**Binary Heap in JavaScript:**

A binary heap is a complete binary tree where every parent node has a value less than or equal to (for a min heap) or greater than or equal to (for a max heap) the values of its children. It is commonly used to implement priority queues and heap sort algorithms efficiently.

**Heap Operations:**

1. Insertion:

* To insert a new element into a heap, it is added at the bottom level, maintaining the complete binary tree property. Then, it is recursively swapped with its parent node until the heap property (min or max) is satisfied.

2. Deletion:

* Deleting the root node from a heap involves removing the root element and replacing it with the last element in the heap. Then, the heap property is restored by recursively swapping the new root node with its children until the heap property is satisfied.

3. Heapify:

* Heapify is an operation that transforms an array into a heap. It starts from the last non-leaf node and applies heapify down operation recursively to ensure that every subtree rooted at that node satisfies the heap property.

**Priority Queue using Heaps:**

A priority queue is an abstract data type that operates similar to a regular queue or stack but where each element has a priority associated with it. Elements with higher priority are dequeued before elements with lower priority.