**Eloquent JavaScript**

**Capter 1: Values and operator**

${} - what is written in the second bracket that is1) computed 2) converted into string 3) write into that position

Binary operator : Operator that takes two value to calculate . +/ -/\*/%

Unary operator: that takes one value to operate. Typeof() is an unary operator.

NaN, Infinity is value .

NaN != NaN cause NaN results from nonsensical operation so two nonsensical operation is not same.

Logical operator: and &&

Or ||

Not !

Type coercion: when js need to operate a wrong type of value, js immediately convert the value to it needs that is not we expect , that is called type coercion. JS do type conversion immediately.

False || null => null

Null || false => false

0, NaN, “”----- false in boolean

Short-Circuit evaluation: when true || x it is true no need for evalueation the x value

False && X => false no need to evaluate right part of logical operation and that is called short-circuit evaluation.

Ternary Operator: ?: to pick one of two values based on third value

**Chapter-2**

**Programme Structure**

Expression: a fragment of code that produces value. Example: 22, “psycology”,

Statement: full sentence. Example:1; !false; let box = 32;

Propmt is just like input of python that takes a value

Whether alert shows the text as an alert in browser.

Switch statement er vitore ekta exprssion nibe , jar theke ekta value pabo.

Oi value ta kon case e ta ache ta search kore ber korbe.

Jokhon ektai expression er jonno onekgulo case hobe only tokhon switch korte hobe. Different expression is not possible.

**Chapter-3**

**Function**

Function can define in three ways.

1. Function functionName(argument1, argument2){code}
2. Let functionName/variable = function(argumen1, argument2){code}
3. Arrow function, let variableName = (argumen1, argumen2) => {code block}

Every time a function is called the context is stored in top of the stack.

Function (a) , if we give 3 arguments it’ll take the first argument and won’t show any error.

**Chapter-4**

**Data Structure:Object and array**

It’s a common practice to declare an object by Const. Cause an object is declared and not reassign. We can modify , add , assign value of an object not the address reference is always same.

This = oi object ta kei mean kore. Object er moddhe jodi oi object er property or method k indicate kori then this keyword use kori.

To check a value is in an array,

List1.includes(value) =>returns true or false

Loop: for (let index = 0; index < list.length ; index++){}

Another way: for( let index of list){}

List.unshift(value) => value list er first e add kore

List.shift() => first value delete kore

To split a string into a list of element we use split function.

Words = sentence.split(“ “) => white space e split korbe

Ekta list k jodi string e convert korte chai then use join.

Sentence = word.join(“ “)

List.lastIndexOf(value) =>last theke khoja shuru kore

An argument = …numbers allowes an array of numbers I.e multiple numbers

**Chapter-5**

**Higher Order function**

**Abstraction:**

when we hide the details and show only what we exactly doing by only calling the function is called abstraction.

Like sum(range(1, 10))

Here sum function is doing summation which we wright is another function no need to show . and range function is also doing a detaile work.

**CallBack:**

A callback function is passed as an argument to another function. We pass the function as object reference in argument and in the function we call the callback function.

Why we use callback function? When we want a work have to be done after another specific work then we use callback. Asyncronous task.

**Anonymous function:**

a function definition without a name in JavaScript is called as an “anonymous function”.

Function is an object.jokhon ekta function define kori, er vitore ja ase shob holo object er property, method. Function er address boshe jokhon function define kori.

What is first class function?

**Higher Order Function:** jei function onno function er upor kaj kore I.e argument hishebe nibe, or return function korbe .

Build-in HOF holo array er map, forEach, filter

Array.map(parameter) ekta function parameter ney ja array er every element er upor oi function ta apply kore. Map, filter method array take change kore na new ekta array create kore