

**Q1. What are the benefits of the built-in array package, if any?**

Answer : Arrays represent multiple data items of the same type using a single name.

In arrays, the elements can be accessed randomly by using the index number.

Arrays allocate memory in contiguous memory locations for all its elements. Hence there is no chance of extra memory being allocated in case of arrays. This avoids memory overflow or shortage of memory in arrays.

**Q2. What are some of the array package's limitations?**

Answer : The number of elements to be stored in an array should be known in advance.

An array is a static structure (which means the array is of fixed size). Once declared the size of the array cannot be modified. The memory which is allocated to it cannot be increased or decreased.

Insertion and deletion are quite difficult in an array as the elements are stored in consecutive memory locations and the shifting operation is costly.

Allocating more memory than the requirement leads to wastage of memory space and less allocation of memory also leads to a problem.

**Q3. Describe the main differences between the array and numpy packages.**

Answer : The array package doesn't provide any help with numerical calculation with the items inside it in number form while NumPy give you a wide variety of numerical operations

An array is a single dimensional entity which hold the numerical data, while numpy can have more than 1 dimension.

In case of array, item can be accessed by its index position and it is easy task while in numpy item is accessed by its column and row index, which makes it slightly time taking. Same goes with appending operation.

In case of array we do not form a tabular structure, while in numpy it forms a tabular structure

**Q4. Explain the distinctions between the empty, ones, and zeros functions.**

Answer : **empty function**: An empty function is a function that does not contain any statement within its body. If you try to write a function definition without any statement in python ,it will return an error. To avoid this, we use **pass** statement. **pass** is a special statement in Python that does nothing. It only works as a dummy statement

**ones**: This function returns a new array of given shape and data type, where the element's value is 1.

**zeros**: This function returns a new array of given shape and data type, where the element's value is 0

**Q5. In the from function function, which is used to construct new arrays, what is the role of the callable argument?**

Answer : Its function is to execute the function over each coordinate and the resulting array. The function is called with N parameters, where N is the rank of shape. Each parameter represents the coordinates of the array varying along a specific axis.

**Q6. What happens when a numpy array is combined with a single-value operand (a scalar, such as an int or a floating-point value) through addition, as in the expression  $A + n$ ?**

Answer : If any scalar value such as integer is added to the numpy array then all the elements inside the array will add that value in it.

**Q7. Can array-to-scalar operations use combined operation-assign operators (such as  $+=$  or  $*=$ )? What is the outcome?**

Answer : It will carry out provided operation on all elements of array.

**Q8. Does a numpy array contain fixed-length strings? What happens if you allocate a longer string to one of these arrays?**

Answer : Yes it is possible that we can include a string of fixed length in numpy array. The dtype of any numpy array containing string values is the maximum length of any string present in the array. Once set, it will only be able to store new string having length not more than the maximum length at the time of the creation. If we try to reassign some another string value having length greater than the maximum length of the existing elements, it simply discards all the values beyond the maximum length except upto those values which are under the limit.

**Q9. What happens when you combine two numpy arrays using an operation like addition (+) or multiplication (\*)? What are the conditions for combining two numpy arrays?**

Answer ; It will simply add or multiply element to element at same position.

**Q10. What is the best way to use a Boolean array to mask another array?**

Answer : Using `masked_where` of numpy package.

**Q11. What are three different ways to get the standard deviation of a wide collection of data using both standard Python and its packages? Sort the three of them by how quickly they execute.**

Answer : `np.std` and `math` package can be used individually.

**12. What is the dimensionality of a Boolean mask-generated array?**

Answer : It will have same dimensionality.