

# Lec10-exceptions-RyanSponzilli

September 26, 2024

## 1 ASTR 310 Lecture 10 - Exceptions

### 1.0.1 Exercise 1: Robust type conversion

Write a Python function `convert_type(s)` that takes a string `s` and converts it to a float if it is a number with a decimal point, converts it to an int if it is an integer, or leaves it as a string otherwise, and returns the result. If the input is not a string or anything fails, return `None`. You will want to use exceptions to prevent the code from aborting. Test your function with the following inputs. [6 pts]

```
'-3.4e12'  
'89.1'  
' .9623'  
'42'  
'hello there'  
6  
['6', 'hello']  
'12.3.5'
```

```
[28]: def convert_types(s):  
        try:  
            return int(s)  
        except:  
            pass  
        try:  
            return float(s)  
        except:  
            if type(s) is str:  
                return s  
            else:  
                return None
```

```
[29]: print(convert_types('-3.3e12'))  
print(convert_types('89.1'))  
print(convert_types(' .9623'))  
print(convert_types('-42'))  
print(convert_types('hello there'))  
print(convert_types('6'))  
print(convert_types(['6', 'hello']))
```

```
print(convert_types('12.3.5'))
```

```
-33000000000000.0  
89.1  
0.9623  
-42  
hello there  
6  
None  
12.3.5
```

### 1.0.2 Exercise 2: validating user input

Write Python code that repeatedly prompts the user for two numbers and prints their ratio, halting when the user types anything other than two numbers. If the second number is zero, politely remind the user that you cannot divide by zero, but continue to accept input. Use exception handling.

Demonstrate your code's error handling capability by testing the failure modes. [7 pts]

```
[2]: while True:  
    num1 = input("First Number: ")  
    num2 = input("Second Number: ")  
  
    try:  
        print(f"Ratio: {float(num1)/float(num2)}")  
    except ZeroDivisionError:  
        print("Error: Please don't make me divide by zero.")  
    except:  
        print("Error: Not valid numbers. Aborting")  
        break
```

```
Ratio: 5.0  
Ratio: 9.0  
Ratio: 7.6240513718622305  
Error: Please don't make me divide by zero.  
Error: Please don't make me divide by zero.  
Ratio: 982.2222222222222  
Error: Not valid numbers. Aborting
```

### 1.0.3 Exercise 3: Godzilla

Write Python code that repeatedly requests input to determine Godzilla's next move. It should display the following menu.

Enter Godzilla's next move:

- 1 - Crush Tokyo City Hall
- 2 - Crush Diet Building
- 3 - Crush Tokyo Tower
- 4 - Fight Mothra
- 5 - Return to the sea

If the user enters numbers 1-4, print the corresponding action (e.g. for entry 3 the program should print something like “Godzilla crushes Tokyo Tower. Ahhh!”). If the user enters 5, exit. If the user enters anything else, print a chastising message. (“That’s not allowed.”) Use exception handling, and write your code in a way that makes it easy to add new menu items. Whether or not the user gave valid input, print a blank line before printing the menu again. Test your code and show the desired behavior. [7 pts]

```
[4]: while True:
    nextMove = input("Enter Godzilla's Next Move: ")

    try:
        nextMoveNum = int(nextMove)
        if nextMoveNum == 1:
            print("Godzilla crushes Tokyo city hall!\n")
        elif nextMoveNum == 2:
            print("Godzilla crushes diet building!\n")
        elif nextMoveNum == 3:
            print("Godzilla crushes Tokyo tower!\n")
        elif nextMoveNum == 4:
            print("Godzilla fights Mothra!\n")
        elif nextMoveNum == 5:
            print("Godzilla returns to the sea.\n")
            break
        else:
            raise ValueError
    except:
        print("That's not allowed!\n")
```

Godzilla crushes Tokyo city hall!

Godzilla crushes diet building!

Godzilla crushes Tokyo tower!

Godzilla fights Mothra!

That's not allowed!

That's not allowed!

Godzilla returns to the sea.