

My name is Chris Alister, I'm from 1MCAB (Reg:2347218, Year:2023-2025) and the Domain which I'll be working on is E-Commerce(Clothing).

1. Write a python program to count the frequency of any specific word (in your domain) in the paragraph.

```
In [ ]: #Paragraph based on my domain
paragraph = '''Ecommerce has revolutionized the clothing industry, providing an inn
With the rise of Ecommerce platforms dedicated to clothing, fashion enthusi
Online clothing stores offer a seamless shopping experience, enabling custo
This digital transformation in the clothing sector has not only expanded th
From trendy streetwear to elegant formalwear, Ecommerce in the clothing dom
'''

#paragraph split into words
s1 = paragraph.split(" ")

#word to be considered
word = "Ecommerce"

#initializing counter
count = 0
for i in s1:
    if i == word:
        count = count + 1

print("The number of occurrences of the word 'Ecommerce' is:", count)
```

The number of occurrences of the word 'Ecommerce' is: 3

->Write a python program to display all the datatypes of selected specific elements in the paragraph. (For example:- name - string, reg.no - int, marks - float, etc.)

```
In [ ]: def infer_data_types(paragraph):

    # Initialize an empty dictionary to store specific elements and their inferred
    data_types_dict = {}

    # Split the paragraph into individual elements based on commas and spaces
    elements = paragraph.split(', ')

    for element in elements:
        # Split each element based on the first space to separate the name from the
        name, value = element.split(' ', 1)

        # Infer the data type of the value
        if value.isdigit():
            data_type = 'int'
        elif value.replace('.', '', 1).isdigit():
            data_type = 'float'
        elif value.lower() in ('true', 'false'):
            data_type = 'bool'
        else:
```

```

        data_type = 'string'

        # Add the specific element and its inferred data type to the dictionary
        data_types_dict[name] = data_type

    return data_types_dict

# Example paragraph with specific elements and values
paragraph = '''name Chris, reg.no 2347218, marks 89.5, is_passed True, age 20'''

# Call the function to infer data types of specific elements in the paragraph
result = infer_data_types(paragraph)

# Display the inferred data types of specific elements
for element, data_type in result.items():
    print(f"{element} - {data_type}")

```

```

name - string
reg.no - int
marks - float
is_passed - bool
age - int

```

->Write a python program to count the number of alphabets, numeric and other special symbols in the paragraph.

```

In [ ]: def count_characters(paragraph):
    # Initialize counters for alphabets, numeric, and other special symbols
    alphabet_count = 0
    numeric_count = 0
    special_count = 0

    for char in paragraph:
        if char.isalpha():
            alphabet_count += 1
        elif char.isdigit():
            numeric_count += 1
        else:
            special_count += 1

    # Create a dictionary with the counts
    character_counts = {
        'Alphabets': alphabet_count,
        'Numeric': numeric_count,
        'Special Symbols': special_count
    }

    return character_counts

# Get user input for the paragraph
paragraph = input("Enter the paragraph: ")

# Call the function to count characters
result = count_characters(paragraph)

```

```
# Display the counts of alphabets, numeric, and special symbols
print("The paragraph typed is: ", paragraph)
print("Alphabets count:", result['Alphabets'])
print("Numeric count:", result['Numeric'])
print("Special symbols count:", result['Special Symbols'])
```

The paragraph typed is: My Domian is Ecommerce Clothing 2023!
 Alphabets count: 27
 Numeric count: 4
 Special symbols count: 6

2)->Create a Set with elements that consists of various data types (int, float, string, Boolean, etc. from your domain) and perform the functions pop(), clear(), discard() and del. Write the insights as docstring.

```
In [ ]: def create_ecommerce_data_types_set():
    ecommerce_data_types_set = {
        "Shirt",
        25,
        34.99,
        True,
        ("Black", "XL"),
        "Summer Sale",
        42.5,
    }
    return ecommerce_data_types_set

data_types_set = create_ecommerce_data_types_set()

# Perform the functions on the Set
popped_element = data_types_set.pop()
print("Popped Element:", popped_element)

data_types_set.clear()
print("Cleared Set:", data_types_set)

data_types_set.discard("Summer Sale")
print("Set after discarding 'Summer Sale':", data_types_set)

del data_types_set
print("Set deleted and no longer accessible.")
```

Popped Element: True
 Cleared Set: set()
 Set after discarding 'Summer Sale': set()
 Set deleted and no longer accessible.

->Update the Set with minimum 5 string attributes of your domain and arrange the Set in descending order.

```
In [ ]: def update_ecommerce_clothing_set():
    ecommerce_clothing_set = {
        "Shirt",
```

```

        "Pants",
        "Dress",
        "Shoes",
        "Hat"
    }

    # Add more string attributes to the set
    ecommerce_clothing_set.update({"Socks", "Tie", "Skirt"})

    return ecommerce_clothing_set

# Update the set with string attributes related to the ecommerce-clothing domain
updated_ecommerce_clothing_set = update_ecommerce_clothing_set()

# Arrange the set in descending order
sorted_ecommerce_clothing_set = sorted(updated_ecommerce_clothing_set, reverse=True)

# Display the sorted set
print("Sorted Set in Descending Order:", sorted_ecommerce_clothing_set)

```

Sorted Set in Descending Order: ['Tie', 'Socks', 'Skirt', 'Shoes', 'Shirt', 'Pants', 'Hat', 'Dress']

->Create a Tuple and Execute the packing and unpacking operations of tuples using the attributes of your domain.

```

In [ ]: def create_ecommerce_clothing_tuple():
        product_tuple = ("Shirt", "Blue", "XL", 25.0, True)
        return product_tuple

# Create a tuple representing product attributes from the ecommerce-clothing domain
ecommerce_clothing_tuple = create_ecommerce_clothing_tuple()

# Unpack the tuple into individual variables representing product attributes
product_name, product_color, product_size, product_price, product_availability = ec

# Display the unpacked variables
print("Product Name:", product_name)
print("Product Color:", product_color)
print("Product Size:", product_size)
print("Product Price:", product_price)
print("Product Availability:", product_availability)

```

Product Name: Shirt
 Product Color: Blue
 Product Size: XL
 Product Price: 25.0
 Product Availability: True

->Enter your domain name as characters and count any number of characters and print the count (for example – ('p','r','o','g','r','a','m') count of 'r' = 2)

```

In [ ]: def count_characters(input_string, target_character):
        return input_string.count(target_character)

```

```

# Get input from the user for the domain name as characters
domain_name = input("Enter your domain name as characters: ")

# Get the target character to count
target_char = input("Enter the character you want to count: ")

# Call the function to count the characters
character_count = count_characters(domain_name, target_char)

# Display the count of the target character
print(f"The Domain: '{domain_name}'")
print(f"Count of '{target_char}': {character_count}")

```

The Domain: 'Ecommerce Clothing Domain'
Count of 'o': 3

->Enter your domain name, execute all the slicing possibilities and also negative indexing.

```

In [ ]: def perform_slicing_and_negative_indexing(input_string):
        # Slicing
        slice1 = input_string[2:5]
        slice2 = input_string[:4]
        slice3 = input_string[6:]
        slice4 = input_string[::-1]

        # Negative indexing
        negative_index1 = input_string[-1]
        negative_index2 = input_string[-3:]

        return slice1, slice2, slice3, slice4, negative_index1, negative_index2

# Get input from the user for the input string
input_string = input("Enter your input string: ")

# Call the function to perform slicing and negative indexing
sliced_results = perform_slicing_and_negative_indexing(input_string)

# Display the results
print("Slice 1:", sliced_results[0])
print("Slice 2:", sliced_results[1])
print("Slice 3:", sliced_results[2])
print("Slice 4:", sliced_results[3])
print("Negative Index 1:", sliced_results[4])
print("Negative Index 2:", sliced_results[5])

```

Slice 1: omm
Slice 2: Ecom
Slice 3: rce Clothing Domain
Slice 4: niamod gnihtolc ecremmocE
Negative Index 1: n
Negative Index 2: ain