Data Structures In JavaScript   Day 21st

A JavaScript Data Structure is a specific technique to organize and store data in a computer so that we can access and modify it efficiently. More accurately, it is a collection of data values, the relationships among them, and the functions or operations that we can apply to the data.

Some of the advantages of using data structures are as follows:

- They simplify the processing of data on a computer.
- They provide an easy way to manage large datasets.
- These are essential for designing efficient program algorithms.

Types of Data Structures in JavaScript.

a). **Linear Data Structure**.

A linear data structure traverses its elements sequentially. We can reach only one data item directly. To access other elements, you need the help of that base element.

b) **Non-linear Data Structure**.

Unlike linear data structures, non-linear data structures don't traverse in a sequence. Every data item connects with numerous other items, reflecting specific relationships.

c). **Static Data Structure**.

Static data structures have a fixed memory size, that is, you need to state the maximum size of the structure well in advance. We cannot allocate the memory later. The example of static data structure is arrays.

Dynamic data structure.

Dynamic data structures differ from static data structures in the way that we can modify the memory size allocated to it. These include *queue, stack and linked lists*.

– **Array**.

According to W3School,JavaScript arrays are used to store multiple values in a single variable.

– **Stack**.

A stack is an ordered list which follows LIFO (last in first out) algorithm. You can access the elements of a stack from only a single end, called **top**. A very common example of a stack, we see in our daily lives, is the stack of chairs. You can only access the top chair at a time. The only way to reach a chair in the middle is by going through all the chairs above it.

– **Queue**.

A queue is another type of ordered list which follows FIFO (first in first out) algorithm. It has two ends, front (elements added) and rear (elements removed). A common example you may see is a queue of cars in a one–way lane. The first car that enters the lane is the first one to exit. Also, the car in the middle cannot exit the queue until all the cars before it.

– **Linked list**.

A linked list maintains a list in the memory. The script has the base address of the first element, that contains the link to the next element in the list. You can only access an element (except the base element) using the previous element. Linked lists store these elements in what we call a node. A node comprises of a data value and the link (address) to the next element. The first node is the head while the last node is the tail of the list.

– **Tree**.

A JavaScript tree is a special data structure that implements the hierarchical tree structure with a root node, child and parent nodes and leaf nodes represented as a set of linked nodes. A tree data structure is a collection of nodes, starting with a root node, with data values in each of the nodes along with the reference to the child nodes.

The root node has no parent node and the leaf nodes have no child nodes. If the child node of a parent node has one or more child nodes, it is a subtree. In the diagram above, the nodes BEF (T1) and DGH (T2) form two subtrees.

– **Graph**.

A graph is a group of a finite number of vertices and edges that connect these vertices. The edges can be 'directed'(directed graph) and 'undirected' (undirected graph). Unlike trees, who maintain a parent–child relationship between their nodes (vertices), the nodes of the graph maintain a complex relationship among them.

– **Hash Table**.

A hash table, or hash map, is a data structure that implements associative arrays since it maps keys to values. It uses a hash function to determine the index of the data value in the array (bucket). These buckets help to identify the storage location of the data we want.

Today we have defined Data Structures and looked at different types of Data Structures in JavaScript, this is the best part of programming. Data Structures are key to good program irregardless of the language used. Tomorrow, on 22rd Day, we will learn about arrays and implement a simple search app.

Best Wishes.

Lux Tech Academy.