NodeJs Buffer Module

* In Node.js, the Buffer module is used for handling binary data. It provides a way to manipulate and interact with streams of raw data, such as images, audio, or network packets, in a more efficient manner.
* It allows you to store, read, modify sequence of bytes.
* Buffer can be created in different ways you can create buffer by specifying its size, or you can initialize it with data from string , array, or other Buffer.
* Once you have buffer you can modify , read the content.
* Think of a Buffer as a container that holds binary data in a specific format. It allows you to read from or write to the data in a controlled way. You can think of it as an array of integers, where each integer represents a single byte of data.
* Here are a few key points to understand about the Buffer module:
* 1. Creation: You can create a Buffer in Node.js using various methods, such as by specifying the data directly, converting from a string, or reading from a file or network socket.
* 2. Data Manipulation: Once you have a Buffer, you can read and modify its content. For example, you can read individual bytes, write new values to specific positions, or perform operations like copying, slicing, and concatenating Buffers.
* 3. Encoding: Buffers can store binary data, but they can also be encoded into different formats, such as Base64 or UTF-8, which allow you to work with text-based data. Node.js provides methods to convert between Buffers and strings using different encodings.
* 4. Efficiency: Buffers are designed to be memory-efficient and performant when working with binary data. They are particularly useful when dealing with large files, network streams, or when performing low-level operations.
* 5. Use Cases: Buffers are commonly used in scenarios where direct manipulation of binary data is necessary, such as when working with file systems, network protocols, encryption/decryption, or when interacting with external libraries that require binary data.
* Overall, the Buffer module in Node.js provides a powerful and efficient way to handle binary data, making it an essential part of many Node.js applications.