Using Try/Except Blocks for Error Handling

The Python interpreter executes codes in try block first. If there are any errors occured in try block, then codes below except block will executes.

```
try:
    f = open('testfile.txt')
except Exception:
    print("Sorry. This file doesn't exist")
# else:
# finally:
```

If we specify the name of raising error in except statement, then the error specified in except statement will be raised among the raised errors.

```
try:
    f = open('testfile.txt')
    var = bad_vars
except FileNotFoundError as e:
    print(e)
    print("Sorry. This file doesn't exist")
except Exception as e:
    # More general errors
    print("Sorry. Something went wrong.")
# else:
# finally:
```

The codes under the else block is executed if no error has occured.

```
try:
    f = open('testfile.txt')
    var = bad_vars
except FileNotFoundError as e:
    print(e)
    print("Sorry. This file doesn't exist")
except Exception as e:
    print("Sorry. Somethig went wrong.")
else :
    print(f.read())
    f.close()
```

The codes under the finally block is executed whether or not an error occurs. The finally statment is mainly used to terminate a database connection.

```
try:
    f = open('testfile.txt')
    var = bad_vars
except FileNotFoundError as e:
    print(e)
    print("Sorry. This file doesn't exist")
except Exception as e:
    print("Sorry. Somethig went wrong.")
else :
    print(f.read())
    f.close()
finally:
    print("Executing Finally...")
```

In [ ]: