

---

# **Black Mamba Documentation**

***Release 1.4.3***

**Robert Vojta**

**Jan 11, 2018**



---

## Contents

---

<b>1</b>	<b>About Black Mamba</b>	<b>3</b>
<b>2</b>	<b>Black Mamba User Guide</b>	<b>5</b>
<b>3</b>	<b>Black Mamba Reference</b>	<b>15</b>
<b>4</b>	<b>Contribution</b>	<b>25</b>
<b>5</b>	<b>Development</b>	<b>27</b>
<b>6</b>	<b>FAQ</b>	<b>29</b>
<b>7</b>	<b>Gallery</b>	<b>31</b>
<b>8</b>	<b>Bundled packages</b>	<b>43</b>
	<b>Python Module Index</b>	<b>47</b>



Welcome stranger. Use one of the following links to continue.



---

## About Black Mamba

---

Black Mamba is Python™ package for [Pythonista](#) application.

Pythonista is great tool, seriously, but it lacks some features like keyboard shortcuts for specific actions. I'm slow without them. So I decided to write set of scripts to *fix* all these issues. To speed up my iteration cycle. To make it as fast as possible. And which snake is the speediest one on the planet? [Black Mamba](#). And now you know why it's called Black Mamba.

For learning how to use Black Mamba, see the [Black Mamba User Guide](#).

For detailed reference documentation of the functions and classes contained in the package, see the [Black Mamba Reference](#).

## 1.1 Related applications

### 1.1.1 Pythonista

Pythonista is a complete development environment for writing Python™ scripts on your iPad or iPhone.

- [Pythonista](#)
- [Pythonista forum](#)

Without the Pythonista there would be no Black Mamba.

Python™ developer? It's a must if you own an iPad.

### 1.1.2 Working Copy

The powerful Git client for iOS that clones, edits, commits, pushes & more.

- [Working Copy](#)

Perfectly cooperates with Drag & Drop script. See [Scripts](#).

### 1.1.3 Dash

Dash gives your iPad and iPhone instant offline access to 150+ API documentation sets.

- [Dash for iOS](#)

Used by Search Dash script. See *Scripts*.

## 1.2 Author

Written in spare time by Robert Vojta. You can find me at:

- [Twitter](#)
- [GitHub](#)
- [Blog](#)

Blog is highly irregular and it's almost one year from my last post. I'm not good at blogging. I always think that I have nothing to say and I blog about things I'm unable to find elsewhere only. Better stick with Twitter or GitHub.



**Release** 1.4.3

**Date** Jan 11, 2018

This guide is intended as an introductory overview of Black Mamba and explains how to install and make use of the most important features of Black Mamba. For detailed reference documentation of the functions and classes contained in the package, see the *Black Mamba Reference*.

## 2.1 Requirements

### 2.1.1 System

- iOS  $\geq$  10
- iPad w/ external keyboard

External keyboard “is a must”. You can use *Scripts* as action items (wrench menu icon), but it’s not as great as with *Keyboard shortcuts*.

---

**Note:** Black Mamba should work on iPhone, but the UI is not optimized for this device at all. You can try it, but it’s not officially supported and never will be.

---

Continue to *Installation*.

## 2.2 Installation

### 2.2.1 Install

```
import requests as r; exec(r.get('http://bit.ly/get-blackmamba').text)
```

Copy the above line, paste it into Pythonista interactive prompt and execute it. Black Mamba will be installed under the `site-packages-3` folder.

### 2.2.2 Update

Black Mamba checks for updates automatically. You can disable check for updates entirely or you can configure time interval. See [Configuration](#) section.

Updates are **not** installed automatically. Black Mamba just informs you about available update. Copy & paste the installation line into Pythonista interactive prompt and execute it. Latest version will be downloaded and installed.

### 2.2.3 PIP

Black Mamba no longer supports installation via `pip`. Latest available release is 1.0.2.

### 2.2.4 Startup

Black Mamba requires from you to put following lines to the `~/Documents/site-packages-3/pythonista_startup.py` file:

```
#!/python3
import blackmamba

blackmamba.main()
```

This is bare minimum. Continue to [Configuration](#) section to learn how to tweak Black Mamba.

## 2.3 Configuration

I strongly suggests to check `config.py:_DEFAULTS` from time to time. Just to check if this documentation isn't outdated.

### 2.3.1 Sections

#### General

Defaults:

```
{
  'general': {
    'jedi': False,
    'register_key_commands': True,
    'page_line_count': 40
  }
}
```

jedi: bool - JEDI is used as a backend for Find usages, Jump to definition and Show documentation *Scripts*. But because JEDI is also used by the Pythonista, JEDI is not thread safe and I don't know when and how it is used, it's disabled by default. You have to set it to True to enable mentioned scripts.

register\_key\_commands: bool - set to False to disable default *Keyboard shortcuts*.

page\_line\_count: int - number of lines to scroll up / down for page up / down.

## Updates

Defaults:

```
'update': {
    'enabled': True,
    'interval': 3600
}
```

enabled: bool - set to False to disable check for updates.

interval: int - check for updates time interval (in seconds).

## File picker

Defaults:

```
'file_picker': {
    'ignore_folders': {
        '': ['.git'],
        '.': ['.Trash', 'Examples',
              'site-packages', 'site-packages-2', 'site-packages-3']
    }
}
```

ignore\_folders: dict - key is a parent directory (not full path, just name) and value is a list of folders to ignore. You can use two special values as keys:

- '' - any parent directory,
- '.' - parent directory is ~/Documents.

Default value says that .git folder inside any folder is ignored. .Trash, Examples, ... folders inside ~/Documents folder are ignored as well.

Affects Open quickly and Run quickly scripts, see *Scripts*.

## Analyzer

Defaults (since 1.1.0):

- ignore\_codes option is removed / ignored
- max\_line\_length option is removed / ignored

```
'analyzer': {
    'hud_alert_delay': 1.0,
    'remove_whitespace': True,
    'flake8': [
        # 1st pass
```

```
[ '--select=E901,E999,F821,F822,F823'],
# 2nd pass
[ '--max-complexity=10', '--max-line-length=127']
]
```

If `flake8` is provided, `ignore_codes` and `max_line_length` options are ignored. `flake8` must contain list of passes and every pass contains list of `flake8` arguments. You can run `flake8` several times with different options in this way.

Defaults (pre 1.1.0):

```
'analyzer': {
    'hud_alert_delay': 1.0,
    'ignore_codes': None,
    'max_line_length': 127,
    'remove_whitespaces': True
}
```

Affects Analyze script, see *Scripts*.

## Tester

Defaults:

```
'tester': {
    'hud_alert_delay': 1.0,
    'hide_console': True
}
```

Affects Run unit tests script, see *Scripts*.

## Drag and Drop

Defaults:

```
'drag_and_drop': {
    'ignore_folders': {
        '': ['.git'],
        '.': ['.Trash', 'Examples',
              'site-packages', 'site-packages-2', 'site-packages-3', 'stash_extensions
↩']
    }
}
```

Affects Drag & Drop script, see *Scripts*.

## Documentation

Defaults:

```
'documentation': {
    'reuse': True,
    'frame': (630, 110, 730, 350)
}
```

`reuse`: `bool` - same overlay view is reused for consequent show documentation calls if set to `True`. Otherwise multiple overlays appear in consequent show documentation calls, but if a symbol's fully qualified name matches existing overlay, this particular overlay is expanded and activated.

`frame`: `tuple(float, float, float, float)` - initial overlay frame in the key window coordinates.

Affects Show documentation script, see *Scripts*.

## 2.3.2 Sample

See `pythonista_startup.py` for more examples.

# 2.4 Keyboard shortcuts

## 2.4.1 General

`Cmd 0` toggles file navigator / library.

## 2.4.2 Dialogs

You can close all dialogs with `Esc` key. Use `Cmd .` if you have a keyboard without `Esc` key.

Dialogs with tables support arrow keys to change selection and `Enter` key to select item.

Some of these dialogs support `Shift Enter` for an alternate action. Example is the Open Quickly script where you can open file in a new tab (`Enter`) or in the current tab (`Shift Enter`).

## 2.4.3 Tabs

`Cmd 1 .. Cmd 9` quickly switches between tabs.

`Ctrl Tab` or `Cmd Shift ]` selects next tab and `Ctrl Shift Tab` or `Cmd Shift [` selects previous tab.

`Cmd W` closes current tab and `Cmd Shift W` closes all tabs except current one.

`Cmd T` creates new empty tab and `Cmd N` shows Pythonista new file dialog.

---

**Note:** Shortcuts `Cmd W`, `Ctrl Tab` and `Ctrl Shift Tab` are no longer registered if you're using Pythonista (> 311013). These three are natively supported. They still do work if you're using older Pythonista version.

---

## 2.4.4 Editor

`Ctrl Up` for page up, `Ctrl Down` for page down. Page line count is configurable, see *Configuration*.

## 2.4.5 Scripts

*Scripts* keyboard shortcuts binding:

Shortcut	Function
Cmd E	Drag & Drop
Cmd /	Toggle comments
Cmd Shift O	Open quickly
Cmd Shift O	Search <a href="#">Dash</a>
Cmd Shift R	Run quickly
Cmd Shift A	Action quickly
Ctrl Shift B	Analyze
Cmd Shift K	Clear annotations
Cmd U	Run unit tests
Cmd Shift L	Outline quickly
Ctrl L	Jump to line
Ctrl Cmd J	Jump to definition
Ctrl Cmd U	Find usages
Ctrl Cmd ?	Show documentation
Ctrl W	Close active overlay <sup>1</sup>
Cmd Option O	Refactor - Organize imports
Cmd Option E	Refactor - Expand star imports
Cmd Option R	Refactor - Rename

Just hold Cmd key and iOS will show you all available shortcuts if you can't remember them.

## 2.4.6 Custom shortcuts

See [blackmamba.uikit.keyboard](#) to learn how to register custom keyboard shortcuts.

## 2.5 Scripts

Following scripts are usable as Pythonista action items (wrench menu icon). In other words, you can use them even without external keyboard.

---

**Note:** Scripts still requires you to call `blackmamba.main()`, see [Startup](#) section.

---

### 2.5.1 Action quickly

Script name `action_quickly.py`.

Shows dialog with user defined action items (wrench icon). You can filter them by title, use arrow keys to change selection and run any of them with Enter key.

Only **user defined action items** are listed.

### 2.5.2 Analyze

Script name `analyze.py`.

---

<sup>1</sup> Show documentation script does use overlays.

Source code analysis with `flake8`. Results are displayed as Pythonista annotations. If there's no error / warning, HUD informs you about that.

This script is configurable, see [Analyzer](#) configuration.

### 2.5.3 Clear annotations

Script name `clear_annotations.py`.

Clears all Pythonista annotations.

### 2.5.4 Close all tabs except current one

Script name `close_all_tabs_except_current_one.py`.

Closes all tabs except current one.

### 2.5.5 Drag & Drop

Script name `drag_and_drop.py`. **iOS >= 11.0 required.**

Shows dialog with opened files and folders tree. You can drag a file / folder to any other iOS application. Works well with [Working Copy](#) and [Kaleidoscope](#) for example.

You can drag a file / folder from any other iOS application as well. But there's one limitation, you can drop them at the folder only.

This script is configurable, see [Drag and Drop](#) configuration.

### 2.5.6 Find usages

Script name `find_usages.py`.

Finds usages of a symbol. If there're no usage, HUD informs you. Otherwise dialog appears where you can select location and scroll to it.

---

**Note:** JEDI must be enabled, see [General](#) configuration.

---

### 2.5.7 Jump to definition

Script name `jump_to_definition.py`.

Jumps to symbol definition. If definition can't be found, HUD informs you. Otherwise it jumps to definition location or shows dialog where you can choose location if more than one definition was found.

---

**Note:** JEDI must be enabled, see [General](#) configuration.

---

### 2.5.8 Jump to line

Script name `jump_to_line.py`.

Shows dialog where you can enter line number to scroll to.

### 2.5.9 New file

Script name `new_file.py`.

If there's no opened file in the Pythonista, new tab is created and *New File...* button tap is emulated.

Otherwise dialog appears where you can enter new file name. File will be created in the current folder. If it already exists, file will be opened instead of creating new one.

### 2.5.10 New tab

Script name `new_tab.py`.

Creates new empty tab.

### 2.5.11 Open quickly

Script name `open_quickly.py`.

Shows dialog with all your files. You can filter these files by directories, file name, etc. Use arrow keys to change selection and then hit `Enter` to open selected file.

If file is already opened, Black Mamba changes selected tab only.

This script is configurable, see *File picker* configuration.

### 2.5.12 Outline quickly

Script name `outline_quickly.py`.

Shows source code outline. You can filter functions, ... by name. Use arrows key to change selection and then hit `Enter` to scroll to the symbol.

### 2.5.13 Run quickly

Script name `run_quickly.py`.

Shows dialog with all your Python files. You can filter these files by directories, file name, etc. Use arrows keys to change selection and then hit `Enter` to run selected file.

This script is configurable, see *File picker* configuration.

### 2.5.14 Run unit tests

Script name `run_unit_tests.py`.

Runs unit tests and show results as Pythonista annotations.

This script is configurable, see *Tester* configuration.



### 2.5.15 Search Dash

Script name `search_dash.py`.

Opens [Dash](#) application with selected text or a symbol around cursor position.

### 2.5.16 Show documentation

Script name `show_documentation.py`.

Shows documentation for the symbol around cursor. If definition can't be found, HUD informs you. If there're more than one definitions, dialog appears where you can select which one to show.

Documentation is displayed as an overlay, which can be closed by the `Ctrl W` shortcut or by tapping on the X button.

This script is configurable, see [Documentation](#) configuration.

---

**Note:** JEDI must be enabled, see [General](#) configuration.

---

### 2.5.17 Toggle comments

Script name `toggle_comments.py`.

Toggles current line / selected lines comments.

### 2.5.18 Refactoring

All refactoring scripts are limited to open single file only.

---

**Note:** It's an experiment. You should use version control system to avoid loosing data.

---

#### Rename

Script name `rename.py`.

Rename identifier.

#### Expand star imports

Script name `expand_star_imports.py`.

Expand star imports.

#### Organize imports

Script name `organize_imports.py`.

Organize imports.

## Futurize

Script name `futurize.py`.

Runs stage 1 of the `futurizer`.

**Release** 1.4.3

**Date** Jan 11, 2018

### 3.1 Versioning

Black Mamba does use [semantic versioning](#) since version 1.0.0. Given a version number `MAJOR.MINOR.PATCH`:

- `MAJOR` version is incremented if there're incompatible API changes,
- `MINOR` version is incremented if there's a new functionality in a backwards-compatible manner,
- `PATCH` version is incremented if there're backwards-compatible bug fixes.

Black Mamba does not use additional labels for pre-release and build metadata as extensions to the `MAJOR.MINOR.PATCH` format.

### 3.2 Modules

Following modules are considered to be the API for Black Mamba from the versioning point of view. Black Mamba includes lot of other modules, but it's not recommended to use them, because they can break even if `MAJOR` is not incremented.

#### 3.2.1 `blackmamba`

Black Mamba initialization module.

Module is Python 3 compatible only.

The only requirement is to call `blackmamba.main()`. Example:

```
import blackmamba
blackmamba.main()
```

**Warning:** Do not add top level imports which depends on Pythonista modules. Module must be importable on any other platform. Add these imports to specific functions decorated with `blackmamba.system.Pythonista` and / or `blackmamba.system.iOS`.

## Reference

`blackmamba.main (config=None)`

Black Mamba initialization.

Call this function from `pythonista_startup.py` (`site-packages-3`) file.

**Parameters** `config` – Optional dictionary, see [Configuration](#)

Example:

```
import blackmamba

config = {
    'general': {
        'jedi': True
    }
}

blackmamba.main(config)
```

See `pythonista_startup.py` for more examples.

### 3.2.2 blackmamba.log

Logging module.

Why custom module instead of `logging`? Several reasons:

- not to interfere with Pythonista logging settings,
- unable to convince Pythonista to use my colors,
- etc.

Default log level is `INFO`. You can use `blackmamba.log.set_level()` to change effective log level. Available log levels are:

- `ERROR`
- `WARNING`
- `INFO`
- `DEBUG`
- `NOTSET`

If you'd like to silent Black Mamba messages, it's recommended to set log level to `ERROR`.

```
import blackmamba.log as log

log.set_level(log.ERROR)
```

**Warning:** This module must not introduce dependency on any other Black Mamba modules and must be importable on any other platform.

## Reference

`blackmamba.log.get_level()`

Return effective log level.

**Returns** Effective log level

`blackmamba.log.set_level(level)`

Set effective log level.

**Parameters** `level` – Level to set

`blackmamba.log.debug(*args, **kwargs)`

Log message with DEBUG level.

**Parameters**

- **args** – Passed to `print()`
- **kwargs** – Passed to `print()`

`blackmamba.log.info(*args, **kwargs)`

Log message with INFO level.

**Parameters**

- **args** – Passed to `print()`
- **kwargs** – Passed to `print()`

`blackmamba.log.warn(*args, **kwargs)`

Log message with WARNING level.

**Parameters**

- **args** – Passed to `print()`
- **kwargs** – Passed to `print()`

`blackmamba.log.error(*args, **kwargs)`

Log message with ERROR level.

**Parameters**

- **args** – Passed to `print()`
- **kwargs** – Passed to `print()`

### 3.2.3 blackmamba.system

System info and decorators.

**Warning:** This module must not introduce dependency on any other Black Mamba modules and must be importable on any other platform as well.

## Reference

`blackmamba.system.PYTHONISTA = False`

bool: True if we're running within Pythonista or False.

`blackmamba.system.IOS = False`

bool: True if we're running within iOS or False.

`blackmamba.system.PYTHONISTA_VERSION = None`

str: Pythonista version or None if we're not within Pythonista.

`blackmamba.system.PYTHONISTA_BUNDLE_VERSION = None`

int: Pythonista bundle version or None if we're not within Pythonista.

`blackmamba.system.PYTHONISTA_VERSION_TUPLE = None`

tuple(int): Pythonista version tuple (3, 1, 1) or None if we're not within Pythonista.

`blackmamba.system.IOS_VERSION = None`

str: iOS version or None if we're not within iOS.

`blackmamba.system.IOS_VERSION_TUPLE = None`

tuple(int): iOS version tuple (11, 0) or None if we're not within iOS.

**class** `blackmamba.system.iOS` (*from\_version=None, to\_version=None*)

Decorator to execute function under specific iOS versions.

Return value is return value of decorated function or None if iOS condition isn't met.

Examples:

Run function only within any iOS version:

```
@iOS()
def run_me():
    pass
```

Run function only within iOS >= 11.0:

```
@iOS('11.0') # or @iOS(from_version='11.0')
def run_me():
    pass
```

Run function only within iOS <= 11.0:

```
@iOS(None, '11.0') # or @iOS(to_version='11.0')
def run_me():
    pass
```

**class** `blackmamba.system.Pythonista` (*from\_version=None, to\_version=None, appex=None*)

Decorator to execute function under specific Pythonista versions.

By default, function is not executed under application extension. You have to pass `appex=True` if you'd like to run some function under appex as well.

Return value is return value of decorated function or None if Pythonista condition isn't met.

Examples:

Run function only within any Pythonista version:

```
@Pythonista()
def run_me():
    pass
```

Run function only within any Pythonista version and allow appex:

```
@Pythonista(appex=True)
def run_me():
    pass
```

Run function only within any Pythonista version and disallow appex:

```
@Pythonista(appex=False)
def run_me():
    pass
```

Run function only within Pythonista >= 3.1.1:

```
@Pythonista('3.1.1') # or @Pythonista(from_version='3.1.1')
def run_me():
    pass
```

Run function only within Pythonista <= 3.2:

```
@Pythonista(None, '3.2') # or @Pythonista(to_version='3.2')
def run_me():
    pass
```

`blackmamba.system.catch_exceptions(func)`

Decorator catching all `Exception` exceptions and writing info to console.

Use this decorator for functions handling keyboard shortcuts, keyboard events, ... to avoid Pythonista crash.

**Parameters** `func` – Function to decorate

**Returns** Return value of `func`

### 3.2.4 blackmamba.uikit.keyboard

Keyboard shortcuts.

#### Key commands

Key command is kind of global shortcut in Pythonista. They're visible if you hold down `Cmd` key for example. Black Mamba does use key commands for shortcuts like toggle comments, etc.

Related functions:

- `UIKeyModifier`
- `UIKeyInput`
- `register_key_command()`

Following example shows how to print *Hallo* with `Cmd H` keyboard shortcut.

```
from blackmamba.uikit.keyboard import (
    register_key_command, UIKeyModifier
)

def hallo():
    print('Hallo')

register_key_command(
    'h',
    UIKeyModifier.command,
    hallo,
    'Print Hallo' # Optional discoverability title (hold down Cmd)
)
```

## Key event handlers

Key event handler is kind of local keyboard shortcut. They're not visible if you hold down Cmd key. Designed to be used in custom dialogs.

Related functions:

- `UIKeyModifier`
- `UIEventKeyCode`
- `register_key_event_handler()`
- `unregister_key_event_handler()`

Following example shows how to close dialog with Cmd . keyboard shortcut.

```
from blackmamba.uikit.keyboard import (
    register_key_event_handler, UIEventKeyCode, UIKeyModifier,
    unregister_key_event_handlers
)

class MyView(ui.View):
    def __init__(self, **kwargs):
        super().__init__(**kwargs)

        def close_me():
            self.close()

        self._handlers = [
            register_key_event_handler(UIEventKeyCode.dot, close_me,
↪modifier=UIKeyModifier.command)
        ]

    def will_close(self):
        unregister_key_event_handlers(self._handlers)
```

## Reference

`blackmamba.uikit.keyboard.is_in_hardware_keyboard_mode()`

Check if HW keyboard is connected.

**Returns** True if HW keyboard is connected otherwise False



**class** blackmamba.uikit.keyboard.UIKeyModifier

Key modifiers enumeration.

Modifiers can be combined like:

```
UIKeyModifier.command | UIKeyModifier.shift
```

See also:

- `register_key_command()`
- `register_key_event_handler()`

**none = 0**

No modifier key

**alphaShift = 65536**

Caps Lock key

**shift = 131072**

Shift key

**control = 262144**

Control key

**alternate = 524288**

Option key

**command = 1048576**

Command key

**numericPad = 2097152**

Key is on numeric pad

**class** blackmamba.uikit.keyboard.UIEventKeyCode

Event key codes.

Not all key codes are listed / included here. Feel free to create pull request with more key codes if you'd like to use them.

See also:

- `register_key_event_handler()`

**right = 79**

Right arrow key

**left = 80**

Left arrow key

**down = 81**

Down arrow key

**up = 82**

Up arrow key

**enter = 40**

Enter key

**space = 44**

Space key

**backspace = 42**

Backspace key

**escape** = 41  
Escape key

**dot** = 55  
Dot key

**class** blackmamba.uikit.keyboard.**UIKeyInput**  
Enumeration of special key input values.

See also:

- `register_key_command()`

**leftArrow** = 'UIKeyInputLeftArrow'  
Left arrow

**rightArrow** = 'UIKeyInputRightArrow'  
Right arrow

**upArrow** = 'UIKeyInputUpArrow'  
Up arrow

**downArrow** = 'UIKeyInputDownArrow'  
Down arrow

blackmamba.uikit.keyboard.**register\_key\_command**(*input*, *modifier\_flags*, *function*, *title=None*)

Register key command.

---

**Note:** There's no function to unregister key commands.

---

#### Parameters

- **input** – str or *UIKeyInput*
- **modifier\_flags** – *UIKeyModifier* or int
- **function** – Function to call
- **title** – Optional discoverability title

**Returns** True if key command was registered otherwise False

blackmamba.uikit.keyboard.**register\_key\_event\_handler**(*key\_code*, *func*, \*, *modifier=<UIKeyModifier.none: 0>*)

Register key event handler.

Usable in dialogs for example. Do not forget to unregister key event handler in `will_close` function of your `ui.View`.

#### Parameters

- **key\_code** – *UIEventKeyCode* or int
- **func** – Function to call
- **modifier** – *UIKeyModifier* or int

**Returns** Handler to use in `unregister_key_event_handler()`

`blackmamba.uikit.keyboard.unregister_key_event_handler(handler)`

Unregister key event handler.

It is safe to call this function multiple times with the same handler. Handler is silently ignored if it's not registered.

**Parameters** `handler` – Handler from `register_key_event_handler()`

`blackmamba.uikit.keyboard.unregister_key_event_handlers(handlers)`

Unregister list of key event handlers.

Convenience function, it just calls `unregister_key_event_handler()` for every handler.

**Parameters** `handlers` – List of handlers



There are several areas where you can help.

### 4.1 Documentation

I'm not native English speaker and some of my sentences sound weird. Fork [blackmamba](#) repository, update [documentation](#) and open pull request. Will happily merge it.

Documentation does use [reStructuredText](#). Here's [quick start](#) guide.

Why not Markdown? [reStructuredText](#) is more powerful and allows me to use links, references and other stuff.

### 4.2 Testing

If you find a bug, please do not hesitate to [file an issue](#). I'm unable to test everything.

### 4.3 Development

Black Mamba is developed in my spare time. I have a full time job, love it and I don't have much time to enhance it. If you'd like to help, see [Development](#).

### 4.4 Questions

Do you have a question? [File an issue](#), I'll add `question` tag and will answer it. Or you can send me direct message on [Twitter](#).



### 5.1 Style

There's no defined style for development. Just stick with the same style you see in other blackmamba modules.

### 5.2 Pull requests

Pull requests must pass flake8 checks:

```
flake8 . --count --select=E901,E999,F821,F822,F823 --show-source --statistics
flake8 . --count --max-complexity=10 --max-line-length=127 --statistics
```

Pull requests must pass tests:

```
PYTHONPATH=. pytest tests
```

Consult [.travis.yml](#) for more details.

### 5.3 Existing issues

Let me know if you'd like to work on [something](#). To avoid situation where I already started working on it.

### 5.4 New ideas

I'm open to new ideas as well. Please, [file an issue](#) first, we can discuss it and then you can implement it.





## 6.1 General

### 6.1.1 Are you affiliated with Pythonista?

No. It's a completely separate project. Although I discuss some of my ideas with [Ole](#) (Pythonista author) from time to time.

### 6.1.2 What if some feature will be added to Pythonista?

I'm perfectly okay with this and will be happy if something from Black Mamba will be natively supported by the Pythonista. That's also the reason why I filled [so many issues](#).

These features will be removed from Black Mamba after some period if this happens.



## CHAPTER 7

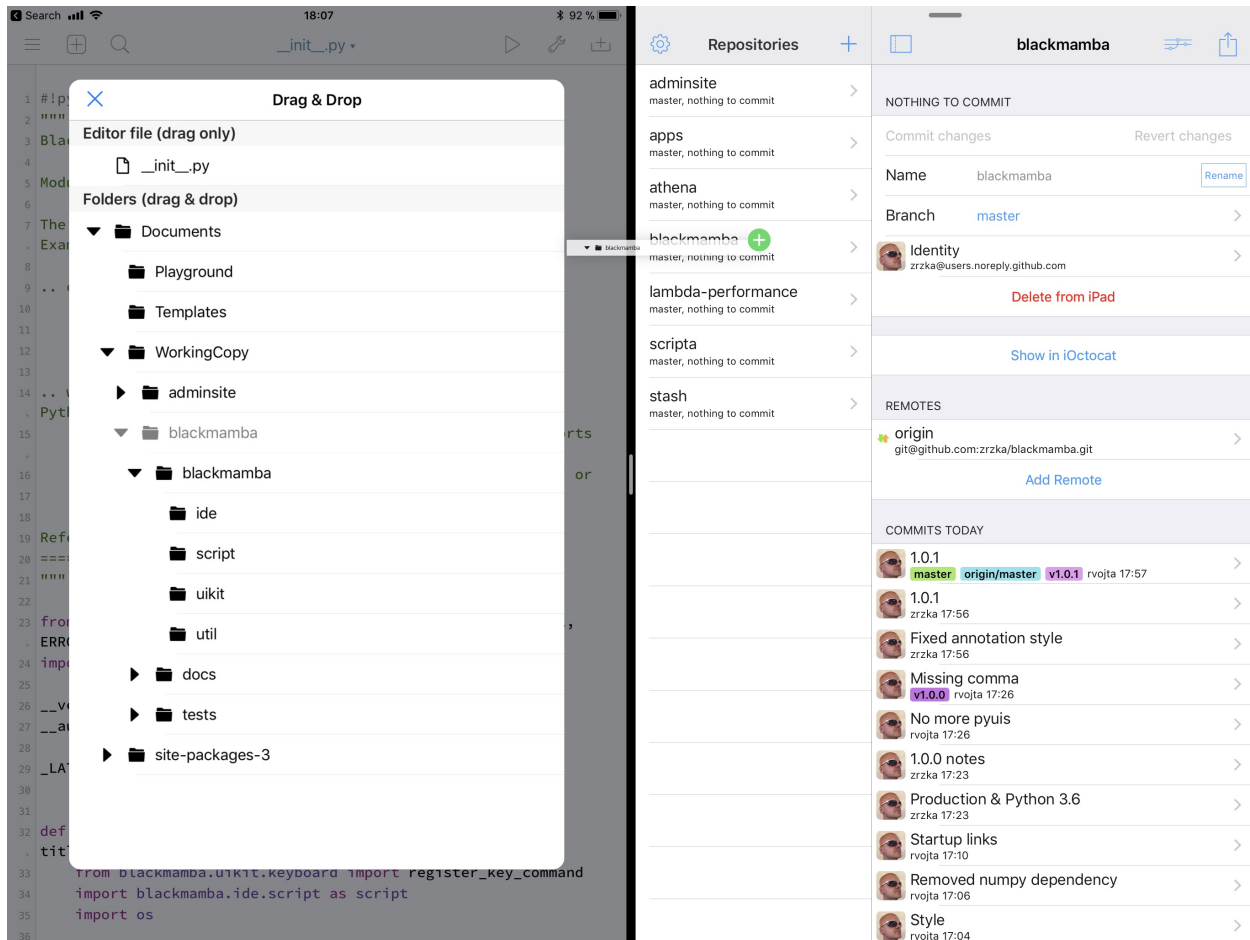
---

### Gallery

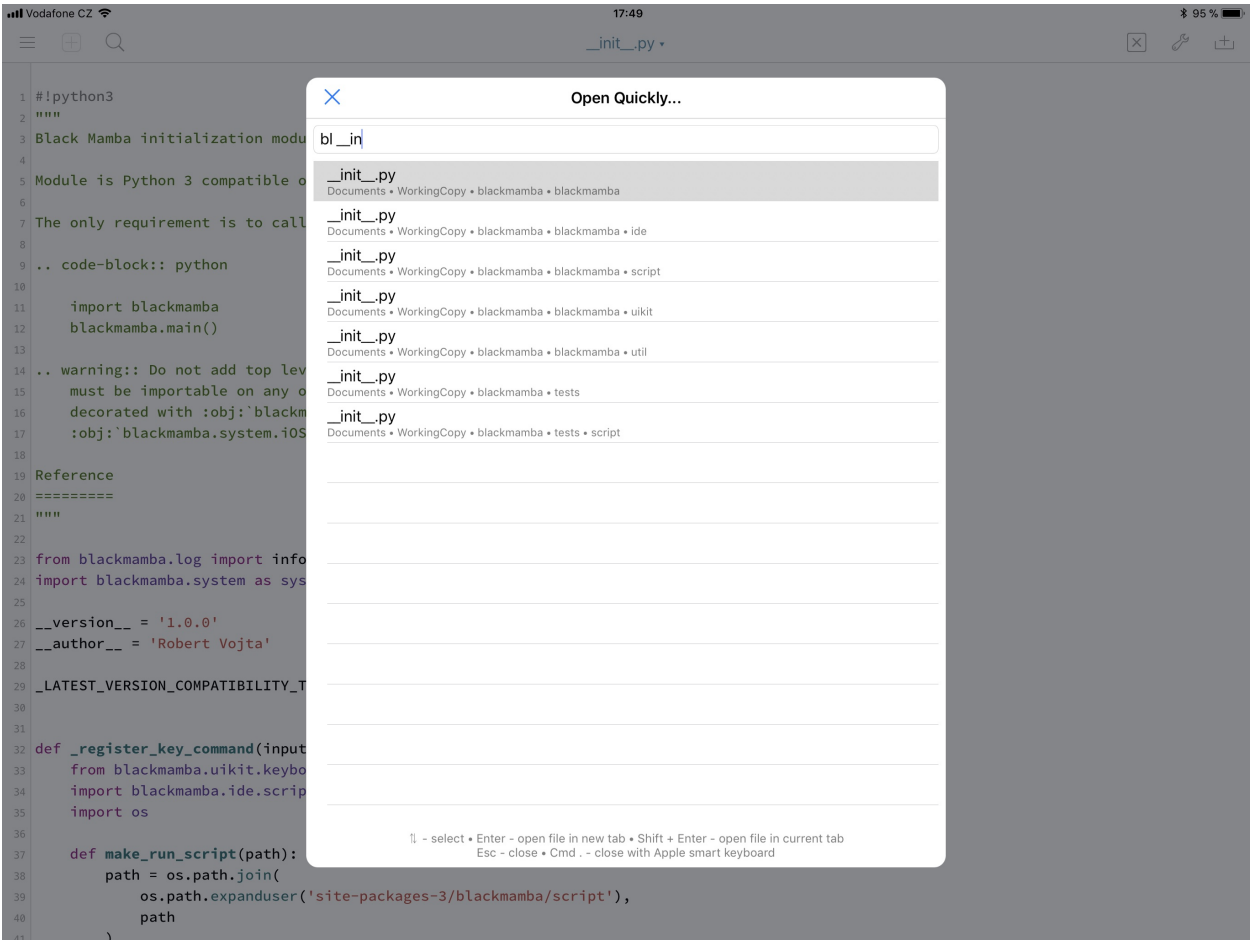
---

Some screenshots to better understand what is the Black Mamba about.

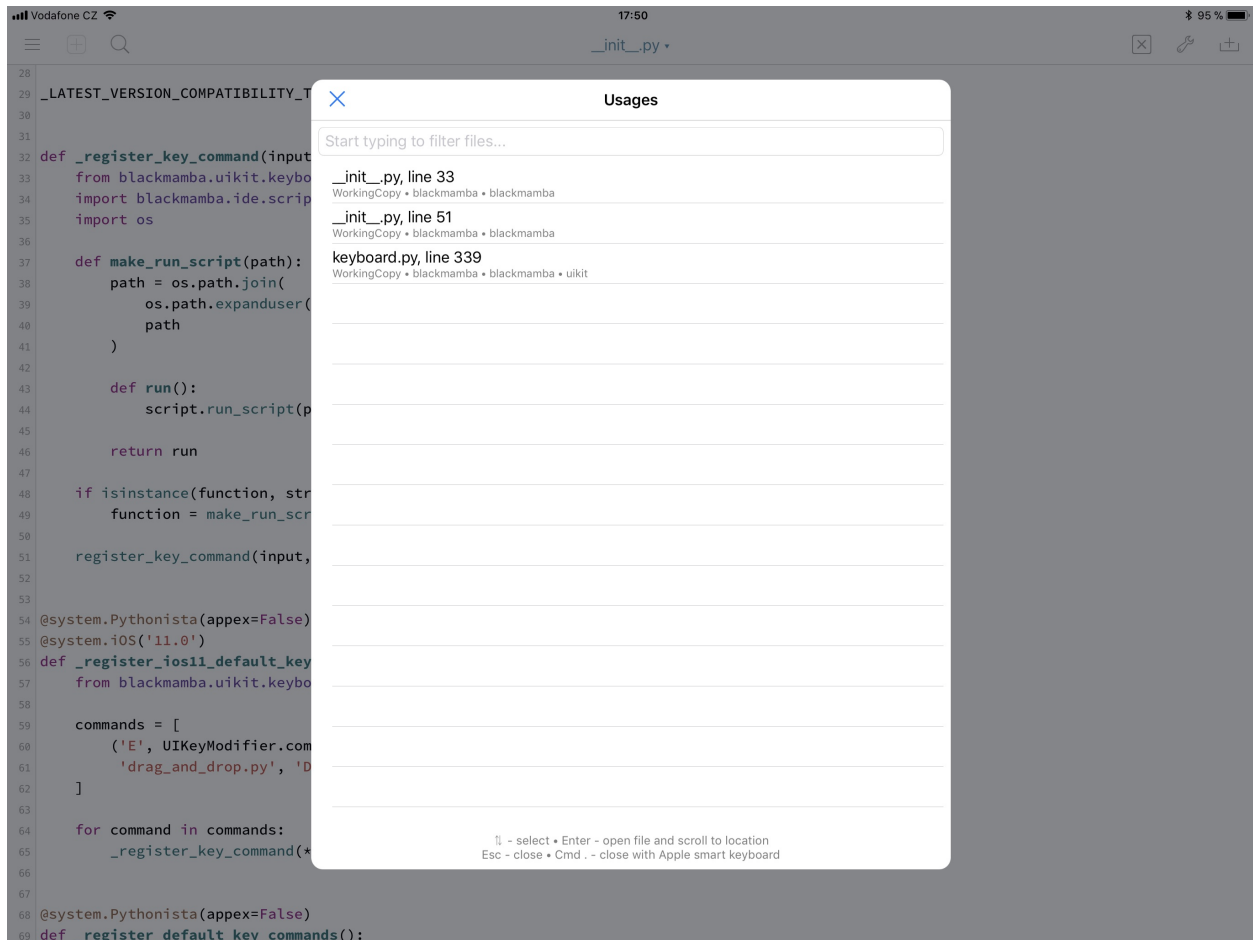
## 7.1 iOS 11 Drag & Drop support



## 7.2 Open quickly



## 7.3 Find usages



## 7.4 Inline unit test results

The screenshot shows a code editor with two tabs: `test_toggle_comments.py` and `run_unit_tests.py`. The `test_toggle_comments.py` tab is active, displaying the following Python code:

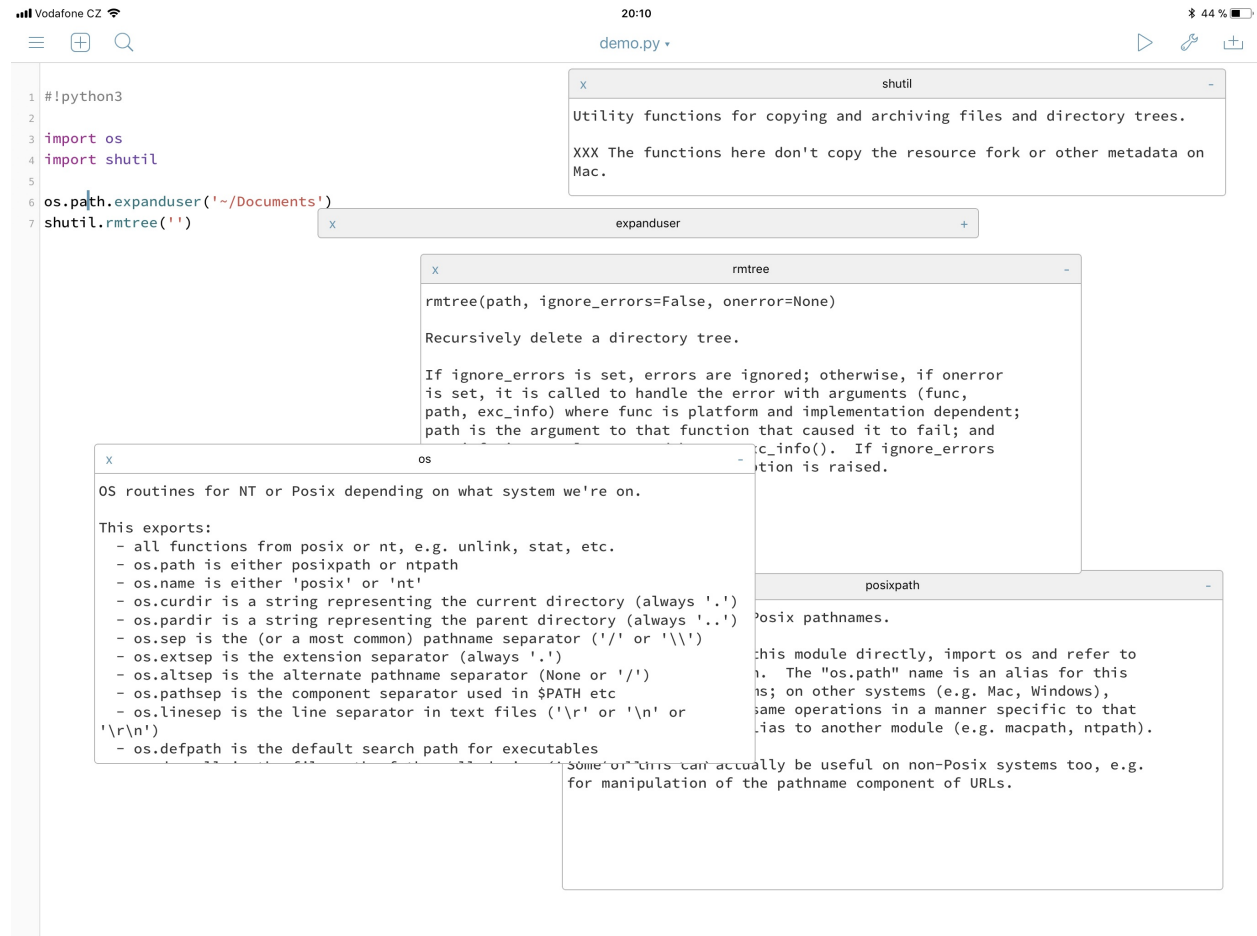
```

30
31 def test_uncomment_line():
32     for key, value in _BASIC_TEST_CASES.items():
33         assert key[1] == _uncomment_line(value)
34
35
36 def test_backward_compatible_uncomment():
37     assert 'a' == _uncomment_line('#a')
38     assert 'a ' == _uncomment_line('#a ')
39     assert '\ta' == _uncomment_line('\ta')
40     assert '\ta ' == _uncomment_line('\ta ')
41
42
43 def test_do_not_comment_commented_lines():
44     assert '# a b c' == _comment_line('# a b c', '')
45     assert '\t# a b c' == _comment_line('\t# a b c', '\t')
46
47
48 def test_do_not_touch_not_commented_lines():
49     assert 'a b c' == _uncomment_line('a b c ')
50     assert 'a b c ' == _uncomment_line('a b c ')
51     assert '\ta b c' == _uncomment_line('\ta b c')
52
53
54 def test_empty_lines():
55     assert '# \n' == _comment_line('\n', ' ')
56     assert '\n' == _uncomment_line(' # \n')
57
58
59 _LINES_TEST_CASES = [
60     (
61         [
62             "def hallo():\n",
63             "    pass\n"
64         ],
65         [
66             "# def hallo():\n",
67             "#     pass\n"
68         ]
69     ),
70 ]

```

On line 49, there is an inline unit test result displayed in a red box: `assert 'a b c' == 'a b c' - a b c + a b c ? +`. The code editor also shows a status bar at the top with "Vodafone CZ", "17:52", and "95 %".

## 7.5 Inline documentation





## 7.6 Inline analyzer results



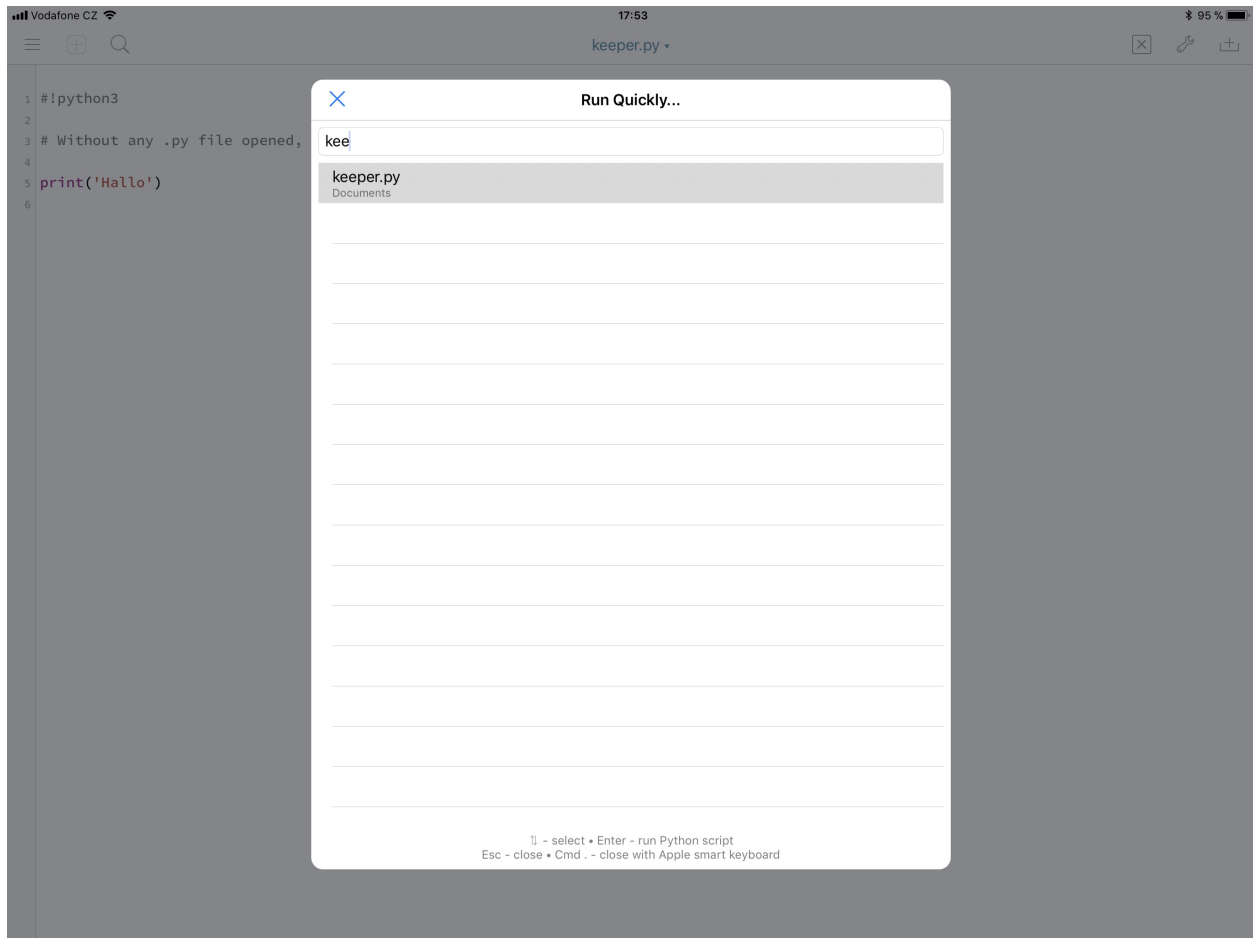
```

188     update_config_with_dict(config)
189     _check_compatibility_and_updates()
190     if get_config_value('general.register_key_commands', True):
191         _register_default_key_commands()
192         _register_iosll_default_key_commands()
193     info('Black Mamba initialized')
194
195
196
197
198 def main(config=None):
199     """
200     Black Mamba initialization.
201
202     Call this function from ``pythonista_startup.py`` (``site-packages-3``) file.
203
204     :param config: Optional dictionary, see :ref:`configuration`
205
206     Example:
207
208     .. code-block:: python
209
210         import blackmamba
211
212         config = {
213             'general': {
214                 'jedi': True
215             }
216         }
217
218         blackmamba.main(config)
219
220     See `pythonista_startup.py` <https://github.com/zrzka/blackmamba/blob/master/pythonista\_startup.py>`_
221     for more examples.
222
223     """
224     _main(config)
225
226
227 if __name__ == '__main__':
228     main()

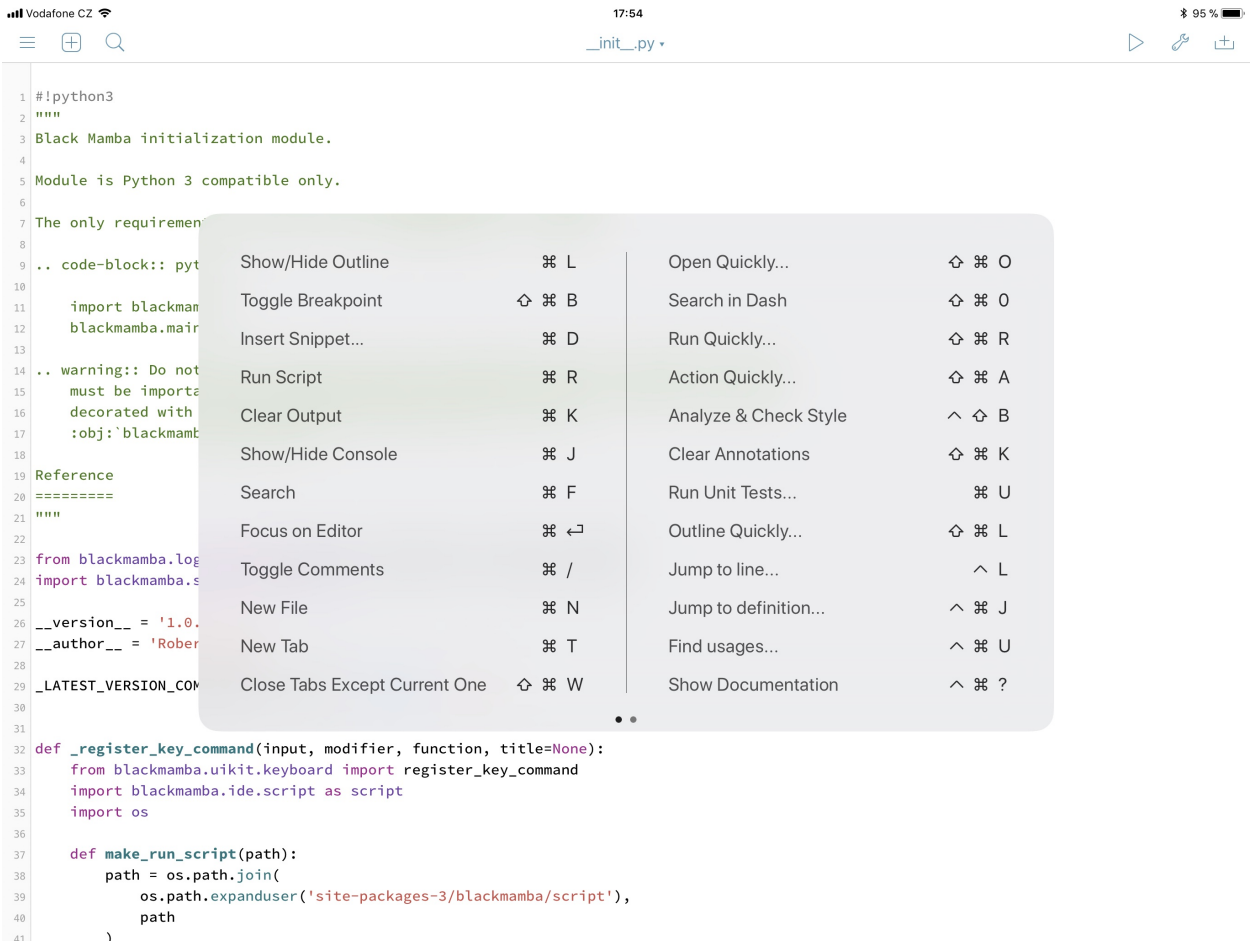
```

E303 too many blank lines (4)

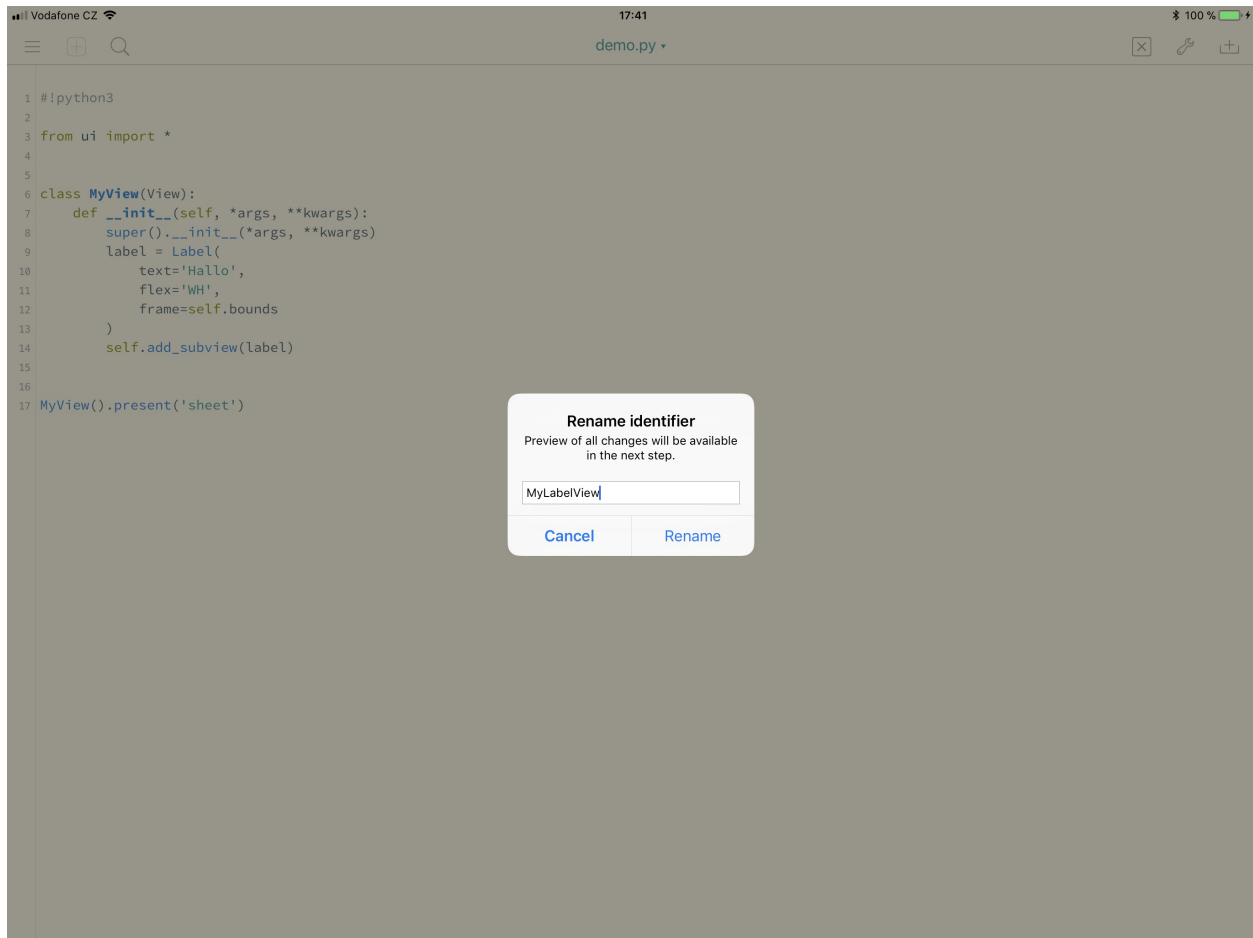
## 7.7 Run quickly

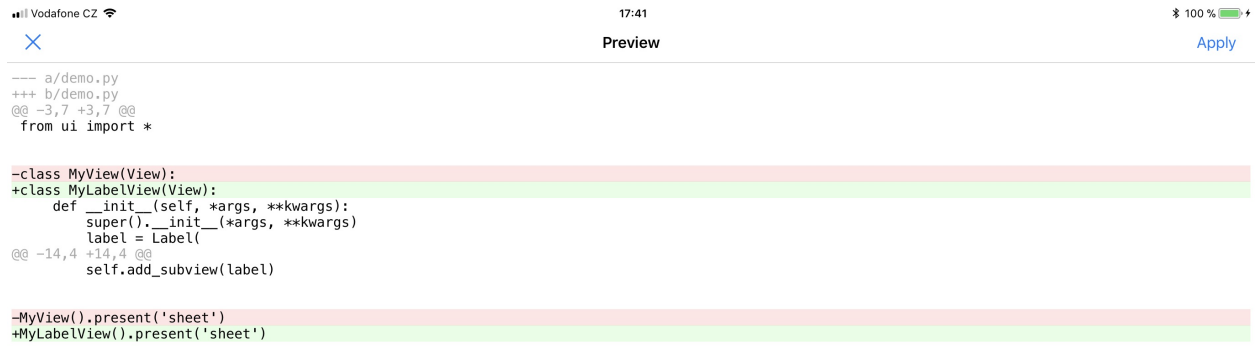


## 7.8 Others



## 7.9 Refactoring







---

## Bundled packages

---

Black Mamba has several bundled packages. Why?

- Not fully working `pip` in Pythonista.
- Pythonista contains some of these packages, but they're outdated (longer release cycle).
- Installation must be as easy as possible (no `pip`, `pip3`, no dependencies).

### 8.1 Packages

List of bundled packages.

#### 8.1.1 docutils

Project homepage.

##### License

See [COPYING](#) for more details.

#### 8.1.2 flake8

Project homepage.

##### License

Copyright (C) 2011-2013 Tarek Ziade <[tarek@ziade.org](mailto:tarek@ziade.org)> Copyright (C) 2012-2016 Ian Cordasco <[graffatcolmin-gov@gmail.com](mailto:graffatcolmin-gov@gmail.com)>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### 8.1.3 future

[Project homepage.](#)

#### License

Copyright (c) 2013-2016 Python Charmers Pty Ltd, Australia

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### 8.1.4 mccabe

[Project homepage.](#)

#### License

Copyright © <year> Ned Batchelder Copyright © 2011-2013 Tarek Ziade <[tarek@ziade.org](mailto:tarek@ziade.org)> Copyright © 2013 Florent Xicluna <[florent.xicluna@gmail.com](mailto:florent.xicluna@gmail.com)>

Licensed under the terms of the Expat License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:



The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### 8.1.5 pycodestyle

[Project homepage.](#)

#### License

Copyright © 2006-2009 Johann C. Rocholl <[johann@rocholl.net](mailto:johann@rocholl.net)> Copyright © 2009-2014 Florent Xicluna <[florent.xicluna@gmail.com](mailto:florent.xicluna@gmail.com)> Copyright © 2014-2016 Ian Lee <[IanLee1521@gmail.com](mailto:IanLee1521@gmail.com)>

Licensed under the terms of the Expat License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### 8.1.6 pyflakes

[Project homepage.](#)

#### License

Copyright 2005-2011 Divmod, Inc. Copyright 2013-2014 Florent Xicluna

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT

HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### **8.1.7 rope**

[Project homepage](#).

#### **License**

Copyright (C) 2015-2016 Nicholas Smith Copyright (C) 2014-2015 Matej Cepl Copyright (C) 2006-2012 Ali Gholami Rudi Copyright (C) 2009-2012 Anton Gritsay

This program is free software; you can redistribute it and/or modify it under the terms of GNU General Public License as published by the Free Software Foundation; either version 2 of the license, or (at your opinion) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

### b

- blackmamba, 15
- blackmamba.config, 6
- blackmamba.log, 16
- blackmamba.script, 10
  - blackmamba.script.action\_quickly, 10
  - blackmamba.script.analyze, 10
  - blackmamba.script.clear\_annotations, 11
  - blackmamba.script.close\_all\_tabs\_except\_current\_one, 11
  - blackmamba.script.drag\_and\_drop, 11
  - blackmamba.script.find\_usages, 11
  - blackmamba.script.jump\_to\_definition, 11
  - blackmamba.script.jump\_to\_line, 12
  - blackmamba.script.new\_file, 12
  - blackmamba.script.new\_tab, 12
  - blackmamba.script.open\_quickly, 12
  - blackmamba.script.outline\_quickly, 12
  - blackmamba.script.refactoring, 13
    - blackmamba.script.refactoring.expand\_star\_imports, 13
    - blackmamba.script.refactoring.futurize, 14
    - blackmamba.script.refactoring.organize\_imports, 13
    - blackmamba.script.refactoring.rename, 13
  - blackmamba.script.run\_quickly, 12
  - blackmamba.script.run\_unit\_tests, 12
  - blackmamba.script.search\_dash, 13
  - blackmamba.script.show\_documentation, 13
  - blackmamba.script.toggle\_comments, 13
- blackmamba.system, 17
- blackmamba.uikit.keyboard, 19



## A

alphaShift (blackmamba.uikit.keyboard.UIKeyModifier attribute), 21  
alternate (blackmamba.uikit.keyboard.UIKeyModifier attribute), 21

## B

backspace (blackmamba.uikit.keyboard.UIEventKeyCode attribute), 21  
blackmamba (module), 15  
blackmamba.config (module), 6  
blackmamba.log (module), 16  
blackmamba.script (module), 10  
blackmamba.script.action\_quickly (module), 10  
blackmamba.script.analyze (module), 10  
blackmamba.script.clear\_annotations (module), 11  
blackmamba.script.close\_all\_tabs\_except\_current\_one (module), 11  
blackmamba.script.drag\_and\_drop (module), 11  
blackmamba.script.find\_usages (module), 11  
blackmamba.script.jump\_to\_definition (module), 11  
blackmamba.script.jump\_to\_line (module), 12  
blackmamba.script.new\_file (module), 12  
blackmamba.script.new\_tab (module), 12  
blackmamba.script.open\_quickly (module), 12  
blackmamba.script.outline\_quickly (module), 12  
blackmamba.script.refactoring (module), 13  
blackmamba.script.refactoring.expand\_star\_imports (module), 13  
blackmamba.script.refactoring.futurize (module), 14  
blackmamba.script.refactoring.organize\_imports (module), 13  
blackmamba.script.refactoring.rename (module), 13  
blackmamba.script.run\_quickly (module), 12  
blackmamba.script.run\_unit\_tests (module), 12  
blackmamba.script.search\_dash (module), 13  
blackmamba.script.show\_documentation (module), 13  
blackmamba.script.toggle\_comments (module), 13  
blackmamba.system (module), 17

blackmamba.uikit.keyboard (module), 19

## C

catch\_exceptions() (in module blackmamba.system), 19  
command (blackmamba.uikit.keyboard.UIKeyModifier attribute), 21  
control (blackmamba.uikit.keyboard.UIKeyModifier attribute), 21

## D

debug() (in module blackmamba.log), 17  
dot (blackmamba.uikit.keyboard.UIEventKeyCode attribute), 22  
down (blackmamba.uikit.keyboard.UIEventKeyCode attribute), 21  
downArrow (blackmamba.uikit.keyboard.UIKeyInput attribute), 22

## E

enter (blackmamba.uikit.keyboard.UIEventKeyCode attribute), 21  
error() (in module blackmamba.log), 17  
escape (blackmamba.uikit.keyboard.UIEventKeyCode attribute), 21

## G

get\_level() (in module blackmamba.log), 17

## I

info() (in module blackmamba.log), 17  
iOS (class in blackmamba.system), 18  
IOS (in module blackmamba.system), 18  
IOS\_VERSION (in module blackmamba.system), 18  
IOS\_VERSION\_TUPLE (in module blackmamba.system), 18  
is\_in\_hardware\_keyboard\_mode() (in module blackmamba.uikit.keyboard), 20

## L

left (blackmamba.uikit.keyboard.UIEventKeyCode attribute), [21](#)

leftArrow (blackmamba.uikit.keyboard.UIKeyInput attribute), [22](#)

## M

main() (in module blackmamba), [16](#)

## N

none (blackmamba.uikit.keyboard.UIKeyModifier attribute), [21](#)

numericPad (blackmamba.uikit.keyboard.UIKeyModifier attribute), [21](#)

## P

Pythonista (class in blackmamba.system), [18](#)

PYTHONISTA (in module blackmamba.system), [18](#)

PYTHONISTA\_BUNDLE\_VERSION (in module blackmamba.system), [18](#)

PYTHONISTA\_VERSION (in module blackmamba.system), [18](#)

PYTHONISTA\_VERSION\_TUPLE (in module blackmamba.system), [18](#)

## R

register\_key\_command() (in module blackmamba.uikit.keyboard), [22](#)

register\_key\_event\_handler() (in module blackmamba.uikit.keyboard), [22](#)

right (blackmamba.uikit.keyboard.UIEventKeyCode attribute), [21](#)

rightArrow (blackmamba.uikit.keyboard.UIKeyInput attribute), [22](#)

## S

set\_level() (in module blackmamba.log), [17](#)

shift (blackmamba.uikit.keyboard.UIKeyModifier attribute), [21](#)

space (blackmamba.uikit.keyboard.UIEventKeyCode attribute), [21](#)

## U

UIEventKeyCode (class in blackmamba.uikit.keyboard), [21](#)

UIKeyInput (class in blackmamba.uikit.keyboard), [22](#)

UIKeyModifier (class in blackmamba.uikit.keyboard), [20](#)

unregister\_key\_event\_handler() (in module blackmamba.uikit.keyboard), [22](#)

unregister\_key\_event\_handlers() (in module blackmamba.uikit.keyboard), [23](#)

up (blackmamba.uikit.keyboard.UIEventKeyCode attribute), [21](#)

upArrow (blackmamba.uikit.keyboard.UIKeyInput attribute), [22](#)

## W

warn() (in module blackmamba.log), [17](#)