
consolemenu Documentation

Release 0.8.0

Author

Mar 05, 2023

1	Installation	3
2	Usage	5
2.1	Getting a selection	6
3	API Reference	7
3.1	ConsoleMenu — Standard menu class	7
3.2	SelectionMenu — Quickly get a selection	10
3.3	MultiSelectMenu — Make multiple selections at one prompt	11
3.4	Items	12
3.4.1	CommandItem	12
3.4.2	ExitItem	12
3.4.3	ExternalItem	12
3.4.4	FunctionItem	13
3.4.5	MenuItem	13
3.4.6	SelectedItem	14
3.4.7	SubmenuItem	14
3.5	Functions	15
3.6	Screen	15
3.7	Menu Components	16
3.7.1	Dimension	16
3.7.2	MenuComponent	16
3.8	Formatting	18
3.8.1	MenuBorderStyle	18
3.8.2	MenuBorderStyleType	18
3.8.3	MenuBorderStyleFactory	19
3.8.4	Menu Borders	20
3.8.4.1	AsciiBorderStyle	20
3.8.4.2	DoubleLineBorderStyle	20
3.8.4.3	DoubleLineOuterLightInnerBorderStyle	21
3.8.4.4	HeavyBorderStyle	21
3.8.4.5	HeavyOuterLightInnerBorderStyle	21
3.8.4.6	LightBorderStyle	21
3.8.5	MenuMargins	21
3.8.6	MenuPadding	22
3.8.7	MenuStyle	22
3.9	Prompts	23

3.9.1	PromptFormatter	23
3.9.2	PromptUtils	24
3.9.3	UserQuit	26
4	Indices and tables	27
	Index	29

Contents:

CHAPTER 1

Installation

Everyone should run:

```
pip install console-menu
```


First things first, import the package:

```
import consolemenu
```

Or just import what you need:

```
from consolemenu import ConsoleMenu
from consolemenu.items import FunctionItem, SubmenuItem, CommandItem
```

Then create a menu:

```
menu = ConsoleMenu("This is a menu!", "It has a subtitle too!")
```

Create menu items for each choice you need:

```
command_item = CommandItem("Run a console command", "touch hello.txt")
function_item = FunctionItem("Call a function", input, ["Enter some input"])
```

To add other menus as submenus, use a *SubmenuItem*, setting the menu property in the constructor so the submenu's parent is set properly:

```
submenu = ConsoleMenu("This is the submenu")
submenu_item = SubmenuItem("Show a submenu", submenu, menu=menu)
```

Add the items to the menu:

```
menu.append_item(command_item)
menu.append_item(function_item)
menu.append_item(submenu_item)
```

Then start the menu:

```
menu.start()
```

After that, the menu will spawn its own thread and go about its business. If you want to wait on the user to finish with the menu before continuing, call:

```
menu.join()
```

To combine these two and simply show a menu and immediately wait for the user to exit the menu, call:

```
menu.show()
```

2.1 Getting a selection

If you have a list of strings, and you want to allow the user to select one, you can use a *SelectionMenu*:

```
from consolemenu import SelectionMenu

a_list = ["red", "blue", "green"]

selection = SelectionMenu.get_selection(a_list)
```

Which is equivalent to:

```
from consolemenu import SelectionMenu

a_list=["red", "blue", "green"]

menu = SelectionMenu(a_list,"Select an option")

menu.show()

menu.join()

selection = menu.selected_option
```

3.1 ConsoleMenu — Standard menu class

```
class consolemenu.ConsoleMenu (title=None, subtitle=None, screen=None, formatter=None, prologue_text=None, epilogue_text=None, clear_screen=True, show_exit_option=True, exit_option_text='Exit', exit_menu_char=None)
```

A class that displays a menu and allows the user to select an option.

Parameters

- **title** (*str*) – The title of the menu, or a method reference that returns a string.
- **subtitle** (*str*) – The subtitle of the menu, or a method reference that returns a string.
- **screen** (*consolemenu.screen.Screen*) – The screen object associated with this menu.
- **formatter** (*MenuFormatBuilder*) – The `MenuFormatBuilder` instance used to format this menu.
- **prologue_text** (*str*) – Text or method reference to include in the “prologue” section of the menu.
- **epilogue_text** (*str*) – Text or method reference to include in the “epilogue” section of the menu.
- **show_exit_option** (*bool*) – Specifies whether this menu should show an exit item by default. Defaults to `True`. Can be overridden when the menu is started.
- **exit_option_text** (*str*) – Text for the Exit menu item. Defaults to `'Exit'`.
- **exit_menu_char** (*str*) – Character to use for exiting the menu. Defaults to `None`.
- **clear_screen** (*bool*) – Set to `False` to disable clearing of screen between menus

cls.currently_active_menu

Class variable that holds the currently active menu or None if no menu is currently active (e.g. when switching between menus)

Type *ConsoleMenu*

items

The list of MenuItem objects that the menu will display

Type list of MenuItem

parent

The parent of this menu

Type *ConsoleMenu*

previous_active_menu

the previously active menu to be restored into the class's currently active menu

Type *ConsoleMenu*

current_option

The currently highlighted menu option

Type int

selected_option

The option that the user has most recently selected

Type int

current_item

The item corresponding to the menu option that is currently highlighted, or None.

Type *consolemenu.items.MenuItem*

selected_item

The item in *items* that the user most recently selected, or None.

Type *consolemenu.items.MenuItem*

start (*show_exit_option=None*)

Start the menu in a new thread and allow the user to interact with it. The thread is a daemon, so *join()* should be called if there's a possibility that the main thread will exit before the menu is done

Parameters **show_exit_option** (*bool*) – Specify whether the exit item should be shown, defaults to the value set in the constructor

join (*timeout=None*)

Should be called at some point after *start()* to block until the menu exits.

Parameters **timeout** (*Number*) – How long to wait before timing out.

show (*show_exit_option=None*)

Calls *start* and then immediately joins.

Parameters **show_exit_option** (*bool*) – Specify whether the exit item should be shown, defaults to the value set in the constructor

append_item (*item*)

Add an item to the end of the menu before the exit item.

Parameters **item** (*MenuItem*) – The item to be added.

add_exit ()

Add the exit item if necessary. Used to make sure there aren't multiple exit items.

Returns True if item needed to be added, False otherwise.

Return type bool

remove_exit ()

Remove the exit item if necessary. Used to make sure we only remove the exit item, not something else.

Returns True if item needed to be removed, False otherwise.

Return type bool

get_input ()

Can be overridden to change the input method. Called in `process_user_input` ()

Returns the ordinal value of a single character

Return type int

process_user_input ()

Gets the next single character and decides what to do with it

draw ()

Refresh the screen and redraw the menu. Should be called whenever something changes that needs to be redrawn.

go_to (*option*)

Go to the option entered by the user as a number

Parameters **option** (*int*) – the option to go to

go_up ()

Go up one, wrap to end if necessary

go_down ()

Go down one, wrap to beginning if necessary

select ()

Select the current item and run it

exit ()

Signal the menu to exit, then block until it's done cleaning up

is_alive ()

Check whether the thread is still alive.

Returns True if the thread is still alive; False otherwise.

Return type bool

wait_for_start (*timeout=None*)

Block until the menu is started.

Parameters **timeout** – How long to wait before timing out.

Returns False if timeout is given and operation times out, True otherwise. None before Python 2.7.

Return type bool

pause ()

Temporarily pause the menu until resume is called.

resume ()

Sets the currently active menu to this one and resumes it.

is_running()

Check if the menu has been started and is not paused.

Returns True if the menu is started and hasn't been paused; False otherwise.

Return type bool

3.2 SelectionMenu — Quickly get a selection

Bases: `consolemenu.ConsoleMenu`

```
class consolemenu.SelectionMenu(strings, title=None, subtitle=None, screen=None,
                                formatter=None, prologue_text=None, epilogue_text=None,
                                show_exit_option=True, exit_option_text='Exit',
                                clear_screen=True)
```

A menu that simplifies item creation, just give it a list of strings and it builds the menu for you

Parameters

- **strings** (list of str) – The list of strings this menu should be built from.
- **title** (str) – The title of the menu.
- **subtitle** (str) – The subtitle of the menu.
- **screen** (`consolemenu.screen.Screen`) – The screen object associated with this menu.
- **formatter** (MenuFormatBuilder) – The MenuFormatBuilder instance used to format this menu.
- **prologue_text** (str) – Text to include in the “prologue” section of the menu.
- **epilogue_text** (str) – Text to include in the “epilogue” section of the menu.
- **show_exit_option** (bool) – Specifies whether this menu should show an exit item by default. Defaults to True. Can be overridden when the menu is started.
- **exit_option_text** (str) – Text for the Exit menu item. Defaults to ‘Exit’.
- **clear_screen** (bool) – Set to False to disable clearing of screen between menus

```
classmethod get_selection(strings, title='Select an option', subtitle=None,
                          show_exit_option=True, _menu=None)
```

Single-method way of getting a selection out of a list of strings.

Parameters

- **strings** (list of str) – The list of strings this menu should be built from.
- **title** (str) – The title of the menu.
- **subtitle** (str) – The subtitle of the menu.
- **show_exit_option** (bool) – Specifies whether this menu should show an exit item by default. Defaults to True.
- **_menu** – Should probably only be used for testing, pass in a list and the created menu used internally by the method will be appended to it

Returns The index of the selected option.

Return type int

3.3 MultiSelectMenu — Make multiple selections at one prompt

Bases: `consolemenu.ConsoleMenu`

```
class consolemenu.MultiSelectMenu (title=None, subtitle=None, formatter=None,
                                     prologue_text=None, epilogue_text=None,
                                     ack_item_completion=True, show_exit_option=True,
                                     exit_option_text='Exit', clear_screen=True)
```

Console menu that allows the selection of multiple menu items at a single prompt.

Parameters

- **title** – The menu title.
- **subtitle** – The menu subtitle.
- **formatter** – The menu formatter instance for styling the menu.
- **prologue_text** – The text to display in the prologue section of the menu.
- **epilogue_text** – The text to display in the epilogue section of the menu.
- **show_exit_option** (*bool*) – Determines if the exit item should be displayed.
- **exit_option_text** (*str*) – Text for the Exit menu item. Defaults to 'Exit'.
- **clear_screen** (*bool*) – Set to False to disable clearing of screen between menus

append_item (*item*)

Add an item to the end of the menu before the exit item.

Note that Multi-Select Menus will not allow a `SubMenuItem` to be added, as multi-select menus are expected to be used only for executing multiple actions.

Parameters **item** (`MenuItem`) – The item to be added

Raises **TypeError** – If the specified `MenuItem` is a `SubMenuItem`.

process_user_input ()

This overrides the method in `ConsoleMenu` to allow for comma-delimited and range inputs.

Examples

All of the following inputs would have the same result:

- 1,2,3,4
- 1-4
- 1-2,3-4
- 1 - 4
- 1, 2, 3, 4

Raises **ValueError** – If the input cannot be correctly parsed.

3.4 Items

3.4.1 CommandItem

Bases: *consolemenu.items.ExternalItem*

class consolemenu.items.**CommandItem**(*text*, *command*, *arguments=None*, *menu=None*,
should_exit=False, *menu_char=None*)

A menu item to execute a console command

Variables

- **text** (*str*) – The text shown for this menu item
- **command** (*str*) – The console command to be executed
- **arguments** (*list[str]*) – An optional list of string arguments to be passed to the command
- **menu** (*ConsoleMenu*) – The menu to which this item belongs
- **should_exit** (*bool*) – Whether the menu should exit once this item’s action is done
- **menu_char** (*str*) – The character used to select this menu item. Optional - defaults to None.

action ()

This class overrides this method

get_return ()

Returns the exit status of the command

Return type int

3.4.2 ExitItem

Bases: *consolemenu.items.MenuItem*

class consolemenu.items.**ExitItem**(*text='Exit'*, *menu=None*, *menu_char=None*)
Used to exit the current menu. Handled by *consolemenu.ConsoleMenu*

show (*index*, *available_width=None*)

ExitItem overrides this method to display appropriate Exit or Return text.

3.4.3 ExternalItem

Bases: *consolemenu.items.MenuItem*

class consolemenu.items.**ExternalItem**(*text*, *menu=None*, *should_exit=False*,
menu_char=None)

A base class for items that need to do stuff on the console outside of the console menu. Sets the terminal back to standard mode until the action is done. Should probably be subclassed.

clean_up ()

This class overrides this method

set_up ()

This class overrides this method

3.4.4 FunctionItem

Bases: `consolemenu.items.ExternalItem`

class `consolemenu.items.FunctionItem` (*text, function, args=None, kwargs=None, menu=None, should_exit=False, menu_char=None*)

A menu item to call a Python function

Variables

- **text** (*str*) – The text shown for this menu item
- **function** – The function to be called
- **args** (*list*) – An optional list of arguments to be passed to the function
- **kwargs** (*dict*) – An optional dictionary of keyword arguments to be passed to the function
- **menu** (`ConsoleMenu`) – The menu to which this item belongs
- **should_exit** (*bool*) – Whether the menu should exit once this item’s action is done
- **menu_char** (*str*) – The character used to select this menu item. Optional - defaults to None.

action ()

This class overrides this method

clean_up ()

This class overrides this method

get_return ()

Returns The return value from the function call

3.4.5 MenuItem

class `consolemenu.items.MenuItem` (*text, menu=None, should_exit=False, menu_char=None*)

A generic menu item

Variables

- **text** (*str*) – The text shown for this menu item
- **menu** (`ConsoleMenu`) – The menu to which this item belongs
- **should_exit** (*bool*) – Whether the menu should exit once this item’s action is done
- **menu_char** (*str*) – The character used to select this menu item. Optional - defaults to None.

action ()

Override to carry out the main action for this item.

clean_up ()

Override to add any cleanup actions necessary for the item

get_return ()

Override to change what the item returns. Otherwise just returns the same value the last selected item did.

set_up ()

Override to add any setup actions necessary for the item

show (*index*)

How this item should be displayed in the menu. Can be overridden, but should keep the same signature.

Default is:

- 1 - Item 1
- 2 - Another Item

Parameters **index** (*int*) – The index of the item in the items list of the menu

Returns The representation of the item to be shown in a menu

Return type str

3.4.6 SelectionItem

Bases: *consolemenu.items.MenuItem*

class consolemenu.items.**SelectionItem** (*text, index, menu=None, menu_char=None*)

The item type used in *consolemenu.SelectionMenu*

Variables

- **text** (*str*) – The text shown for this menu item
- **index** (*int*) – The index of this item in the list used to initialize the *consolemenu.SelectionMenu*
- **menu** (*ConsoleMenu*) – The menu to which this item belongs
- **menu_char** (*str*) – The character used to select this menu item. Optional - defaults to None.

get_return ()

Returns The index of this item in the list of strings

Return type int

3.4.7 SubmenuItem

Bases: *consolemenu.items.MenuItem*

class consolemenu.items.**SubmenuItem** (*text, submenu, menu=None, should_exit=False, menu_char=None*)

A menu item to open a submenu

Variables

- **text** (*str*) – The text shown for this menu item
- **submenu** (*ConsoleMenu*) – The submenu to be opened when this item is selected
- **menu** (*ConsoleMenu*) – The menu to which this item belongs
- **should_exit** (*bool*) – Whether the menu should exit once this item's action is done
- **menu_char** (*str*) – The character used to select this menu item. Optional - defaults to None.

action ()

This class overrides this method

clean_up()

This class overrides this method

get_return()

Returns The returned value in the submenu

get_submenu()

We unwrap the submenu variable in case it is a reference to a method that returns a submenu

set_menu(menu)

Sets the menu of this item. Should be used instead of directly accessing the menu attribute for this class.

Parameters menu (`ConsoleMenu`) – the menu

set_up()

This class overrides this method

3.5 Functions

`consolemenu.clear_terminal()`

Call the platform specific function to clear the terminal: `cls` on windows, `reset` otherwise

3.6 Screen

class `consolemenu.screen.Screen`

Class representing a console screen.

static clear()

Clear the screen.

static flush()

Flush any buffered standard output to screen.

input(prompt=)

Prompt the end user for input.

Parameters prompt (`str`, optional) – The message to display as the prompt.

Returns The input provided by the user.

static printf(*args)

Print the specified arguments to the screen.

Parameters *args – Variable length argument list.

static println(*args)

Print the specified arguments to the screen, including an appended newline character.

Parameters *args – Variable length argument list.

screen_height

The screen height in rows.

Type `int`

screen_width

The screen width in columns.

Type `int`

3.7 Menu Components

3.7.1 Dimension

class `consolemenu.menu_component.Dimension` (*width=0, height=0, dimension=None*)

The Dimension class encapsulates the height and width of a component.

Parameters

- **width** (*int*) – the width of the Dimension, in columns.
- **height** (*int*) – the height of the Dimension, in rows.
- **dimension** (`Dimension`, *optional*) – an existing Dimension from which to duplicate the height and width.

3.7.2 MenuComponent

class `consolemenu.menu_component.MenuComponent` (*menu_style, max_dimension=None*)

Base class for a menu component.

Parameters

- **menu_style** (`MenuStyle`) – the style for this component.
- **max_dimension** (`Dimension`) – the maximum Dimension (width x height) for the menu. Defaults to width=80 and height=40 if not specified.

Raises `TypeError` – if `menu_style` is not a `MenuStyle`.

border_style

The border style for this component.

Type `consolemenu.format.MenuBorderStyle`

calculate_border_width()

Calculate the width of the menu border. This will be the width of the maximum allowable dimensions (usually the screen size), minus the left and right margins and the newline character. For example, given a maximum width of 80 characters, with left and right margins both set to 1, the border width would be 77 ($80 - 1 - 1 - 1 = 77$).

Returns the menu border width in columns.

Return type `int`

calculate_content_width()

Calculate the width of inner content of the border. This will be the width of the menu borders, minus the left and right padding, and minus the two vertical border characters. For example, given a border width of 77, with left and right margins each set to 2, the content width would be 71 ($77 - 2 - 2 - 2 = 71$).

Returns the inner content width in columns.

Return type `int`

generate()

Generate this component.

Yields *str* – The next string of characters for drawing this component.

inner_horizontal_border()

The complete inner horizontal border section, including the left and right border verticals.

Returns The complete inner horizontal border.

Return type str

inner horizontals ()

The string of inner horizontal border characters of the required length for this component (not including the menu margins or verticals).

Returns The inner horizontal characters.

Return type str

margins

The margins for this component.

Type *consolemenu.format.MenuMargins*

max_dimension

The maximum dimension for the menu.

Type *Dimension*

outer_horizontal_border_bottom ()

The complete outer bottom horizontal border section, including left and right margins.

Returns The bottom menu border.

Return type str

outer_horizontal_border_top ()

The complete outer top horizontal border section, including left and right margins.

Returns The top menu border.

Return type str

outer_horizontals ()

The string of outer horizontal border characters of the required length for this component (not including the menu margins or verticals).

Returns The outer horizontal characters.

Return type str

padding

The padding for this component.

Type *consolemenu.format.MenuPadding*

row (*content=""*, *align='left'*, *indent_len=0*)

A row of the menu, which comprises the left and right verticals plus the given content. If the content is larger than the allotted space for a single row, the content is wrapped onto multiple lines, while also respecting user-included newline characters.

Returns One or more rows of this menu component with the specified content.

Return type str

style

The style for this component.

Type *consolemenu.format.MenuStyle*

3.8 Formatting

3.8.1 MenuBorderStyle

class consolemenu.format.MenuBorderStyle

Base class for console menu border. Each property should be overridden by a subclass.

bottom_left_corner

The outer, bottom left corner of the menu.

bottom_right_corner

The outer, bottom right corner of the menu.

inner_horizontal

The character for inner horizontal section lines.

inner_vertical

The character for inner vertical section lines.

intersection

The character for intersecting inner vertical and inner horizontal lines (a “+” shape).

outer_horizontal

The character for outer, horizontal lines (the top and bottom lines of the menu).

outer_horizontal_inner_down

The character for a top horizontal line with a downward inner line (a “T” shape).

outer_horizontal_inner_up

The character for a bottom horizontal line with an upward inner line (an inverted “T” shape).

outer_vertical

The character for an outer vertical line of the menu (the left and right sides of the menu).

outer_vertical_inner_left

The character for an outer vertical line, with a protruding inner line to the left.

outer_vertical_inner_right

The character for an outer vertical line, with a protruding inner line to the right.

top_left_corner

The top left corner of the menu.

top_right_corner

The top right corner of the menu.

3.8.2 MenuBorderStyleType

class consolemenu.format.MenuBorderStyleType

Defines the various menu border styles, as expected by the border factory.

ASCII_BORDER = 0

Menu Border using pure ASCII characters. Usable on all platforms.

Type int

LIGHT_BORDER = 1

Menu Border using the “light” box drawing characters. Should be usable on all platforms.

Type int

HEAVY_BORDER = 2

Menu Border using the “heavy” box drawing characters. NOTE: On Windows, this border style will work ONLY on Python 3.6 and later. It will raise a UnicodeEncodeError exception on earlier Python versions. If requesting this border style via the MenuBorderStyleFactory when on Windows/Python 3.5 or earlier, this border style will be substituted by the *DOUBLE_LINE_BORDER*.

Type int

DOUBLE_LINE_BORDER = 3

Menu Border using “double-line” box drawing characters.

Type int

HEAVY_OUTER_LIGHT_INNER_BORDER = 4

Menu Border using the “heavy” box drawing characters for the outer border elements, and “light” box-drawing characters for the inner border elements. NOTE: On Windows, this border style will work ONLY on Python 3.6 and later. It will raise a UnicodeEncodeError exception on earlier Python versions. If requesting this border style via the MenuBorderStyleFactory when on Windows/Python 3.5 or earlier, this border style will be substituted by the *DOUBLE_LINE_BORDER*.

Type int

DOUBLE_LINE_OUTER_LIGHT_INNER_BORDER = 5

Menu Border using the “double-line” box drawing characters for the outer border elements, and “light” box-drawing characters for the inner border elements.

Type int

3.8.3 MenuBorderStyleFactory

class consolemenu.format.MenuBorderStyleFactory

Factory class for creating MenuBorderStyle instances.

create_ascii_border()

Create an ASCII border style.

Returns a new instance of AsciiBorderStyle.

Return type AsciiBorderStyle

create_border(border_style_type)

Create a new MenuBorderStyle instance based on the given border style type.

Parameters border_style_type (*int*) – an integer value from *MenuBorderStyleType*.

Returns a new MenuBorderStyle instance of the specified style.

Return type MenuBorderStyle

create_doubleline_border()

Create a border style using “double-line” box drawing characters.

Returns a new instance of DoubleLineBorderStyle.

Return type DoubleLineBorderStyle

create_doubleline_outer_light_inner_border()

Create a border style using “double-line” box drawing characters for outer border elements, and “light” box drawing characters for inner border elements.

Returns a new instance of DoubleLineOuterLightInnerBorderStyle

Return type DoubleLineOuterLightInnerBorderStyle

create_heavy_border ()

Create a border style using “heavy” box drawing characters.

NOTE: The Heavy border style will work on Windows ONLY when using Python 3.6 or later. If on Windows and using an earlier version of Python, the heavy border will be substituted with the DOUBLE_LINE_BORDER.

Returns a new instance of HeavyBorderStyle, unless on Windows and pre-Python 3.5, in which case a new instance of DoubleLineStyle will be returned.

Return type HeavyBorderStyle or DoubleLineStyle

create_heavy_outer_light_inner_border ()

Create a border style using “heavy” box drawing characters for outer border elements, and “light” box drawing characters for inner border elements.

NOTE: The Heavy border style will work on Windows ONLY when using Python 3.6 or later. If on Windows and using an earlier version of Python, the heavy border will be substituted with the DOUBLE_LINE_BORDER.

Returns a new instance of HeavyOuterLightInnerBorderStyle, unless on Windows and pre-Python 3.5, in which case a new instance of DoubleLineOuterLightInnerBorderStyle will be returned.

Return type HeavyOuterLightInnerBorderStyle or
DoubleLineOuterLightInnerBorderStyle

create_light_border ()

Create a border style using “light” box drawing characters.

Returns a new instance of LightBorderStyle

Return type LightBorderStyle

static is_win_python35_or_earlier ()

Convenience method to determine if the current platform is Windows and Python version 3.5 or earlier.

Returns True if the current platform is Windows and the Python interpreter is 3.5 or earlier; False otherwise.

Return type bool

3.8.4 Menu Borders

3.8.4.1 AsciiBorderStyle

class consolemenu.format.AsciiBorderStyle

A Menu Border Style using only ASCII characters.

3.8.4.2 DoubleLineStyle

class consolemenu.format.DoubleLineStyle

MenuBorderStyle class using “double-line” box drawing characters.

3.8.4.3 DoubleLineOuterLightInnerBorderStyle

class `consolemenu.format.DoubleLineOuterLightInnerBorderStyle`

MenuBarStyle class using Unicode “double-line” box drawing characters for the outer borders, and “light” box drawing characters for the inner borders.

3.8.4.4 HeavyBorderStyle

class `consolemenu.format.HeavyBorderStyle`

MenuBarStyle class using Unicode “heavy” box drawing characters.

3.8.4.5 HeavyOuterLightInnerBorderStyle

class `consolemenu.format.HeavyOuterLightInnerBorderStyle`

MenuBarStyle class using Unicode “heavy” box drawing characters for the outer borders, and “light” box drawing characters for the inner borders.

3.8.4.6 LightBorderStyle

class `consolemenu.format.LightBorderStyle`

MenuBarStyle class using Unicode “light” box drawing characters.

3.8.5 MenuMargins

class `consolemenu.format.MenuMargins` (*top=1, left=2, bottom=0, right=2*)

Class for menu margins. A margin is the area between the maximum specified dimensions (which is usually the width and height of the screen) and the menu border.

Parameters

- **top** (*int*) – The top margin.
- **left** (*int*) – The left margin.
- **bottom** (*int*) – The bottom margin.
- **right** (*int*) – The right margin.

bottom

The bottom margin.

Returns The bottom margin.

Return type `int`

left

The left margin.

Returns The left margin.

Return type `int`

right

The right margin.

Returns The right margin.

Return type `int`

top

The top margin.

Returns The top margin.

Return type int

3.8.6 MenuPadding

class consolemenu.format.**MenuPadding** (*top=1, left=2, bottom=1, right=2*)

Class for menu padding. Padding is the area between the menu border and the content of the menu.

Parameters

- **top** (*int*) – The top padding.
- **left** (*int*) – The left padding.
- **bottom** (*int*) – The bottom padding.
- **right** (*int*) – The right padding.

bottom

The bottom padding.

Returns The bottom padding.

Return type int

left

The left padding.

Returns The left padding.

Return type int

right

The right padding.

Returns The right padding.

Return type int

top

The top padding.

Returns The top padding.

Return type int

3.8.7 MenuStyle

class consolemenu.format.**MenuStyle** (*margins=None, padding=None, border_style=None, border_style_type=None, border_style_factory=None*)

Class for specifying all menu styling, such as margins, padding, and border style.

Parameters

- **margins** (*MenuMargins*) – The menu margin settings.
- **padding** (*MenuPadding*) – The menu padding.
- ((*border_style*) – obj:MenuBorderStyle): The menu border style. Takes precedence over *border_style_type* if both are specified.

- **border_style_type** (*int*) – The border style type as defined by *MenuBorderStyleType*.
- **border_style_factory** (*MenuBorderStyleFactory*) – The factory instance to use to create the borders.

border_style

The border style instance.

Returns The *MenuBorderStyle* instance.

Return type *MenuBorderStyle*

border_style_factory

The border style factory instance.

Returns The *MenuBorderStyleFactory* instance.

Return type *MenuBorderStyleFactory*

margins

The margins instance.

Returns The *MenuMargins* instance.

Return type *MenuMargins*

padding

The padding instance.

Returns The *MenuPadding* instance.

Return type *MenuPadding*

3.9 Prompts

3.9.1 PromptFormatter

class `consolemenu.prompt_utils.PromptFormatter`

Class for formatting a text input prompt, to allow overriding the message as desired.

Default answers will appear in [square brackets] and allow the user to return that answer by simply pressing the Enter button.

If a 'Quit' option is desired, set *enable_quit* to True and provide a *quit_string* (default is 'q') and a *quit_message* (default is '(enter q to Quit)').

static format_prompt (*prompt=None, default=None, enable_quit=False, quit_string='q', quit_message='(enter q to Quit)')*

Format the message presented to the user during input prompting.

Parameters

- **prompt** (*str*) – The message to ask the user.
- **default** (*str, optional*) – The default answer if user does not provide explicit input.
- **enable_quit** (*bool, optional*) – Flag to determine whether a Quit option will be presented.
- **quit_string** (*str, optional*) – The string the user must input to quit (default is 'q').

- **quit_message** (*str*, *optional*) – The message to the user explaining how to Quit.

Returns The formatted prompt string.

Return type str

3.9.2 PromptUtils

class consolemenu.prompt_utils.**PromptUtils** (*screen*, *prompt_formatter=None*)

Utility class with various routines for prompting for user input.

Creates a new instance of ConsoleUtils with the specified console. If no console was specified, creates a new default console using the ConsoleFactory.

Parameters

- **screen** (*consolemenu.screen.Screen*) – The Screen instance.
- **prompt_formatter** (*PromptFormatter*, *optional*) – The instance of PromptFormatter for displaying the prompt.

clear ()

Clear the screen.

confirm_answer (*answer*, *message=None*)

Prompts the user to confirm a question with a yes/no prompt. If no message is specified, the default message is: “You entered { }. Is this correct?”

Parameters

- **answer** (*str*) – The answer to confirm.
- **message** (*str*, *optional*) – Optional message if a different confirmation prompt is desired.

Returns True if the user confirmed Yes, or False if user specified No.

Return type bool

enter_to_continue (*message=None*)

A console prompt to ask the user to ‘Press [Enter] to continue’.

Parameters **message** (*str*, *optional*) – A message to display in place of the default.

input (*prompt=None*, *default=None*, *validators=None*, *enable_quit=False*, *quit_string='q'*, *quit_message='(enter q to Quit)'*)

Generic prompt the user for input.

Parameters

- **prompt** (*str*) – The message to prompt the user.
- **default** (*str*, *optional*) – The default value to suggest as an answer.
- **validators** (*BaseValidator*, *optional*) – The list of validators to perform input validation.
- **enable_quit** (*bool*, *optional*) – Specifies whether the user can cancel out of the input prompt.
- **quit_string** (*str*, *optional*) – The string which the user must input in order to quit.
- **quit_message** (*str*, *optional*) – The message to explain how to quit.

Returns an InputResult tuple.

Return type InputResult

input_password (*message=None*)

Prompt the user for a password or other confidential data.

This is equivalent to the input() method, but does not echo inputted characters to the screen.

Parameters **message** (*str*) – The prompt message.

Returns The password provided by the user.

Return type str

printf (**args*)

Prints the specified arguments to the screen.

Parameters ***args** – Variable length argument list.

println (**args*)

Prints the specified arguments to the screen, followed by a newline character.

Parameters ***args** – Variable length argument list.

prompt_and_confirm_password (*message*)

Prompt for a password using the given message, then prompt a second time for a confirmation password, and verify both provided passwords match. If the passwords do not match, an error is displayed, “Passwords do not match”, and the user must input both passwords again.

Parameters **message** (*str*) – The prompt message.

Returns The password.

Return type str

prompt_for_bilateral_choice (*prompt, option1, option2*)

Prompt the user for a response that must be one of the two supplied choices.

NOTE: The user input verification is case-insensitive, but will return the original case provided by the given options.

Parameters

- **prompt** (*str*) – The prompt to present the choices to the user.
- **option1** (*str*) – The first option.
- **option2** (*str*) – The second option.

Returns The choice selected by the user.

Return type str

prompt_for_numbered_choice (*choices, title=None, prompt='>'*)

Displays a numbered vertical list of choices from the provided list of strings.

Parameters

- **choices** (*list of str*) – The list of choices to display.
- **title** (*str, optional*) – Optional title to display above the numbered list.
- **prompt** (*str*) – The prompt string. Default is “>”.

Returns The index of selected choice.

Return type int

prompt_for_trilateral_choice (*prompt, option1, option2, option3*)

Prompt the user for a response that must be one of the three supplied choices.

NOTE: The user input verification is case-insensitive, but will return the original case provided by the given options.

Parameters

- **prompt** (*str*) – The prompt to present the choices to the user.
- **option1** (*str*) – The first option.
- **option2** (*str*) – The second option.
- **option3** (*str*) – The third option.

Returns The choice selected by the user.

Return type str

prompt_for_yes_or_no (*prompt*)

Prompts the user with the specified question, and expects a yes (y) or no (n) response, returning a boolean value representing the user's answer.

Parameters **prompt** (*str*) – The prompt to display to the user.

Returns True for yes, False for no.

Return type bool

screen

The Screen instance.

Type `consolemenu.screen.Screen`

validate_input (*input_string, validators*)

Validate the given input string against the specified list of validators.

Parameters

- **input_string** (*str*) – The input string to verify.
- **validators** (list of `BaseValidator`) – The list of validators.

Returns The validation result. True if the input is valid; False otherwise.

Return type bool

Raises `InvalidValidator` – If the list of validators contains an invalid `BaseValidator` class.

3.9.3 UserQuit

class `consolemenu.prompt_utils.UserQuit`

Exception raised when a user chooses to Quit from an input prompt.

CHAPTER 4

Indices and tables

- `genindex`
- `modindex`
- `search`

A

action() (*consolemenu.items.CommandItem* method), 12

action() (*consolemenu.items.FunctionItem* method), 13

action() (*consolemenu.items.MenuItem* method), 13

action() (*consolemenu.items.SubmenuItem* method), 14

add_exit() (*consolemenu.ConsoleMenu* method), 8

append_item() (*consolemenu.ConsoleMenu* method), 8

append_item() (*consolemenu.MultiSelectMenu* method), 11

ASCII_BORDER (*consolemenu.format.MenuBorderStyleType* attribute), 18

B

border_style (*consolemenu.format.MenuStyle* attribute), 23

border_style (*consolemenu.menu_component.MenuComponent* attribute), 16

border_style_factory (*consolemenu.format.MenuStyle* attribute), 23

bottom (*consolemenu.format.MenuMargins* attribute), 21

bottom (*consolemenu.format.MenuPadding* attribute), 22

bottom_left_corner (*consolemenu.format.MenuBorderStyle* attribute), 18

bottom_right_corner (*consolemenu.format.MenuBorderStyle* attribute), 18

C

calculate_border_width() (*consolemenu.menu_component.MenuComponent* method), 16

calculate_content_width() (*consolemenu.menu_component.MenuComponent* method), 16

clean_up() (*consolemenu.items.ExternalItem* method), 12

clean_up() (*consolemenu.items.FunctionItem* method), 13

clean_up() (*consolemenu.items.MenuItem* method), 13

clean_up() (*consolemenu.items.SubmenuItem* method), 14

clear() (*consolemenu.prompt_utils.PromptUtils* method), 24

clear() (*consolemenu.screen.Screen* static method), 15

CommandItem (*class in consolemenu.items*), 12

confirm_answer() (*consolemenu.prompt_utils.PromptUtils* method), 24

ConsoleMenu (*class in consolemenu*), 7

create_ascii_border() (*consolemenu.format.MenuBorderStyleFactory* method), 19

create_border() (*consolemenu.format.MenuBorderStyleFactory* method), 19

create_doubleline_border() (*consolemenu.format.MenuBorderStyleFactory* method), 19

create_doubleline_outer_light_inner_border() (*consolemenu.format.MenuBorderStyleFactory* method), 19

create_heavy_border() (*consolemenu.format.MenuBorderStyleFactory* method), 20

create_heavy_outer_light_inner_border() (*consolemenu.format.MenuBorderStyleFactory* method), 20

create_light_border() (*consolemenu.format.MenuBorderStyleFactory* method), 20

method), 20
 current_item (consolemenu.ConsoleMenu attribute), 8
 current_option (consolemenu.ConsoleMenu attribute), 8
 currently_active_menu (consolemenu.ConsoleMenu.cls attribute), 7

D

Dimension (class in consolemenu.menu_component), 16
 DOUBLE_LINE_BORDER (consolemenu.format.MenuBorderStyleType attribute), 19
 DOUBLE_LINE_OUTER_LIGHT_INNER_BORDER (consolemenu.format.MenuBorderStyleType attribute), 19
 draw () (consolemenu.ConsoleMenu method), 9

E

enter_to_continue () (consolemenu.prompt_utils.PromptUtils method), 24
 exit () (consolemenu.ConsoleMenu method), 9
 ExitItem (class in consolemenu.items), 12
 ExternalItem (class in consolemenu.items), 12

F

flush () (consolemenu.screen.Screen static method), 15
 format_prompt () (consolemenu.prompt_utils.PromptFormatter static method), 23
 FunctionItem (class in consolemenu.items), 13

G

generate () (consolemenu.menu_component.MenuComponent method), 16
 get_input () (consolemenu.ConsoleMenu method), 9
 get_return () (consolemenu.items.CommandItem method), 12
 get_return () (consolemenu.items.FunctionItem method), 13
 get_return () (consolemenu.items.MenuItem method), 13
 get_return () (consolemenu.items.SelectionItem method), 14
 get_return () (consolemenu.items.SubmenuItem method), 15
 get_selection () (consolemenu.SelectionMenu class method), 10
 get_submenu () (consolemenu.items.SubmenuItem method), 15

go_down () (consolemenu.ConsoleMenu method), 9
 go_to () (consolemenu.ConsoleMenu method), 9
 go_up () (consolemenu.ConsoleMenu method), 9

H

HEAVY_BORDER (consolemenu.format.MenuBorderStyleType attribute), 18
 HEAVY_OUTER_LIGHT_INNER_BORDER (consolemenu.format.MenuBorderStyleType attribute), 19

I

inner_horizontal (consolemenu.format.MenuBorderStyle attribute), 18
 inner_horizontal_border () (consolemenu.menu_component.MenuComponent method), 16
 inner horizontals () (consolemenu.menu_component.MenuComponent method), 17
 inner_vertical (consolemenu.format.MenuBorderStyle attribute), 18
 input () (consolemenu.prompt_utils.PromptUtils method), 24
 input () (consolemenu.screen.Screen method), 15
 input_password () (consolemenu.prompt_utils.PromptUtils method), 25
 intersection (consolemenu.format.MenuBorderStyle attribute), 18
 is_alive () (consolemenu.ConsoleMenu method), 9
 is_running () (consolemenu.ConsoleMenu method), 9
 is_win_python35_or_earlier () (consolemenu.format.MenuBorderStyleFactory static method), 20
 items (consolemenu.ConsoleMenu attribute), 8

J

join () (consolemenu.ConsoleMenu method), 8

L

left (consolemenu.format.MenuMargins attribute), 21
 left (consolemenu.format.MenuPadding attribute), 22
 LIGHT_BORDER (consolemenu.format.MenuBorderStyleType attribute), 18

M

margins (consolemenu.format.MenuStyle attribute), 23

margins (*consolemenu.menu_component.MenuComponent* attribute), 17
 max_dimension (*consolemenu.menu_component.MenuComponent* attribute), 17
 MenuBorderStyle (*class in consolemenu.format*), 18
 MenuBorderStyleFactory (*class in consolemenu.format*), 19
 MenuBorderStyleType (*class in consolemenu.format*), 18
 MenuComponent (*class in consolemenu.menu_component*), 16
 MenuItem (*class in consolemenu.items*), 13
 MenuMargins (*class in consolemenu.format*), 21
 MenuPadding (*class in consolemenu.format*), 22
 MenuStyle (*class in consolemenu.format*), 22
 MultiSelectMenu (*class in consolemenu*), 11

O

outer_horizontal (*consolemenu.format.MenuBorderStyle* attribute), 18
 outer_horizontal_border_bottom() (*consolemenu.menu_component.MenuComponent* method), 17
 outer_horizontal_border_top() (*consolemenu.menu_component.MenuComponent* method), 17
 outer_horizontal_inner_down (*consolemenu.format.MenuBorderStyle* attribute), 18
 outer_horizontal_inner_up (*consolemenu.format.MenuBorderStyle* attribute), 18
 outer horizontals() (*consolemenu.menu_component.MenuComponent* method), 17
 outer_vertical (*consolemenu.format.MenuBorderStyle* attribute), 18
 outer_vertical_inner_left (*consolemenu.format.MenuBorderStyle* attribute), 18
 outer_vertical_inner_right (*consolemenu.format.MenuBorderStyle* attribute), 18

P

padding (*consolemenu.format.MenuStyle* attribute), 23
 padding (*consolemenu.menu_component.MenuComponent* attribute), 17
 parent (*consolemenu.ConsoleMenu* attribute), 8
 pause() (*consolemenu.ConsoleMenu* method), 9

previous_active_menu (*consolemenu.ConsoleMenu* attribute), 8
 printf() (*consolemenu.prompt_utils.PromptUtils* method), 25
 printf() (*consolemenu.screen.Screen* static method), 15
 println() (*consolemenu.prompt_utils.PromptUtils* method), 25
 println() (*consolemenu.screen.Screen* static method), 15
 process_user_input() (*consolemenu.ConsoleMenu* method), 9
 process_user_input() (*consolemenu.MultiSelectMenu* method), 11
 prompt_and_confirm_password() (*consolemenu.prompt_utils.PromptUtils* method), 25
 prompt_for_bilateral_choice() (*consolemenu.prompt_utils.PromptUtils* method), 25
 prompt_for_numbered_choice() (*consolemenu.prompt_utils.PromptUtils* method), 25
 prompt_for_trilateral_choice() (*consolemenu.prompt_utils.PromptUtils* method), 25
 prompt_for_yes_or_no() (*consolemenu.prompt_utils.PromptUtils* method), 26
 PromptFormatter (*class in consolemenu.prompt_utils*), 23
 PromptUtils (*class in consolemenu.prompt_utils*), 24

R

remove_exit() (*consolemenu.ConsoleMenu* method), 9
 resume() (*consolemenu.ConsoleMenu* method), 9
 right (*consolemenu.format.MenuMargins* attribute), 21
 right (*consolemenu.format.MenuPadding* attribute), 22
 row() (*consolemenu.menu_component.MenuComponent* method), 17

S

Screen (*class in consolemenu.screen*), 15
 screen (*consolemenu.prompt_utils.PromptUtils* attribute), 26
 screen_height (*consolemenu.screen.Screen* attribute), 15
 screen_width (*consolemenu.screen.Screen* attribute), 15
 select() (*consolemenu.ConsoleMenu* method), 9
 selected_item (*consolemenu.ConsoleMenu* attribute), 8

`selected_option` (*consolemenu.ConsoleMenu* attribute), 8
`SelectedItem` (*class in consolemenu.items*), 14
`SelectionMenu` (*class in consolemenu*), 10
`set_menu()` (*consolemenu.items.SubmenuItem* method), 15
`set_up()` (*consolemenu.items.ExternalItem* method), 12
`set_up()` (*consolemenu.items.MenuItem* method), 13
`set_up()` (*consolemenu.items.SubmenuItem* method), 15
`show()` (*consolemenu.ConsoleMenu* method), 8
`show()` (*consolemenu.items.ExitItem* method), 12
`show()` (*consolemenu.items.MenuItem* method), 13
`start()` (*consolemenu.ConsoleMenu* method), 8
`style` (*consolemenu.menu_component.MenuComponent* attribute), 17
`SubmenuItem` (*class in consolemenu.items*), 14

T

`top` (*consolemenu.format.MenuMargins* attribute), 21
`top` (*consolemenu.format.MenuPadding* attribute), 22
`top_left_corner` (*consolemenu.format.MenuBorderStyle* attribute), 18
`top_right_corner` (*consolemenu.format.MenuBorderStyle* attribute), 18

U

`UserQuit` (*class in consolemenu.prompt_utils*), 26

V

`validate_input()` (*consolemenu.prompt_utils.PromptUtils* method), 26

W

`wait_for_start()` (*consolemenu.ConsoleMenu* method), 9