ruben-console-utilities

Release 0.3.0

Nov 28, 2022

Contents

1 Features **3**

2 Goals **5**

3 Installation **7**

4 Usage **9**

5 Documentation **11**

5.1 API Reference 11

Python Module Index **17**

Index **19**

This is a package containing utility functions for command-line applications.

Warning: As this library is still in version 0.y.z, keep in mind that the API may change at any time. See Item 4 of the Semantic Versioning Specification: https://semver.org/#spec-item-4.

CHAPTER 1

Features

- Inputting an integer, float or boolean
- Selecting from a list of options (by entering an integer or character)

CHAPTER 2

Goals

- **Simplicity**: Straightforward, high-level functions with sensible defaults. Most parameters are optional, so you can hit the ground running.
- **Flexible**: An extensive range of options are provided, making the library useful in a wide range of situations.

CHAPTER 3

Installation

Install as usual:

```
pip install ruben-console-utilities
```

You may wish to create a virtual environment beforehand.

CHAPTER 4

Usage

Here a few examples:

```
>>> import consoleutilities as cu
>>> cu.input_option_int(["Export as PDF", "Export as HTML", "Export as TeX"])
[0]: Export as PDF
[1]: Export as HTML
[2]: Export as TeX
Enter integer [0..2]: >? 0
0
>>> cu.input_int("Pick a number between 1 and 10: ", 1, 10, include_max=True)
Pick a number between 1 and 10: >? 8
8
>>> cu.input_boolean("Specify whether to trust this host")
Specify whether to trust this host [y/N]: >? abc123
False
>>> cu.input_boolean("Specify whether to enable HTTPS", default=True)
Specify whether to enable HTTPS [Y/n]: >?
True
```

CHAPTER 5

Documentation

5.1 API Reference

5.1.1 console_utilities Module

`console_utilities.`**`input_boolean`**(*prompt: str*, *default: Optional[bool] = False*, *error_message: Optional[str] = None*, *true_string: str = 'y'*, *false_string: str = 'n'*) → bool

Allows the user to input a boolean.

Specifically, this function does the following:

1. Prints the prompt
2. Waits for the user to enter a value
3. Checks the value is valid (i.e. equal to `true_string` or `false_string`).
4. If valid, the corresponding boolean value is returned (i.e. `True` for `true_string` and `False` for `false_string`). If not valid, the default value will be returned. If not valid and a default has not been provided, the error message is printed and the function loops until a valid value is entered.

Note that case is ignored.

Parameters

- **`prompt`** – Prompt message to print on waiting for input. If `None`, the default prompt will be used.
- **`default`** – Specifies a default value to use on entering an invalid value (i.e. a value other than `true_string` or `false_string`). If `True`, an invalid value will be considered `True`. If `False`, an invalid value will be considered `False`. If `None`, the user must enter either `true_string` or `false_string`.
- **`error_message`** – Error message to print on entering an invalid value. If `None`, the default error message will be used. Ignored if `default` is not `None`.
- **`true_string`** – The string used to represent `True`.

- **false_string** – The string used to represent `False`.

Returns The boolean value entered by the user.

`console_utilities.`**`input_float`**(*prompt: Optional[str] = None*, *min_value: Optional[float] = None*, *max_value: Optional[float] = None*, *error_message: Optional[str] = None*, *include_min: bool = True*, *include_max: bool = False*, *default: Optional[float] = None*) → float

Allows the user to input a float, possibly within a specified range.

Specifically, this function does the following:

1. Prints the prompt
2. Waits for the user to enter a value
3. Checks the value is valid (i.e. is a valid float and in the specified range, if applicable).
4. If valid, the value is returned. If not valid, the error message is printed and the function loops until a valid value is entered. Hence, this function should only ever return a valid value.

Parameters

- **prompt** – Prompt message to print on waiting for input. If `None`, the default prompt will be used.
- **min_value** – Minimum value allowed. Inclusive if `include_min` is `True`; exclusive otherwise. Use `None` if a minimum value is not required.
- **max_value** – Maximum value allowed. Inclusive if `include_max` is `True`; exclusive otherwise. Use `None` if a maximum value is not required.
- **error_message** – Error message to print on entering an invalid value. If `None`, the default error message will be used.
- **include_min** – Specifies whether the `min_value` is inclusive. Use `True` for inclusive or `False` for exclusive.
- **include_max** – Specifies whether the `max_value` is inclusive. Use `True` for inclusive or `False` for exclusive.
- **default** – Specifies a default value to return on entering an invalid value. If `None`, the function will loop until a valid value is entered.

Returns The valid float value entered by the user.

`console_utilities.`**`input_int`**(*prompt: Optional[str] = None*, *min_value: Optional[int] = None*, *max_value: Optional[int] = None*, *error_message: Optional[str] = None*, *include_max: bool = False*, *default: Optional[int] = None*) → int

Allows the user to input an integer, possibly within a specified range.

Specifically, this function does the following:

1. Prints the prompt
2. Waits for the user to enter a value
3. Checks the value is valid (i.e. is a valid integer and in the specified range, if applicable).
4. If valid, the value is returned. If not valid, the error message is printed and the function loops until a valid value is entered. Hence, this function should only ever return a valid value.

Parameters

- **prompt** – Prompt message to print on waiting for input. If `None`, the default prompt will be used.
- **min_value** – Minimum value allowed. Inclusive. Use `None` if a minimum value is not required.
- **max_value** – Maximum value allowed. Inclusive if `include_max` is `True`; exclusive otherwise. Use `None` if a maximum value is not required.
- **error_message** – Error message to print on entering an invalid value. If `None`, the default error message will be used.
- **include_max** – Specifies whether the `max_value` is inclusive. Use `True` for inclusive or `False` for exclusive.
- **default** – Specifies a default value to return on entering an invalid value. If `None`, the function will loop until a valid value is entered.

Returns The valid integer value entered by the user.

console_utilities.**input_multiple_int**(*prompt: Optional[str] = None, min_value: Optional[int] = None, max_value: Optional[int] = None, error_message: Optional[str] = None, include_max: bool = False, default: Optional[List[int]] = None, separator: Optional[str] = None, allow_empty: bool = True*) → List[int]

Allows the user to input multiple integers, optionally with each in a specified range.

Parameters

- **prompt** – Prompt message to print on waiting for input. If `None`, the default prompt will be used.
- **min_value** – Minimum value allowed. Inclusive. Use `None` if a minimum value is not required.
- **max_value** – Maximum value allowed. Inclusive if `include_max` is `True`; exclusive otherwise. Use `None` if a maximum value is not required.
- **error_message** – Error message to print on entering an invalid value. If `None`, the default error message will be used.
- **include_max** – Specifies whether the `max_value` is inclusive. Use `True` for inclusive or `False` for exclusive.
- **default** – Specifies a default value to return on entering an invalid value. If `None`, the function will loop until a valid value is entered.
- **separator** – Specifies the separator string to use when splitting the input into multiple values. If `None` or an empty string, then whitespace will be used.
- **allow_empty** – Specifies whether no values may be entered. If `True`, zero or more values may be entered; if `False`, one or more values may be entered.

Returns The valid integer values entered by the user.

console_utilities.**input_multiple_option_int**(*options: List[str], prompt: Optional[str] = None, error_message: Optional[str] = None, separator: Optional[str] = None, allow_empty: bool = True*) → List[int]

Allows the user to select multiple options from a list of options by entering the options' indices.

Parameters

- **options** – List of option names. Example: `["Export to PDF", "Export to HTML"]`.
- **prompt** – Prompt message to print on waiting for input. If `None`, the default prompt will be used.
- **error_message** – Error message to print on entering an invalid value. If `None`, the default error message will be used.
- **separator** – Specifies the separator string to use when splitting the input into multiple values. If `None` or an empty string, then whitespace will be used.
- **allow_empty** – Specifies whether no values may be entered. If `True`, zero or more values may be entered; if `False`, one or more values may be entered.

Returns The indices of the selected options.

`console_utilities.`**`input_option_char`**(*options: List[str], chars: List[str], prompt: str = None, error_message: str = None, ignore_case: bool = True, default: Optional[str] = None*) → str

Allows the user to select from a list of options by entering a character.

Specifically, this function does the following:

1. Displays a list of option names with their associated characters
2. Prints the prompt
3. Waits for the user to enter a value
4. Checks the value is valid (i.e. is a valid integer and in the specified range, if applicable).
5. If valid, the value is returned; else, the error message is printed and this process is repeated until a valid value is entered. Hence, this function should only ever return a valid value.

`options` and `chars` are expected to be lists of the same length. Additionally, `chars` must not contain duplicates, to avoid ambiguity. If `ignore_case` is `True` then case is also ignored when checking for duplicates.

Parameters

- **options** – List of option names. Example: `["Export to PDF", "Export to HTML"]`.
- **chars** – List of characters to associate with each option. Example: `["p", "h"]`.
- **prompt** – Prompt message to print on waiting for input. If `None`, the default prompt will be used.
- **error_message** – Error message to print on entering an invalid value. If `None`, the default error message will be used.
- **ignore_case** – Specifies whether case should be ignored. If `True`, this also means that the characters in `chars` will be output in lower case to reduce confusion for the user. (To me, displaying some characters in upper case and some in lower case implies that case is not ignored.)
- **default** – Specifies a default value to return on entering an invalid value. If `None`, the function will loop until a valid value is entered.

Returns The character corresponding to the selected option.

Todo:

- Replace the `options` and `chars` lists with a single list of objects.

---

console_utilities.**input_option_int**(*options: List[str], prompt: Optional[str] = None, error_message: Optional[str] = None*) → int

Allows the user to select a single option from a list of options by entering the option's index.

Specifically, this function does the following:

1. Displays a list of option names with their indices
2. Waits for the user to input an index (i.e. an integer between 0 and `len(options) - 1`) using *`input_option_int()`*

Parameters

- **options** – List of option names. Example: `["Export to PDF", "Export to HTML"]`.
- **prompt** – Prompt message to print on waiting for input. If `None`, the default prompt will be used.
- **error_message** – Error message to print on entering an invalid value. If `None`, the default error message will be used.
- **default** – Specifies a default value to return on entering an invalid value. If `None`, the function will loop until a valid value is entered.

Returns The index of the selected option.

Python Module Index

C

console_utilities, 11

Index

C

`console_utilities` (*module*), 11

I

`input_boolean()` (*in module console_utilities*), 11
`input_float()` (*in module console_utilities*), 12
`input_int()` (*in module console_utilities*), 12
`input_multiple_int()` (*in module console_utilities*), 13
`input_multiple_option_int()` (*in module console_utilities*), 13
`input_option_char()` (*in module console_utilities*), 14
`input_option_int()` (*in module console_utilities*), 14