

Command line arguments that make you smile

@martinmelin

martin@tictail.com

Command line arguments that make you smile :-)

@martinmelin

martin@tictail.com

```
$ python --version
```

```
Python 2.7.5
```

```
$ python --version
```

```
Python 2.7.5
```

UI for your
command-line program

Who still writes
command-line programs?

Everyone should

But we usually just call them scripts

Scripts are awesome

Very few things are actually one-offs

One-off scripts spread knowledge

Similar things can be solved by looking at old scripts

Scripts save
you from yourself

```
$ one-off-script.py
```

```
RuntimeError: Stupid mistake
```

Scripts are much
better with arguments

```
$ one-off-script.py --dry-run
```

```
RuntimeError: Stupid mistake
```

... but adding arguments
is painful, so we don't

It doesn't have to be

but first, the status quo:

argparse

standard library recommendation


```
$ python prog.py -h
```

```
usage: prog.py [-h] [--sum] N [N ...]
```

Process some integers.

positional arguments:

N an integer for the accumulator

optional arguments:

-h, --help show this help message and exit
--sum sum the integers (default:
 find the max)

Let's parse!

```
import argparse
```

```
parser = argparse.ArgumentParser(  
    description='Process some integers.')
```

```
import argparse
```

```
parser = argparse.ArgumentParser(  
    description='Process some integers.')
```

```
parser.add_argument(  
    'integers',  
    metavar='N',  
    type=int,  
    nargs='+',  
    help='an integer for the accumulator')
```

```
import argparse

parser = argparse.ArgumentParser(
    description='Process some integers.')

parser.add_argument(
    'integers',
    metavar='N',
    type=int,
    nargs='+',
    help='an integer for the accumulator')

parser.add_argument(
    '--sum',
    dest='accumulate',
    action='store_const',
    const=sum,
    default=max,
    help='sum the integers (default: find the max)')
```

```
import argparse

parser = argparse.ArgumentParser(
    description='Process some integers.')

parser.add_argument(
    'integers',
    metavar='N',
    type=int,
    nargs='+',
    help='an integer for the accumulator')

parser.add_argument(
    '--sum',
    dest='accumulate',
    action='store_const',
    const=sum,
    default=max,
    help='sum the integers (default: find the max)')

args = parser.parse_args()
```

```
import argparse

parser = argparse.ArgumentParser(
    description='Process some integers.')

parser.add_argument(
    'integers',
    metavar='N',
    type=int,
    nargs='+',
    help='an integer for the accumulator')

parser.add_argument(
    '--sum',
    dest='accumulate',
    action='store_const',
    const=sum,
    default=max,
    help='sum the integers (default: find the max)')

args = parser.parse_args()
```

... and we're done

Phew!

All that ugly code for:

```
prog.py [-h] [--sum] N [N ...]
```

There's a better way!

Better practices[™]

docopt

made by Vladimir Keleshev

```
$ python prog.py -h
```

```
usage: prog.py [-h] [--sum] N [N ...]
```

Process some integers.

positional arguments:

N an integer for the accumulator

optional arguments:

-h, --help show this help message and exit
--sum sum the integers (default:
 find the max)

Let's parse!

```
"""usage: prog.py [-h] [--sum] N [N ...]
```

```
Process some integers.
```

```
positional arguments:
```

```
  N                an integer for the accumulator
```

```
optional arguments:
```

```
  -h, --help      show this help message and exit
```

```
  --sum           sum the integers (default:  
                                                         find the max)
```

```
"""
```

```
"""usage: prog.py [-h] [--sum] N [N ...]
```

```
Process some integers.
```

```
positional arguments:
```

```
  N                an integer for the accumulator
```

```
optional arguments:
```

```
  -h, --help      show this help message and exit
```

```
  --sum           sum the integers (default:  
                                                         find the max)
```

```
"""
```

```
import docopt
```

```
"""usage: prog.py [-h] [--sum] N [N ...]
```

```
Process some integers.
```

```
positional arguments:
```

```
  N                an integer for the accumulator
```

```
optional arguments:
```

```
  -h, --help      show this help message and exit
```

```
  --sum           sum the integers (default:  
                                                         find the max)
```

```
"""
```

```
import docopt
```

```
args = docopt.docopt(__doc__)
```


... and we're done

```
$ python prog.py --sum 1 2 3  
{'--help': False,  
  '--sum': True,  
  'N': ['1', '2', '3']}
```

```
$ python prog.py --sum 1 2 3
{'--help': False,
 '--sum': True,
 'N': ['1', '2', '3']}
```

```
$ python prog.py --help
usage: prog.py [-h] [--sum] N [N ...]
```

Process some integers.

positional arguments:

N an integer for the accumulator

optional arguments:

-h, --help show this help message and exit
--sum sum the integers (default:
 find the max)

Write for humans,
let the computer figure it out

Write scripts

Write scripts

Use docopt

Write scripts

Use docopt

Smile :-)

Thanks!

@martinmelin

martin@tictail.com