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 anything fails, return None. You will want to use exceptions to prevent the code from aborting. Test your function with the inputs

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

$[6pts]$

In [3]:
```python
#Code Here
def convert_type(s):
    try:
        float_conversion = float(s)
        if float_conversion.is_integer():
            return int(float_conversion)
        else:
            return float_conversion
    except ValueError:
        return s

    except TypeError:
        return None


test_inputs = ['-3.4e12', '89.1', '.9623', '42', 'hello there', 6, ['6', 'hello']]
test_results = [convert_type(i) for i in test_inputs]

test_results
```

Out[3]:
```
[-3400000000000, 89.1, 0.9623, 42, 'hello there', 6, None]
```

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. $[7pts]$

In [6]:
```python
#Code Here
#Code Here
while True:
    try:
        num1 = input("Enter Value 1: ")
```

```
        num2 = input("Enter Value 2: ")

        num1 = float(num1)
        num2 = float(num2)

        if num2 == 0:
            print("Cannot divide by zero, please try again.")
            continue

        r = num1 / num2
        print(f"The ratio is: {r} for {num1} and {num2}")

    except ValueError:
        print("Non-numeric input detected, exiting.")
        break
```

```
The ratio is: 2.0 for 8.0 and 4.0
The ratio is: 1408.4558491340954 for 123456789.0 and 87654.0
The ratio is: 83666477415.94597 for 5.678987654323457e+16 and 678765.0
Cannot divide by zero, please try again.
The ratio is: 180.3846153846154 for 2345.0 and 13.0
Non-numeric input detected, exiting.
```

Exercise 3: Godzilla

Write Python code that repeatedly requests input to determine Godzilla's next move. It should display the 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 ("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. $[7pts]$

In [10]:
```
def godzilla_game():
    actions = {
        '1': 'Crush Tokyo City Hall',
        '2': 'Crush Diet Building',
        '3': 'Crush Tokyo Tower',
        '4': 'Fight Mothra'
    }

    while True:
        print("\nEnter Godzilla's next move:")
        print("1 - Crush Tokyo City Hall")
        print("2 - Crush Diet Building")
```

```
        print("3 - Crush Tokyo Tower")
        print("4 - Fight Mothra")
        print("5 - Return to the sea")

        try:
            choice = input()
            if choice == '5':
                print("Godzilla returns to the sea.")
                break
            elif choice in actions:
                print(f"Godzilla {actions[choice]}. Ahhh!")
            else:
                raise ValueError("That's not allowed.")
        except ValueError as e:
            print(e)
        # print("\n")
godzilla_game()
```

```
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
That's not allowed.

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
Godzilla Crush Tokyo Tower. Ahhh!

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
Godzilla returns to the sea.
```

In [ ]:

```
%%capture
# Here we use a script to generate pdf and save it to google drive.

# After executing this cell, you will be asked to link to your GoogleDrive account.
# Then, the pdf will be generated and saved to your GoogleDrive account and you nee

from google.colab import drive
drive.mount('/content/drive')
# install tex; first run may take several minutes
! apt-get install texlive-xetex
# file path and save location below are default; please change if they do not match
! jupyter nbconvert --output-dir='/content/drive/MyDrive/' '/content/drive/MyDrive/
```