
dmenu

Release 0.2.1

Mar 28, 2017

Contents

Python Module Index

3

A dmenu wrapper.

dmenu is a dynamic menu for X, originally designed for dwm. It manages large numbers of user-defined menu items efficiently.

```
dmenu.show(items, command='dmenu', bottom=None, fast=None, case_insensitive=None, lines=None,
            monitor=None, prompt=None, font=None, background=None, foreground=None, back-
            ground_selected=None, foreground_selected=None)
```

Present a dmenu to the user.

Parameters

- **items** (*Iterable[str]*) – defines the menu items being presented to the user. items should not contain the newline character.
- **command** (*Optional[str]*) – defines the path to the dmenu executable. Defaults to 'dmenu'.
- **bottom** (*Optional[bool]*) – dmenu appears at the bottom of the screen.
- **fast** (*Optional[bool]*) – dmenu grabs the keyboard before reading stdin. This is faster, but will lock up X until stdin reaches end-of-file.
- **case_insensitive** (*Optional[bool]*) – dmenu matches menu items case insensitively.
- **lines** (*Optional[int]*) – dmenu lists items vertically, with the given number of lines.
- **monitor** (*Optional[int]*) – dmenu is displayed on the monitor number supplied. Monitor numbers are starting from 0.
- **prompt** (*Optional[str]*) – defines the prompt to be displayed to the left of the input field.
- **font** (*Optional[str]*) – defines the font or font set used. eg. “fixed” or “Monospace-12:normal” (an xft font)
- **background** (*Optional[str]*) – defines the normal background color. #RGB, #RRGGBB, and X color names are supported.
- **foreground** (*Optional[str]*) – defines the normal foreground color.
- **background_selected** (*Optional[str]*) – defines the selected background color.
- **foreground_selected** (*Optional[str]*) – defines the selected foreground color.

Raises

- *DmenuCommandError*
- *DmenuUsageError*

Returns The user’s selected menu item, their own typed item, or None if they hit escape.

Examples

```
>>> import dmenu
```

```
>>> dmenu.show(['a', 'b', 'c'])
'a' # user selected a
```

```
>>> dmenu.show(['a', 'b', 'c'], prompt='pick a letter')
'b' # user selected b
```

```
>>> dmenu.show(['a', 'b', 'c'])
None # user hit escape
```

```
>>> dmenu.show(['a', 'b', 'c'])
'd' # user typed their own selection, d
```

```
>>> dmenu.show(['a', 'b', 'c'], command='not_a_valid_dmenu')
Traceback (most recent call last):
...
dmenu.dmenu.DmenuCommandError: The provided dmenu command could not be used ([
↳'not_a_valid_dmenu']): [Errno 2] No such file or directory: 'not_a_valid_dmenu'
```

```
>>> dmenu.show(['a', 'b', 'c'], monitor=2)
Traceback (most recent call last):
...
dmenu.dmenu.DmenuUsageError: This version of dmenu does not support your usage ([
↳'dmenu', '-m', '2']):
usage: dmenu [-b] [-f] [-i] [-l lines] [-p prompt] [-fn font]
           [-nb color] [-nf color] [-sb color] [-sf color] [-v]
```

Consider configuring show using partial application:

```
>>> import functools
>>> show = functools.partial(dmenu.show, bottom=True)
>>> show(['we', 'show', 'up', 'below'])
>>> show(['us', 'too'])
```

`dmenu.version` (*command='dmenu'*)

The dmenu command's version message.

Raises *DmenuCommandError*

Example

```
>>> import dmenu
>>> dmenu.version()
'dmenu-4.5, © 2006-2012 dmenu engineers, see LICENSE for details'
```

class `dmenu.DmenuError`

The base class for dmenu errors.

class `dmenu.DmenuCommandError` (*args, error*)

The dmenu command failed.

class `dmenu.DmenuUsageError` (*args, usage*)

The dmenu command does not support your usage.

d

dmenu, ??

D

`dmenu` (module), 1

`DmenuCommandError` (class in `dmenu`), 2

`DmenuError` (class in `dmenu`), 2

`DmenuUsageError` (class in `dmenu`), 2

S

`show()` (in module `dmenu`), 1

V

`version()` (in module `dmenu`), 2