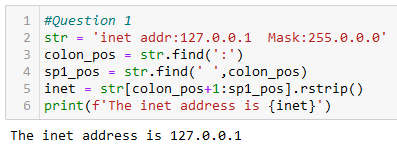
**Module 6 Assignment**

**Prompt:**

**1. What function is used to get the ASCII code of a character ?**

Ord()

2. Take the following Python code that stores a string: *str = 'inet addr:127.0.0.1  Mask:255.0.0.0'.* Use find and string slicing to extract the portion of the string after the colon inet address character and then use the rstrip function to remove any trailing characters.



3a. Using a for loop through a string, count the number of internet addresses in the string below by using the split method.

str = \

'''

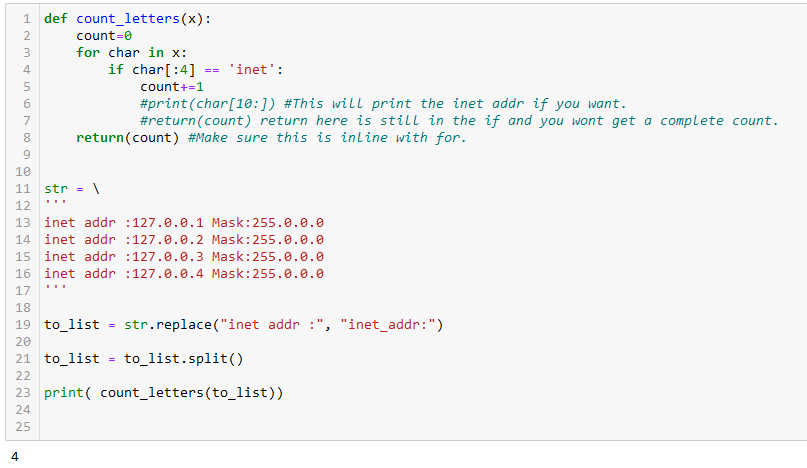
inet addr :127.0.0.1 Mask:255.0.0.0

inet addr :127.0.0.2 Mask:255.0.0.0

inet addr :127.0.0.3 Mask:255.0.0.0

inet addr :127.0.0.4 Mask:255.0.0.0

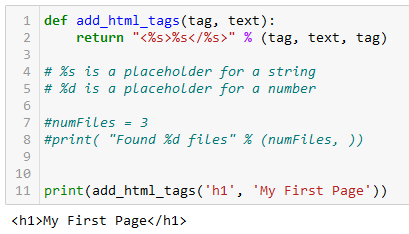
'''



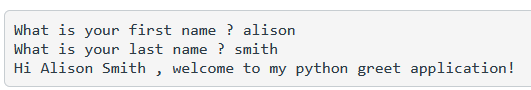
3b. Use the count() method to return the number of occurrences of the substring in the given string.

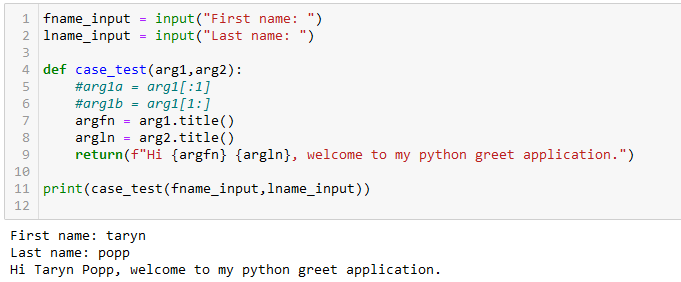
4. Write a Python function to create the HTML string with tags around the word(s). Sample function and result are shown below:



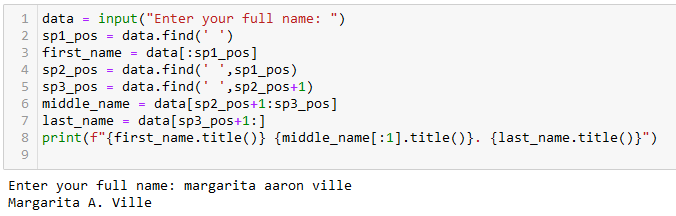


5. Write a Python script that takes input from the user and displays that input back in upper and lower cases.

Console Output  




6. Write a program that takes your full name as input and displays the abbreviations of the middle name except the first and last name which is displayed as it is. For example, if your name is Elvis Aaron Presley, then the output should be Elvis A. Presley.



7. Download the [List of Top 100 Famous People](https://www.biographyonline.net/people/famous-100.html)

[Links to an external site.](https://www.biographyonline.net/people/famous-100.html) compiled by biographyonline.net which contains a list of famous people - including famous actors, politicians, entrepreneurs, writers, artists and humanitarians chosen mainly from the nineteenth, twentieth or twenty-first centuries. After downloading the list, you want to query if an Italian painter and scientist (italian, painter, scientist) is in the list:

**famous\_list = ''' \**

Marilyn Monroe (1926 – 1962) American actress, singer, model

Abraham Lincoln (1809 – 1865) US President during American civil war

Nelson Mandela (1918 – 2013)  South African President anti-apartheid campaigner

John F. Kennedy (1917 – 1963) US President 1961 – 1963

Martin Luther King (1929 – 1968)  American civil rights campaigner

Queen Elizabeth II (1926 – ) British monarch since 1954

Winston Churchill (1874 – 1965) British Prime Minister during WWII

Donald Trump (1946 – ) Businessman, US President.

Bill Gates (1955 – ) American businessman, founder of Microsoft

Muhammad Ali (1942 – 2016) American Boxer and civil rights campaigner

Mahatma Gandhi (1869 – 1948) Leader of Indian independence movement

Margaret Thatcher (1925 – 2013) British Prime Minister 1979 – 1990

Mother Teresa (1910 – 1997) Macedonian Catholic missionary nun

Christopher Columbus (1451 – 1506) Italian explorer

Charles Darwin (1809 – 1882) British scientist, theory of evolution

Elvis Presley (1935 – 1977) American musician

Albert Einstein (1879 – 1955) German scientist, theory of relativity

Paul McCartney (1942 – ) British musician, member of Beatles

Queen Victoria ( 1819 – 1901) British monarch 1837 – 1901

Pope Francis (1936 – ) First pope from the Americas

'''

Using the variable **famous\_list**, write a program to check if a famous individual is in the list above, if they are then print: **Sorry, the individual did not make the top 20 cut!** Otherwise print: **Yup, the individual did make the top 20 cut.**

**Console:**

Please Enter the name of the famous individual? Albert Einstein

Yup, Albert Einstein did make the Top 20 cut!

Please Enter the name of the famous individual? leonardo Da vinci

Sorry, Leonardo Da Vinci did not make the Top 20 cut!

import pandas as pd  
import re  
  
#df = pd.read\_csv('D:\Python Learning\Continuing Education COMP660\Module 6\FamousPeopleList.csv')  
#print(df.head(5))  
  
famous\_list = ''' \  
Marilyn Monroe (1926 – 1962) American actress, singer, model  
Abraham Lincoln (1809 – 1865) US President during American civil war  
Nelson Mandela (1918 – 2013) South African President anti-apartheid campaigner  
John F. Kennedy (1917 – 1963) US President 1961 – 1963  
Martin Luther King (1929 – 1968) American civil rights campaigner  
Queen Elizabeth II (1926 – ) British monarch since 1954  
Winston Churchill (1874 – 1965) British Prime Minister during WWII  
Donald Trump (1946 – ) Businessman, US President.  
Bill Gates (1955 – ) American businessman, founder of Microsoft  
Muhammad Ali (1942 – 2016) American Boxer and civil rights campaigner  
Mahatma Gandhi (1869 – 1948) Leader of Indian independence movement  
Margaret Thatcher (1925 – 2013) British Prime Minister 1979 – 1990  
Mother Teresa (1910 – 1997) Macedonian Catholic missionary nun  
Christopher Columbus (1451 – 1506) Italian explorer  
Charles Darwin (1809 – 1882) British scientist, theory of evolution  
Elvis Presley (1935 – 1977) American musician  
Albert Einstein (1879 – 1955) German scientist, theory of relativity  
Paul McCartney (1942 – ) British musician, member of Beatles  
Queen Victoria ( 1819 – 1901) British monarch 1837 – 1901  
Pope Francis (1936 – ) First pope from the Americas  
Charles Francis (1936 – 1942) Fake entry for testing dupes  
'''  
famous\_list = famous\_list.upper()  
famous\_list = famous\_list.replace('(','|')  
famous\_list = famous\_list.replace(')','|')  
#Replace carriage return with pipe  
famous\_list = famous\_list.replace('\n','|')  
famous\_list = famous\_list.split('|')  
  
'''So the issue is that some of the people's descriptions have date ranges in them,  
 which means we can't split the text by ' - '.  
 Below we use three different loops to get every 3rd item in the list, offset by the starting item.  
 We end up with three separate dataframes for name, lifespan, and desc. Then we have to merge them.'''  
  
def GetNames(text, n):  
 builtstring = ""  
 for i in range(0, len(text)):  
 if (i + 3) % n == 0:  
 # print(text[i])  
 builtstring = builtstring + text[i] + '|'  
 return builtstring  
  
def GetDates(text, n):  
 builtstring = ""  
 for i in range(0, len(text)):  
 if (i + 2) % n == 0:  
 # print(text[i])  
 builtstring = builtstring + text[i] + '|'  
 return builtstring  
  
def GetDesc(text, n):  
 builtstring = ""  
 for i in range(2, len(text)):  
 if (i + 4) % n == 0:  
 # print(text[i])  
 builtstring = builtstring + text[i] + '|'  
 return builtstring  
  
#Run each function to get an individual list of each parsed column.  
#There's probably a more elegant way of doing this.  
names = GetNames(famous\_list,3).split('|')  
dates = GetDates(famous\_list,3).split('|')  
desc = GetDesc(famous\_list,3).split('|')  
  
#Here we're converting each list into a data frame and naming them  
#according to the names in the cols\_list.  
cols\_list = ['name', 'lifespan', 'description']  
df\_names = pd.DataFrame(data = names, columns= [cols\_list[0]])  
df\_lifespan = pd.DataFrame(data = dates, columns= [cols\_list[1]])  
df\_desc = pd.DataFrame(data = desc, columns= [cols\_list[2]])  
  
#Here we're 'unioning' columns of each df  
df\_m = pd.concat([df\_names, df\_lifespan], axis=1)  
df\_fp = pd.concat([df\_m,df\_desc], axis = 1)  
  
#Ways to print single columns  
#print(df\_fp['lifespan'])  
#print(df\_fp.name.to\_string(index=False))  
  
search\_str =input('Enter some comma-separated characteristics to search for. e.g. italian,painter,scientist ')  
search\_upper = search\_str.upper()  
search\_final = search\_upper.replace(",","|")  
  
search\_results = df\_fp[df\_fp['description'].str.contains(search\_final, na=False)]  
  
if search\_results.empty:  
 print(f"Sorry, there were no hits on your search: {search\_str}")  
else:  
 print(search\_results.to\_string(index=False))  
  
#If I had more time I would try to figure out how to do this search without converting  
#the whole famous list to upper case.  
  
  
#This is the solution to the 2nd half of the question  
text = input('Please Enter the name of the famous individual: ')  
#Convert to camel case if needed  
text\_camel = text.upper()  
search\_value = df\_fp[df\_fp['name'].str.contains(text\_camel)]  
  
#get\_name = search\_value.name[0]  
#print(get\_name)  
  
if search\_value.empty:  
 print(f"Sorry, {text\_camel.title()} did not make the top 20 cut!")  
else:  
 print(f"Yup, {text\_camel.title()} did make the Top 20 cut!")

