

Step0 - File Input and Exception Handling

May 8, 2018

0.0.1 File Input/Output

File is a named location on disk to store related information. When we want to read from or write to a file we need to open it first. When we are done, it needs to be closed, so that resources that are tied with the file are freed.

So in Python, a file operation takes place in the following order. 1. Open a file 2. Read or write (perform operation) 3. Close the file

```
In [4]: # Read input from console, input will read numeric values and throw error if a string is
        num = input("Enter a number: ")
        print type(num)
```

```
Enter a number: 2
<type 'int'>
```

```
In [2]: # Read input from console, raw_input will read input as string
        num = raw_input("Enter a number: ")
        print type(num)
```

```
Enter a number: 2
<type 'str'>
```

0.0.2 Writing to a file

```
In [ ]: # Set working directory
        import os

        # Set working directory
        os.chdir('C:\\Users\\Manoh\\Documents')

        # Below code will create a file named vehicles and add the items. \n is a newline character
        vehicles = ['scooter\n', 'bike\n', 'car\n']
        f = open('vehicles.txt', 'w')
        f.writelines(vehicles)
        f.close()
```

0.0.3 Reading from file

```
In [ ]: f = open('vehicles.txt')
        print f.readlines()
        f.close()
```

0.0.4 Exception Handling

An exception is an error that happens during the execution of a program that disrupts the normal flow of the program's instructions. When a Python script encounters a situation that it cannot cope with, it raises an exception. Your program should be designed to handle both expected and unexpected errors.

try and except A critical operation which can raise exception is placed inside the try clause and the code that handles exception is written in except clause

```
In [11]: import sys

        try:
            a = 1
            b = 1
            print "Result of a/b: ", a / b
        except (ZeroDivisionError):
            print("Can't divide by zero")
        except (TypeError):
            print("Wrong data type, division is allowed on numeric data type only")
        except:
            print "Unexpected error occurred", '\n', "Error Type: ", sys.exc_info()[0], '\n',
```

Result of a/b: 1

```
In [12]: try:
            a = 1
            b = 0
            print(a / b)
        except (ZeroDivisionError):
            print("Can't divide by zero")
        except (TypeError):
            print("Wrong data type, division is allowed on numeric data type only")
        except:
            print "Unexpected error occurred", '\n', "Error Type: ", sys.exc_info()[0], '\n',
```

Can't divide by zero

```
In [13]: try:
            a = 1
            b = 0
```

```

    print(A / b)
except (ZeroDivisionError):
    print("Can't divide by zero")
except (TypeError):
    print("Wrong data type, division is allowed on numeric data type only")
except:
    print "Unexpected error occurred", '\n', "Error Type: ", sys.exc_info()[0], '\n',

```

```

Unexpected error occurred
Error Type: <type 'exceptions.NameError'>
Error Msg:  name 'A' is not defined

```

Finally This is an optional clause which is intended to define clean-up actions that must be executed under all circumstances.

```

In [16]: try:
        f = open('C:\\Users\\Manoh\\Documents\\vechicles.txt')
        print f.readline()
        i = int(s.strip())
    except IOError as e:
        print "I/O error({0}): {1}".format(e.errno, e.strerror)
    except ValueError:
        print "Could not convert data to an integer."
    except:
        print "Unexpected error occurred", '\n', "Error Type: ", sys.exc_info()[0], '\n',
    finally:
        f.close()
        print "file has been closed"

```

```
scooter
```

```

Unexpected error occurred
Error Type: <type 'exceptions.NameError'>
Error Msg:  name 's' is not defined
file has been closed

```

Reference: Mastering machine learning with python in six-steps