**BLOG ON JAVA SCRIPT.**

Every programming language has built-in data structures. Data structures are a method of storing and organizing data for efficient access and manipulation by all users and devices. The common data structures in JavaScript include; stacks, arrays, queues, linked lists and trees. Each of these makes it possible to call upon data existing in a specific state and use it in several ways.

Algorithms are specific tasks laid out in a step-to-step manner. Basically, data structures allow algorithms to use data stored in the computer. Data structures must be written properly in order for the data stored within to be accessed.

**DATA STRUCTURES IN JAVASCRIPT.**

**ADVANTAGES OF DATA STRUCTURES.**

* Enables us to manage and utilize large datasets.
* Search for particular data from a database.
* Handle multiple requests from users at once.
* Simplify and speed up data processing.
* Design algorithms that are tailored towards particular programs
* **STACK.**

A stack data structure holds a list of elements and works on the principle of LIFO. This means the element that is added most recently is the element that is moved first. Push and pop are the main operations in stack.

* **QUEUE**

Though similar to stack, queue occurs when you store the requests in the order they were received in.

Queues use the principle of FIFO. This is helpful as a buffer for requests, each request in the order it was received until it can be processed.

* **LINKED LIST**

Linked lists use a referencing system instead of using physical placements of data in the memory. They store the elements in nodes. These nodes contain a pointer to the subsequent node, which creates a link between all the nodes. With a linked list system, you can efficiently insert and remove the items without reorganising.

* **TREES**

A Tree has a root node and other nodes branch from this root node. A root node has various child nodes, which have reference to all the elements. The child nodes branch off from other child nodes. When a node has linked child nodes, it is called an internal node. Odes that that do not have child nodes are called external nodes. A tree structure is helpful in storing hierarchical data, and it can help with searches where one needs to order the data.

* **ARRAYS**
* **Tom odell another love**