1. What will be the output of the following code?

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
print(arr[1:4])
```

a) [1, 2, 3]
b) [2, 3, 4]
c) [3, 4, 5]
d) Error
Answer: b) [2, 3, 4]

2. Which of the following functions gives the mean of a NumPy array?

a) np.average()
b) np.mean()
c) np.median()
d) Both a and b
Answer: d) Both a and b

1. Explain the difference between NumPy arrays and Python lists. Why are NumPy arrays preferred in data science? Give one example code snippet to demonstrate the difference.

A:

- NumPy arrays are homogeneous (store elements of the same type) and optimized for mathematical operations, while Python lists can store heterogeneous data but are slower for numerical computation.
- NumPy arrays use contiguous memory blocks, enabling vectorized operations, while lists require iterative looping

Example:

```
import numpy as n
# Python list multiplication
list1 = [1, 2, 3, 4]
list_result = [x*2 for x in list1]
# NumPy array multiplication
arr = np.array([1, 2, 3, 4])
arr_result = arr * 2
print("List result:", list_result)   # [2, 4, 6, 8]
print("NumPy result:", arr_result)   # [2 4 6 8]
```

2. Differentiate between lists, tuples, and strings as sequence data types in Python with suitable examples.

A:

In Python, lists, tuples, and strings are all examples of sequence data types, but they differ in their properties and usage. A **list** is a mutable sequence, meaning its elements can be modified after creation. Lists are written using square brackets [ ] and are commonly used when we need a collection of items that can change, such as **L = [1, 2, 3],** where we can update L[0] = 10 to get [10, 2, 3]. On the other hand, a **tuple** is an immutable sequence, written using parentheses ( ). Once created, the elements of a tuple cannot be changed, making it suitable for fixed collections of values, for example**, T = (1, 2, 3).** Attempting to modify T[0] would result in an error. Similarly, a **string** is also an immutable sequence, but it specifically stores textual data and is enclosed in single or double quotes, such as **s = "python".** While we can access characters individually using indexing, like s[0] which gives 'p', we cannot modify the characters since strings are immutable. Thus, the key difference lies in mutability lists are mutable, while tuples and strings are immutable and in their typical usage, with strings being specialized for handling text.