
colorterm Documentation

Release 0.3

Aurélien Matouillot

February 12, 2014

colorterm is a package to write some formatted message in your terminal. It supports 16 colors. You can set:

- formatting like underline, bold, ...
- color
- background color

Supported format

You can visit [this page](#) to see the rendering of each formats.

1.1 formatting

- bold
- dim
- underline
- blink
- reverse
- hidden

1.2 color

The same colors are supported for the foreground like the background

- default
- black
- red
- green
- yellow
- blue
- magenta
- cyan
- light_gray
- dark_gray
- light_red
- light_green
- light_yellow

- light_blue
- light_magenta
- light_cyan
- white

Usage

2.1 Predefined functions

```
from colorterm import colorterm
print colorterm.underline_red_on_yellow('Hello world')
```

You can combine all the formatting and the colors. See examples:

```
from colorterm import colorterm
print colorterm.underline('Hello world')
print colorterm.red('Hello world')
print colorterm.on_yellow('Hello world')
print colorterm.red_on_yellow('Hello world')
print colorterm.underline_red_on_yellow('Hello world')
```

2.2 Formatter

You can use the formatter to make some custom renderings:

```
from colorterm import formatter
print formatter('{red}Hello{/red} {underline}world')
```

You can use the formatting and colors like you want.

Note: You can't use for example {red_on_yellow} in the formatter, you should use them separated like in the following example

```
from colorterm import formatter
print formatter('{red}{on_yellow}Hello{/red} {underline}world')
```

Output redirection

colorterm supports correctly the output redirection to another program. Example with the a python file named test.py which contains:

```
from colorterm import colorterm
print colorterm.underline('Hello world')
```

```
$ python test.py
```

Now redirect the output to a less:

```
$ python test.py | less
```

Table formatting

colorterm support to display table as output:

```
from colorterm import Table
table = Table('ID', 'Name')
rows = [('id1', 'name1'), ('id2', 'name2')]
for ident, name in rows:
    table.add_row({
        'ID': ident,
        'Name': name,
    })
print table.display()
```

It will display the following table in your shell:

4.1 Table options

column_separator

Default: ' '. The column separator, you can put '|' for example.

header_convert

Default: colorterm.underline. A function to apply on the display of the header.

Example:

```
from colorterm import Table, colorterm
table = Table('ID', 'Name', column_separator = ' | ', header_convert=colorterm.red_underline)
rows = [('id1', 'name1'), ('id2', 'name2')]
for ident, name in rows:
    table.add_row({
        'ID': ident,
        'Name': name,
    })
print table.display()
```

Output:

4.2 Column options

`convert`

Default: None. A function to apply formatting on the cells from this column

`align`

Default: 'left'. Where to display the text of the cells from this column. One of 'left' or 'right'.

Example:

```
from colorterm import Table, colorterm
table = Table('ID',
              {'name': 'Name',
               'convert': colorterm.red,
               'align': 'right'})
rows = [('id1', 'name1'), ('id2', 'name2')]
for ident, name in rows:
    table.add_row({
        'ID': ident,
        'Name': name,
    })
print table.display()
```

Output:

4.3 Cell options

`convert`

Default: None. A function to apply formatting to the cell

`align`

Default: 'left'. Where to display the text. One of 'left', 'right'.

Example:

```
from colorterm import Table, colorterm
table = Table('ID', 'Long name')
rows = [
    ('id1', {
        'value': 'name1',
        'convert': colorterm.red,
        'align': 'right'}),
    ('id2', 'name2')]
for ident, name in rows:
    table.add_row({
        'ID': ident,
        'Long name': name,
    })
print table.display()
```

Output:

4.4 Row options

`convert`

Default: None. A function to apply formatting to the row

Example:

```
from colorterm import Table, colorterm
table = Table('ID', 'Name')
rows = [
    ('id1', 'name1'),
    ('id2', 'name2')]
for ident, name in rows:
    table.add_row({
        'ID': ident,
        'Name': name,
    }, convert=colorterm.red)
print table.display()
```

Output: