## Exceptions

* A way to deal with certain exceptional situations (disk errors, printer on fire, divide by zero, etc.)
* ‘raise’ is how to deal with these exceptions
  + E.g., “raise IndexError(‘Hello!’)” will handle an IndexError
* The exception is “raised” all the way up to the Call Stack until it is handled
* Best practice is for an exception to be handled at the highest point of the call stack, or whoever called the original function
* Do not handle an exception unless you have a method to recover from it

## Files

* Use the built-in “open” function to access a file on the file system
* Use the “read” function to read the entire file
* A file should be closed if the program remains running to release the file handle
* As you read portions of a file object, the previously used portions are discarded
* The “readline” function reads the first line of a file – calling it again reads the next line, etc.
  + “readlines” inserts all lines in the file as a list
* The “strip” function can be used to remove CR or LF from end of a line
* To open a file for writing, pass w as the second argument to the open function – f = open(‘out.txt’,’w’)
* The write function passes its input and writes it to the file
  + A file must be closed in order to commit changes to disk
  + TIP: Use “finally:” to close the file
* To open a file as binary, use ‘rb’ as second argument to open
  + Binary data behaves exactly like a string with a “b” at the front
  + If a string contains non-ASCII bits, it must be stored as binary to handle the encoding
  + The “encode” function will convert the string to the specified encoding
  + The “decode” function will convert a binary object back to a string