

---

# **scripting Documentation**

***Release 0.1***

**Gabriele Lanaro**

**Aug 02, 2017**



---

## Contents

---

<b>1</b>	<b>Reference</b>	<b>3</b>
<b>2</b>	<b>Indices and tables</b>	<b>5</b>
	<b>Python Module Index</b>	<b>7</b>



Scripting is a fantastic python module to produce python shell scripts fast and clean, with support for command line arguments and useful unix-like commands.

Contents:



Shell-like commands for scripting development

`scripting.commands.archive(src, dst, format='tar')`

Create an archive *dst* from the directory *src*. The format can be specified with the *format* parameter that can be:

- tar
- gzip
- bz2
- zip

`scripting.commands.basename(src)`

Unix basename

`scripting.commands.cp(*args, **kw)`

Copy files and directories

**Parameters** **src** – can be a string or an iterable and is the file/s to copy

;param **dst**: the destination file or directory

`scripting.commands.find(glob, startdir='.')`

Return a list of files that matches *glob*

**Parameters**

- **glob** – is the pattern to search for like *\*.py* or *\*.py?*
- **startdir** – the path from where to search

`scripting.commands.mkdir(*args, **kw)`

Make a directory

**Parameters** **parent** – if True, create all the directory structure

`scripting.commands.rm(*args, **kw)`

Remove a file or a directory

**Parameters** **src** – a string or an iterable, representing the file/s to remove

`scripting.commands.sh(cmdl, args=[])`

Simple function to run shell\_commands it takes a shell command line *cmdln*, optionally further arguments *args*.

Returns a tuple, (exitcode, output, error)

`scripting.commands.take_str_or_list(f)`

Allow the function decorated to take as the first argument a string or an iterable.

If the first arg is a string, call *f* with the string as the first argument, if the first argument is an iterable, call *f* on each item of the iterable and return a list of the results.

`scripting.commands.touch(*args, **kw)`

Create *src* file if it doesn't exists or update his modification time according to *times*.

`scripting.commands.unpack(src, dst, format=None)`

Unpack the archive *src* in the destination directory *dst*. The format can be *tar*, *gzip*, *bz2*, *zip* or *None* to auto-select the format based on the extension.

Module to control the interaction with the shell Bash

`scripting.bash.source(fname)`

Emulate the source command in bash, reading the environment from a script.



## CHAPTER 2

---

### Indices and tables

---

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



### S

`scripting.bash`, 4

`scripting.commands`, 3



### A

archive() (in module scripting.commands), 3

### B

basename() (in module scripting.commands), 3

### C

cp() (in module scripting.commands), 3

### F

find() (in module scripting.commands), 3

### M

mkdir() (in module scripting.commands), 3

### R

rm() (in module scripting.commands), 3

### S

scripting.bash (module), 4

scripting.commands (module), 3

sh() (in module scripting.commands), 4

source() (in module scripting.bash), 4

### T

take\_str\_or\_list() (in module scripting.commands), 4

touch() (in module scripting.commands), 4

### U

unpack() (in module scripting.commands), 4