**ITP 125 – Homework #3**

**Tanuja Mohan**

**Automate the Boring Stuff**

**Chapter 4**

1. [ ] signifies that a list is being created. A list begins with the opening square bracket and ends with a closing square bracket.
2. spam[2] = ‘hello’
3. It evaluates to the value 3
4. Since negative indexes count from the end this evaluates to ‘d’
5. It evaluates to [‘a’, ‘b’]
6. Evaluates to 1
7. bacon now contains [3.14, ‘cat’, 11, ‘cat’, True, 99]
8. bacon now contains [3.14, 11, ‘cat’, True]
9. List concatenation operator is + and the operator for replication is \*
10. Append adds values to the end of the list and insert can add a value anywhere in the list depending on what value is passed in as a parameter
11. You can delete a value in a list by either using the del statement or the remove() function.
12. You can call the len() function on both lists and string and both have index values. They both can also be concatenated or replicated.
13. Lists are mutable meaning that you can add, remove, and change values in them and are written using square brackets. Tuples are immutable meaning that we cannot change them at all and are written using parentheses.
14. (42,)
15. tuple() and list()
16. The variables contain references to the list values
17. Copy.copy() does a shallow copy of the list and copy.deepcopy() does a deep copy of the list.

**Chapter 5**

1. An empty dictionary just has {}
2. {‘foo’: 42}
3. The difference between a dictionary and a list is that a dictionary is unordered while a list is ordered
4. Trying to access a key that does not exist gives a KeyError error
5. There is no difference because the in operator checks to see if a value exists as a key in the dictionary
6. ‘cat’ in spam checks to see if ‘cat’ is a key in the dictionary and ‘cat’ in spam.values() checks if ‘cat’ is a value for a key in the dictionary
7. spam.setdefault(‘color’, ‘black’)
8. pprint.pprint()

**Chapter 6**

1. Escape characters allow us to represent string character values that are impossible/difficult to type into code.
2. \n is the newline character and \t is the tab character
3. \\ represents a backslash character
4. It is not a problem because we used double quotes to represent the beginning and end of the string
5. You can use multiline strings instead
6. ‘e’, ‘Hello’, ‘Hello’, ‘lo world!’
7. ‘HELLO’, True, ‘hello’
8. [‘Remember, ‘ , ‘remember, ‘ , ‘the’, ‘fifth’, ‘of’, ‘ November.’] and ‘There-can-be-only-one.’
9. Right justify is rjust(), left justify is ljust(), and to center is center()
10. You can trim whitespace from the beginning using lstrip() and from the end by using rstrip()