

optparse!

Hunter Blanks, hblanks@monetate.com / [@tildehblanks](https://twitter.com/tildehblanks)
github.com/hblanks/talks

13.5. argparse -- Parser for command line options

13.5.1.2018-08-08

Deprecation since version 3.7 The `argparse` module is deprecated and will not be developed further. Development will continue with the `argparse` module.

Source code: [argparse.py](#)

`argparse` is a high-level command-line parser that generates `ArgumentParser` objects that handle `add_argument()`. `ArgumentParser` objects are used to parse command-line arguments and generate `ArgumentError` exceptions. `ArgumentParser` objects are also used to generate `ArgumentError` exceptions. `ArgumentParser` objects are also used to generate `ArgumentError` exceptions. `ArgumentParser` objects are also used to generate `ArgumentError` exceptions.

Here is an example of using `argparse` to parse command-line arguments:

```
import argparse
import sys

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument('x', type=int, help='x')
    parser.add_argument('--y', type=int, help='y')
    parser.add_argument('--z', type=int, help='z')
    args = parser.parse_args()
    print('x: %d, y: %d, z: %d' % (args.x, args.y, args.z))

if __name__ == '__main__':
    main()
```

Arbeitsblätter online

11

Image-Long (vertical) and (wide ...)

Image-Long (vertical) and (wide ...)

The image-Long structure
Image-Long

Image-Long (vertical) and (wide ...)

```
image = Image.open('image.jpg', 'r')
image.save('image.jpg', 'JPEG')
image.save('image.png', 'PNG')
image.save('image.gif', 'GIF')
image.save('image.tiff', 'TIFF')
```

your python version, full path may be warranted

ArgumentParser object

will become sys.argv[0]

```
if __name__ == '__main__':  
    parser = argparse.ArgumentParser(  
        description='A simple program that does nothing but prints the command line arguments.'    )
```

```
    parser.add_argument('n', type=int, help='The number of arguments to print.')  
    parser.add_argument('--verbose', '-v', action='store_true', help='Whether to print the command line arguments or not.')  
    args = parser.parse_args()
```

```
    # Print the command line arguments  
    if args.verbose:  
        print('Command line arguments: ', sys.argv[1:])
```

```
    # Print the number of arguments  
    print('Number of arguments: ', args.n)
```

```
    # Print the command line arguments  
    if args.verbose:  
        print('Command line arguments: ', sys.argv[1:])  
    else:  
        print('Command line arguments: ', sys.argv[1:])
```

takes lots of arguments

```

def __init__(self, name, email):
    self.name = name
    self.email = email
    self.friends = []
    return self

```

```

if __name__ == '__main__':
    john = Person('John', 'john@example.com')
    john.add_friend(Person('Jane', 'jane@example.com'))
    john.add_friend(Person('Bob', 'bob@example.com'))
    john.print_friends()

```

```

class Person:
    def __init__(self, name, email):
        self.name = name
        self.email = email
        self.friends = []

```

```

    def add_friend(self, friend):
        self.friends.append(friend)

```

```

    def print_friends(self):

```

you can print the help whenever you want
(NB: it will go to stdout)

