\

Typed Arrays

Contents of the arrays are strictly controlled. Every element in an array is going to be either 8,16,32, float 64 bit introdure.

Usecase:

If you are creating web graph,.i.e being able to do 3D graph on the web. This required being able to the pass as array of numbers. You want to store a pixel value like red blue green value for every pixel. Things like that you want to pass the data to some sort of native interface that was expecting to receive the only numbers thats specific size.

->Typed Arrays are used by:WebGL,Canvas,Web Audio API, XMLHttpRequests,Fetch API,Web Sockets,Web workers,Media Source API and File APIs

--->Here one byte is equal to 8 bits.

Int8Array: 8 bit array intodure(Every introdure in array must be 8 bit) we can use value negative 128 to 127

Uint8Array:Unsigned inrodure.... Unsigned means no zeros...we can use 0 to 255.

Uinit8ClamppedArray(0-255): here clampped means it is built-in data validation. if you try to put number which is less than zero then it will convert it to zero. If you put the number which is greater than 255 then it will automatically becomes 255.

similarly we have 8 bit, 16 bit signed and unsigned.

Int16Array,Uint16Array,Int32Array,Uint32Array,Float32Array,Float64Array

The container for all above is called Array Buffer

ArrayBuffer:

The ArrayBuffer is a data type that is used to represent a generic, fixed-length binary data buffer. You can't directly manipulate the contents of an ArrayBuffer; instead, you create a typed array view or a DataView which represents the buffer in a specific format, and use that to read and write the contents of the buffer.

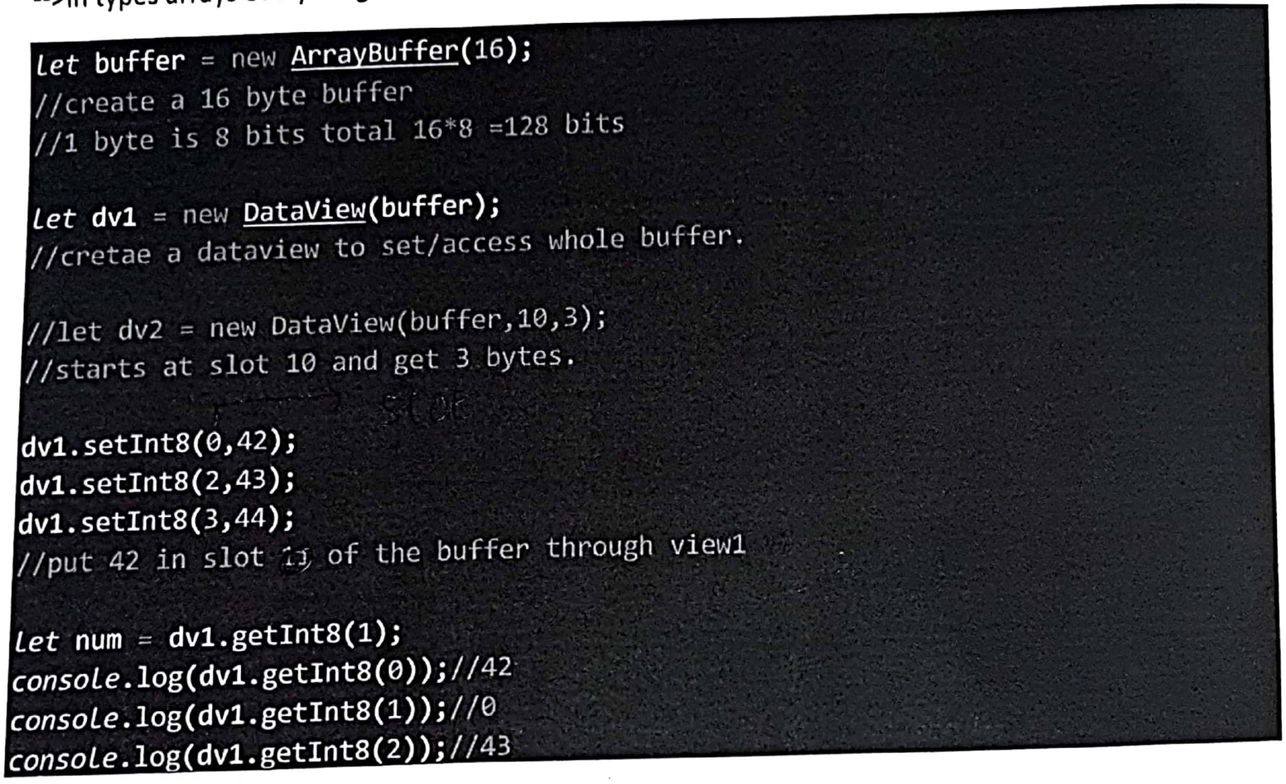
->ArrayBuffer is used to represent chunk of data of raw buffer data.

->You are not allowed actually go into the arraybuffer and say hey take the bunch of data and change them. if you want to interact with array buffer then you have to use data view.

DataView:

Data view contains get and set methods

-->In types arrays everything in the array should have same data type.



```
let buffer = new ArrayBuffer(16);
//create a 16 byte buffer
//1 byte is 8 bits total 16*8 =128 bits

let dv1 = new DataView(buffer);
//cretae a dataview to set/access whole buffer.

//let dv2 = new DataView(buffer,10,3);
//starts at slot 10 and get 3 bytes.
                          slot
dv1.setInt8(0,42);
dv1.setInt8(2,43);
dv1.setInt8(3,44);
//put 42 in slot 11 of the buffer through view1

let num = dv1.getInt8(1);
console.log(dv1.getInt8(0));//42
console.log(dv1.getInt8(1));//0
console.log(dv1.getInt8(2));//43
```