

# Testing - basic implementation

```
import interpreter

# dictionary from input to expected output
tests = {
    "1": 1,
    "(1)": 1,
    ... # many more tests
}

for i, (test, expected) in enumerate(tests.items()):
    result = interpreter.interpret(test, verbose=verbose)
    if result != expected:
        print("Failed test {}: expected {}, got {} on input {}".format(
            i, expected, result, test
        ))
```

# Additional nice-to-haves

- Pretty-printing for AST nodes - actually display as a kind of "tree", like we've seen with directories previously
- Command-line arguments for verbosity, running test suite, etc.

```
((1 + 2) * (3 + 4))
```

```
MulExp
├─ AddExp
│   ├── Number(1)
│   └── Number(2)
└─ AddExp
    ├── Number(3)
    └── Number(4)
```

```
$ python3 main.py -h
usage: main.py [-h] [--test] [--verbose] [file]
```

Simple Calculator

positional arguments:

file                    Path to source file to interpret

options:

-h, --help            show this help message and exit

--test                Run the test suite

--verbose            Enable verbose output