

Python Basics for Data Science Practice Questions

Instructions:

- Use your understanding of python basics to answer the following questions.
- The sample code provided is a guide and not a strict approach.
- You may use a Jupyter notebook to present your answers.
- Solutions will be presented in the next class and corrections provided.

Questions:

1. Create a function to count the frequency of words in a text.

- **Hint:**

- Write a function that takes a string of text as input and counts the frequency of each word.
- Use a dictionary to store the word counts.
- Return the dictionary with words as keys and their counts as values.
- Handle cases like punctuation, case sensitivity, and empty strings.

Python

```
def word_frequency(text):
    dict_of_word_freq = {}    #define the dictionary to store the word count
    str_text = str(text).lower() #convert text to a string of lower case
    words = text.split() #split text into list of words separated
    for word in words:    #loop through word in the string
        ...                #remove punctuation like (.,:) using strip
                           #or replace method
        ...                #Use if to check if the word already exist in the
                           #dictionary, else assign count 1
                           #if it exist add 1 to the count

    return dict_of_word_freq
```

2. Create a function that acts as a simple calculator.

- **Hint:**

- Write a function that takes two numbers and an operator ('+', '-', '*', '/') as input.
- Use conditional statements to perform the correct arithmetic operation based on the operator.
- Return the result of the calculation.

- Handle cases like division by zero and invalid operators.

Python

```
def simple_calculator(num1, num2, operator):
    if operator == '+':
        return num1 + num2
    ...
    ... #add conditions for minus and multiplication
    elif operator == '/':
        if ... #check num2 is divisible by zero
            return num1/num2
        else:
            return "Cannot divide by zero"
    else:
        return "Invalid operator"
```

3. Create a function that validates and converts data types.

- **Hint:**

- Write a function that checks if input data is of a certain type (e.g., integer, string).
- If the data is not of the expected type, attempt to convert it (e.g., convert a string to an integer).
- Return the validated or converted data, or an error message if the conversion is not possible.
- Include checks for common data types like integers, floats, and strings.

Python

```
def validate_and_convert(data, expected_type):
    if ... #if data type is equal to the expected type
        return data
    try:
        return expected_type(data)
    except ValueError:
        return f"Cannot convert {data} to {expected_type.__name__}"
```

4. Create a function to convert temperatures between Celsius and Fahrenheit.

- **Hint:**

- Write a function that takes a temperature value and its unit (Celsius or Fahrenheit) as input.
- Use conditional statements to determine the conversion formula.
- Convert the temperature to the opposite unit and return the result.

- Handle edge cases like invalid inputs or incorrect unit types.
- Use the below formula.

$$\text{degreesFahrenheit} = \text{degreesCelsius} \times (9/5) + 32.$$

5. Create a function to check the strength of a password.

- **Hint:**

- Write a function that takes a password as input and evaluates its strength based on length, use of special characters, numbers, and uppercase/lowercase letters.
- Use conditional statements to check for different criteria (e.g., minimum length, presence of numbers).
- Return a rating (e.g., "Weak," "Moderate," "Strong") based on the evaluation.
- Handle edge cases like empty strings or extremely short passwords.

Python

```
def check_password_strength(password):
    if ...                #check if length is less than minimum (say 8)
        return "Weak"
    elif ...              #check if there's NO digit
        return "Moderate"
    elif ...              #check if there's NO Uppercase
        return "Moderate"
    elif ...              #check if there's special character (
        return "Moderate"
    else:
        return "Strong"
```