Core Functionality

The argparse module's support for command-line interfaces is built around an instance of argparse.ArgumentParser. It is a container for argument specifications and has options that apply to the parser as whole:

```
parser = argparse.ArgumentParser(
                    prog='ProgramName',
                    description='What the program does',
                    epilog='Text at the bottom of help')
```

The ArgumentParser.add_argument() method attaches individual argument specifications to the parser. It supports positional arguments, options that accept values, and on/off flags:

```
parser.add_argument('filename')           # positional argument
parser.add_argument('-c', '--count')      # option that takes a value
parser.add_argument('-v', '--verbose',
                    action='store_true')  # on/off flag
```

The ArgumentParser.parse_args() method runs the parser and places the extracted data in a argparse.Namespace object:

```
args = parser.parse_args()
print(args.filename, args.count, args.verbose)
```

Quick Links for add_argument()

Name	Description	Values
action	Specify how an argument should be handled	`store', 'store_const', 'store_true', 'append', 'append_const', 'count', 'help', 'version'`
choices	Limit values to a specific set of choices	['foo', 'bar'], range(1, 10), or Container instance
const	Store a constant value	

default	Default value used when an argument is not provided	Defaults to `None`
dest	Specify the attribute name used in the result namespace	
help	Help message for an argument	
metavar	Alternate display name for the argument as shown in help	
nargs	Number of times the argument can be used	int, '?', '*', or '+'
required	Indicate whether an argument is required or optional	`True` or `False`
type	Automatically convert an argument to the given type	`int`, `float`, `argparse.FileType('w')`, or callable function