# Python 3 Exceptions bitio.py Quick Reference

Eddie Guo

October 2019

## 1 Introduction to bitio.py

### 1.1 Topics Covered

(i) BitReader

(ii) EOFError

(iii) BitWriter

## 2 BitReader

- `BitReader(input_stream)`
  - Creates instance of BitReader class
  - This obj will read from input_stream
- `readbit()` reads next bit in input_stream, & returns it as 1 or 0
- `readbits(n)` reads n bits & returns them as seq of bits eval as integer

```
import bitio

with open('simple.txt', 'rb') as fin:
    mybitreader = bitio.BitReader(fin)

    # read in a byte, one bit at a time
    for i in range(8):
        my_bit = mybitreader.readbit()
        print(my_bit, end = '')
    print()

    # read in a byte all at once
    my_bite = mybitreader.readbits(8)
    print(my_byte)
```

### 2.1 How to Read to End of File?

- EOFError raised when there are no more bits to read from the input_stream

```
import bitio

with open('simple.txt', 'rb') as fin:
    mybitreader = bitio.BitReader(fin)
    end_of_file = False

    while note end_of_file:
        try:
            bit = mybitreader.readbit()
            print(bit, end = '')
        except EOFError:
            end_of_file = True
```

## 3 BitWriter

- `BitWriter(output_stream)`
  - Creates instance of BitWriter class
  - This obj will write to output_stream
- `writebit(bit)`
  - If bit is True, writes 1 to output_stream
  - If bit is False, write 0 to output_stream
- `writebits(integer_value, n)` writes the n least significant bits of integer_value to output_stream starting w/ most significant of these bits
- `flush()`
  - Forces any bits waiting in buffer to output_stream
  - ALWAYS call when finished writing to write any partial bytes to output_stream
  - Any incomplete bytes automatically padded w/ extra 0s in least significant bits

```
import bitio

seq1 = '01101000'
seq2 = [ord('E'), ord('L'), ord('L'), ord('O')]

with open('message.txt', 'wb') as fout:
    mybitwriter = bitio.BitWriter(fout)

    for single_bit in seq1:
        if single_bit == '1':
            mybitwriter.writebit(True)
        elif single_bit == '0':
            mybitwriter.writebit(False)
    mybitwriter.flush()  # don't forget at the end!

    for single_byte in seq2:
        mybitwriter.writebits(single_byte, 8)
    mybitwriter.flush() # don't forget at the end!

# output
hELLO
```