

```
#####
# Module for basic terminal color and cursor manipulations via an easy to use API.
# Under the hood ANSI escape sequences are printed to stdout of the terminal.
#
# Some details about ANSI escape sequences can be found here:
# - https://de.wikipedia.org/wiki/ANSI-Escapesequenz
# - https://docs.microsoft.com/en-us/windows/console/console-virtual-terminal-sequences
#
# @module:      cterm
# @platform:    Windows 10
# @author:      cwsoft
# @python:      3.8 or higher
#####
```

Classes

[builtins.object](#)
[Cursor](#)
[Terminal](#)
[enum.Enum\(builtins.object\)](#)
[Ansi](#)
[Colors](#)
[Styles](#)

class **Ansi**([enum.Enum](#))

[Ansi](#)(value, names=None, *, module=None, qualname=None, type=None, start=1)

Basic ANSI control sequences.

Method resolution order:

[Ansi](#)
[enum.Enum](#)
[builtins.object](#)

Data and other attributes defined here:

CSI = <Ansi.CSI: '\x1b['>

OSC = <Ansi.OSC: '\x1b]'>

Data descriptors inherited from [enum.Enum](#):

name

The name of the Enum member.

value

The value of the Enum member.

Readonly properties inherited from [enum.EnumMeta](#):

__members__

Returns a mapping of member name->value.

This mapping lists all enum members, including aliases. Note that this is a read-only view of the internal mapping.

class **Colors**([enum.Enum](#))

[Colors](#)(value, names=None, *, module=None, qualname=None, type=None, start=1)

Basic terminal foreground color codes. For background colors simply increment by 10.

Method resolution order:

[Colors](#)
[enum.Enum](#)
[builtins.object](#)

Data and other attributes defined here:

BLACK = <Colors.BLACK: 30>
BLUE = <Colors.BLUE: 34>
BRIGHT_BLUE = <Colors.BRIGHT_BLUE: 94>
BRIGHT_CYAN = <Colors.BRIGHT_CYAN: 96>
BRIGHT_GREEN = <Colors.BRIGHT_GREEN: 92>
BRIGHT_MAGENTA = <Colors.BRIGHT_MAGENTA: 95>
BRIGHT_RED = <Colors.BRIGHT_RED: 91>
BRIGHT_WHITE = <Colors.BRIGHT_WHITE: 97>
BRIGHT_YELLOW = <Colors.BRIGHT_YELLOW: 93>
CYAN = <Colors.CYAN: 36>
GREEN = <Colors.GREEN: 32>
GREY = <Colors.GREY: 90>
MAGENTA = <Colors.MAGENTA: 35>
RED = <Colors.RED: 31>
RESET = <Colors.RESET: 39>
WHITE = <Colors.WHITE: 37>
YELLOW = <Colors.YELLOW: 33>

Data descriptors inherited from [enum.Enum](#):

name

The name of the Enum member.

value

The value of the Enum member.

Readonly properties inherited from [enum.EnumMeta](#):

__members__

Returns a mapping of member name->value.

This mapping lists all enum members, including aliases. Note that this is a read-only view of the internal mapping.

class [Cursor\(builtins.object\)](#)

Static class allowing basic cursor operations supported by most terminals.

Static methods defined here:

disable()

Disable (hide) terminal cursor.

down(pos=1)

Move cursor down by pos rows.

enable()

Enables (show) terminal cursor.

left(pos=1)

Move cursor to the left by n-cols (assuming LTR languages).

restore_pos()

Restore cursor position from last stored position in memory.

right(pos=1)

Move cursor to the right by pos cols (assuming LTR languages).

set_pos(row=1, col=1)

Set cursor position to specified terminal row, col coordinates.

store_pos()

Store actual cursor position in memory.

up(pos=1)

Move cursor up by pos rows.

Data descriptors defined here:

__dict__

dictionary for instance variables (if defined)

__weakref__

list of weak references to the object (if defined)

class **Styles**([enum.Enum](#))

[Styles](#)(value, names=None, *, module=None, qualname=None, type=None, start=1)

Basic terminal styles supported by most terminals.

Method resolution order:

[Styles](#)

[enum.Enum](#)

[builtins.object](#)

Data and other attributes defined here:

BOLD = <Styles.BOLD: 1>

RESET = <Styles.RESET: 0>

REVERSE = <Styles.REVERSE: 7>

UNDERLINE = <Styles.UNDERLINE: 4>

Data descriptors inherited from [enum.Enum](#):

name

The name of the Enum member.

value

The value of the Enum member.

Readonly properties inherited from [enum.EnumMeta](#):

__members__

Returns a mapping of member name->value.

This mapping lists all enum members, including aliases. Note that this is a read-only view of the internal mapping.

class **Terminal**([builtins.object](#))

Static class to modify terminal colors and cursor position and output formatted text.

Static methods defined here:

clear(mode=<Clear.ALL: 2>)

Clear terminal screen. Mode must be of [Enum Terminal](#).Clear.

clear_line(mode=<Clear.ALL: 2>)

Clear terminal screen. Mode must be of [Enum Terminal](#).Clear.

initialize(forecolor=<Colors.RESET: 39>, backcolor=<Colors.RESET: 39>)

Initialize terminal window (reset colors, clear output, set cursor to top-left position).

set_color(forecolor=<Colors.RESET: 39>, backcolor=<Colors.RESET: 39>)

Set terminal fore- and background color to specified values. [Colors](#) must be of [Enum Colors](#).

Example: [set_color](#)(forecolor=[Colors](#).RED, backcolor=[Colors](#).YELLOW).

set_style(*styles)

Set terminal font styles to specified values. Font styles must be of [Enum Styles](#).

Example: `set_style(Styles.BOLD, Styles.UNDERLINE)`.

write(text, end='\n', row=None, col=None, forecolor=<Colors.RESET: 39>, backcolor=<Colors.RESET: 39>, auto_reset=<AutoReset.OFF: 0>)

Write text to specified terminal position using specified fore- and background color.

Use [Enum Terminal](#).AutoReset to reset colors and/or cursor position after writing to console.

Data descriptors defined here:

__dict__

dictionary for instance variables (if defined)

__weakref__

list of weak references to the object (if defined)

Data and other attributes defined here:

AutoReset = <enum 'AutoReset'>

Supported auto_reset modes as used in terminal write method.

Clear = <enum 'Clear'>

Supported clear modes used in the terminal clear methods.