**Practice Questions: Strings**

1. **Slicing - Basic:** Given the string text = "Programming", write the code to extract and print the substring "gram".
2. **Slicing - Negative Indexing:** Given the string data = "Python\_Expert", write the code to extract and print the substring "Expert" using only **negative indices**.
3. **Length and Access:** Store your full name in a variable full\_name. Print both the total length of the string and the character at the very first index.
4. **Case Conversion:** Write a program that asks the user for a word and then prints that word three times: once in **all uppercase**, once in **all lowercase**, and once with only the **first letter capitalized**.
5. **Splitting and Joining:** Given the string sentence = "Python is powerful", first **split** the string into a list of words. Then, **join** the words back into a single string using a hyphen (-) as the separator.
6. **Finding and Indexing:** Given haystack = "banana, apple, orange", find the starting **index** of the word "apple" within the string. Then, use the .replace() method to change "apple" to "kiwi" and print the result.
7. **Stripping Characters:** Given the string messy = " \*\*\*Hello World\*\*\* \n", use a single operation to remove **both** the leading/trailing whitespace and the asterisk characters (\*).
8. **Checking Prefix/Suffix:** Write a conditional statement that checks if the string filename = "report.pdf" **ends** with .pdf **AND** if it **starts** with report. Print "File is valid" if both conditions are true.
9. **String Multiplication:** Use string multiplication to create and print a string consisting of ten dollar signs ($).
10. **String Formatting (f-string and alignment):** Given item = "Laptop" and price = 999.99, use a single **f-string** to print the output as:

Item: Laptop | Price: $999.99

(Ensure the price has exactly two decimal places).

**Practice Questions: Lists**

1. **Creation and Access:** Create a list named planets containing the names of four planets. Print the entire list, and then print only the third planet using its index.
2. **Modification and Deletion:** Given the list tasks = ["buy milk", "clean house", "pay bills", "wash car"]. First, **change** the second item to "buy bread". Then, **delete** the last item using the del keyword. Print the final list.
3. **Appending and Inserting:** Start with an empty list called schedule. Use the .append() method to add two items. Then, use the .insert() method to place a third item at the beginning (index 0). Print the resulting list.
4. **Slicing - Sublist:** Given the list numbers = [10, 20, 30, 40, 50, 60], create and print a new list called middle containing only the elements 20,30,40, and 50 using **slicing**.
5. **Iteration with for loop:** Write a for loop to iterate over the list fruits = ["apple", "banana", "cherry"] and print each fruit name on a separate line.
6. **Checking for Membership:** Write a conditional statement that checks if the string "lemon" is present in the list citrus = ["orange", "grapefruit", "lime"]. Print "Lemon found!" if true, otherwise print "No lemon here."
7. **Sorting and Reversing:** Given scores = [95, 78, 88, 92, 85], use a method to sort the list in **ascending order** (in-place). Then, use another method to **reverse** the order of the sorted list. Print the final result.
8. **Finding and Counting:** Given the list items = ['A', 'B', 'A', 'C', 'B', 'A']. Print the **index** of the first occurrence of 'C'. Then, print the total **count** of the element 'A'.
9. **List Copying (Shallow):** Create a list original = [1, 2, 3]. Create a **shallow copy** of this list named new\_list (do NOT use simple assignment =). Change the second element of new\_list to 99. Print both original and new\_list to show they are independent.
10. **Nested Lists and Access:** Create a list named matrix that contains two nested lists: the first nested list should be [1,2], and the second should be [3,4]. Write the code to access and print the number 4.

**Practice Questions: Tuples**

1. **Creation and Immutability:** Create a tuple named coordinates with the values (10,20). Write the code that would be used to try and change the second element to 30 (i.e., coordinates[1] = 30). What kind of error would this code produce?
2. **Access and Length:** Given the tuple alphabet = ('a', 'b', 'c', 'd', 'e'), print the total number of elements in the tuple. Then, access and print the second-to-last element using **negative indexing**.
3. **Tuple Unpacking:** Given the tuple student = ("Alice", 25, "Physics"), write a single line of code to unpack the values into three separate variables named name, age, and major. Print the value of major.
4. **Slicing:** Given the tuple data = (100, 200, 300, 400, 500), extract and print a new tuple containing only the middle three elements: (200,300,400).
5. **Single-Element Tuple:** Write the correct Python syntax to create a tuple named single\_item that contains only the string "Hello". Print the tuple and its type to verify its structure.
6. **Nested Tuples:** Given the tuple nested\_data = ( (1, 2), (3, 4), (5, 6) ), write the code to access and print the number 4.
7. **Looping:** Write a for loop to iterate over the tuple colors = ('red', 'green', 'blue') and print each element on a new line, along with its length.
8. **Checking for Existence:** Write a conditional statement that checks if the value 42 is present in the tuple scores = (15, 22, 42, 38). Print "Score found" if it is, and "Score not present" otherwise.
9. **Converting to List:** Given the tuple immutable\_data = ('a', 'b', 'c'), convert it into a mutable list named mutable\_list. Add the element 'd' to the end of mutable\_list. Print the final list.
10. **Finding and Counting:** Given the tuple path = ('N', 'E', 'S', 'W', 'N', 'E'), use the appropriate tuple methods to find the **index** of the first occurrence of 'S'. Then, print the total **count** of the element 'N'.

**Practice Questions: Dictionaries**

1. **Creation and Access:** Create a dictionary named inventory to store the count of three items: "apples" (30), "bananas" (15), and "oranges" (45). Write the code to access and print the count of "oranges".
2. **Adding and Updating:** Start with the dictionary user = {'username': 'coder', 'level': 5}. Write the code to **add** a new key-value pair, 'score': 1000, and then **update** the value of 'level' to 6. Print the final dictionary.
3. **Safe Access (The .get() method):** Given the dictionary config = {'timeout': 60, 'retries': 3}, write a print statement using the .get() method to retrieve the value for the key 'timeout'. Then, write another .get() call to retrieve a value for the non-existent key 'log\_level', providing a **default value** of "INFO".
4. **Removing Items (.pop()):** Given the dictionary data = {'a': 10, 'b': 20, 'c': 30}, use the .pop() method to remove the key-value pair for key 'b'. Print the value that was removed and the resulting dictionary.
5. **Iteration Over Keys:** Write a for loop to iterate over the **keys** of the dictionary stock = {'coffee': 150, 'tea': 80, 'sugar': 200} and print each key (the item name) on a new line.
6. **Iteration Over Values:** Write a for loop to iterate over the **values** of the stock dictionary from Question 5 and calculate the **total quantity** of all items. Print the total.
7. **Iteration Over Items:** Write a for loop using the .items() method to iterate over both the **key and value** of stock. Inside the loop, print a sentence for each item in the format: "The stock for [Key] is [Value]."
8. **Nested Dictionary Access:** Given the nested dictionary project = {'name': 'WebApp', 'details': {'version': 1.2, 'status': 'complete'}}, write the code to access and print the value of the nested key 'version' (i.e., 1.2).
9. **Dictionary Comprehension:** Use a **dictionary comprehension** to create a new dictionary where the keys are the numbers 1,2,3,4, and 5, and the corresponding values are the square of the key. Print the resulting dictionary.
10. **Using .keys() and Membership:** Given the dictionary settings = {'theme': 'dark', 'font': 'sans-serif'}, use the .keys() method along with the in operator to check if the dictionary contains the key 'theme'. Print whether the key was found or not.

**Practice Questions: Sets**

1. **Creation and Uniqueness:** Create a set named unique\_numbers from the following list: [5, 2, 5, 8, 2, 1]. Print the resulting set and explain why it has fewer elements than the starting list.
2. **Adding and Removing:** Start with the set letters = {'a', 'b'}. Use a method to **add** the element 'c'. Then, use the .discard() method to remove the element 'a'. Print the final set.
3. **Union Operation:** Given set\_a = {1, 2, 3} and set\_b = {3, 4, 5}, write the code to compute and print the **union** of the two sets using the pipe operator (|).
4. **Intersection Operation:** Given the same sets from Question 3, write the code to compute and print the **intersection** of the two sets, showing the common element(s).
5. **Difference Operation:** Given all\_chars = {'a', 'b', 'c', 'd'} and vowels = {'a', 'e', 'i'}, write the code to find and print the set containing elements in all\_chars but **not** in vowels (i.e., the difference).
6. **Symmetric Difference:** Given list1 = [1, 2, 3, 4] and list2 = [3, 4, 5, 6]. Convert both lists to sets and then compute the **symmetric difference** (^) between them. Print the result.
7. **Checking Subsets:** Write a conditional statement to check if small\_set = {10, 20} is a **subset** of large\_set = {10, 20, 30, 40}. Print a message indicating whether the condition is True or False.
8. **Converting from Tuple:** Given a tuple of duplicate names: names\_tuple = ("Tom", "Sam", "Tom", "Jane", "Sam"). Convert this tuple to a set to eliminate duplicates, then convert the result **back to a list** to get a sequence of unique names. Print the final unique list.
9. **Set Mutability (Conceptual):** Explain in one sentence why a list, which is a mutable type, **cannot** be an element of a set (e.g., my\_set = {1, [2, 3]}).
10. **Set Comprehension:** Use a **set comprehension** to create a set containing the squares of all odd numbers between 1 and 10 (inclusive). Print the resulting set.