How do I access command line arguments in Python?

I use python to create my project settings setup, but I need help getting the command line arguments.

I tried this on the terminal:

```
$python myfile.py var1 var2 var3
```

In my Python file, I want to use all variables that are input.

```
import sys

print(sys.argv)
```

More specifically, if you run `python example.py one two three`:

```
>>> import sys
>>> print(sys.argv)
['example.py', 'one', 'two', 'three']
```

```
import sys

sys.argv[1:]
```

will give you a list of arguments (not including the name of the python file)

You can use `sys.argv` to get the arguments as a list.
If you need to access individual elements, you can use

```
sys.argv[i]
```

where `i` is index, `0` will give you the python filename being executed. Any index after that are the arguments passed.

Python code:

```
import sys

# main
param_1= sys.argv[1]
param_2= sys.argv[2]
param_3= sys.argv[3]
print 'Params=', param_1, param_2, param_3
```

Invocation:

```
$python myfile.py var1 var2 var3
```

Output:

```
Params= var1 var2 var3
```

If you call it like this: `$ python myfile.py var1 var2 var3`

```
import sys
```

```
var1 = sys.argv[1]
var2 = sys.argv[2]
var3 = sys.argv[3]
```

Similar to arrays you also have `sys.argv[0]` which is always the current working directory.

Some additional things that I can think of.

As @allsyed said sys.argv gives a list of components (including program name), so if you want to know the number of elements passed through command line you can use len() to determine it. Based on this, you can design exception/error messages if user didn't pass specific number of parameters.

Also if you looking for a better way to handle command line arguments, I would suggest you look at https://docs.python.org/2/howto/argparse.html