
AdafruitShell Library Documentation

Release 1.0

Melissa LeBlanc-Williams

Sep 14, 2023

CONTENTS

1	Dependencies	3
2	Installing from PyPI	5
3	Contributing	7
4	Documentation	9
5	Table of Contents	11
5.1	adafruit_shell	11
5.1.1	Implementation Notes	11
6	Indices and tables	15
	Python Module Index	17
	Index	19

Python helper for running Shell scripts in Python

**CHAPTER
ONE**

DEPENDENCIES

This driver depends on:

- Linux

**CHAPTER
TWO**

INSTALLING FROM PYPI

On supported GNU/Linux systems like the Raspberry Pi, you can install the driver locally [from PyPI](#). To install for current user:

```
pip3 install adafruit-python-shell
```

To install system-wide (this may be required in some cases):

```
sudo pip3 install adafruit-python-shell
```

To install in a virtual environment in your current project:

```
mkdir project-name && cd project-name
python3 -m venv .env
source .env/bin/activate
pip3 install adafruit-python-shell
```

**CHAPTER
THREE**

CONTRIBUTING

Contributions are welcome! Please read our [Code of Conduct](#) before contributing to help this project stay welcoming.

**CHAPTER
FOUR**

DOCUMENTATION

For information on building library documentation, please check out [this guide](#).

TABLE OF CONTENTS

5.1 adafruit_shell

Python helper for running Shell scripts in Python

- Author(s): Melissa LeBlanc-Williams

5.1.1 Implementation Notes

Software and Dependencies:

- Linux

`class adafruit_shell.Shell`

Class to help with converting Shell scripts over to Python. Having all the functions in one place makes updates easier and code shorter.

`property args`

Get a list of supplied arguments

`argument_exists(arg, prefix='-')`

Check if the given argument was supplied

`bail(message=None, **kwargs)`

Exit and display an error message if given

`chdir(directory)`

Change directory

`check_kernel_update_reboot_required()`

Checks if the pi needs to be rebooted since the last kernel update

`check_kernel_userspace_mismatch()`

Check if the userspace is 64-bit and kernel is 32-bit

`chmod(location, mode)`

Change the permissions of a file or directory

`chown(location, user, group=None, recursive=False)`

Change the owner of a file or directory

`static clear()`

Clear the screen

copy(*source, destination*)

Move a file or directory from source to destination

static date()

Return a string containing the current date and time

error(*message, **kwargs*)

Display some information

exists(*location*)

Check if a path or file exists

static exit(*status_code=0*)

Exit and return the status code to the OS

static get_arch()

Return a string containing the architecture

static get_architecture()

Get the type of Processor

static get_board_model()

Use PlatformDetect to get the board model

get_os()

Return a string containing the release which we can use to compare in the script

get_raspbian_version()

Return a string containing the raspbian version

static getcwd()

Get the Current Working Directory

grep(*search_term, location*)

Run the grep command and return the result

property group

Get or set the current group that is displayed in color along with messages

static home_dir()

Return the User's home directory

info(*message, **kwargs*)

Display a message with the group in green

static is_arm64()

Check if Platform.machine() returns ARM 64

static is_arhf()

Check if Platform.machine() (same as uname -m) returns an ARM platform that supports hardware floating point

static is_armv6()

Check if Platform.machine() returns ARM v6

static is_armv7()

Check if Platform.machine() returns ARM v7

```
static is_armv8()
    Check if Platform.machine() returns ARM v8

static is_linux()
    Check that we are running linux

static is_python3()
    Check if we are running Python 3 or later

static is_raspberry_pi()
    Use PlatformDetect to check if this is a Raspberry Pi

is_raspberry_pi_os()
    Check if we are running Raspberry Pi OS or Raspbian

static is_root()
    Return whether the current user is logged in as root or has super user access

isdir(location)
    Check if a location exists and is a directory

static kernel_minimum(version)
    Check that we are running on at least the specified version

move(source, destination)
    Move a file or directory from source to destination

static path(file_path)
    Return the relative path. This works for paths starting with ~

pattern_replace(location, pattern, replace='', multi_line=False)
    Similar to sed, but uses pure python multi_line will search the entire file as a large text glob, but certain
    regex patterns such as ^ and $ will not work on a line-by-line basis

pattern_search(location, pattern, multi_line=False, return_match=False)
    Similar to grep, but uses pure python multi_line will search the entire file as a large text glob, but certain
    regex patterns such as ^ and $ will not work on a line-by-line basis returns True/False if found

popd()
    Change directory

static print_colored(message, color)
    Print out a message in a specific color

prompt(message, *, default=None, force_arg=None, force_arg_value=True)
    A Yes/No prompt that accepts optional defaults Returns True for Yes and False for No

prompt_reboot(default='y', **kwargs)
    Prompt the user for a reboot

pushd(directory)
    Change directory

static reboot()
    Reboot the system
```

reconfig(*file*, *pattern*, *replacement*)

Given a filename, a regex pattern to match and a replacement string, perform replacement if found, else append replacement to end of file.

static release()

Return the latest kernel release version

remove(*location*)

Remove a file or directory if it exists

require_root()

Check if the current user has root access and exit if not.

run_command(*cmd*, *suppress_message=False*, *return_output=False*, *run_as_user=None*)

Run a shell command and show the output as it runs

static script()

Return the name of the script that is running

static select_n(*message*, *selections*)

Display a list of selections for the user to enter

warn(*message*, ***kwargs*)

Display a message with the group in yellow

write_text_file(*path*, *content*, *append=True*)

Write the contents to a file at the specified path

**CHAPTER
SIX**

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

a

adafruit_shell, [11](#)

INDEX

A

adafruit_shell
 module, 11
args (*adafruit_shell.Shell property*), 11
argument_exists() (*adafruit_shell.Shell method*), 11

B

bail() (*adafruit_shell.Shell method*), 11

C
chdir() (*adafruit_shell.Shell method*), 11
check_kernel_update_reboot_required()
 (*adafruit_shell.Shell method*), 11
check_kernel_userspace_mismatch()
 (*adafruit_shell.Shell method*), 11
chmod() (*adafruit_shell.Shell method*), 11
chown() (*adafruit_shell.Shell method*), 11
clear() (*adafruit_shell.Shell static method*), 11
copy() (*adafruit_shell.Shell method*), 11

D

date() (*adafruit_shell.Shell static method*), 12

E

error() (*adafruit_shell.Shell method*), 12
exists() (*adafruit_shell.Shell method*), 12
exit() (*adafruit_shell.Shell static method*), 12

G

get_arch() (*adafruit_shell.Shell static method*), 12
get_architecture() (*adafruit_shell.Shell static method*), 12
get_board_model() (*adafruit_shell.Shell static method*), 12
get_os() (*adafruit_shell.Shell method*), 12
get_raspbian_version() (*adafruit_shell.Shell method*), 12
getcwd() (*adafruit_shell.Shell static method*), 12
grep() (*adafruit_shell.Shell method*), 12
group (*adafruit_shell.Shell property*), 12

H

home_dir() (*adafruit_shell.Shell static method*), 12

I

info() (*adafruit_shell.Shell method*), 12
is_arm64() (*adafruit_shell.Shell static method*), 12
is_armhf() (*adafruit_shell.Shell static method*), 12
is_armv6() (*adafruit_shell.Shell static method*), 12
is_armv7() (*adafruit_shell.Shell static method*), 12
is_armv8() (*adafruit_shell.Shell static method*), 12
is_linux() (*adafruit_shell.Shell static method*), 13
is_python3() (*adafruit_shell.Shell static method*), 13
is_raspberry_pi() (*adafruit_shell.Shell static method*), 13
is_raspberry_pi_os() (*adafruit_shell.Shell method*), 13
is_root() (*adafruit_shell.Shell static method*), 13
isdir() (*adafruit_shell.Shell method*), 13

K

kernel_minimum() (*adafruit_shell.Shell static method*), 13

M

module
 adafruit_shell, 11
move() (*adafruit_shell.Shell method*), 13

P

path() (*adafruit_shell.Shell static method*), 13
pattern_replace() (*adafruit_shell.Shell method*), 13
pattern_search() (*adafruit_shell.Shell method*), 13
popd() (*adafruit_shell.Shell method*), 13
print_colored() (*adafruit_shell.Shell static method*), 13
prompt() (*adafruit_shell.Shell method*), 13
prompt_reboot() (*adafruit_shell.Shell method*), 13
pushd() (*adafruit_shell.Shell method*), 13

R

reboot() (*adafruit_shell.Shell static method*), 13

`reconfig()` (*adafruit_shell.Shell method*), [13](#)
`release()` (*adafruit_shell.Shell static method*), [14](#)
`remove()` (*adafruit_shell.Shell method*), [14](#)
`require_root()` (*adafruit_shell.Shell method*), [14](#)
`run_command()` (*adafruit_shell.Shell method*), [14](#)

S

`script()` (*adafruit_shell.Shell static method*), [14](#)
`select_n()` (*adafruit_shell.Shell static method*), [14](#)
`Shell` (*class in adafruit_shell*), [11](#)

W

`warn()` (*adafruit_shell.Shell method*), [14](#)
`write_text_file()` (*adafruit_shell.Shell method*), [14](#)