Adding Arguments

Let's write some code that adds three new arguments that our parser can understand. We'll add an argument that is required and two that are not. We'll also look at adding a default and a required type. Here's the code:

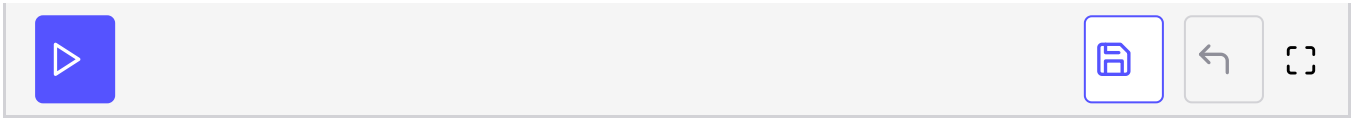
```
# arg_demo2.py

import argparse


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

    # required argument
    parser.add_argument('-x', action="store", required=True,
                        help='Help text for option X')
    # optional arguments
    parser.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()
```



Now let's run it a few times so you can see what happens:

```
mike@pc:~/py/argsparsing$ python arg_demo2.py
usage: arg_demo2.py [-h] -x X [-y Y] [-z Z]
arg_demo2.py: error: argument -x is required

mike@pc:~/py/argsparsing$ python arg_demo2.py -x something
Namespace(x='something', y=False, z=None)

mike@pc:~/py/argsparsing$ python arg_demo2.py -x something -y text
Namespace(x='something', y='text', z=None)
```

```
mike@pc:~/py/argsparsing$ python arg_demo2.py -x something -z text
usage: arg_demo2.py [-h] -x X [-y Y] [-z Z]
arg_demo2.py: error: argument -z: invalid int value: 'text'


mike@pc:~/py/argsparsing$ python arg_demo2.py -x something -z 10
Namespace(x='something', y=False, z=10)
```

As you can see, if you run the code without passing it any arguments, you will get an error. Next we pass it just the required argument so you can see what the defaults are for the other two. Then we try passing "text" to the ‘-y’ argument and that gets stored, so we know it doesn’t require a Boolean. The last two examples show what happens when you pass an invalid and a valid value to the ‘-z’ argument.

By the way, the argument names do not have to be one character in length. You can change those something more descriptive, like ‘arg1 ` or ‘simulator’ or whatever you want.