**Python Shell**

import os

class Shell\_Prompt():

def shell\_prompt(self):

return '$'

def list\_files(self):

return [file for file in os.listdir(os.path.realpath(os.path.join(os.getcwd(), os.path.dirname(\_\_file\_\_)))) if os.path.isfile(file)]

def print\_files(self):

for x in self.list\_files():

print(x)

@staticmethod

def input\_error():

print("Input Error")

return None

@staticmethod

def maths(opp, x, y):

return {

'add': lambda: x + y,

'sub': lambda: x - y,

'mul': lambda: x \* y,

'div': lambda: x / y,

}.get(opp)()

Shell = Shell\_Prompt()

while True:

switch = {

"LIST": lambda: Shell.print\_files(),

"ADD": lambda: print(Shell.maths("add", int(command[1]), int(command[2]))) if len(command) == 3 else Shell.input\_error(),

"SUB": lambda: print(Shell.maths("sub", int(command[1]), int(command[2]))) if len(command) == 3 else Shell.input\_error(),

"MUL": lambda: print(Shell.maths("mul", int(command[1]), int(command[2]))) if len(command) == 3 else Shell.input\_error(),

"DIV": lambda: print(Shell.maths("div", int(command[1]), int(command[2]))) if len(command) == 3 else Shell.input\_error(),

"EXIT": exit

}

command = [x for x in input(Shell.shell\_prompt).split(' ')]

switch.get(command[0], Shell.input\_error)()

**Bandit Report**

Test results:

No issues identified.

Code scanned:

Total lines of code: 33

Total lines skipped (#nosec): 0

Run metrics:

Total issues (by severity):

Undefined: 0

Low: 0

Medium: 0

High: 0

Total issues (by confidence):

Undefined: 0

Low: 0

Medium: 0

High: 0

Files skipped (0):

Potential Security Flaws still exist

* Type Conversion by passing in a type the program is not expecting we could cause the
* Command Extension