5.2 Python Homework

Data Types

1. Create a file and name it 5_2-Homework
    - A Jupyter Notebook file would probably work best as you would want to type some answers to questions and code for this homework.
    - Decorate / beautify your markdown in any way your like.

2. Open a Markdown cell in your Jupyter Notebook and answer these questions please:
    - Use a markdown numbered list to answer these questions.
    - What is an algorithm?
        i. a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.
    - Why is Python considered to be a high-level language?
        i. When compiled, other languages turn into Assembly and run directly in the processor. Hence, **being an interpreted language, which is not subject to the processor**, makes Python a high-level programming language. Also, Python is a high-level programming language that is known for its ease of readability.
    - Is Python a Natural or a Formal language?
        i. Formal languages are important to computer science (and to many other fields). **All programming languages, like Java, C, and Python, are formal languages**.
        ii. Python is **an interpreted, interactive, object-oriented programming language**. It incorporates modules, exceptions, dynamic typing, very high level dynamic data types, and classes. It supports multiple programming paradigms beyond object-oriented programming, such as procedural and functional programming.
        iii. Natural languages are the languages people speak, such as English, Spanish, and French. They were not designed by people (although people try to impose some order on them); they evolved naturally. Formal languages are languages that are designed by people for specific applications.
    - Variable names may not start with certain characters - name two.

    i. A variable name must start with a letter or an underscore character (_) A variable name cannot start with a digit. A variable name can only contain alpha-numeric characters and underscores (a-z, A-Z , 0-9 , and _) Variable names are case-sensitive (age, Age and AGE are three different variables)
    ii. Variables cannot start with numbers or symbols other than the underscore.
- What is a Semantic error?
    i. A semantic error **occurs when a program works without raising an Exception, but gives a different than the expected result**. The underlying defects are usually more difficult to eliminate than defects that result in error messages.
    ii. Semantic errors are **problems with a program that runs without producing error messages but doesn't do the right thing**. Example: An expression may not be evaluated in the order you expect, yielding an incorrect result.
- What is the #1 rule of coding / debugging?
    i. Rule #1: **Assume nothing**.
    ii. Check the values of all the variables involved. Read the stack trace carefully. Step through your code. Nine times out of ten, the bug will hide in the one area of the problem you think you can take for granted
- list 5 Python reserved words.

Reserved words in Python

Here is the list of all the reserved words in Python.
Note - This list may change with different versions of Python. Python 3 has 33 while Pyt
2 has 30 reserved words. The print was removed from Python 2 keywords and added as
built-in Python function.

False	def	if	raise
None	del	import	return
True	elif	in	try
and	else	is	while
as	except	lambda	with
assert	finally	nonlocal	yield
break	for	not	
class	form	or	
continue	global	pass	

i.

- add and name, a web link for any Python coding article, in your markdown cell
    i. Geeked

3. Open a code cell in your Jupyter Notebook and write the following code:
    - Write a multi-line comment with your name, favorite food, and dream job on 3 different lines.
        i. print('Holly')
        ii. print('Sushi')
        iii. print('Product manager')
    - assign 5 different data types to 5 different variables. At least one data type must be a string.

```
data =
pd.read_csv(r"/Users/hollyrobinson/Documents/dap2022/dapcap/GitHub
/DAP-Capstone-Categorizing-and-Statistics/data_new.csv",
dtype={'Undergraduate Major': 'category', 'Starting Median
Salary':'str','Mid-Career Median Salary':'str','Percent change
from Starting to Mid-Career Salary':'float16','Mid-Career 10th
Percentile Salary':'str','Mid-Career 25th Percentile
```

```
    Salary':'str','Mid-Career 75th Percentile
    Salary':'str','Mid-Career 90th Percentile
    Salary':'str','Colleges':'category'})
```

ii.

- print the length of your string.

Method:6:

- Using sum() and list comprehension function. We use list comprehension for iterati over the string and sum function to sum total characters in string

Python3


```
# Python code to demonstrate string length
# using sum


def findLen(string):
    return sum( 1 for i in string);


# Driver Code
string = 'geeks'
print(findLen(string))
```

Output:

```
5
```

i.

Method:5:

- Using reduce method. Reduce method is used to iterate over the string and return a result of collection element provided to the reduce function. We will iterate over th string character by character and count 1 to result each time.

Python3

```
# Python code to demonstrate string length
# using reduce

import functools

def findLen(string):
    return functools.reduce(lambda x,y: x+1, string, 0)


# Driver Code
string = 'geeks'
print(findLen(string))
```


Output:

```
5
```

Method#4:

- Using string methods join and count. The join method of strings takes in an iterable returns a string which is the concatenation of the strings in the iterable. The separa between the elements is the original string on which the method is called. Using joi counting the joined string in the original string will also result in the length of the s

Python3

```
# Python code to demonstrate string length
# using join and count

# Returns length of string
def findLen(str):
    if not str:
        return 0
    else:
        some_random_str = 'py'
        return ((some_random_str).join(str)).count(some_random_str)

str = "geeks"
print(findLen(str))
```

Output:

Method#3:

- Using while loop and Slicing. We slice a string making it shorter by 1 at each iteratio
  eventually resul

  hile loop stops. Maintaining a co
  the number of iterations will result in the length of the string.

Python3

```python
# Python code to demonstrate string length
# using while loop.

# Returns length of string
def findLen(str):
    counter = 0
    while str[counter:]:
        counter += 1
    return counter

str = "geeks"
print(findLen(str))
```

Output:

```
5
```

- Using for loop and in operator. A string can be iterated over, directly in a for loop. Maintaining a count of the number of iterations will result in the length of the string

Python3

```python
# Python code to demonstrate string length
# using for loop

# Returns length of string
def findLen(str):
    counter = 0
    for i in str:
        counter += 1
    return counter


str = "geeks"
print(findLen(str))
```


Output:

```
5
```

Methods#1:

- Using the built-in function len. The built-in function len returns the number of item
  container.

Python3

K-Means Elbow Method code for Python – Pre...
predictivehacks.com/.../k-means-elbow-meth...

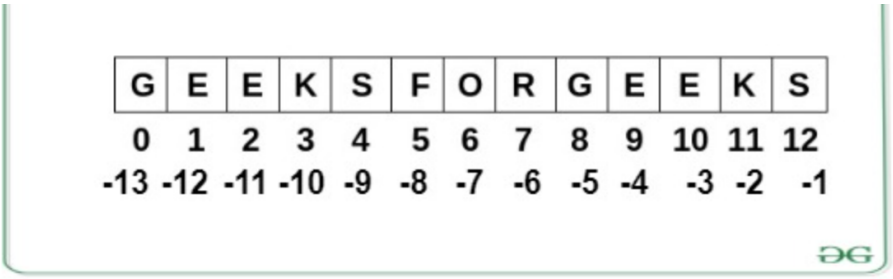
```
# Python code to demonstrate string length
# using len

str = "geeks"
print(len(str))
```

Output:

```
5
```

- print the index value of the 4th character in your string.



Python3

```
# Python Program to Access
# characters of String

String1 = "GeeksForGeeks"
print("Initial String: ")
print(String1)

# Printing First character
print("\nFirst character of String is: ")
print(String1[0])

# Printing Last cha
print("\nLast charac
print(String1[-1])
```


i.

- Create a new variable called savvy, and assign it the string with this phrase "Learning Data Analytics and Python is Awesome!"
    i. _savvy = print("Learning Data Analytics and Python is Awesome!")
- Return a range of characters that slices the above string from the beginning of "ing" up to before "tics"
    i. print(String[5:18])
- Replace "Awesome" with "great" in the string

i.

Python String | replace()

Difficulty Level : Easy • Last Updated : 27 Sep, 2022

Read Discuss

The **replace()** in Python returns a copy of the string where all occurrences of a substri
replaced with another substring.

Syntax of replace()

Syntax: *string.replace(old, new, count)*

Parameters:

- ***old –*** *old substring you want to replace.*
- ***new –*** *new substring which would replace the old substring.*
- ***count –*** *(**Optional**) the number of times you want to replace the old substring w*
  the new substring.

Return Value : *It returns a copy of the string where all occurrences of a substring a*
replaced with another substring.

Replace All Instances of a Single Character using replace()

In this example, we are only replacing a single character from a given string. The Pytho replace() method is case-sensitive, and therefore it performs a case-sensitive substrir substitution, i.e. R in FOR is unchanged.

Python3

```
string = "grrks FOR grrks"

# replace all instances of 'r' (old) with 'e' (new)
new_string = string.replace("r", "e" )

print(string)
print(new_string)
```

Output :

```
grrks FOR grrks
geeks FOR geeks
```

Replace All Instances of a String

Here, we replaced all the geeks with GeeksforGeeks using replace() function.

Python3

```
string = "geeks for geeks \ngeeks for geeks"
print(string)

# Prints the string by r
# 3 occurrence of Geeks
print(string.replace("geeks", "GeeksforGeeks"))
```


Output :

```
geeks for geeks
geeks for geeks
GeeksforGeeks for GeeksforGeeks
GeeksforGeeks for GeeksforGeeks
```

Replace Only a Certain Number of Instances using replace()

In this example, we are replacing certain numbers of words. i.e. “ek” with “a” with **coun**

Python3

```
string = "geeks for geeks geeks geeks geeks"

# Prints the string by replacing
# e by a
print(string.replace("e", "a"))

# Prints the string by replacing only
# 3 occurrence of ek by a
print(string.replace("ek", "a", 3))
```

Output:

```
gaaks for gaaks gaaks gaaks gaaks
geas for geas geas geeks geeks
```

Learn SQL | Codecademy
codecademy.com/courses/learn-sql/.../sql

- Test and print the savvy string to see it contains "Python"
    i. word = 'geeks for geeks'
    ii. print(word.find('for'))
- Create and assign 3 more variables called name, age and length using the multi-variable naming method.
    i. Name = Age
    ii. Age = Length
    iii. Length = Name
    iv. ????
- Format a new string called 'miniBio' using curly brackets to complete this phrase... "Hi my name is (name), I am so (tall) and (so) old today.
    i. miniBio =
- print 'miniBio'
- cast and print the age variable to a float.

4. Save your file in your homework folder, and push it to GitHub.
    - Once you complete all your homework for the week, create a ticket to let the TAs know you have completed it.

- Remember to add the link to your homework, to the ticket.