# Indexed Collection:

Indexed Collections are collections that have numeric indices, meaning data are ordered by an index value.

1. Arrays
2. Typed Arrays:-

JS typed arrays are ~~pretty~~ array-like objects that provide a mechanism for reading and writing raw binary data in memory buffers.

A memory buffer is a region of memory that is allocated for the purpose of temporarily storing data. It essentially, is a contiguous block of memory where we can read from or write to binary data.

Typed arrays are useful when interacting with platform features, such as audio and video manipulation, access to raw data using websockets, and so forth.

Each entry in JS-typed arrays is a raw binary value in one of number of supported formats, from 8-bit integers to 64-bit floating-point numbers.

Implementation is split into buffers and views. A buffer is an object representing a chunk of data, with no format and no mechanism for accessing contents. A view is used to access the memory contained in a buffer. A view provides a context - that is, a data type, starting offset, and number of elements.

* When the JS engine scans a script file, it makes an environment called the Execution Context that handles the entire transformation and execution of the code.

## Buffers -

Types are ArrayBuffer and SharedArrayBuffer. They are generic objects that just contain raw data. Supported actions:

a) Allocate - a new memory span with 0 is initialised, as soon as a new buffer is created.

b) Copy - Array's slice() method can be used to copy a portion of memory.

c) Transfer - Using the transfer() and transferToFixedLength() methods you can transfer the ownership of the memory span to a new buffer object. After the transfer, the original buffer is no longer useful or usable. A SharedArrayBuffer cannot be transfered because it already is shared by all execution context.

d) Resize - Using the resize() method, you can resize the memory span (either claim more space, as long as it doesn't pass the pre-set maxByteLength limit, or release some memory space).

## Views -

There are two types of views: typed array views and DataView. Typed arrays provide utility methods that allow you to conveniently transform binary data. DataView is more low-level and allows granular control of how data is accessed.

| Type | Value Range | Size in bytes | Web IDL Type |
|---|---|---|---|
| Int8Array | -128 to 127 | 1 | byte |
| Uint8Array | 0 to 255 | 1 | octet |
| Uint8ClampedArray | 0 to 255 | 1 | octet |
| Int16Array | -32768 to 32767 | 2 | short |
| Uint16Array | 0 to 65535 | 2 | unsigned short |
| Int32Array | -2147483648 to 2147483647 | 4 | long |
| Uint32Array | 0 to 4294967295 | 4 | unsigned long |
| Float32Array | -3.4e38 to 3.4e38 | 4 | unrestricted float |
| Float64Array | -1.8e308 to 1.8e308 | 8 | unrestricted double |
| BigInt64Array | $-2^{63}$ to $2^{63}-1$ | 8 | bigint |
| BigUint64Array | 0 to $2^{64}-1$ | 8 | bigint |