

---

# **PrintTags Documentation**

***Release 1.5.0***

**Michael Lockyer**

**Aug 25, 2020**



---

## Contents

---

<b>1</b>	<b>Getting Started</b>	<b>3</b>
<b>2</b>	<b>API Documentation</b>	<b>5</b>
	<b>Python Module Index</b>	<b>13</b>
	<b>Index</b>	<b>15</b>



# [PrintTags]

>>> [info] Lightweight, tagged, and color-coded



# CHAPTER 1

---

## Getting Started

---

PrintTags is a lightweight package designed to act as an alternative to the built-in Python 3 print function. It prints color coded, tagged messages that can be useful in debugging, or if you just prefer a cleaner appearance in your terminal.

First, install PrintTags using pip:

```
pip install PrintTags
```

Then simply import it, and call the desired print function:

```
import PrintTags as pt

pt.info('My message')
```

There are also color methods that will print a colored message directly:

```
pt.green('My message')
```

PrintTags is designed to be backward compatible with Python's default print function. This means all functions within the PrintTags namespace accept the same keyword arguments as print:

```
pt.success('positional', 'arguments', sep=' ', end='\n', file=None, flush=True)
```

These functions also include additional keyword arguments that are used to customize the output:

```
# Prints using a user defined tag
pt.success('positional', 'arguments', tag='[custom_success]')
# Prepends a datetime stamp to the output
pt.success('positional', 'arguments', add_datetime=True)
# Prepends a prefix value to the output. This will not be
# treated as a positional argument and therefore will not be
# separated by "sep" argument.
pt.success('positional', 'arguments', prefix='some_prefix')
pt.success('positional', 'arguments', prefix='some_prefix')
```

All methods listed above will colorize the input string and print it to the console. If you need only to colorize a string without printing it, just import the *Colors* module and call the appropriate color method:

```
from PrintTags import Colors

# Will return "My message" wrapped in the associated ANSI escape code
blue_message = Colors.blue('My message')
```

Continue to the [API Documentation](#) for more information.



**class** `PrintTags.Colors`

Contains all the base methods responsible for wrapping input strings in ANSI escape codes

**static** `black` (*value: str*) → str

Colorizes a string to black

**Parameters** `value` (*str*) – The string to colorize

**Returns** The colorized string

**Return type** str

**static** `blue` (*value: str*) → str

Colorizes a string to blue

**Parameters** `value` (*str*) – The string to colorize

**Returns** The colorized string

**Return type** str

**static** `cyan` (*value: str*) → str

Colorizes a string to cyan

**Parameters** `value` (*str*) – The string to colorize

**Returns** The colorized string

**Return type** str

**static** `green` (*value: str*) → str

Colorizes a string to green

**Parameters** `value` (*str*) – The string to colorize

**Returns** The colorized string

**Return type** str

**static** `magenta` (*value: str*) → str

Colorizes a string to magenta

**Parameters** **value** (*str*) – The string to colorize

**Returns** The colorized string

**Return type** str

**static red** (*value: str*) → str

Colorizes a string to red

**Parameters** **value** (*str*) – The string to colorize

**Returns** The colorized string

**Return type** str

**static white** (*value: str*) → str

Colorizes a string to white

**Parameters** **value** (*str*) – The string to colorize

**Returns** The colorized string

**Return type** str

**static yellow** (*value: str*) → str

Colorizes a string to yellow

**Parameters** **value** (*str*) – The string to colorize

**Returns** The colorized string

**Return type** str

`PrintTags.black(*args, add_datetime: bool = False, prefix: Optional[str] = None, sep: str = ' ', end: str = '\n', closed_ok: bool = False, file: Optional[TextIO] = None, flush: bool = False) → None`

Prints values in black.

**Args:** `add_datetime` (bool, optional): Whether or not a datetime timestamp should be printed. Default *False*. `prefix` (any, optional): A string interpolatable value that should be prepended to the print. Default *None*. `sep` (str, optional): String inserted between values, default is a space. Default ' '. `end` (str, optional): String appended after the last value, default is a newline. Default '\n'.

‘.

**closed\_ok** (bool, optional): Whether or not the `ValueError` raised by a closed `stdout` should be suppressed. Default *False*.

`file`: A file-like object (stream, optional): Defaults to the current `sys.stdout`. Default *None*. `flush` (bool, optional): Whether to forcibly flush the stream. Default *False*.

`PrintTags.red(*args, add_datetime: bool = False, prefix: Optional[str] = None, sep: str = ' ', end: str = '\n', closed_ok: bool = False, file: Optional[TextIO] = None, flush: bool = False) → None`

Prints values in red.

**Args:** `add_datetime` (bool, optional): Whether or not a datetime timestamp should be printed. Default *False*. `prefix` (any, optional): A string interpolatable value that should be prepended to the print. Default *None*. `sep` (str, optional): String inserted between values, default is a space. Default ' '. `end` (str, optional): String appended after the last value, default is a newline. Default '\n'.

‘.

**closed\_ok (bool, optional):** Whether or not the `ValueError` raised by a closed stdout should be suppressed. Default *False*.

file: A file-like object (stream, optional): Defaults to the current `sys.stdout`. Default *None*. flush (bool, optional): Whether to forcibly flush the stream. Default *False*.

```
PrintTags.green(*args, add_datetime: bool = False, prefix: Optional[str] = None, sep: str = ' ', end: str
               = '\n', closed_ok: bool = False, file: Optional[TextIO] = None, flush: bool = False)
               → None
```

Prints values in green.

**Args:** add\_datetime (bool, optional): Whether or not a datetime timestamp should be printed. Default *False*. prefix (any, optional): A string interpolatable value that should be prepended to the print. Default *None*. sep (str, optional): String inserted between values, default is a space. Default ‘ ‘. end (str, optional): String appended after the last value, default is a newline. Default ‘

‘.

**closed\_ok (bool, optional):** Whether or not the `ValueError` raised by a closed stdout should be suppressed. Default *False*.

file: A file-like object (stream, optional): Defaults to the current `sys.stdout`. Default *None*. flush (bool, optional): Whether to forcibly flush the stream. Default *False*.

```
PrintTags.yellow(*args, add_datetime: bool = False, prefix: Optional[str] = None, sep: str = ' ', end:
                 str = '\n', closed_ok: bool = False, file: Optional[TextIO] = None, flush: bool =
                 False) → None
```

Prints values in yellow.

**Args:** add\_datetime (bool, optional): Whether or not a datetime timestamp should be printed. Default *False*. prefix (any, optional): A string interpolatable value that should be prepended to the print. Default *None*. sep (str, optional): String inserted between values, default is a space. Default ‘ ‘. end (str, optional): String appended after the last value, default is a newline. Default ‘

‘.

**closed\_ok (bool, optional):** Whether or not the `ValueError` raised by a closed stdout should be suppressed. Default *False*.

file: A file-like object (stream, optional): Defaults to the current `sys.stdout`. Default *None*. flush (bool, optional): Whether to forcibly flush the stream. Default *False*.

```
PrintTags.blue(*args, add_datetime: bool = False, prefix: Optional[str] = None, sep: str = ' ', end: str
               = '\n', closed_ok: bool = False, file: Optional[TextIO] = None, flush: bool = False) →
               None
```

Prints values in blue.

**Args:** add\_datetime (bool, optional): Whether or not a datetime timestamp should be printed. Default *False*. prefix (any, optional): A string interpolatable value that should be prepended to the print. Default *None*. sep (str, optional): String inserted between values, default is a space. Default ‘ ‘. end (str, optional): String appended after the last value, default is a newline. Default ‘

‘.

**closed\_ok (bool, optional):** Whether or not the `ValueError` raised by a closed stdout should be suppressed. Default *False*.

file: A file-like object (stream, optional): Defaults to the current `sys.stdout`. Default *None*. flush (bool, optional): Whether to forcibly flush the stream. Default *False*.

```
PrintTags.magenta(*args, add_datetime: bool = False, prefix: Optional[str] = None, sep: str = ' ', end: str = '\n', closed_ok: bool = False, file: Optional[TextIO] = None, flush: bool = False) → None
```

Prints values in magenta.

**Args:** add\_datetime (bool, optional): Whether or not a datetime timestamp should be printed. Default *False*. prefix (any, optional): A string interpolatable value that should be prepended to the print. Default *None*. sep (str, optional): String inserted between values, default is a space. Default ' '. end (str, optional): String appended after the last value, default is a newline. Default ' '.

‘.

**closed\_ok (bool, optional):** Whether or not the `ValueError` raised by a closed stdout should be suppressed. Default *False*.

file: A file-like object (stream, optional): Defaults to the current `sys.stdout`. Default *None*. flush (bool, optional): Whether to forcibly flush the stream. Default *False*.

```
PrintTags.cyan(*args, add_datetime: bool = False, prefix: Optional[str] = None, sep: str = ' ', end: str = '\n', closed_ok: bool = False, file: Optional[TextIO] = None, flush: bool = False) → None
```

Prints values in cyan.

**Args:** add\_datetime (bool, optional): Whether or not a datetime timestamp should be printed. Default *False*. prefix (any, optional): A string interpolatable value that should be prepended to the print. Default *None*. sep (str, optional): String inserted between values, default is a space. Default ' '. end (str, optional): String appended after the last value, default is a newline. Default ' '.

‘.

**closed\_ok (bool, optional):** Whether or not the `ValueError` raised by a closed stdout should be suppressed. Default *False*.

file: A file-like object (stream, optional): Defaults to the current `sys.stdout`. Default *None*. flush (bool, optional): Whether to forcibly flush the stream. Default *False*.

```
PrintTags.white(*args, add_datetime: bool = False, prefix: Optional[str] = None, sep: str = ' ', end: str = '\n', closed_ok: bool = False, file: Optional[TextIO] = None, flush: bool = False) → None
```

Prints values in white.

**Args:** add\_datetime (bool, optional): Whether or not a datetime timestamp should be printed. Default *False*. prefix (any, optional): A string interpolatable value that should be prepended to the print. Default *None*. sep (str, optional): String inserted between values, default is a space. Default ' '. end (str, optional): String appended after the last value, default is a newline. Default ' '.

‘.

**closed\_ok (bool, optional):** Whether or not the `ValueError` raised by a closed stdout should be suppressed. Default *False*.

file: A file-like object (stream, optional): Defaults to the current `sys.stdout`. Default *None*. flush (bool, optional): Whether to forcibly flush the stream. Default *False*.

```
PrintTags.info(*args, tag_text: Optional[str] = 'info', add_datetime: bool = False, prefix: Optional[str] = None, sep: str = ' ', end: str = '\n', closed_ok: bool = False, file: Optional[TextIO] = None, flush: bool = False) → None
```

Used for printing basic information.

**Args:**

**tag\_text (str, optional):** The text content of the tag that will be prepended to the print.

*None* for no tag. Default *'info'*.

add\_datetime (bool, optional): Whether or not a datetime timestamp should be printed. Default

*False*. prefix (str, optional): A string interpolatable value that will be prepended to the print.

Default *None*. sep (str, optional): string inserted between values, default is a space. Default *' '*.

end (str, optional): string appended after the last value, default is a newline. Default *'\n'*.

“.

**closed\_ok (bool, optional):** Whether or not the `ValueError` raised by a closed stdout should be suppressed. Default *False*.

file (TextIO, optional): defaults to the current `sys.stdout`. Default *None*. flush (bool, optional): whether to forcibly flush the stream. Default *False*.

```
PrintTags.success(*args, tag_text: Optional[str] = 'success', add_datetime: bool = False, prefix: Optional[str] = None, sep: str = ' ', end: str = '\n', closed_ok: bool = False, file: Optional[TextIO] = None, flush: bool = False) → None
```

Used to indicate successful execution.

**Args:**

**tag\_text (str, optional):** The text content of the tag that will be prepended to the print.

*None* for no tag. Default *'success'*.

add\_datetime (bool, optional): Whether or not a datetime timestamp should be printed. Default

*False*. prefix (str, optional): A string interpolatable value that will be prepended to the print.

Default *None*. sep (str, optional): string inserted between values, default is a space. Default *' '*.

end (str, optional): string appended after the last value, default is a newline. Default *'\n'*.

“.

**closed\_ok (bool, optional):** Whether or not the `ValueError` raised by a closed stdout should be suppressed. Default *False*.

file (TextIO, optional): defaults to the current `sys.stdout`. Default *None*. flush (bool, optional): whether to forcibly flush the stream. Default *False*.

```
PrintTags.notice(*args, tag_text: Optional[str] = 'notice', add_datetime: bool = False, prefix: Optional[str] = None, sep: str = ' ', end: str = '\n', closed_ok: bool = False, file: Optional[TextIO] = None, flush: bool = False) → None
```

Used to print important information.

**Args:**

**tag\_text (str, optional):** The text content of the tag that will be prepended to the print.

*None* for no tag. Default *'notice'*.

add\_datetime (bool, optional): Whether or not a datetime timestamp should be printed. Default

*False*. prefix (str, optional): A string interpolatable value that will be prepended to the print.

Default *None*. sep (str, optional): string inserted between values, default is a space. Default *' '*.

end (str, optional): string appended after the last value, default is a newline. Default *'\n'*.

“.

**closed\_ok (bool, optional):** Whether or not the *ValueError* raised by a closed stdout should be suppressed. Default *False*.

file (TextIO, optional): defaults to the current *sys.stdout*. Default *None*. flush (bool, optional): whether to forcibly flush the stream. Default *False*.

`PrintTags.timeout (*args, tag_text: Optional[str] = 'timeout', add_datetime: bool = False, prefix: Optional[str] = None, sep: str = ' ', end: str = '\n', closed_ok: bool = False, file: Optional[TextIO] = None, flush: bool = False) → None`

Used to indicate a timeout.

#### Args:

**tag\_text (str, optional):** The text content of the tag that will be prepended to the print.

*None* for no tag. Default *'timeout'*.

add\_datetime (bool, optional): Whether or not a datetime timestamp should be printed. Default

*False*. prefix (str, optional): A string interpolatable value that will be prepended to the print.

Default *None*. sep (str, optional): string inserted between values, default is a space. Default *' '*.

end (str, optional): string appended after the last value, default is a newline. Default *'\n'*.

“.

**closed\_ok (bool, optional):** Whether or not the *ValueError* raised by a closed stdout should be suppressed. Default *False*.

file (TextIO, optional): defaults to the current *sys.stdout*. Default *None*. flush (bool, optional): whether to forcibly flush the stream. Default *False*.

`PrintTags.warn (*args, tag_text: Optional[str] = 'warn', add_datetime: bool = False, prefix: Optional[str] = None, sep: str = ' ', end: str = '\n', closed_ok: bool = False, file: Optional[TextIO] = None, flush: bool = False) → None`

Used to highlight that there may be an issue, or that code has improperly executed.

#### Args:

**tag\_text (str, optional):** The text content of the tag that will be prepended to the print.

*None* for no tag. Default *'warn'*.

add\_datetime (bool, optional): Whether or not a datetime timestamp should be printed. Default

*False*. prefix (str, optional): A string interpolatable value that will be prepended to the print.

Default *None*. sep (str, optional): string inserted between values, default is a space. Default *' '*.

end (str, optional): string appended after the last value, default is a newline. Default *'\n'*.

“.

**closed\_ok (bool, optional):** Whether or not the *ValueError* raised by a closed stdout should be suppressed. Default *False*.

file (TextIO, optional): defaults to the current sys.stdout. Default *None*. flush (bool, optional): whether to forcibly flush the stream. Default *False*.

```
PrintTags.error(*args, tag_text: Optional[str] = 'error', add_datetime: bool = False, prefix: Optional[str] = None, sep: str = ' ', end: str = '\n', closed_ok: bool = False, file: Optional[TextIO] = None, flush: bool = False) → None
```

Can be used to print the description or message associated with an exception.

#### Args:

**tag\_text (str, optional):** The text content of the tag that will be prepended to the print.  
*None* for no tag. Default *'error'*.

add\_datetime (bool, optional): Whether or not a datetime timestamp should be printed. Default *False*. prefix (str, optional): A string interpolatable value that will be prepended to the print. Default *None*. sep (str, optional): string inserted between values, default is a space. Default ' '. end (str, optional): string appended after the last value, default is a newline. Default '\n'.

“.

**closed\_ok (bool, optional):** Whether or not the *ValueError* raised by a closed stdout should be suppressed. Default *False*.

file (TextIO, optional): defaults to the current sys.stdout. Default *None*. flush (bool, optional): whether to forcibly flush the stream. Default *False*.





**p**

`PrintTags`, 5



## B

`black()` (*in module `PrintTags`*), 6  
`black()` (*`PrintTags.Colors` static method*), 5  
`blue()` (*in module `PrintTags`*), 7  
`blue()` (*`PrintTags.Colors` static method*), 5

## C

`Colors` (*class in `PrintTags`*), 5  
`cyan()` (*in module `PrintTags`*), 8  
`cyan()` (*`PrintTags.Colors` static method*), 5

## E

`error()` (*in module `PrintTags`*), 11

## G

`green()` (*in module `PrintTags`*), 7  
`green()` (*`PrintTags.Colors` static method*), 5

## I

`info()` (*in module `PrintTags`*), 9

## M

`magenta()` (*in module `PrintTags`*), 8  
`magenta()` (*`PrintTags.Colors` static method*), 5

## N

`notice()` (*in module `PrintTags`*), 9

## P

`PrintTags` (*module*), 5

## R

`red()` (*in module `PrintTags`*), 6  
`red()` (*`PrintTags.Colors` static method*), 6

## S

`success()` (*in module `PrintTags`*), 9

## T

`timeout()` (*in module `PrintTags`*), 10

## W

`warn()` (*in module `PrintTags`*), 10  
`white()` (*in module `PrintTags`*), 8  
`white()` (*`PrintTags.Colors` static method*), 6

## Y

`yellow()` (*in module `PrintTags`*), 7  
`yellow()` (*`PrintTags.Colors` static method*), 6