

Homework 5

DUE DATE: October 27

Question 1: Write a program that asks the user for a password, and then uses regular expressions to make sure the password string it is passed is strong enough. A strong enough password is defined as one that is at least 8 characters long, contains both uppercase and lowercase characters, and has at least one digit. Test to show its response to a strong enough password and to a password that is not strong enough.

Question 2: In this question you are going to scrape the website:

<https://www.basketball-reference.com/playoffs/>

First import the proper libraries and create a BeautifulSoup object to the webpage.

- Create a list called “finals” that stores for each year from 1977-2021, the year, the NBA champion, the NBA runner-up, and the finals MVP. Then turn your list into a DataFrame, and print the DataFrame.
- Create a DataFrame with two columns. The first column is the unique names of all teams that won an NBA championship since 1977, and the second column is the total number of championships that they won in that time span. The DataFrame should be sorted based on total number of championships from most to least. Print out the sorted DataFrame.
- Create a bar chart of the total number of championships for each team in (b) that has won an NBA championship.
- Create a second DataFrame that has the unique names of all the Finals MVP winners since 1977 and the total number of MVPs that they each have won. It again should be sorted from most MVPs to least. Print out the sorted DataFrame.
- Create a second list called “playoffs” that stores for each year from 1977-2021, the year, the playoff leader in points (their name) and their total number of points. Then turn your list into a DataFrame, and print the DataFrame. (Hint: To extract the total number of points from outside the parenthesis, you need to call “re.search('\([^\)]+\)',name).group(1)”, where “name” is the name of the string where you store the player’s name with their total number of points.)
- Sort the DataFrame based on the Points column, from the player with the most points in the playoffs to the playoff leader with the least number of points. Print out the sorted DataFrame.

Question 3: In this question you are going to scrape the total number of Wins, Losses, and Winning Percentage for the Warriors for each year between 1977-2014 from Basketball Reference. You are going to do this by accessing the league page for each year which is found at the following address:

https://www.basketball-reference.com/leagues/NBA_YEAR.html

where *YEAR* is the year you are interested in looking up (e.g. 2000). Create a list called *seasons* and cycle through each year from 1977-2014 and store in *seasons* the year, wins, losses, and winning percentage for the Golden State Warriors.

- a) Turn the list into a DataFrame. Then convert the *Wins* column of the DataFrame to type integer. Then print out the DataFrame.
- b) Calculate and print out the mean, median, and standard deviation of wins for the Warriors from 1977-2014.
- c) Sort the seasons based on winning percentage and print out the years in descending order from the year the team had the highest winning percentage to the year the team had the lowest winning percentage.
- d) Plot the Warriors wins each year as a line chart.
- e) Plot the Warriors wins each year using a horizontal bar chart.
- f) Prompt the user to enter a year between 1977-2014. Check the input to make sure that it is between those 2 years. Then print out the number of wins the Warriors received in the user inputted year.

Question 4: You are going to access a simple API created by space travel enthusiasts, called Open Notify. The website for the API is <http://open-notify.org/>. To access the API run the code in the box below. Then answer the following questions.

- a) Print a statement that reads, "There are currently X people in space," where you get the correct number for X.
- b) Print the names of each person currently in space, in alphabetical order.

Question 5: Create a Mad Libs program that starts with the string:

"The ADJECTIVE panda walked to the NOUN and then VERB. A nearby NOUN was unaffected by these events."

Then lets users add their own text to replace the words *ADJECTIVE*, *NOUN*, *VERB*, and *NOUN*. Then print out the resulting string with the user inputted words filled in.

Question 6: Create a timed multiplication quiz that poses 5 multiplication problems to the user ranging from 0x0 to 9x9. Your program should do all of the following:

- a) For each of the 5 questions, randomly select 2 integers from 0 to 9 and print out the multiplication problem to the user as "#X N1 x N2 =" where X is the questions number and N1 and N2 are the randomly selected integers. (Ex. "#2 5 x 5 =")
- b) Use *pyinputplus* to assess the user's answer. If the user's response is correct then print out "Correct!" If the user's response doesn't match the correct answer, then the program should reject the answer, display "Incorrect!" as a string, and then prompt the user to answer again.
- c) Add the feature that if the user doesn't answer within 8 seconds or they exceed 3 tries, then they get the question incorrect.
- d) Add the feature that you keep track of the number of correct answers (answers that they get in under 8 seconds and in less than 3 tries) and print out at the end of the quiz, the number of correct answers out of 5.