PART 1: Python

1) What is dynamic typing in Python?

In Python, you don't need to declare the type of a variable (like int or string) when you create it. The type is determined at runtime based on the value assigned. You can also reassign a variable to a different type later.

- **Example:** x = 5 (int), then x = "Hello" (string) is perfectly valid.

2) What is the difference between mutable and immutable data types?

- **Mutable:** Can be changed (e.g., Lists, Dictionaries, Sets).
- **Immutable:** Cannot be changed; any "change" creates a new object (e.g., Strings, Tuples, Integers).

3) What are local and global variables?

- **Local Variables:** Defined inside a function. They can only be accessed within that specific function.
- **Global Variables:** Defined outside any function. They can be accessed throughout the entire script.

4) What is the purpose of the return statement in a function?

The return statement exits a function and sends a result back to the "caller." Without a return statement, a function completes its task but provides a value of None to the rest of the program.

5) What is list comprehension?

A concise, one-line way to create new lists based on existing sequences. It is faster and more readable than using a traditional for loop.

- **Example:** squares = [x**2 for x in range(5)]

6) Difference between break, continue, and pass

Break: Exits the loop entirely.

Continue: Skips the current iteration and moves to the next one.

Pass: A "do-nothing" placeholder used when code is syntactically required but you don't want to execute anything.

7) What is recursion? Give a simple example.

Recursion is when a function calls itself to solve a smaller version of the same problem. It must have a "base case" to prevent an infinite loop.

- **Example:** Calculating Factorial.

```
def factorial(n):
  if n == 1: return 1 # Base case
  return n * factorial(n - 1)
```

8) What is a lambda function?

A small, anonymous function defined without a name using the lambda keyword. It can have any number of arguments but only one expression.

- **Example:** add = lambda x, y: x + y

9) What is the difference between is and ==?

- **== (Equality):** Checks if the values of two objects are equal.
- **is (Identity):** Checks if two variables point to the exact same object in memory.

10) What are docstrings and why are they important?

Docstrings (Documentation Strings) are literal strings written using triple quotes """...""" at the start of a function, class, or module.

- **Importance:** They explain what the code does, helping other developers understand your logic. They can be accessed using the help() function or __doc__ attribute.

PART 2: NumPy

1. What is vectorization in NumPy?

Vectorization is the process of performing operations on entire arrays at once without using explicit for loops. It is much faster because it leverages optimized C code under the hood.

- **Example:** Multiplying two arrays element-wise (arr1 * arr2) instead of looping through every index.

2. What is the difference between reshape() and resize()?

- **reshape():** Changes the shape of an array but returns a new view. It does not change the original array's data.
- **resize():** Changes the shape of the array in-place. If the new size is larger, it may fill extra elements with zeros (depending on the method used).

3. What are NumPy dimensions and axes?

Dimensions (ndim): The number of indices needed to access an element (e.g., 1D is a list, 2D is a table).

Axes: Directions along which operations happen.

- **Axis 0:** Operates vertically (down the rows).
- **Axis 1:** Operates horizontally (across the columns).

4. What is slicing in NumPy arrays?

Slicing allows you to extract a portion of an array using the syntax [start:stop:step]. In NumPy, slicing returns a "view" of the data, meaning changes to the slice will affect the original array.

- **Example:** arr[1:5] gets elements from index 1 to 4.

5. How does NumPy help in mathematical computations for AI?

NumPy is the backbone of AI because it handles the Linear Algebra required for neural networks. It efficiently manages large matrices (weights/biases) and performs high-speed calculations (Dot Products) that are essential for training models.

PART 3: Pandas

1. What is data cleaning in Pandas?

Data cleaning is the process of fixing or removing "bad" data in a dataset. This includes:

- Removing duplicates.
- Fixing structural errors (typos).
- Handling missing values or outliers.
- Ensuring consistent formatting.

2. What is the difference between loc and iloc?

- **loc (Label-based):** Used to select data by the name of the row or column.
- **iloc (Integer-based):** Used to select data by its numerical index (position starting at 0).

3. What are missing values and how does Pandas handle them?

Missing values are represented as NaN (Not a Number) in Pandas.

- **Handling them:**
    - dropna(): Removes rows or columns with missing values.
    - fillna(): Replaces missing values with a specific value (like the mean or median).

4. What is groupby() and why is it used?

The groupby() method allows you to split data into groups based on some criteria, apply a function (like sum or mean), and combine the results. It is used for data aggregation.

- **Example:** Finding the average sales per city.

5. How does Pandas help in data preprocessing for AI models?

Pandas is used to transform raw data into a format AI models can understand. Key tasks include:

- **Feature Selection:** Choosing the most relevant columns.
- **Encoding:** Turning text (categorical data) into numbers.
- **Scaling:** Normalizing data so all features are on a similar scale.