ansicolor Documentation

Release 0.2.4

Martin Matusiak

Jun 03, 2017

Contents

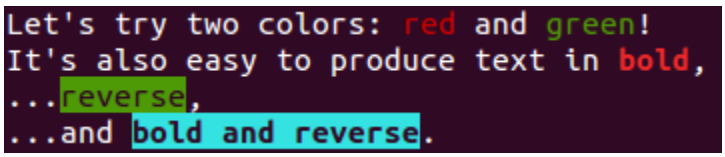
CHAPTER 1

User guide

Getting started with colors

To highlight using colors:

```
from ansicolor import cyan
from ansicolor import green
from ansicolor import red
from ansicolor import white

print("Let's try two colors: %s and %s!" % (red("red"), green("green")))
print("It's also easy to produce text in %s," % (red("bold", bold=True)))
print("...%s," % (green("reverse", reverse=True)))
print("...and %s." % (cyan("bold and reverse", bold=True, reverse=True)))
```

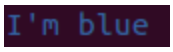


This will emit ansi escapes into the string: one when starting a color, another to reset the color back to the default:

```
>>> from ansicolor import green

>>> green("green")
'\x1b[0;0;32mgreen\x1b[0;0m'
```

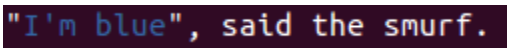
If I want to be able to pass a color as an argument I can also use the `colorize` function:

```
from ansicolor import Colors
from ansicolor import colorize

print(colorize("I'm blue", Colors.Blue))
```



I can also apply color on a portion of a string:

```
from ansicolor import Colors
from ansicolor import colorize

print(colorize('''"I'm blue", said the smurf.''', Colors.Blue, start=1, end=9))
```



Using highlights

Quite often colors are used not to format all of the text in color, but to highlight certain parts of it. The function *ansicolor.highlight_string* takes this a bit further by allowing you to pass in a list of pairs that represent the start and end offsets in the string that you want highlighted.

```
import re

from ansicolor import highlight_string

text = """
"What giants?" asked Sancho Panza.
"The ones you can see over there," answered his master, "with the huge arms, some of␣
↪which are very nearly two leagues long."
"Now look, your grace," said Sancho, "what you see over there aren't giants, but␣
↪windmills, and what seems to be arms are just their sails, that go around in the␣
↪wind and turn the millstone."
"Obviously," replied Don Quijote, "you don't know much about adventures."
""".strip()

def get_line_indices(text):
    odds, evens = [], []
    for i, match in enumerate(re.finditer('(?m)^.*$', text)):
        start = match.start()
        end = match.end()
        if (i + 1) % 2 == 1:
            odds.append((start, end))
        else:
            evens.append((start, end))
    return odds, evens

def get_word_indices(regex, text):
    pairs = []
    for i, match in enumerate(re.finditer(regex, text)):
        start = match.start()
        end = match.end()
        pairs.append((start, end))
    return pairs

odds, evens = get_line_indices(text)
characters = get_word_indices('(?i)(don quijote|master|sancho panza|sancho)', text)

print(">> highlight only odds:")
```

```
print(highlight_string(text, odds))

print("\n>> highlight both:")
print(highlight_string(text, odds, evens))

print("\n>> highlight both + characters:")
print(highlight_string(text, odds, evens, characters))
```

```
>> highlight only odds:
"What giants?" asked Sancho Panza.
"The ones you can see over there," answered his master, "with the huge arms, som
e of which are very nearly two leagues long."
"Now look, your grace," said Sancho, "what you see over there aren't giants, but
 windmills, and what seems to be arms are just their sails, that go around in th
e wind and turn the millstone."
"Obviously," replied Don Quijote, "you don't know much about adventures."

>> highlight both:
"What giants?" asked Sancho Panza.
"The ones you can see over there," answered his master, "with the huge arms, som
e of which are very nearly two leagues long."
"Now look, your grace," said Sancho, "what you see over there aren't giants, but
 windmills, and what seems to be arms are just their sails, that go around in th
e wind and turn the millstone."
"Obviously," replied Don Quijote, "you don't know much about adventures."

>> highlight both + characters:
"What giants?" asked Sancho Panza.
"The ones you can see over there," answered his master, "with the huge arms, som
e of which are very nearly two leagues long."
"Now look, your grace," said Sancho, "what you see over there aren't giants, but
 windmills, and what seems to be arms are just their sails, that go around in th
e wind and turn the millstone."
"Obviously," replied Don Quijote, "you don't know much about adventures."
```

Every list of pairs that is passed in is considered a highlighting layer and gets a new color. Where layers overlap this is indicated by applying:

- bold (two layers overlap),
- reverse (three layers overlap),
- bold and reverse (four layers overlap).

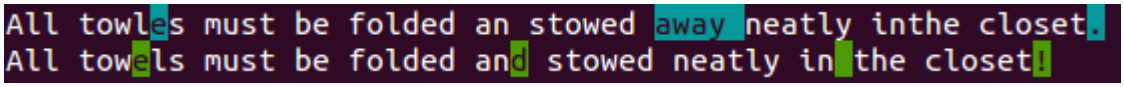
Four layers is the maximum, because after that there is no further distinction possible. See *ansicolor.demos.demo_highlight* for an exhaustive example.

Colored diffs

One practical use of colors is to make differences in text more visible. *ansicolor.colordiff* computes a diff of two strings and returns a marked-up version of them that highlights the characters that differ between the two.

```
from ansicolor import Colors
from ansicolor import colordiff
```

```
statement = "All towles must be folded an stowed away neatly inthe closet."
reviewed = "All towels must be folded and stowed neatly in the closet!"

first, second = colordiff(statement, reviewed,
                          color_x=Colors.Cyan, color_y=Colors.Green)

print(first)
print(second)
```



Working with marked-up strings

Stripping markup

Sometimes I may have a string that contains markup and I'll want to do something with it that concerns only the text, so I can strip the markup:

```
>>> from ansicolor import red
>>> from ansicolor import strip_escapes
>>> from ansicolor import yellow

>>> message = "My favorite colors are %s and %s" % (yellow("yellow"), red("red"))
>>> print("The length of this text is not: %d" % len(message))
The length of this text is not: 67
>>> print("The length of this text is: %d" % len(strip_escapes(message)))
The length of this text is: 37
```

Producing output

Printing marked-up strings directly is not appropriate for all use cases. When writing to a file it's generally not desirable to print ansi escapes meant for a terminal. The two functions *ansicolor.write_out* and *ansicolor.write_err* omit ansi escapes if the file being written to is not a tty.

```
from ansicolor import red
from ansicolor import write_out

write_out(red("This looks red in a terminal.\n"))
```

CHAPTER 2

API documentation

ansicolor package

Submodules

ansicolor.ansicolor module

ansicolor.ansicolor.**black**(*s*, *bold=False*, *reverse=False*)
Colorize string in black

Parameters

- **s** (*string*) – The string to colorize.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

ansicolor.ansicolor.**blue**(*s*, *bold=False*, *reverse=False*)
Colorize string in blue

Parameters

- **s** (*string*) – The string to colorize.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

ansicolor.ansicolor.**cyan**(*s*, *bold=False*, *reverse=False*)
Colorize string in cyan

Parameters

- **s** (*string*) – The string to colorize.

- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

ansicolor.ansicolor.**green**(*s*, *bold=False*, *reverse=False*)
Colorize string in green

Parameters

- **s** (*string*) – The string to colorize.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

ansicolor.ansicolor.**magenta**(*s*, *bold=False*, *reverse=False*)
Colorize string in magenta

Parameters

- **s** (*string*) – The string to colorize.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

ansicolor.ansicolor.**red**(*s*, *bold=False*, *reverse=False*)
Colorize string in red

Parameters

- **s** (*string*) – The string to colorize.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

ansicolor.ansicolor.**white**(*s*, *bold=False*, *reverse=False*)
Colorize string in white

Parameters

- **s** (*string*) – The string to colorize.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

ansicolor.ansicolor.**yellow**(*s*, *bold=False*, *reverse=False*)
Colorize string in yellow

Parameters

- **s** (*string*) – The string to colorize.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

ansicolor.ansicolor.**colorize**(*s*, *color*, *bold=False*, *reverse=False*, *start=None*, *end=None*)
Colorize a string with the color given.

Parameters

- **s** (*string*) – The string to colorize.
- **color** (*Colors* class) – The color to use.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.
- **start** (*int*) – Index at which to start coloring.
- **end** (*int*) – Index at which to end coloring.

Return type string

ansicolor.ansicolor.**wrap_string**(*s*, *pos*, *color*, *bold=False*, *reverse=False*)
Colorize the string up to a position.

Parameters

- **s** (*string*) – The string to colorize.
- **pos** (*int*) – The position at which to stop.
- **color** (*Colors* class) – The color to use.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

Deprecated since version 0.2.2: This function has been deprecated in favor of *colorize()*.

ansicolor.ansicolor.**get_code**(*color*, *bold=False*, *reverse=False*)
Returns the escape code for styling with the given color, in bold and/or reverse.

Parameters

- **color** (*Colors* class) – The color to use.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

ansicolor.ansicolor.**highlight_string**(*s*, **spanlists*, ***kw*)
Highlight spans in a string using a list of (begin, end) pairs. Each list is treated as a layer of highlighting. Up to four layers of highlighting are supported.

Parameters

- **s** (*string*) – The string to highlight
- **spanlists** (*list*) – A list of tuples on the form `[(begin, end)*]*`
- **kw** – May include: *bold*, *reverse*, *color*, *colors* and *nocolor*

Return type string

Deprecated since version 0.2.3: The *color* parameter has been deprecated in favor of *colors*.

ansicolor.ansicolor.**get_highlighter**(*colorid*)
Map a color index to a highlighting color.

Parameters **coloriid** (*int*) – The index.

Return type *Colors*

ansicolor.ansicolor.**strip_escapes**(*s*)
Strips escapes from the string.

Parameters **s** (*string*) – The string.

Return type string

ansicolor.ansicolor.**justify_formatted**(*s*, *justify_func*, *width*)
Justify a formatted string to a width using a function (eg. `string.ljust`).

Parameters

- **s** (*string*) – The formatted string.
- **justify_func** – The justify function.
- **width** (*int*) – The width at which to justify.

Return type string

ansicolor.ansicolor.**colordiff**(*x*, *y*, *color_x=<class 'ansicolor.ansicolor.Cyan'>*, *color_y=<class 'ansicolor.ansicolor.Green'>*, *debug=False*)
Formats a diff of two strings using the longest common subsequence by highlighting characters that differ between the strings.

Returns the strings *x* and *y* with highlighting.

Parameters

- **x** (*string*) – The first string.
- **y** (*string*) – The second string.
- **color_x** (*Colors* class) – The color to use for the first string.
- **color_y** (*Colors* class) – The color to use for the second string.
- **debug** (*bool*) – Whether to print debug output underway.

Return type tuple

ansicolor.ansicolor.**set_term_title**(*s*)
Set the title of a terminal window.

Parameters **s** (*string*) – The title.

ansicolor.ansicolor.**write_out**(*s*)
Write a string to `sys.stdout`, strip escapes if output is a pipe.

Parameters **s** (*string*) – The title.

ansicolor.ansicolor.**write_err**(*s*)
Write a string to `sys.stderr`, strip escapes if output is a pipe.

Parameters **s** (*string*) – The title.

class ansicolor.ansicolor.**Colors**
Bases: `object`

Container class for colors

class **Black**
Bases: `object`

id = 0

class Colors.**Blue**
Bases: object

id = 4

class Colors.**Cyan**
Bases: object

id = 6

class Colors.**Green**
Bases: object

id = 2

class Colors.**Magenta**
Bases: object

id = 5

class Colors.**Red**
Bases: object

id = 1

class Colors.**White**
Bases: object

id = 7

class Colors.**Yellow**
Bases: object

id = 3

classmethod Colors.**iter**()

classmethod Colors.**new**(*colorname*)

ansicolor.demos module

ansicolor.demos.**demo_color**()

ansicolor.demos.**demo_diff**()

ansicolor.demos.**demo_highlight**()

ansicolor.demos.**demo_highlight_reverse**()

Module contents

ansicolor.**black**(*s*, *bold=False*, *reverse=False*)
Colorize string in black

Parameters

- **s** (*string*) – The string to colorize.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

ansicolor.**blue**(*s*, *bold=False*, *reverse=False*)
Colorize string in blue

Parameters

- **s** (*string*) – The string to colorize.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

ansicolor.**cyan**(*s*, *bold=False*, *reverse=False*)
Colorize string in cyan

Parameters

- **s** (*string*) – The string to colorize.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

ansicolor.**green**(*s*, *bold=False*, *reverse=False*)
Colorize string in green

Parameters

- **s** (*string*) – The string to colorize.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

ansicolor.**magenta**(*s*, *bold=False*, *reverse=False*)
Colorize string in magenta

Parameters

- **s** (*string*) – The string to colorize.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

ansicolor.**red**(*s*, *bold=False*, *reverse=False*)
Colorize string in red

Parameters

- **s** (*string*) – The string to colorize.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

ansicolor.**white**(*s*, *bold=False*, *reverse=False*)
Colorize string in white

Parameters

- **s** (*string*) – The string to colorize.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

ansicolor.**yellow**(*s*, *bold=False*, *reverse=False*)
Colorize string in yellow

Parameters

- **s** (*string*) – The string to colorize.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

ansicolor.**colorize**(*s*, *color*, *bold=False*, *reverse=False*, *start=None*, *end=None*)
Colorize a string with the color given.

Parameters

- **s** (*string*) – The string to colorize.
- **color** (*Colors* class) – The color to use.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.
- **start** (*int*) – Index at which to start coloring.
- **end** (*int*) – Index at which to end coloring.

Return type string

ansicolor.**wrap_string**(*s*, *pos*, *color*, *bold=False*, *reverse=False*)
Colorize the string up to a position.

Parameters

- **s** (*string*) – The string to colorize.
- **pos** (*int*) – The position at which to stop.
- **color** (*Colors* class) – The color to use.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

Deprecated since version 0.2.2: This function has been deprecated in favor of *colorize()*.

ansicolor.**get_code**(*color*, *bold=False*, *reverse=False*)
Returns the escape code for styling with the given color, in bold and/or reverse.

Parameters

- **color** (*Colors* class) – The color to use.
- **bold** (*bool*) – Whether to mark up in bold.
- **reverse** (*bool*) – Whether to mark up in reverse video.

Return type string

ansicolor.**highlight_string**(*s*, **spanlists*, ***kw*)
Highlight spans in a string using a list of (begin, end) pairs. Each list is treated as a layer of highlighting. Up to four layers of highlighting are supported.

Parameters

- **s** (*string*) – The string to highlight
- **spanlists** (*list*) – A list of tuples on the form `[(begin, end)*]*`
- **kw** – May include: *bold*, *reverse*, *color*, *colors* and *nocolor*

Return type string

Deprecated since version 0.2.3: The *color* parameter has been deprecated in favor of *colors*.

ansicolor.**get_highlighter**(*colorid*)
Map a color index to a highlighting color.

Parameters **colorid** (*int*) – The index.

Return type *Colors*

ansicolor.**strip_escapes**(*s*)
Strips escapes from the string.

Parameters **s** (*string*) – The string.

Return type string

ansicolor.**justify_formatted**(*s*, *justify_func*, *width*)
Justify a formatted string to a width using a function (eg. `string.ljust`).

Parameters

- **s** (*string*) – The formatted string.
- **justify_func** – The justify function.
- **width** (*int*) – The width at which to justify.

Return type string

ansicolor.**colordiff**(*x*, *y*, *color_x=<class 'ansicolor.ansicolor.Cyan'>*, *color_y=<class 'ansicolor.ansicolor.Green'>*, *debug=False*)
Formats a diff of two strings using the longest common subsequence by highlighting characters that differ between the strings.

Returns the strings *x* and *y* with highlighting.

Parameters

- **x** (*string*) – The first string.
- **y** (*string*) – The second string.
- **color_x** (*Colors* class) – The color to use for the first string.
- **color_y** (*Colors* class) – The color to use for the second string.
- **debug** (*bool*) – Whether to print debug output underway.

Return type tuple

ansicolor.**set_term_title**(*s*)
Set the title of a terminal window.

Parameters **s** (*string*) – The title.

ansicolor.**write_out**(*s*)
Write a string to sys.stdout, strip escapes if output is a pipe.

Parameters **s** (*string*) – The title.

ansicolor.**write_err**(*s*)
Write a string to sys.stderr, strip escapes if output is a pipe.

Parameters **s** (*string*) – The title.

class ansicolor.**Colors**
Bases: object

Container class for colors

class **Black**
Bases: object

id = 0

class Colors.**Blue**
Bases: object

id = 4

class Colors.**Cyan**
Bases: object

id = 6

class Colors.**Green**
Bases: object

id = 2

class Colors.**Magenta**
Bases: object

id = 5

class Colors.**Red**
Bases: object

id = 1

class Colors.**White**
Bases: object

id = 7

class Colors.**Yellow**
Bases: object

id = 3

classmethod Colors.**iter**()

classmethod Colors.**new**(*colorname*)

CHAPTER 3

Project documentation

Release notes

0.2.4

- First version supporting Python 3.4!

0.2.3

- *ansicolor.highlight_string()* accepts a new kwarg *colors*. *color* has been deprecated.

0.2.2

- *ansicolor.colorize()* now accepts `start` and `end` kwargs. *ansicolor.wrap_string()* has been deprecated.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index

M

N

R

S

W

Y