Wade Moulton

09/25/2020

CS 4720 W01

Assignment 3 Report

Weblink: <http://studentweb.kennesaw.edu/~jmoulto2/>

**Assignment Description**

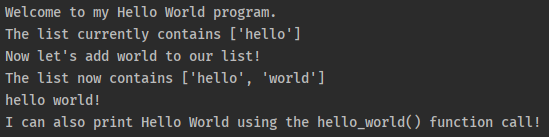
Assignment 3 is centered around creating python scripts. The three scripts for this assignment were *helloworld.py, wordguesser.py, and weathermap.py*. Each of these scripts helped to understand different fundamentals for the python language. Along with this report I have also provided an Assignment 3 link on my student webpage to display the source code as well as sample outputs making use of a library called **prismjs** to ‘prettify’ the code display on the webpage. All source code can be found at the bottom of this document.

**Section 1**

*Script file: helloworld.py*

The hello world script is all about learning to make the most simplistic of scripts common to any beginner course of all programming languages. To make my version slightly more creative I have made use of f strings as well as functions and lists from our topics of study this module.

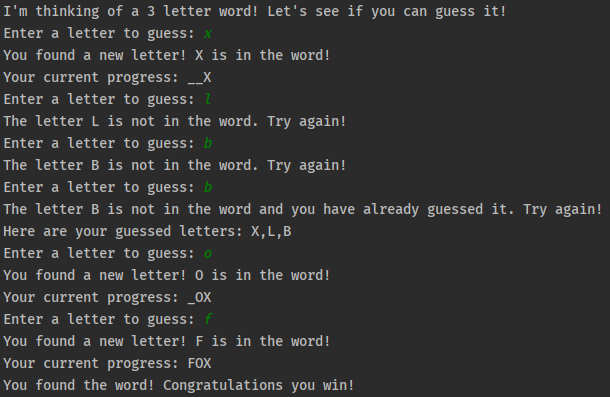
*helloworld.py* output:



**Section 2**

*Script files: wordguesser.py*

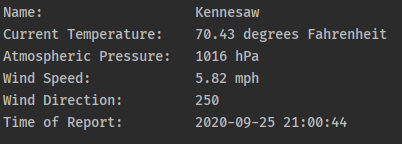
The word guesser script makes use of several topics from this module, such as: lists, functions, a while loop, and conditional statements. To accomplish this task a user must first enter a letter. The main function will check to see if the letter is alphabetical and if not will prompt with an error and ask for input again. Once a letter has been entered this goes to the *guessLetter* function, which then determines if the letter is in the word or not and then if the letter has been guessed already. For each of the different possible outcomes a message will be printed and the user will continue to get a prompt to enter letters until the final word matches the initial word.



**Section 3**

*Script files: weathermap.py*

The weather map script is an example for using an external web API to return some type of data; in the case of this assignment, weather data for a given zip code. For this script to function I had to first get an API key from openweathermap.org and then created variables at the beginning of the script to easily change the API key as well as setting a user id and a zip code. In the main function of the script the request is made to the web URL and the response is decoded to JSON format. Once the response is in JSON format, each item is printed as directed with the values coming straight from the JSON.



**Section 4**

To expand this assignment to my student web page I have added the source code and sample outputs on the Assignment 3 link. I have also provided a link at the top to download this report. To style the code snippets with syntax highlighting I am also using **prismjs** and including it in the html file for Assignment 3. Here is a sample from my student web page:



**Source Code**

# Helloworld.py

# Name: Wade Moulton  
# Date: 09/25/2020  
# Class: CS 4720  
# Section: W01  
# Instructor: Dr. Sarah North  
  
hw\_list = ['hello']  
  
  
def hello\_world():  
 return 'Hello World'  
  
  
def main():  
 print(f"Welcome to my Hello World program.")  
 print(f"The list currently contains {hw\_list}")  
 print(f"Now let's add world to our list!")  
 hw\_list.append('world')  
 print(f"The list now contains {hw\_list}")  
 print(f"{hw\_list[0]} {hw\_list[1]}!")  
 print(f"I can also print {hello\_world()} using the hello\_world() function call!")  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

# Wordguesser.py

# Name: Wade Moulton  
# Date: 09/25/2020  
# Class: CS 4720  
# Section: W01  
# Instructor: Dr. Sarah North  
  
# set the word once and then convert to list for easier iteration  
word = 'FOX'  
word\_list = list(word)  
final\_word = []  
guessed = []  
  
def guessLetter(ltr):  
 # convert letter to uppercase  
 ltr = ltr.upper()  
  
 # first check if the letter is in the word or not  
 # if letter is in word, change all occurrences of blank letters in final\_word to match at correct index  
 # then check if the letter has been guessed, if not add to guessed array  
 # give user feedback based on conditions  
 if ltr in word\_list or ltr.upper() in word\_list:  
 for i in range(len(word\_list)):  
 if word\_list[i] == ltr or word\_list[i].lower == ltr:  
 final\_word[i] = ltr  
 if not (ltr in guessed or ltr.lower() in guessed):  
 guessed.append(ltr)  
 print(f"You found a new letter! {ltr} is in the word!")  
 print(f"Your current progress: {''.join(str(x) for x in final\_word)}")  
 else:  
 print(f"The letter {ltr} is in the word, but you already guessed it. Try again!")  
 print(f"Here are your guessed letters: {','.join(str(x) for x in guessed)}")  
 print(f"Your current progress: {''.join(str(x) for x in final\_word)}")  
 else:  
 if not (ltr in guessed or ltr.lower() in guessed):  
 guessed.append(ltr)  
 print(f"The letter {ltr} is not in the word. Try again!")  
 else:  
 print(f"The letter {ltr} is not in the word and you have already guessed it. Try again!")  
 print(f"Here are your guessed letters: {','.join(str(x) for x in guessed)}")  
  
  
def main():  
 print(f"I'm thinking of a {len(word)} letter word! Let's see if you can guess it!")  
 for i in range(len(word\_list)):  
 final\_word.append('\_')  
 while word\_list != final\_word:  
 ltr = input('Enter a letter to guess: ').lower()  
 # check if input is alphabetic  
 if ltr.isalpha():  
 guessLetter(ltr)  
 else:  
 print('Input must be a letter. Try again!')  
  
 print("You found the word! Congratulations you win!")  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

# Weathermap.py

# Name: Wade Moulton  
# Date: 09/25/2020  
# Class: CS 4720  
# Section: W01  
# Instructor: Dr. Sarah North  
  
from datetime import datetime  
import urllib  
import requests  
import json  
  
# set variables  
user\_id = 'Boatshoes24'  
user\_apiid = 'daa9f87894cb7014c886cfa3fac6c927'  
zip\_code = 30144  
  
# set url for http request  
getURL = f'http://api.openweathermap.org/data/2.5/weather?zip={zip\_code},us&appid={user\_apiid}&units=imperial'  
  
  
def main():  
 # retrieve request and decode to json  
 response = urllib.request.urlopen(getURL)  
 decoded\_response = response.read().decode('utf-8')  
 json\_res = json.loads(decoded\_response)  
  
 print(f"Name:\t\t\t\t\t{json\_res['name']}")  
 print(f"Current Temperature:\t{json\_res['main']['temp']} degrees Fahrenheit")  
 print(f"Atmospheric Pressure:\t{json\_res['main']['pressure']} hPa")  
 print(f"Wind Speed:\t\t\t\t{json\_res['wind']['speed']} mph")  
 print(f"Wind Direction:\t\t\t{json\_res['wind']['deg']}")  
 timestamp = json\_res['dt']  
 dt\_object = datetime.fromtimestamp(timestamp)  
 print(f"Time of Report:\t\t\t{dt\_object}")  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()