

writing better
command-line
interfaces
with
docopt

```
Usage: cocoa new <path> [--theme=<THEME>]
       cocoa build [--continuous]
       cocoa serve [--host=127.0.0.1] [--port=9000]
       cocoa (-h | --help | --version)
```

```
#!/usr/bin/env python
```

```
"""
```

```
Usage: cocoa new <path> [--theme=<THEME>]
       cocoa build [--continuous]
       cocoa serve [--host=127.0.0.1] [--port=9000]
       cocoa (-h | --help | --version)
```

```
"""
```

```
import argparse
```

```
parser = argparse.ArgumentParser(prog='cocoa')
parser.add_argument('--version', action='version', version='1.2.0rc1')
subparsers = parser.add_subparsers()
```

```
parser_init = subparsers.add_parser('new')
parser_init.add_argument('<path>')
parser_init.add_argument('--theme')
```

```
parser_build = subparsers.add_parser('build')
parser_build.add_argument('--continuous')
```

```
parser_serve = subparsers.add_parser('serve')
parser_serve.add_argument('--host', default='127.0.0.1')
parser_serve.add_argument('--port', default=9000)
```

```
print(parser.parse_args())
```

“ If you have to refer to the documentation every time you use a module, find (or build) a new module

– *Kenneth Reitz*

```
#!/usr/bin/env python
```

```
''''''
```

```
Usage: cocoa new <path> [--theme=<THEME>]
```

```
      cocoa build [--continuous]
```

```
      cocoa serve [--host=127.0.0.1] [--port=9000]
```

```
      cocoa (-h | --help | --version)
```

```
''''''
```

```
from docopt import docopt
```

```
print(docopt(__doc__, version='1.2.0rc1'))
```

```
#!/usr/bin/env python
```

```
''''''
```

```
Usage: cocoa new <path> [--theme=<THEME>]
       cocoa build [--continuous]
       cocoa serve [--host=127.0.0.1] [--port=9000]
       cocoa (-h | --help | --version)
```

```
''''''
```

```
from docopt import docopt
```

```
print(docopt(__doc__, version='1.2.0rc1'))
```

```
{ '--continuous': False,          '<directory>': None,
  '--help':       False,          'build':       False,
  '--host':       '0.0.0.0',      'init':        False,
  '--port':       None,           'serve':       True,
  '--theme':      None,
  '--version':    False           }
```

```
Usage: cocoa new <path> [--theme=<THEME>]
       cocoa build [--continuous]
       cocoa serve [--host=127.0.0.1] [--port=9000]
       cocoa (-h | --help | --version)
```

```
Usage: cocoa new <path> [--theme=<THEME>]
       cocoa build [--continuous]
       cocoa serve [--host=127.0.0.1] [--port=9000]
       cocoa (-h | --help | --version)
```


A tool for building static websites.

```
Usage: cocoa new <path> [--theme=<THEME>]
       cocoa build [--continuous]
       cocoa serve [--host=127.0.0.1] [--port=9000]
       cocoa (-h | --help | --version)
```

Options:

--version	show the program's version number
-h --help	show this help message
--theme=<THEME>	name of the site theme to use
--continuous	watch the site directory and rebuild on changes
--host=<HOST>	hostname to bind to; use 0.0.0.0 to serve globally
--port=<PORT>	port to run on

Usage: my_program command --option <arg>
my_program [<optional-arg>]
my_program --another-option=<with-arg>
my_program (--either-this| --or-that)
my_program <repeating-arg> <repeating-arg> ...



Search Projects

Search

Help Donate Login Register

docopt 0.6.2

✓ Latest Version

`pip install docopt`

Jun 16, 2014

Pythonic argument parser, that will make you smile

Navigation

Project Description

Release History

Download Files

Project Links

Homepage

Meta

License: MIT License

Author: Vladimir Keleshev

option, arguments, parsing, optparse, argparse, getopt

docopt creates beautiful command-line interfaces

Video introduction to docopt: [PyCon UK 2012: Create *beautiful* command-line interfaces with Python](#)

New in version 0.6.1:

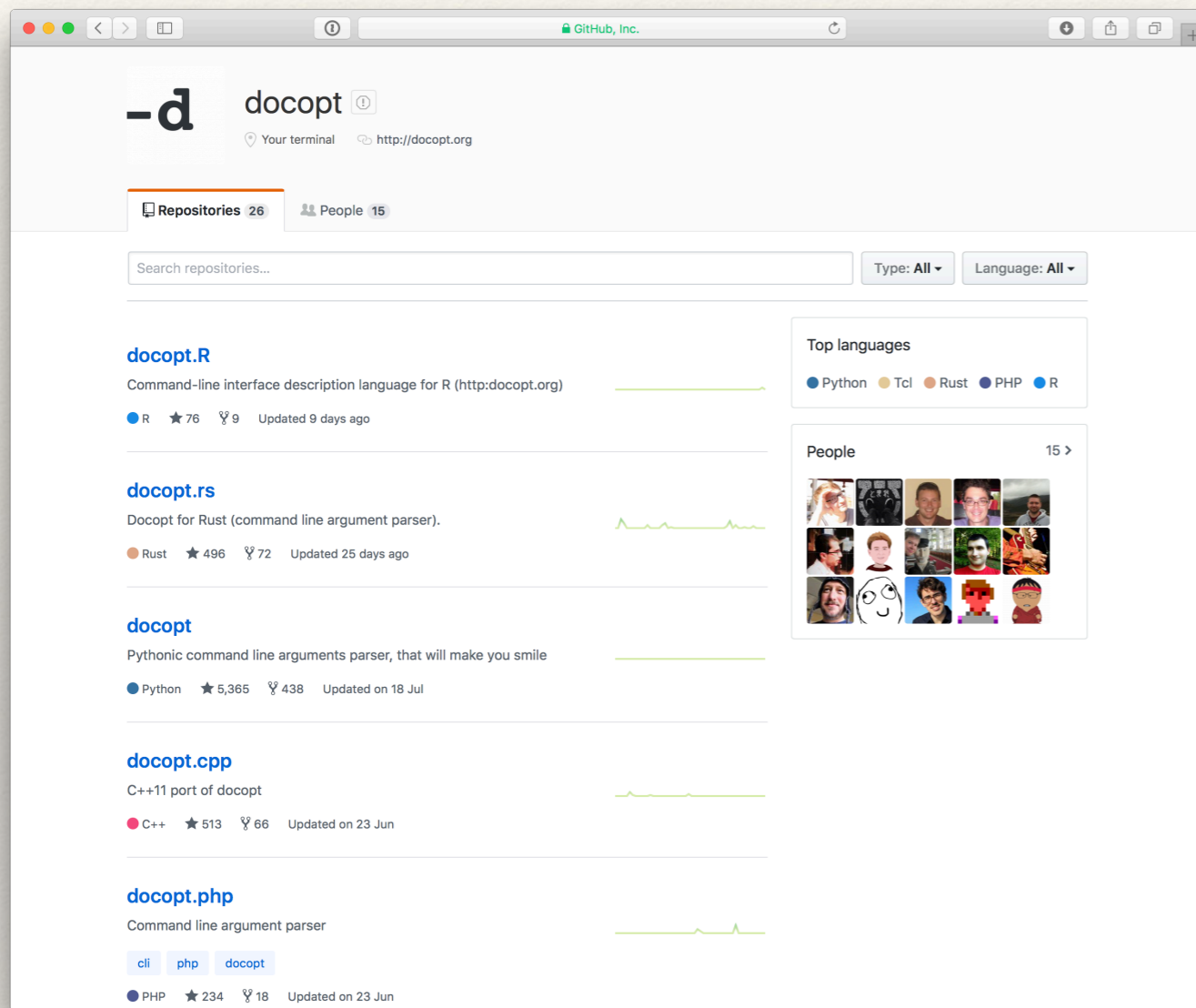
- Fix issue #85 which caused improper handling of `[options]` shortcut if it was present several times.

New in version 0.6.0:

- New argument `options_first`, disallows interspersing options and arguments. If you supply `options_first=True` to `docopt`, it will interpret all arguments as positional arguments after first positional argument.
- If option with argument could be repeated, its default value will be interpreted as space-separated list. E.g. with `[default: ./here ./there]` will be interpreted as `['./here', './there']`.

Breaking changes:

- Meaning of `[options]` shortcut slightly changed. Previously it ment “any known option”. Now it means “any option not in usage-pattern”. This avoids the situation when an option is allowed to be repeated unintentionally.



C
C#
C++
Clojure
CoffeeScript
D
F#
Go
Haskell
Haxe
Java

Julia
Lua
Nim
PHP
Python
R
Ruby
Rust
Scala
Swift
TCL

```
<?php
```

```
$doc = <<<DOC
```

```
A tool for building static websites.
```

```
Usage: cocoa.php new <path> [--theme=<THEME>]
```

```
      cocoa.php build [--continuous]
```

```
      cocoa.php serve [--host=127.0.0.1] [--port=9000]
```

```
      cocoa.php (-h | --help | --version)
```

```
Options:
```

```
  --version          show the program's version number
```

```
  -h --help         show this help message
```

```
  --theme=<THEME>  name of the site theme to use
```

```
  --continuous     watch the site directory and rebuild on changes
```

```
  --host=<HOST>    hostname to bind to; use 0.0.0.0 to serve  
                  globally
```

```
  --port=<PORT>    port to run on
```

```
DOC;
```

```
require('path/to/src/docopt.php');
```

```
$args = Docopt::handle($doc, array('version'=>'Cocoa 1.2.0rc1'));
```

```
foreach ($args as $k=>$v)
```

```
    echo $k.': '.json_encode($v).PHP_EOL;
```

<http://docopt.org>