

Assignment No-2

Based on UNIT-2

CSE114: Application based programming in python

Syllabus

---

List, Tuple and Dictionaries

Lists and Nested List: Introduction, Accessing list, Operations, Working with lists, Library Function and Methods with Lists

Strings: Introduction, Accessing items of a string, Operations, Working, Library Functions and Methods with strings.

Tuple: Introduction, Accessing tuples, Operations, Working, Library Functions and Methods with Tuples.

Sets: Introduction, Operations, Working, functions with sets. Difference between set and lists.

Dictionaries :Introduction, Accessing values in dictionaries, Working with dictionaries, Library Functions

---

1. Write a Python program to concatenate two lists and then perform element-wise multiplication of the resulting list by 3.
2. Given a nested list, extract and print the last element of each inner list using list indexing.
3. Create a Python script that takes a string as input and prints the string in reverse order, removing any spaces.
4. Define two tuples and perform the following operations: concatenate them, find the length of each tuple, and print the result.
5. Develop a Python program that takes two sets as input and prints the union and intersection of the sets.
6. Explain the purpose of the `append()` and `extend()` methods in Python lists. Provide examples demonstrating their usage.
7. Explore and demonstrate the usage of the `split()` method with a string containing multiple words. Print the resulting list.
8. Discuss the role of the `count()` method for tuples. Provide a scenario where this method could be useful and write a code snippet to demonstrate it.
9. Compare and contrast the difference between sets and lists. Provide specific examples illustrating when to use sets over lists and vice versa.
10. Create a Python dictionary representing a student's information (e.g., name, age, grade). Use the `get()` method to retrieve and print a specific value, handling the case where the key may not exist.

11. Write a Python function that takes two lists as parameters and returns a new list containing only the common elements between the two input lists. Ensure the result has no duplicate elements.
12. Implement a function that takes a string as input and returns a new string where each word is reversed. For example, if the input is "Hello World," the output should be "olleH dlroW."
13. Write a Python program that takes two sets as input and determines whether one set is a subset of the other. Print a message indicating the result.
14. Design a Python script that represents a simple student database using a dictionary. Allow the user to add new students, update their grades, and retrieve the average grade for all students in the database.
15. Create a Python program that initializes a tuple of tuples, where each inner tuple represents a student with (name, age, grade). Sort the students based on their grades in descending order and return the sorted list.
16. Write a Python function that takes a list of strings as input and returns a new list containing only the strings with more than five characters. Additionally, reverse the characters in each string.
17. Write a Python script that takes a dictionary containing student names as keys and their corresponding grades as values. Sort the dictionary based on grades in descending order and print the sorted results.
18. Given a list of tuples representing books with (title, author, publication year), create a Python function that sorts the list based on the publication year in ascending order. Return the sorted list.