SECTION 10: ERRORS AND EXCEPTIONS HANDLING - 46 minutes, 6 parts

3/6 Errors and Exception Homework - Solutions

- **Error handling for squaring strings**
  - -> we are handling the errors and exceptions
  - -> running the code alone returns an error message <- if we try and square a string
  - -> run this code, if it does not work, then return this error message

```
]: try:
       for i in ['a','b','c']:
           print(i**2)
   except:
       print("General error! Watch out!")
```

```
General error! Watch out!
```

- **Using a "finally" block to print "all done"**
  - -> in the second example, we have a block of code which we have now indented inside an except statement

```
try:
    x = 5
    y = 0
    z = x/y
except:
    print("Error!!")
finally:
    print("All done")
```

- **Printing out an integer squared, with a while loop**
  - -> we are defining a function
  - -> inside this, a try block
  - -> we are asking for an integer
  - -> then printing out an error statement in case of exception
  - -> we need break statements when we define while loops
  - -> we can also use waiting instead <- while waiting
  - -> it is waiting for the correct response
  - -> then in the problem solving approach, we always test our solution in order to see if if works or not

```
]: def ask():

    while True:
        try:
            n = int(input("Enter a number"))
        except:
            print("Please try again! \n")
            continue
        else:
            break

    print("Your number squared is: ")
    print(n**2)
```

```
: def ask():

    # Waiting for correct response
    waiting = True
    while waiting:
        try:
            n = int(input("Enter a number"))
        except:
            print("Please try again! \n")
            continue
        else:
            waiting = False

    print("Your number squared is: ")
    print(n**2)
```