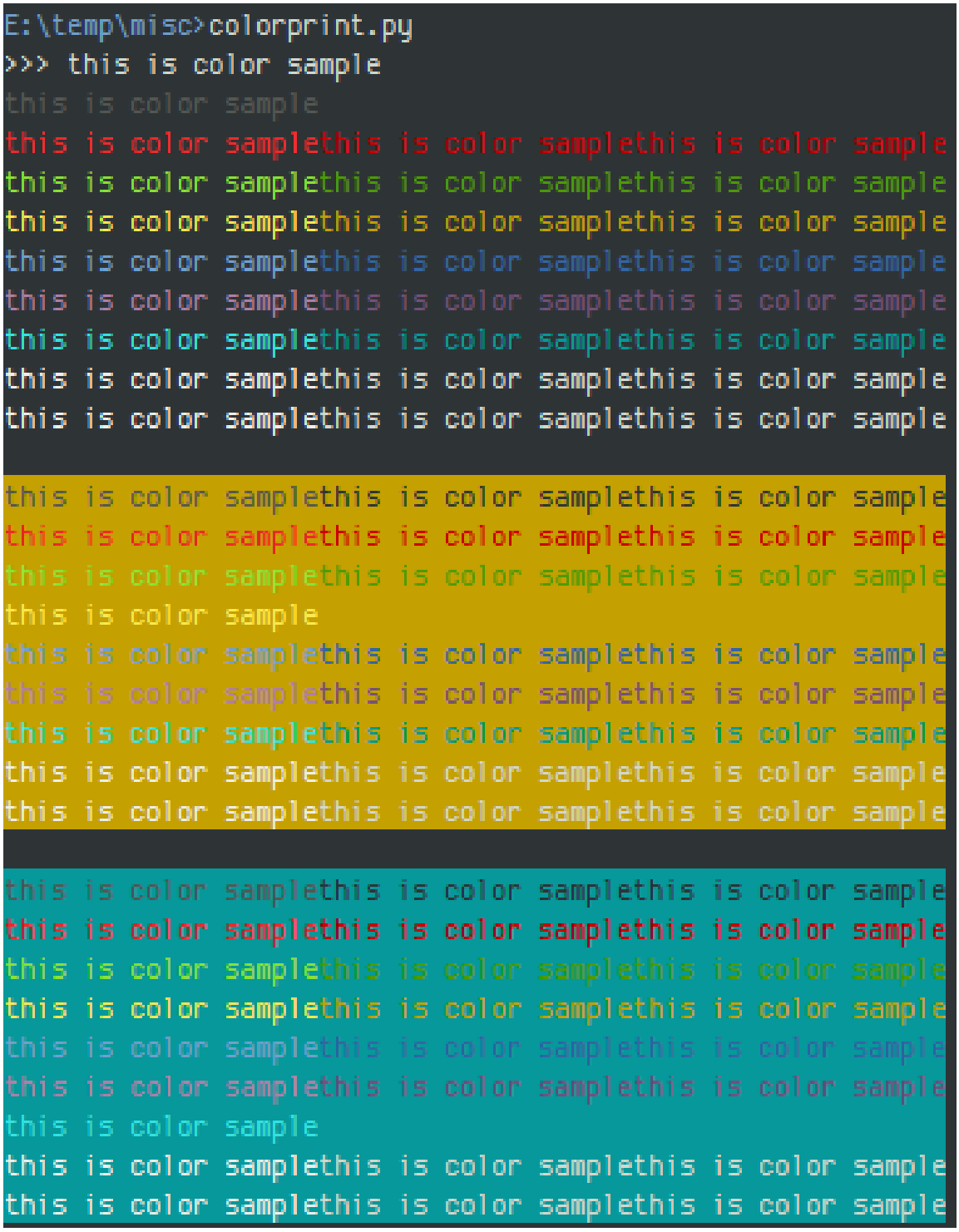
G.H Raisoni College of Engineering and Management Wagholi, Pune Department of First Year, B.Tech **Project - Print Coloured Text**	
Name : Swayam Terode	**Division : C**
Roll Number : C70	**Subject : Programming for Problem Solving (UITP102)**
Registration Number : AIFT20201101047	

The Below Code Prints Coloured Text

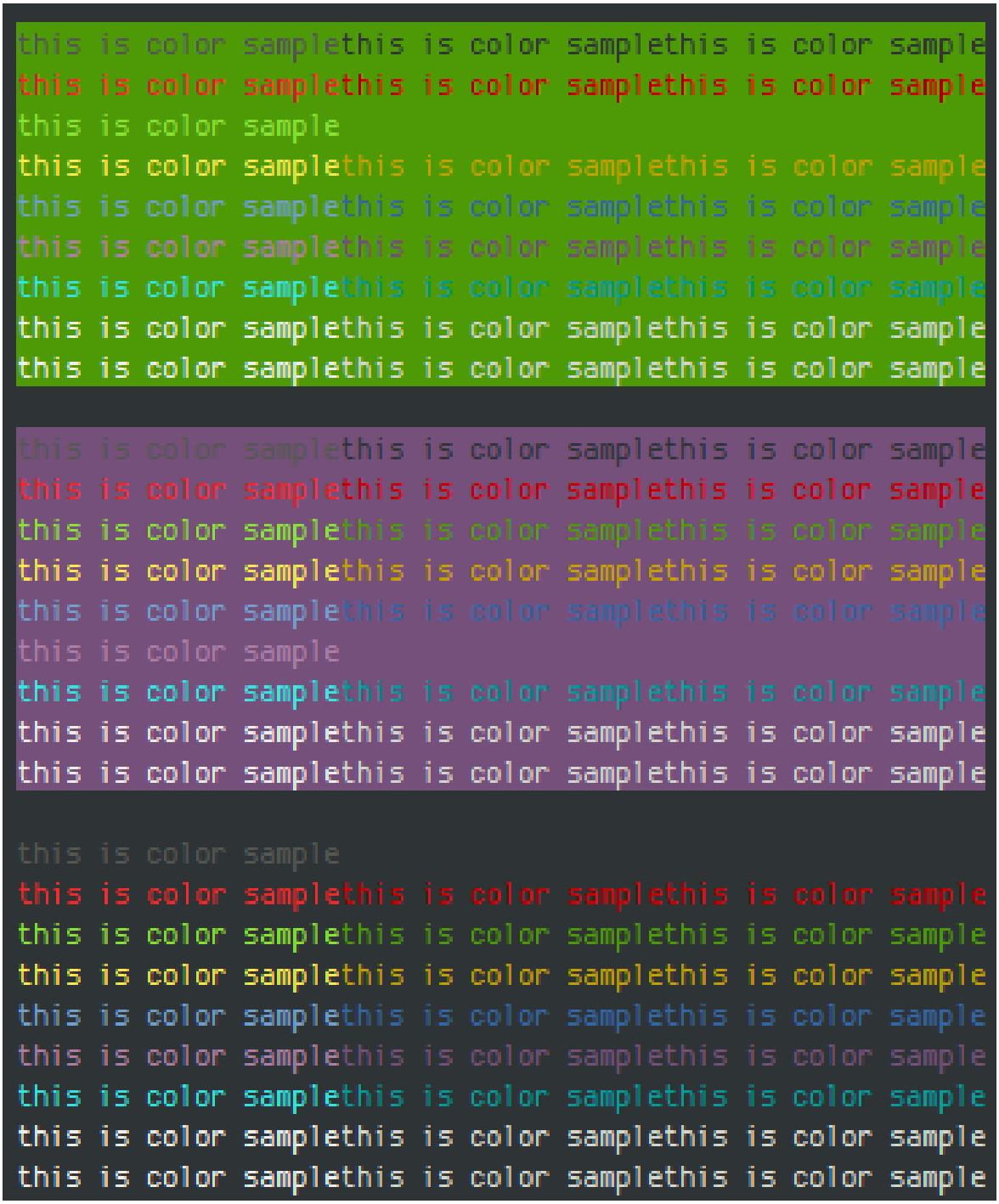
```
COLORS =
dict(zip(range(1,
10), 'black red
green yellow
blue magenta'
                    ' cyan white reset'.split()))
class BG:
    black = '\033[40m'
    red = '\033[41m'
    green = '\033[42m'
    yellow = '\033[43m'
    blue = '\033[44m'
    magenta = '\033[45m'
    cyan = '\033[46m'
    white = '\033[47m'
    reset = '\033[49m'
class FG:
    black = '\033[30m'
    red = '\033[31m'
    green = '\033[32m'
    yellow = '\033[33m'
    blue = '\033[34m'
    magenta = '\033[35m'
    cyan = '\033[36m'
    white = '\033[37m'
```

```
    reset = '\033[39m'
class BRIGHT:
    bright = '\033[1m'
    dim = '\033[2m'
    normal = '\033[22m'
class Color:
    resetall = '\033[0m'
    @classmethod
    def colored(self, msg, foreground='white',
background='black', bright=0):
        xBRIGHT, xBG, xFG = '', '', ''
        if bright is 1:
            xBRIGHT = BRIGHT.bright
        elif bright is 2:
            xBRIGHT = BRIGHT.dim
        else:
            xBRIGHT = BRIGHT.normal
        if hasattr(BG, background):
            xBG = getattr(BG, background)
        if hasattr(FG, foreground):
            xFG = getattr(FG, foreground)
        return '{}{}{}{}{}'.format(
              xBRIGHT, xBG, xFG, msg, Color.resetall)
if __name__ == '__main__':
    msg = input('>>> ')
    colors = list(COLORS.values())
    for i in range(0, len(colors), 3):
        for fg in colors:
            line = ' '.join([''.join(Color.colored(msg, fg, colors[i+x],
y)
                             for y in (1, 2, 0))
                            for x in range(3)])
            print(line)
        print()
```

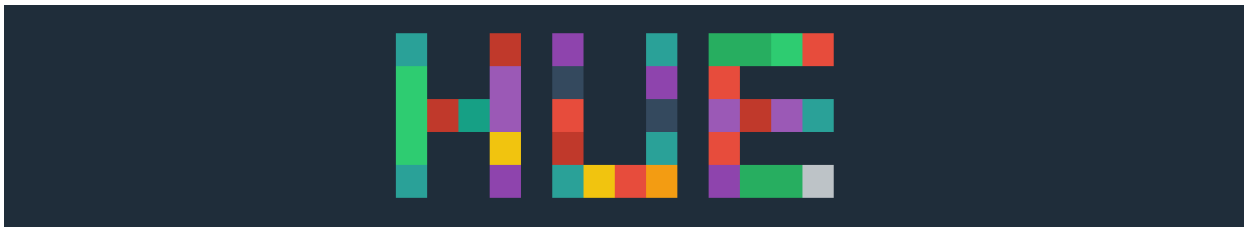
The below picture are the screenshots of the above code




this is color samplethis is color samplethis is color sample
this is color sample
this is color samplethis is color samplethis is color sample
this is color samplethis is color samplethis is color sample
this is color samplethis is color samplethis is color sample
this is color samplethis is color samplethis is color sample
this is color samplethis is color samplethis is color sample
this is color samplethis is color samplethis is color sample
this is color samplethis is color samplethis is color sample
this is color samplethis is color samplethis is color sample
this is color samplethis is color samplethis is color sample
this is color samplethis is color samplethis is color sample
this is color sample
this is color samplethis is color samplethis is color sample
this is color samplethis is color samplethis is color sample
this is color samplethis is color samplethis is color sample
this is color samplethis is color samplethis is color sample
this is color samplethis is color samplethis is color sample
this is color samplethis is color samplethis is color sample
this is color samplethis is color samplethis is color sample
this is color samplethis is color samplethis is color sample
this is color samplethis is color samplethis is color sample
this is color sample
this is color sample



This is a simple code this can also be done with the help of HUE which is explained below:

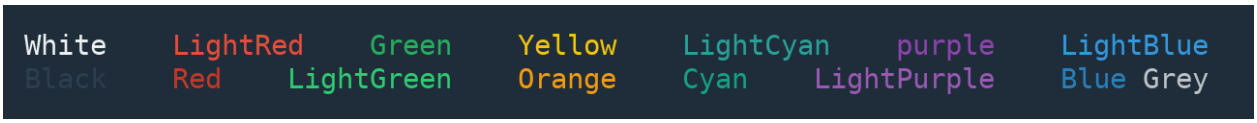


Hue provides a minimal and powerful interface to print colored text and labels in the terminal.It works with Python 2 as well as Python 3.

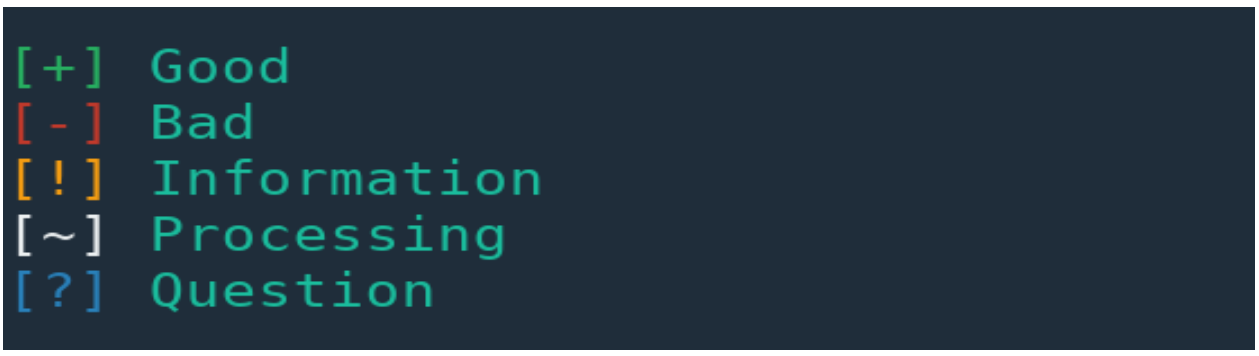
Supported Stuff

Following styles are supported

Following colors are supported



Following labels are supported



Installation

You can install hue with pip as follows:

pip install huepy

or with easy_install:

easy_install huepy

Usage

First of all, import everything that Hue has to offer as follows:

from huepy import *

Printing colored text is as simple as doing
print(red('This string is red'))
Easy right? But what if you want to print italic text? You can simply do this
print(italic('This string is in italic'))
You can also combine styles and colors
print(bold(red('This string is bold and red')))

Output:

```
>>> print bold(red('This string is bold and red'))
This string is bold and red
```

And what is the use of those labels?

I have been using these labels in projects as a minimal output schema.
If some error occured in your program or something else bad happened you don't need to print the whole line in red. With hue, you can simply do this

print(bad('An error occured.'))

Take a look at the output of all the
labels

```
>>> from huepy import *
>>> print bad('An error occured.')
[-] An error occured.
>>> print good('Attempt Successful.')
[+] Attempt Successful.
>>> print info('There are 69 files in total.')
[!] There are 69 files in total.
>>> print run('Scanning the target...')
[~] Scanning the target...
>>> print que('Are you sure about it?')
[?] Are you sure about it?
```

List of all colors

white, grey, black, green, lightgreen, cyan, lightcyan, red, lightred,
blue, lightblue, purple, light purple, orange, yellow

List of all styles

bold, bg, under, strike, italic

List of all labels

info, que, run, bad, good
Note: Windows versions below windows 10 do not support ANSI escape sequences so the colors will not be printed in command prompt.

Why hue

Because its awesome! Lets print a red colored string in popular coloring libraries:

Colorama

```
from colorama import Fore
print(Fore.RED + 'This string is red')
```

Termcolor

```
import sys
from termcolor import colored, cprint
print(colored('This string is red', 'red'))
```

Hue

```
from hue import *
print(red('This string is red'))
```

Here's comparison table:

	Hue	Colorama	Termcolor
Compatibility	Unix & Windows 10	Unix & Windows	Unix
Ease of use	10/10	4/10	5/10
Bright Colors	Yes	No	No

Note: Colorama and Termcolor print bold styled strings when asked for bright colored strings. On the other hand, Hue supports both bright and bold strings. Also the *Ease to use* ratings are a result of my own experience and may differ for others.