In [1]:
```
###Module 03: Live Session Exercise: Exception Handling###
## Kip McCharen (cam7cu)
import os
import time
import random
```

In [2]:
```
def doerrors(a, yourname="", encouragement=""):
    """Function to handle errors correction from arguments passed in as
        text. Username and encouragement text accepted. """
    #Creating anonymous function which will print:
    #   1. input that failed, 2. error class name, 3. the error message
    erout = lambda x,y: print(f"\tFailed to execute: {x}\n{type(err).__name__}: {y}")
    try:
        exec(a) #First try to execute the input passed as a string
    #There does not appear to be a way to automatically generate
    # this long list of the different error exceptions, oh well.
    except TypeError as err:
        erout(a,err) #Call the anonymously defined function erout()
    except ValueError as err:
        erout(a,err)
    except ZeroDivisionError as err:
        erout(a,err)
    except FileNotFoundError as err:
        erout(a,err)
    except IOError as err:
        erout(a,err)
    except ArithmeticError as err:
        erout(a,err)
    except Exception as err:
        erout(a,err)
    finally:
        if yourname != "" and encouragement != "":
            #If a name and encouragement were passed, print those
            print(f"\t{yourname}, {encouragement}")
        else:
            print("\tRotten luck.") #Otherwise print Rotten luck.
        #Add an extra line space to make output more clear,
        # and wait 2 seconds so users can read the output as it happens
        print("\n")
        time.sleep(2)
```

In [3]:

```
def interact_with_user():
    """Function to interact positively with the user, and call the
        doerrors() function. At random chance, a randomized
        encouragement string is added after each error from an
        iterator of short sentences. """
    #Initialize variables to do all the things

    #1. List of encouragements
    encouragement = ["Hang in there.    ", "Don't give up.", \
    "Keep pushing.", "Keep fighting!", "Stay strong.", "Never give up.", \
    "Never say 'die'.", "You can do it!", "Follow your dreams.", \
    "Reach for the stars.", "  Do the impossible.", \
    "Always believe in yourself. ", "    The sky is the limit."]
    #Shuffle encouragements so we don't know which one comes first
    random.shuffle(encouragement)
    #Strip any whitespace and make lowercase to append to a sentence
    encouragement = iter([x.lower().strip() for x in encouragement])

    #2. List of inputs to run and call the errors in order
    args = [r"'2'+2", r"int('xyz')", r"x=100/0", r"open('madeupfile.txt')", \
        r"os.remove(filepath = os.path.dirname(os.path.abspath(__file__)))", \
            "2.718281 ** 1000", "print 'whoooo'"]

    #Now talk to the user!
    yourname = input("Hi there, what's your name?    -> ")
    yourname = "Gerethy"
    print(f"great! J/K I'm a computer not a person. " +
        f"I'll just call you {yourname}.")
    time.sleep(2)
    print(f"OK {yourname}, let's test some errors here.\n")
    time.sleep(2)
    for a in args: #iterate through list of inputs to execute
        #random chance to insert the name and encouragement
        if bool(random.getrandbits(1)):
            doerrors(a, yourname, next(encouragement))
        else:
            doerrors(a)
    #finish the session positively
    time.sleep(2)
    print(f"Well {yourname}, successfully produced errors! " +
        f"Is that a success or not?")
    time.sleep(3)
    print(f"Honestly, who's to say. Thanks for your help anyways. " +
        f"Don't forget, {next(encouragement)}")
```

In [4]:
```
interact_with_user()
```

```
great! J/K I'm a computer not a person. I'll just call you Gerethy.
OK Gerethy, let's test some errors here.

        Failed to execute: '2'+2
TypeError: can only concatenate str (not "int") to str
        Gerethy, never say ‘die’.


        Failed to execute: int('xyz')
ValueError: invalid literal for int() with base 10: 'xyz'
        Rotten luck.


        Failed to execute: x=100/0
ZeroDivisionError: division by zero
        Rotten luck.


        Failed to execute: open('madeupfile.txt')
FileNotFoundError: [Errno 2] No such file or directory: 'madeupfile.txt'
        Rotten luck.


        Failed to execute: os.remove(filepath = os.path.dirname(os.path.abspath(__file_
_)))
NameError: name '__file__' is not defined
        Gerethy, hang in there.


        Failed to execute: 2.718281 ** 1000
OverflowError: (34, 'Result too large')
        Gerethy, do the impossible.


        Failed to execute: print 'whoooo'
SyntaxError: Missing parentheses in call to 'print'. Did you mean print('whoooo')? (<st
ring>, line 1)
        Gerethy, follow your dreams.


Well Gerethy, successfully produced errors! Is that a success or not?
Honestly, who's to say. Thanks for your help anyways. Don't forget, you can do it!
```

In [ ]:
```

```