

Q1: Assign your name to the variable name.

```
[]  
  
name = ???
```

Q2: Assign your age (real or fake) to the variable age.

```
[]  
  
age = ???
```

Q3: Assign a boolean value to the variable has_android_phone.

```
[]  
  
has_android_phone = ???
```

You can check the values of these variables by running the next cell.

```
[]  
  
name, age, has_android_phone
```

Q4: Create a dictionary person with keys "Name", "Age", "HasAndroidPhone" and values using the variables defined above.

```
[]
```

```
person = ???
```

Let's use the person dictionary to print a nice message.

```
[]
```

```
print("{} is aged {}, and owns an {}".format(  
    person["Name"],  
    person["Age"],  
    "Android phone" if person["HasAndroidPhone"] else "iPhone"  
))
```

Problem 2 - Working with Lists

Q5: Create a list containing the following 3 elements:

- your favorite color
 - the number of pets you have
 - a boolean value describing whether you have previous programming experience
-

```
[]
```

```
my_list = ???
```

Let's see what the list looks like:

```
[]
```

```
my_list
```

Q6: Complete the following print and if statements by accessing the appropriate elements from my_list.

Hint: Use the list indexing notation [].

```
[]
```

```
print('My favorite color is', ???)
```

```
[]
```

```
print('I have {} pet(s)'.format(???))
```

```
[]
```

```
if ???:
```

```
    print("I have previous programming experience")
```

```
else:
```

```
    print("I do not have previous programming experience")
```

Q7: Add your favorite single digit number to the end of the list using the appropriate list method.

```
[]
```

```
my_list.???
```

Let's see if the number shows up in the list.

```
[]
```

```
my_list
```

Q8: Remove the first element of the list, using the appropriate list method.

Hint: Check out methods of list here:

https://www.w3schools.com/python/python_ref_list.asp

```
[]
```

```
my_list.???
```

```
[]
```

```
my_list
```

Q9: Complete the print statement below to display the number of elements in my_list.

```
[]
```

```
print("The list has {} elements.".format(???))
```

Problem 3 - Conditions and loops

Q10: Calculate and display the sum of all the numbers divisible by 7 between 18 and 534 i.e. $21+28+35+\dots+525+532$.

Hint: One way to do this is to loop over a range using for and use an if statement inside it.

```
[ ]
```

```
# store the final answer in this variable
```

```
sum_of_numbers = 0
```

```
# perform the calculation here
```

```
???
```

```
[ ]
```

```
print('The sum of all the numbers divisible by 7 between 18 and 534 is', sum_of_numbers)
```

Problem 4 - Flying to the Bahamas

Q11(a): A travel company wants to fly a plane to the Bahamas. Flying the plane costs 5000 dollars. So far, 29 people have signed up for the trip. If the company charges 200 dollars per ticket, what is the profit made by the company?

Fill in values or arithmetic expressions for the variables below.

```
[ ]
```

```
cost_of_flying_plane = ???
```

```
[]
```

```
number_of_passengers = ???
```

```
[]
```

```
price_of_ticket = ???
```

```
[]
```

```
profit = ???
```

```
[]
```

```
print('The company makes of a profit of {} dollars'.format(profit))
```

Q11(b): Out of the 29 people who took the flight, only 12 buy tickets to return from the Bahamas on the same plane. If the flying the plane back also costs 5000 dollars, and does the company make an overall profit or loss? The company charges the same fee of 200 dollars per ticket for the return flight.

Use an `if` statement to display the result.

```
[]
```

```
# this is optional
```

```
???
```

```
[]
```

```
# this is optional
```

```
if ???:
```

```
    print("The company makes an overall profit of {} dollars".format(???))
```

```
else:
```

```
    print("The company makes an overall loss of {} dollars".format(???))
```

Double-click (or enter) to edit

Problem 5 - Twitter Sentiment Analysis(Optional)

Are you ready to perform some *Data Analysis with Python*? In this problem, we'll analyze some fictional tweets and find out whether the overall sentiment of Twitter users is happy or sad. This is a simplified version of an important real world problem called *sentiment analysis*.

Before we begin, we need a list of tweets to analyze. We're picking a small number of tweets here, but the exact same analysis can also be done for thousands, or even millions of tweets. The collection of data that we perform analysis on is often called a *dataset*.

```
[]
```

```
tweets = [
```

```
    "Wow, what a great day today!! #sunshine",
```

```
    "I feel sad about the things going on around us. #covid19",
```

```
    "I'm really excited to learn Python with @JovianML #zerotopandas",
```

```
"This is a really nice song. #linkinpark",  
"The python programming language is useful for data science",  
"Why do bad things happen to me?",  
"Apple announces the release of the new iPhone 12. Fans are excited.",  
"Spent my day with family!! #happy",  
"Check out my blog post on common string operations in Python. #zerotopandas",  
"Freecodecamp has great coding tutorials. #skillup"  
]
```

Let's begin by answering a very simple but important question about our dataset.

Q12: How many tweets does the dataset contain?

```
[]  
  
number_of_tweets = ???
```

Let's create two lists of words: happy_words and sad_words. We will use these to check if a tweet is happy or sad.

```
[]  
  
happy_words = ['great', 'excited', 'happy', 'nice', 'wonderful', 'amazing', 'good', 'best']
```

```
[]  
  
sad_words = ['sad', 'bad', 'tragic', 'unhappy', 'worst']
```

To identify whether a tweet is happy, we can simply check if contains any of the words from `happy_words`. Here's an example:

```
[ ]
```

```
sample_tweet = tweets[0]
```

```
[ ]
```

```
sample_tweet
```

```
[ ]
```

```
is_tweet_happy = False
```

```
# Get a word from happy_words
```

```
for word in happy_words:
```

```
    # Check if the tweet contains the word
```

```
    if word in sample_tweet:
```

```
        # Word found! Mark the tweet as happy
```

```
        is_tweet_happy = True
```

Do you understand what we're doing above?

For each word in the list of happy words, we check if it is a part of the selected tweet. If the word is indeed a part of the tweet, we set the variable `is_tweet_happy` to `True`.

```
[]
```

```
is_tweet_happy
```

Q13: Determine the number of tweets in the dataset that can be classified as happy.

Hint: You'll need to use a loop inside another loop to do this. Use the code from the example shown above.

```
[]
```

```
# store the final answer in this variable
```

```
number_of_happy_tweets = 0
```

```
# perform the calculations here
```

```
???
```

```
[]
```

```
print("Number of happy tweets:", number_of_happy_tweets)
```

Q14: What fraction of the total number of tweets are happy?

For example, if 2 out of 10 tweets are happy, then the answer is $2/10$ i.e. 0.2 .

```
[]
```

```
happy_fraction = ???
```

```
[]
```

```
print("The fraction of happy tweets is:", happy_fraction)
```

To identify whether a tweet is sad, we can simply check if contains any of the words from sad_words.

Q15: Determine the number of tweets in the dataset that can be classified as sad.

```
[]
```

```
# store the final answer in this variable
```

```
number_of_sad_tweets = 0
```

```
# perform the calculations here
```

```
???
```

```
[]
```

```
print("Number of sad tweets:", number_of_sad_tweets)
```

Q16: What fraction of the total number of tweets are sad?

```
[]
```

```
sad_fraction = ???
```

```
[]
```

```
print("The fraction of sad tweets is:", sad_fraction)
```

Great work, even with some basic analysis, we already know a lot about the sentiment of the tweets given to us. Let us now define a metric called "sentiment score", to summarize the overall sentiment of the tweets.

Q16(b): Calculate the sentiment score, which is defined as the difference between the fraction of happy tweets and the fraction of sad tweets.

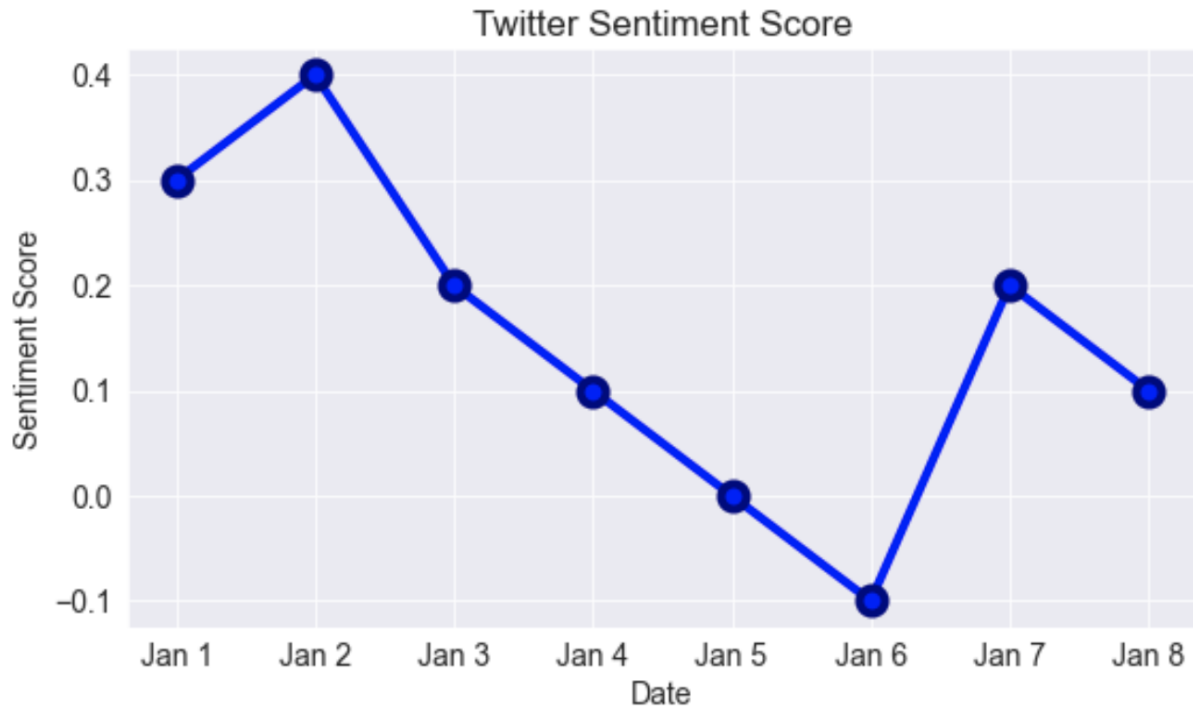
```
[]
```

```
sentiment_score = ???
```

```
[]
```

```
print("The sentiment score for the given tweets is", sentiment_score)
```

In a real world scenario, we could calculate & record the sentiment score for all the tweets sent out every day. This information can be used to plot a graph and study the trends in the changing sentiment of the world. The following graph was creating using the Python data visualization library `matplotlib`, which we'll cover later in the course.



What does the sentiment score represent? Based on the value of the sentiment score, can you identify if the overall sentiment of the dataset is happy or sad?

Q16(c): Display whether the overall sentiment of the given dataset of tweets is happy or sad, using the sentiment score.

[]

if ???:

 print("The overall sentiment is happy")

else:

 print("The overall sentiment is sad")

Finally, it's also important to track how many tweets are neutral i.e. neither happy nor sad. If a large fraction of tweets are marked neutral, maybe we need to improve our lists of happy and sad words.

Q16d (Optional): What is the fraction of tweets that are neutral i.e. neither happy nor sad.

```
[]
```

```
# store the final answer in this variable
```

```
number_of_neutral_tweets = 0
```

```
# perform the calculation here
```

```
???
```

```
[]
```

```
[]
```

```
[]
```

```
neutral_fraction = ???
```

```
[]
```

```
print('The fraction of neutral tweets is', neutral_fraction)
```

Ponder upon these questions and try some experiments to hone your skills further:

- What are the limitations of our approach? When will it go wrong or give incorrect results?
- How can we improve our approach to address the limitations?
- What are some other questions you would like to ask, given a list of tweets?
- Try collecting some real tweets from your Twitter timeline and repeat this analysis. Do the results make sense?

IMPORTANT NOTE: If you want to try out these experiments, please create a new notebook using the "New Notebook" button on your Jovian.ml profile, to avoid making unintended changes to your assignment submission notebook.
