Recovering from Errors

In order to stop run-time errors from crashing your programs you can use the try/except statements

In [1]:

```
try:
    number = int(input("Enter an integer: "))
except ValueError:
    print("The is not an integer")
```

```
Enter an integer: a
The is not an integer
```

With the try/except statements the program would crash

In [2]:

```
number = int(input("Enter an integer: "))
```

```
Enter an integer: b

---------------------------------------------------------------------------
--
ValueError                                Traceback (most recent call las
t)
<ipython-input-2-d222549eef79> in <module>
----> 1 number = int(input("Enter an integer: "))

ValueError: invalid literal for int() with base 10: 'b'
```

To handle different type of possible errors you can use more than one except statements

In [3]:

```
try:
    number = int(input("Enter a non-zero integer: "))
    print(f"10 / {number} = {10.0 / number}")
except ValueError:
    print("You did not enter an integer")
except ZeroDivisionError:
    print("You cannot enter 0")
```

```
Enter a non-zero integer: 0
You cannot enter 0
```