Options that Conflict

What do you do if you have options that conflict with each other? A common example would be running your application in verbose mode versus quiet mode. You can run it in either mode, but not both. How do we prevent the user from running it that way though? It's actually quite easy via the **mutually_exclusive_group** function. Let's pretend that options **x** and **y** cannot run at the same time and modify our code accordingly:

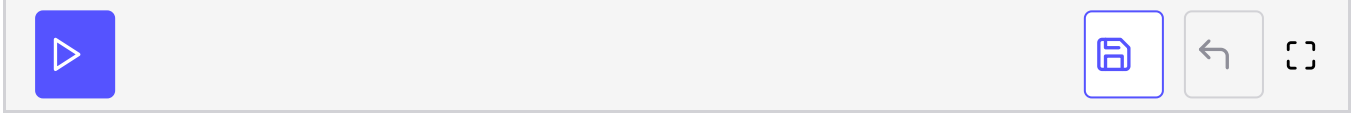
```
import argparse


def get_args():
    """"""
    parser = argparse.ArgumentParser(
        description="A simple argument parser",
        epilog="This is where you might put example usage"
    )

    group = parser.add_mutually_exclusive_group()
    group.add_argument('-x', '--execute', action="store",
                       help='Help text for option X')
    group.add_argument('-y', help='Help text for option Y', default=False)

    parser.add_argument('-z', help='Help text for option Z', type=int)
    print(parser.parse_args())

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



You will note that we have to create a mutually exclusive group. Then we add the options that need to be mutually exclusive to that group. The rest go into the regular parser group. Let's try running the code with both options like this:

```
python arg_demo3.py -x 10 -y 2
```

When I did this command, I ended up getting the following output:

```
usage: arg_demo3.py [-h] [-x EXECUTE | -y Y] [-z Z]
arg_demo2.py: error: argument -y: not allowed with argument -x/--execute
```

Obviously that didn't work and the argparse module told us why.

Wrapping Up

You now know how to create the basics of an argument parser. There are many other aspects of this module that you might be interested in, such as defining an alternate destination name for the argument to be saved to, using different prefixes (i.e. '+' instead of '-'), creating argument groups and more. I recommend checking out the documentation for more details.