# Debugging

## Outline

Debugging

- Exception handling
- Assertions
- Python debugger

## Try to run some code & it breaks

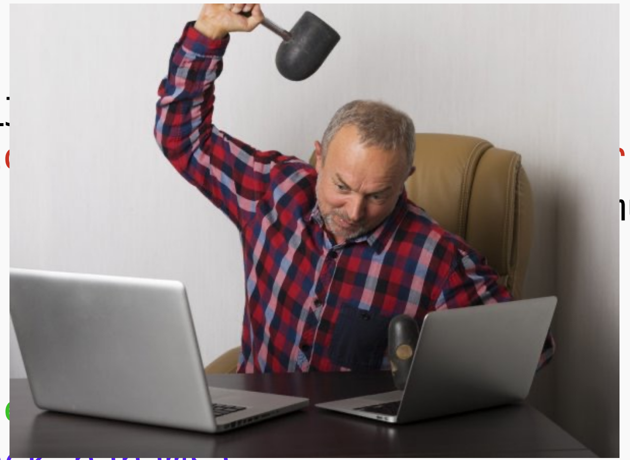
```
In [4]: model = LinearRegression()


In [5]: model.fit(5,100)
>>>

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-5-9e6d49d34619> in <module>
----> 1 model.fit(5,100)


~/opt/anaconda3/lib/python3.8/site-packages/sklearn/linear_model/_base.py in fit(self, X, y,
sample_weight)
    503
    504         n_jobs_ = self.n_j
--> 505         X, y = self._vali                             rse=['csr', 'csc', 'coo'],
    506                                                      ulti_output=True)
    507


~/opt/anaconda3/lib/python3.8/site                          in _validate_data(self, X, y,
reset, validate_separately, **check_params)
```



## The anatomy of broken code

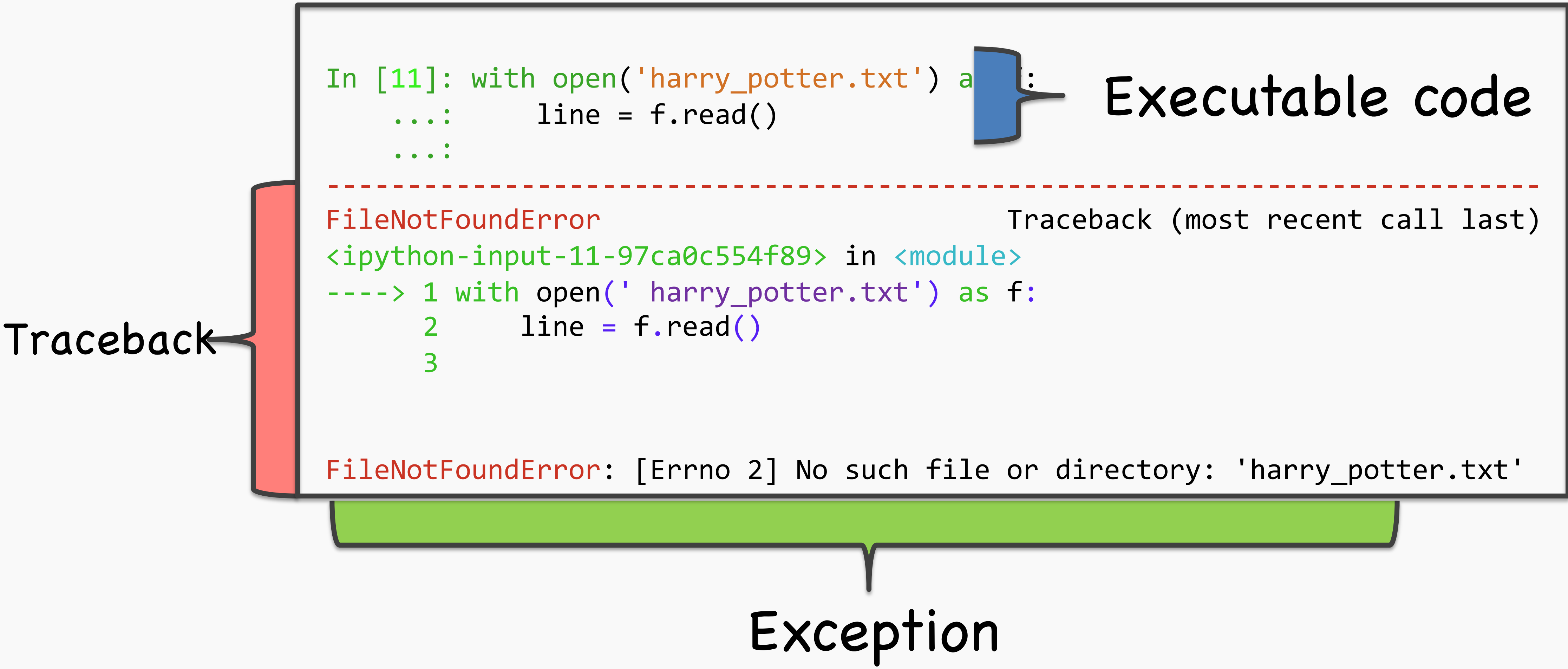
```
In [11]: with open('harry_potter.txt') as f:
    ...:     line = f.read()
    ...:
```

Executable code

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-11-97ca0c554f89> in <module>
----> 1 with open('harry_potter.txt') as f:
      2     line = f.read()
      3


FileNotFoundError: [Errno 2] No such file or directory: 'harry_potter.txt'
```

## The anatomy of broken code



```
In [11]: with open('harry_potter.txt') a    :
    ...:     line = f.read()
    ...:
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-11-97ca0c554f89> in <module>
----> 1 with open(' harry_potter.txt') as f:
      2     line = f.read()
      3


FileNotFoundError: [Errno 2] No such file or directory: 'harry_potter.txt'
```

## Traceback

## Traceback

Defining the inception function

```
# Call stack
   ...: def inception():
   ...:     dream1()
   ...:
   ...:
   ...: def dream1():
   ...:     dream2()
   ...:
   ...:
   ...: def dre
   ...:     dre
   ...:
   ...:
   ...: def dream3():
   ...:     raise
Exception("Dream is collapsing")
```

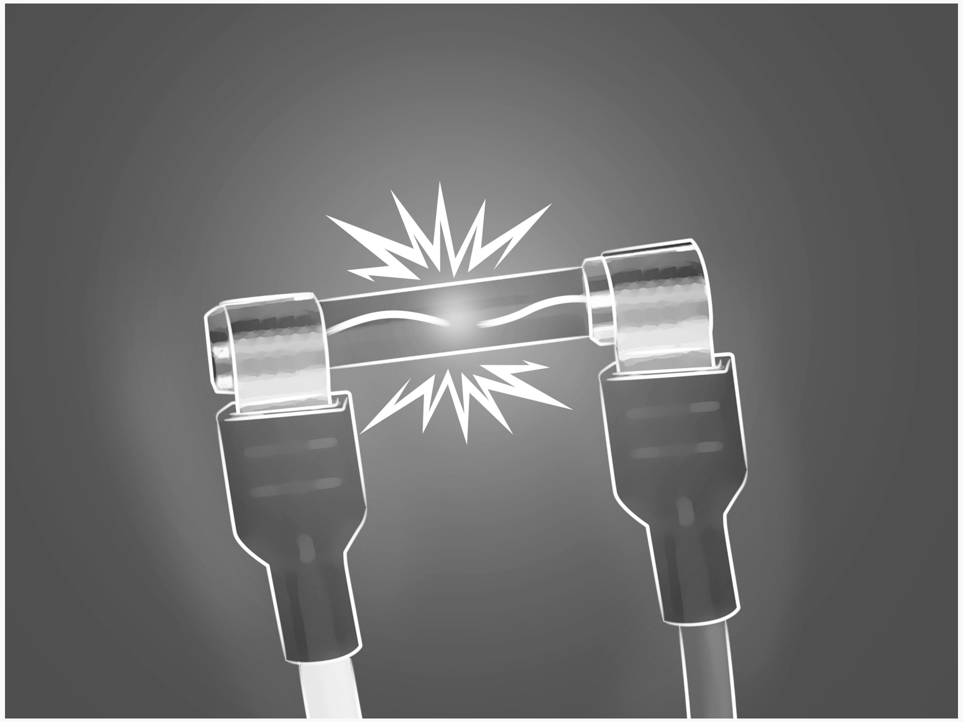
What is an Exception

```
In [13]:
inception()
---------------------------------------------------------------------------
Exception                                 Traceback (most recent call last)
<ipython-input-13-fce33c2cc0f5> in <module>
----> 1 inception()
~/Dropbox/PyDS/Session6/debugging.py in inception()
      3 # Call stack
      4 def inception():
----> 5     dream1()
      6
      7
~/Dropbox/PyDS/Session6/debugging.py in dream1()
      7
      8 def dream1():
----> 9     dream2()
     10
     11
            ion6/debugging.py in dream2()
                 ):
---> 13     dream3()
     14
~/Dropbox/PyDS/Session6/debugging.py in dream3()
     15
     16 def dream3():
---> 17     raise Exception("Dream is collapsing")
     18
Exception: Dream is collapsing
```

## Exception

## Exception

- An Exception is like fuse, set in place, to avoid your code from doing something that it isn't supposed to do.
- For e.g., if you try to divide by zero, python will raise a `ZeroDivisionError` exception.
- Like python functions, there are some built-in exceptions, but you can raise your own as well.



## Exceptions in Python

KeyboardInterrupt

SyntaxError

NameError

IndentationError

IndexError

KeyError

ZeroDivisionError

FileNotFoundError

AssertionError

## Errors in Python

KeyboardInterrupt

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-3-ffdadd16ced2> in <module>
      1 while True:
----> 2     print('hello')
      3


KeyboardInterrupt:
```

## Errors in Python

KeyboardInterrupt

SyntaxError

NameError

IndentationError

IndexError

KeyError

ZeroDivisionError

FileNotFoundError

AssertionError

## Errors in Python

SyntaxError

```
for i in range(10)
>>>
--------------------------------------------------------------------------
  File "<ipython-input-5-9bf3d452bb2a>", line 1
    for i in range(10)
                      ^
SyntaxError: invalid syntax
```

## Errors in Python

| | | |
|---|---|---|
| KeyboardInterrupt | SyntaxError | NameError |
| IndentationError | IndexError | KeyError |
| ZeroDivisionError | FileNotFoundError | AssertionError |

## Errors in Python

NameError

```
for i in varlist:
   ...:     print(i)
>>>
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-6-aaa6b2ddcca4> in <module>
----> 1 for i in varlist:
      2     print(i)
      3


NameError: name ‘varlist' is not defined
```

## Errors in Python

- KeyboardInterrupt
- SyntaxError
- NameError
- IndentationError
- IndexError
- KeyError
- ZeroDivisionError
- FileNotFoundError
- AssertionError

## Errors in Python

AssertionError

```
assert condition, (optional)Message to print
```

```
assert 5 > 6, ‘It is greater'
>>>
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
<ipython-input-7-e5860cc9eaa5> in <module>
----> 1 assert 5 > 6, 'It is greater'


AssertionError: It is greater
```

## Raising Exceptions

You can raise exceptions in your own code using the keyword `raise`

```
raise Exception('Optional Message')
```

## Raising Exceptions

You can raise exceptions in your own code using the keyword `raise`

```
raise Exception('Optional Message')
```

```
def goalspermatch(matches: int, goals :int):
   ...:     return goals/matches

goalspermatch(matches = 3,goals = 7.6)
>>>
2.5333333333333333
```

```
def goalspermatch(matches: int, goals :int):
    ...:     if (type(goals)==int):
    ...:         return goals/matches
    ...:     else:
    ...:         raise Exception('Goals cannot be a non-integer value')

goalspermatch(matches = 3,goals = 7.6)
```

## Raising Exceptions

You can raise exceptions in your own code using the keyword `raise`

```
def goalspermatch(matches: int, goals :int):
   ...:     return goals/matches

goalspermatch(matches = 3,goals = 7.6)
>>>
2.5333333333333333
```

```
def goalspermatch(matches: int, goals :int):
    ...:     if (type(goals)==int):
    ...:         return goals/matches
    ...:     else:
    ...:         raise Exception('Goals cannot be a non-integer value')

goalspermatch(matches = 3,goals = 7.6)
>>>

---------------------------------------------------------------------------
Exception                                 Traceback (most recent call last)
<ipython-input-22-4072d42f3a05> in <module>
----> 1 goalspermatch(matches=3,goals=7.6)


<ipython-input-21-7df8c4ea59d6> in goalspermatch(matches, goals)
      3         return goals/matches
      4     else:
----> 5         raise Exception('Goals cannot be a non-integer value')
      6


Exception: Goals cannot be a non-integer value
```

## How to make your code run ?

## Try/Except block

## Try/Except

If at first your code doesn't run, try again

- The `try/except` block allows you to skip code if it encounters an exception.
- Like an `if... else` block, it skips execution to the `except` part of the code and continues execution.
- The Except block can be modified to account for a specific type of error as well (e.g. `ZeroDivisionError`)

```
Some code here
try:
    statement 1
    statement 2
    statement n
except:
    Do something
Rest of the Code
```