

Forum:[Hack Forums](#)

Thread: [colorama module | The Beautiful Colorful Module | Hydra's Tutorial](#)

Hydra_ - June 24, 2013, 9:48 pm

[img]http://s4.postimage.org/5ftaj2zrb/Hello_HF_2_copy.png[/img]

Colorama is a pretty cool module for making colorful text in your console application, imagine your console with colors, it would be GREAT ? for example check the pictures of some of my programs with those colors

Console Applications ([Click to View](#))

[img]http://s17.postimg.org/iezj4bt73/ALL_THE_THING.png[/img]

[img]http://s9.postimage.org/4vxc7i3u7/image.png[/img]

[img]http://s16.postimage.org/ecs5nrug5/pass_3.png[/img]

And more ...

Many people like my programs console with those colors, like Rhyno and Catalyst .. and I appreciate that ?

So, How to do such beautiful console?

Download colorama module

First of all we have to download the module. You can get the module from the following link:

[Download](#)

Get started

Colorama let u control of the Foreground, Background and the Style of the text, but to start controlling all of these we have to import and initiate it:

```
Code from colorama import *  
init()
```

As I told u before there are three basic controls for the text, the color, the background and the style. To manage the colors of text we use the following:

Code Fore.RED This is to make the foreground text RED.

Code Back.RED This is to make the background text RED.

Colorama supports the following colors

RED

WHITE

BLACK

GREEN

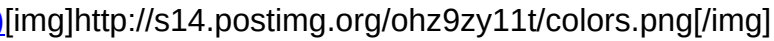
YELLOW

BLUE

MAGENTA

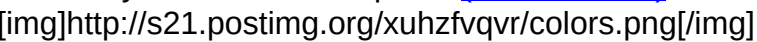
CYAN

So let's see an example, the following example will print python in RED, then it would print it in WHITE and the background in RED.

Example#1 [\(Click to View\)](#)

But these colors are some kinda dark, let's lighten them and make them BRIGHT and to do that we use:

Code `Style.BRIGHT`Example#2 [\(Click to View\)](#)

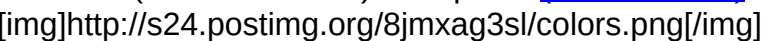


As we noticed the 'python' word are bright now in the two lines. ;)

But what if we want to brighten just the line which `Style.BRIGHT` would be written in?!

Okay, at the of the script we initiate colorama by typing `init()` and notice it took no parameters, but to brighten one line we would set a parameter in `init()` called `autoreset` and we would set `True` as a value for it.

Code `init(autoreset=True)`Example#3 [\(Click to View\)](#)



Can u see the difference now, the first line is bright and the second one is dark, so if you initiated the script with the `autoreset` as `True`, and you want some lines in BRIGHT, you would have to write `Style.BRIGHT` at the first of each of those lines.

Now if we want to reset the foreground color we use:

Code `Fore.RESET`And if we want to reset the background color we use:

Code `Back.RESET`And to reset foreground, background and style we use:

Code `Style.RESET_ALL`

By this line you have just finished the basic stuff of colorama module to make wonderful colorful console applications, go and try it and start making your own colorful apps. For more info you can check [the official page of colorama module](#).

If you need any help with colorama or anything in python, Feel free to PM me.

Thanks for reading.

```
||
||
||
\_/
\_/
V
```

Please don't suggest people use `from foo import *`; it pollutes the global namespace and can break things without you knowing why really easily.

It's fine to support select functions/modules/classes/whatever (`from foo import bar, baz, qux`), but `import *` is just tacky.

`__init__` - June 25, 2013, 6:40 am

(Jun 25, 2013 03:53 AM)plausibility Wrote: [\[CITE\]](#) Please don't suggest people use `from foo import *`; it pollutes the global namespace and can break things without you knowing why really easily.

It's fine to support select functions/modules/classes/whatever (`from foo import bar, baz, qux`), but `import *` is just tacky.

I agree with this, unless the classes are prefixed with something recognizable for example all PyQt classes start with the letter 'Q'. Also it's much easier to type

```
Code app = QApplication(sys.argv)
than
```

```
Code app = PyQt4.QtCore.QApplication(sys.argv)
```

Hydra_ - June 25, 2013, 7:40 pm

Oh yeah, that's a good advice which I completely missed! ;)

PigBacon - June 26, 2013, 12:36 am

(Jun 25, 2013 06:40 AM)`__init__` Wrote: [\[CITE\]](#) I agree with this, unless the classes are prefixed with something recognizable for example all PyQt classes start with the letter 'Q'. Also it's much easier to type

```
Code app = QApplication(sys.argv)
than
```

```
Code app = PyQt4.QtCore.QApplication(sys.argv)
```

Definitely. Under given circumstances, it's appropriate to do that (albeit it's good if the module sets `__all__`, so not to completely trash your globals).

In this instance, I wouldn't say it's a good idea.

`init()`? `Style`? `Fore`? These are all names that aren't uncommon; `init()` especially.