Week 11 - interpreting error messages, debugging

Math16b

1 List of Common Errors

1. ```
   NameError: name 'some_name' is not defined
   NameError: global name 'some_name' is not defined
   ```

   This error occurs when you use a variable name which has not been declared before.

2. ```
   IndentationError: unindent does not match any outer indentation level
   IndentationError: unexpected indent
   IndentationError: expected an indented block
   ```

   In order to indent the code correctly, one must consistently use one and only one of either: tab or: four spaces. These errors occur if there are tabs mixed with four spaces. These can also occur if there is inconsistent space-indenting within a single block.

3. ```
   SyntaxError: invalid syntax
   ```

   These can happen in a variety of sitations. Some of them are:

   - Forgetting the parens around the arguments to print
   - Forgetting the colon at the end of the condition in an if statement
   - Trying to use a reserved word as a variable name

4. ```
   TypeError: unsupported operand type(s) for +: 'int' and 'str'
   ```

   A type error can occur when you do an operation on a data type which is not correct. For example, in the above shown error, the problem arises because int+string does not make sense.

5. ```
   AttributeError: 'module' object has no attribute 'sparse'
   ```

   An attribute in Python means some property that is associated with a particular type of object. In other words, the attributes of a given object are the variables and functions it has in it. Attribute errors in Python are generally raised when you try to access or call an attribute that a particular object type doesn't possess.

6. ```
   ValueError: could not convert string to float: 'string'
   ```

   A Value error is raised when a built-in operation or function receives an argument that has the right type but an inappropriate value. In the above example, the float function can take a string, ie float('5'). The error arises when the value 'string' in float('string') is an inappropriate (non-convertible) string.
   Or for example when we try to add two numpy arrays of different sizes.

7. ```
   KeyError: 'name'
   ```

   These errors occur when dealing with python dictionaries. A keyError raised when a dictionary key is not found in the set of existing keys.

8. ```
   IndexError: list index out of range
   ```

   This error occurs when we are trying to refer to some index that doesn't exist.

9. ```
   ModuleNotFoundError
   ```

   This occurs when a module is not found.
   ```

10. `ImportError`

    This occurs when a specified function can not be found.

11. `ZeroDivisionError: division by zero`

    This occurs when you divide by zero.

2 Exercises

The code snippets below raise one of the above errors. For each code snippet, indicate the error it raises.

1. ```
   for i in range(10)
       print(i**2)
   ```

2. ```
   import numpy as np
   A = numpy.array([1,2,3])
   ```

3. ```
   a = 1
   print(A)
   ```

4. ```
   l = ['a',2, True]
   print(l(0))
   ```

5. ```
   ls = [1,2,3,4]
   for i in ls:
       print(ls[i]+ls[i+1])
   ```

6. ```
   print(2+'2')
   ```

7. ```
   l = 1
   l.append(3)
   ```
   ```