Andrew Pederson

Stretch Journal

7301 Intro to IT

Week 3: To start this project, I purchased the Udemy course *Complete Python Bootcamp: Go from zero to hero in Python 3.* I had no knowledge of Python. The first thing I did was watch the course introduction videos and learned a bit of background information on Python. Then I got Python set-up on my computer. To do this, I had to learn new Command Line terms. I've always wanted to learn more about the Command Line, so that was fun! Then I got Jupyter Notebook installed on my computer, which is what I'll be using to code.

Jupyter Notebook (Anaconda3)   2/9/2020 5:28 PM   Shortcut   3 KB

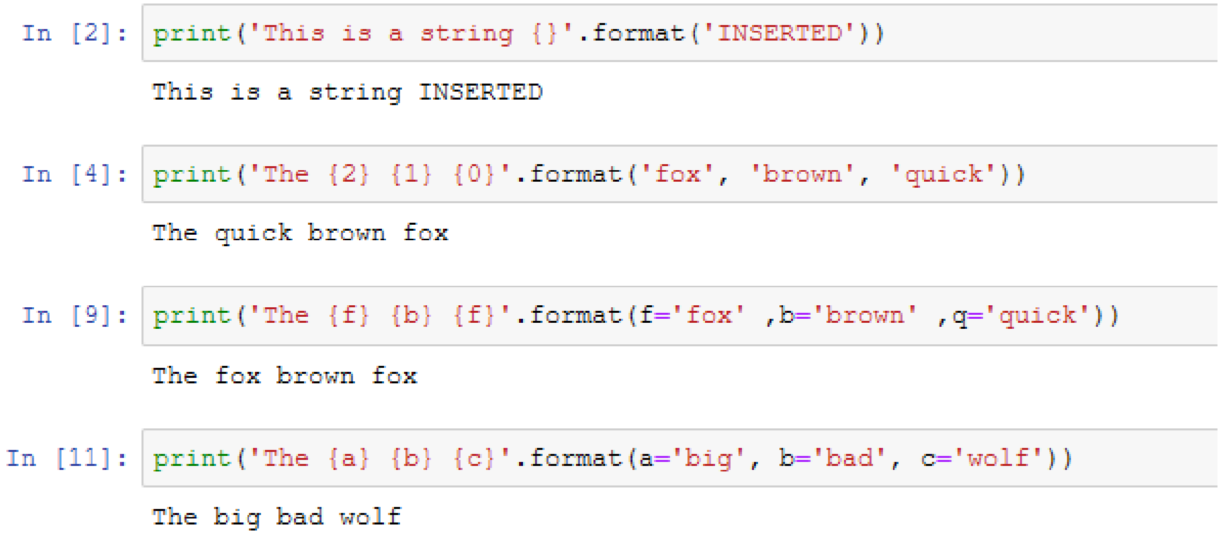
Week 4: This week I started to learn some basic Python. I learned about the different Python Data Types, which are integers, floating points, strings, lists, dictionaries, tuples, sets, and Booleans. I also had my first few Coding Exercise.

Week 5: Strings are sequences of characters in either apostrophes (' ') or quotes (" "). You can print strings by using the print function. Example: print("Hello World"). I also learned how to slice and index strings.

Index Example: 'Hello World'[-3] returns the letter 'r'. d =- 1, l = -2, r = -3

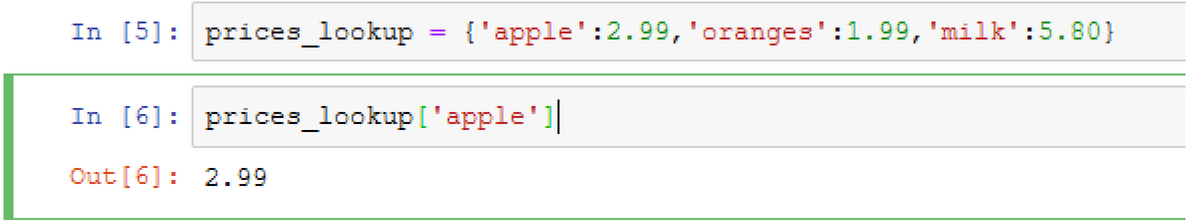
Slice Example: 'tinker'[1:4] returns the phrase 'ink' (characters 2, 3, 4). The slice goes up to the variable after the colon.

Week 6: I learned how to do print formatting with strings. This section was significantly denser than the first few sections. I got through it, but I will need to revisit it later. I learned the .format() method and the f-string (formatted string literals) method. Example (string interpolation):



```
In [2]: print('This is a string {}'.format('INSERTED'))

        This is a string INSERTED

In [4]: print('The {2} {1} {0}'.format('fox', 'brown', 'quick'))

        The quick brown fox

In [9]: print('The {f} {b} {f}'.format(f='fox' ,b='brown' ,q='quick'))

        The fox brown fox

In [11]: print('The {a} {b} {c}'.format(a='big', b='bad', c='wolf'))

        The big bad wolf
```

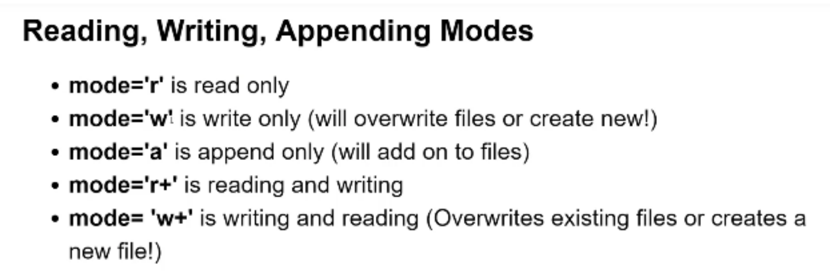
Week 7: This week I learned about lists, dictionaries, sets, and tuples. Lists are ordered sequences that can hold a variety of object types, like strings, numbers, floating points, etc. I learned how to append items on to lists by using the .append() method and remove items with the .pop() method. I also learned how to sort lists with the .sort() method. Dictionaries are like lists, but they are unordered and use key-pairings to assign values to objects. This lets you call objects without knowing where they exist in the index. Dictionaries can contain numbers, or even other dictionaries! Example dictionary that I made:



```
In [5]: prices_lookup = {'apple':2.99,'oranges':1.99,'milk':5.80}

In [6]: prices_lookup['apple']

Out[6]: 2.99
```

Tuples are list that are "immutable," meaning they can't be changed once their index is set. Tuples use parentheses, unlike lists which use brackets. Sets are like lists but all the variables are unique.

Week 8: This week I learned about Booleans, which are just true or false statements. Python requires "True" and "False" to be capitalized to be considered Booleans. I learned how to create .txt files inside of Jupyter Notebooks. Then, I learned how to interact with a .txt file inside of Jupyter Notebooks using these modes:



Reading, Writing, Appending Modes

- **mode='r'** is read only
- **mode='w'** is write only (will overwrite files or create new!)
- **mode='a'** is append only (will add on to files)
- **mode='r+'** is reading and writing
- **mode= 'w+'** is writing and reading (Overwrites existing files or creates a new file!)

I ended today’s session by taking the test on this section. It required a lot of review; there were parts of the test that I did not remember at all. I think I’ll take this test again next week. Hopefully, I’ll have retained more of it! Today I spent the longest time working on this stretch assignment (3+ hours).

Week 9: This week I took the previous week’s test again; this time it went a lot better. Overall, I think this course is good, but I wish it encouraged students to slow down and take time applying each concept. I find I learn best when I apply my knowledge to something, whereas this course tends to dump a ton of information on you at once. You are supposed to work alongside it in Jupyter Notebooks, which is essential, but I wish the instructor would say: “Please stop here and try making your own version of this concept.” Journaling alongside the course does help with that though. Anyway, after taking the test, I learned about comparison and Boolean operators. This section was straight forward compared to other lessons, so despite my last complaint, I didn’t need to really work on applying this section. I then started learning about if, elif, and else statements.

Week 10: This week I spent much more time with if, elif, and else statements in section 5. This is where it started to feel like programming. Basically, in this section, I learned to create if-then statements that cause a method to happen to a variable if the condition is satisfied. Example:

```
name = ‘Andy’

if name == ‘Andy’:
        print(‘Hello Andy.’)
else print(‘This is not Andy.’)
```

In this example, since the name in the first line is defined as ‘Andy,’ the if statement would print ‘Hello Andy.’ because the condition of name being equal to Andy is satisfied. If the top line was: name = ‘Bob’, then the if statement would produce the else result of ‘This is not Andy.’ However, I could add an elif statement to the code, like:

```
elif name == ‘Bob’:
        print(‘Hello Bob.’)
```

If I then set the name variable to = ‘Bob’ like this example:

```
name = ‘Bob’

if name == ‘Andy’:
        print(‘Hello Andy.’)
elif name == ‘Bob’:
        print(‘Hello Bob.’)
else print(‘This is not Andy.’)
```

then this piece of code would print the result ‘Hello Bob.’

Week 11: This week I started learning about For Loops and While Loops. For Loops let you iterate over the elements (like lists, strings, or dictionaries) in objects. This lets you interact with elements in an object. For loops are structured similarly to if statements but use the keyword ‘for’ instead of ‘if.’ For example:

```
mylist = [1,2,3]

for item_name in my_list:
        print(item_name)
```

This will return a result of:

```
1
2
3
```

So, I assigned the list [1,2,3] to the variable “mylist”. Then I created the variable “item_name” which is a placeholder for all the items in mylist.

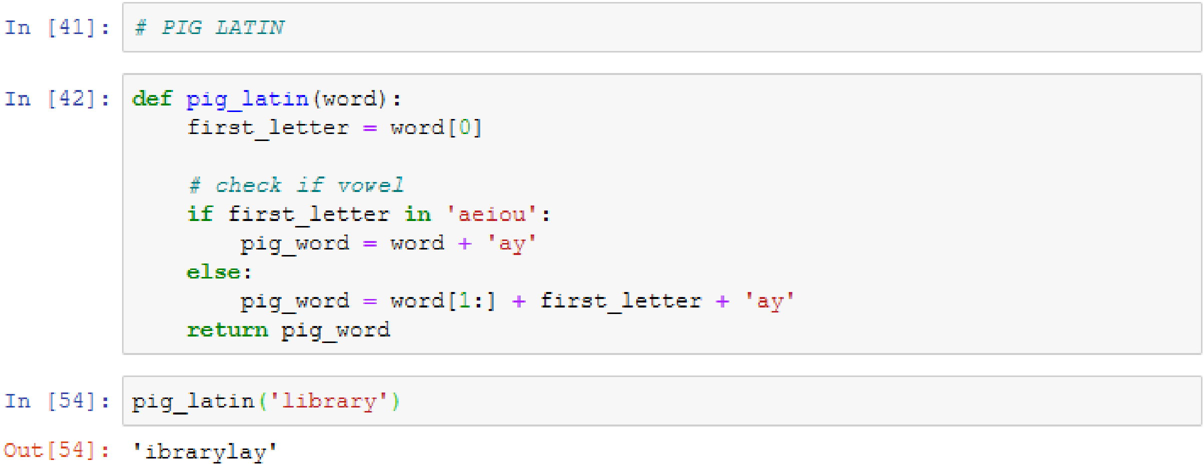
Week 12: I pushed through to the end of Section 5 on loops. I did not do well on the test at the end of the section. So, I went back and watched all the videos for the section again and did some coding exercises. I spent several hours working on this. I ended up taking the test a few times until I felt satisfied with my results. This section was significantly more challenging than the rest of the course so far. I also spent some time outside of the Udemy course reading about programming and watching fundamental programming videos on YouTube. This Python course has taught me a lot, but it also made me feel like I’m missing some of the fundamentals of programming that the course expected of me.

Week 13: I started this week by reviewing what I learned in Section 5. I haven’t retained all of it, but I can feel myself getting better. I’m starting to have more of an intuitive understanding of the syntax and vocabulary of Python. I think that part of my frustration is that I can’t memorize everything that I’ve learned, but I remind myself that even advanced programmers look things up all the time. After reviewing Section 5, I moved on to Section 6: Methods and Functions.

Week 14: Unfortunately, I did not get any work done on this assignment this week. I spent all my time working on my final papers. Hopefully, I'll have more time next week after I finish these papers.

Week 15: Since I finished my two final papers, I was able to do a lot more work on this Python stretch assignment. I did a quick review of Section 5 and the beginning of Section 6. After the review, I got much deeper into Section 6. Functions are incredibly power tools. Functions let you create a block of code that can be executed many times, instead of having to make repeated blocks of code. I did 10 coding exercises for this section, which were challenging, but still approachable. I was able to get most of them right on my first or second try. This gave me a confidence boost after how hard the previous test was.

Here is an example function:



```
In [41]: # PIG LATIN

In [42]: def pig_latin(word):
             first_letter = word[0]

             # check if vowel
             if first_letter in 'aeiou':
                 pig_word = word + 'ay'
             else:
                 pig_word = word[1:] + first_letter + 'ay'
             return pig_word

In [54]: pig_latin('library')

Out[54]: 'ibrarylay'
```

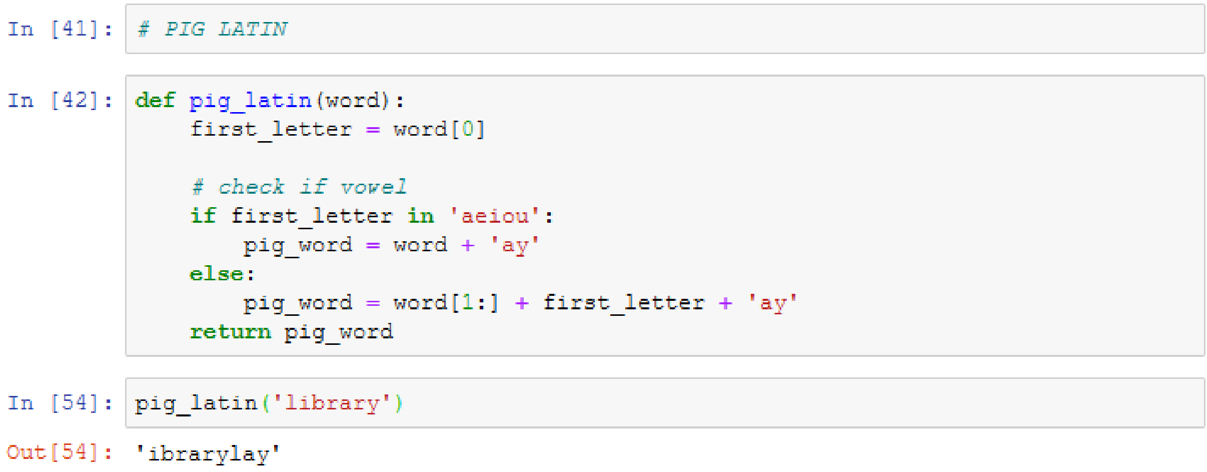
This function lets you enter a string of text into the function and returns the word as Pig Latin. 'library' becomes 'ibrarylay.' The function checks to see if the first letter of the string is a vowel ('aeiou'). If it does start with a vowel, the function returns the string with 'ay' at the end. If the string starts with a consonant, it removes the first letter of the string, adds to the end, and adds 'ay' (pig_word = word[1:] + first letter + 'ay').

Stretch Assignment Final Reflection: Udemy Complete Python Bootcamp

I chose Udemy's Complete Python Bootcamp as my stretch assignment. Udemy is an instruction site with many courses that focus on coding and other tech skills. I chose Python as a coding language because its uses in data science. I have no background in programming (I only have an introductory knowledge of HTML/XML).

My goal was to finish the course's first milestone project in section 7. I did not quite get there. I am currently a little over halfway through section. Even though I didn't reach that goal, I did put a lot of time and work into this project. Each section has about 1-2 hours' worth of videos that I watched and coded along with. While I was able to keep up with the instructor, I found myself emulating him instead of really thinking about what he was teaching, so by the end of the section there was quite a bit of knowledge that I had not retained. I ended up watching all the videos two or three times and taking the quizzes/test five or six times. This course taught me a lot, but I find I don't really retain things unless I am able to apply the knowledge. To get the full value out of the lessons, I would have spent more time working on my own coding examples to better apply the knowledge.

Here is the most practical example of a block of code I made (with help from the videos). This block of code is a function that translates strings of text into pig latin. For example, 'library' becomes 'ibrarylay.' The function checks to see if the first letter of the string is a vowel ('aeiou'). If it does start with a vowel, the function returns the string with 'ay' at the end. If the string starts with a consonant, it removes the first letter of the string, adds to the end, and adds 'ay' (pig_word = word[1:] + first letter + 'ay'). It accomplishes this through 'if' and 'else' statements.



```
In [41]: # PIG LATIN

In [42]: def pig_latin(word):
             first_letter = word[0]

             # check if vowel
             if first_letter in 'aeiou':
                 pig_word = word + 'ay'
             else:
                 pig_word = word[1:] + first_letter + 'ay'
             return pig_word

In [54]: pig_latin('library')

Out[54]: 'ibrarylay'
```

I'm going to keep working on this Udemy course after the end of this semester, with the intent of reaching my original goal of finishing the first milestone project. After that, I'll probably switch to W3 School's Python tutorial and go over everything I learned in Udemy again to really solidify what I learned.