Module 31-JS Recap and Basic ES6, ES2015

31-0 Introduction

JavaScript ES6 (also known as ECMAScript 2015 or ECMAScript 6) is **the newer version of JavaScript that was introduced in 2015**. ECMAScript is the standard that JavaScript programming language uses. ECMAScript provides the specification on how JavaScript programming language should work.

31-1 Simple Recap of JavaScript six core parts

```
// variable
var deposit = 400;

// condition
if (deposit > 500) {

}
else if (deposit < 200) {

}
else if (deposit == 500) {

}
else if (deposit != 500) {

}
else if (deposit >= 500) {

}
else if (deposit <= 500) {

}
else {

}

// array
const numbers = [45, 6587, 124, 45, 1, 365];
const numberCount=numbers.length;
numbers.pop();
numbers.push(111);
numbers[2]=555;
```

```
// check whether 222 included in the array
if(numbers.indexOf(222)!=-1)
{

}
if(numbers.includes(222))
{

}

// loop
// while,for
for(const numbers of numbers)
{

}

// function
function fullName(first,second)
{
    const name=first+' '+second;
}

const person=fullName("Hamid","Hosen");

// object
const bottle = { color: 'yellow', contain: 'water', price: 50 }
```

31-2 Recap of DOM Manipulation and event handler

```
document.getElementById("add-
border").addEventListener('click',function(){

    const container = document.getElementById("friend-container");
    container.style.border="2px solid red";
    container.style.margin="10px";
    container.style.padding="10px";
    container.style.borderRadius="10px";
});

function addBackgroundColor()
{
```

```
    const friend = document.getElementsByClassName("friend");

    for(const friends of friend){
        friends.style.backgroundColor='green';
    }
}

document.getElementById("add-
friend").addEventListener('click',function(){

    const container = document.getElementById("friend-container");
    const friendDiv=document.createElement('div');
    friendDiv.classList.add("friend");

    friendDiv.innerHTML = `
            <h3 class="friend-name">New Friend</h3>
            <p>Quam, sapiente.</p>
    `;
    container.appendChild(friendDiv);
});
```

31-3 Breakup with var to have a relationship with Let and const

ECMAScript=ES6(6-version) is **the scripting language that forms the basis of JavaScript**. ECMAScript standardized by the ECMA International standards organization in the ECMA-262 and ECMA-402 specifications. The following ECMAScript standards have been approved or are being worked on: Name.

ES6 or ES2015

Technical Committee 39

What is a "TC39"? TC39 stands for "**Technical Committee 39**" and is the committee responsible for iterating on and evolving the ECMAScript language specification. The committee generally meets around 6 times a year to discuss progress on pending proposals and collectively work on moving forward with changes to the spec.

31-4 Function default parameter for not provided values

```
function add(num1, num2 = 33) {
```

```
  // option 2
  // num2 = num2 || 0;
  // option 1
  // if (num2 == undefined) {
  //     num2 = 0;
  // }
  const total = num1 + num2;
  return total;
}
const result = add(15, 0);
console.log(result);

function fullName(first, last = "Chowdhury") {
  const name = first + " " + last;
  return name;
}

console.log(fullName("Hamid","Hosen"));
```

31-5 Template string, multiple line string, dynamic string

String Interpolation-> Syntax: ${...}

স্ট্রিং এর মধ্যে ভেরিয়েবল এর নাম দিয়ে ভেরিয়েবলের মান বসানোকে string interpolation বলে;

```
const priya = "Asif Akbar";
const meye = "Meye tumi ki dukkho chino";
const kobita = `kobita tumi sopno charini`;
const multiLine =
  "This is my first line. \n" +
  "this is my second line.\n" +
  "third line text here\n" +
  "fourth line text here";

// console.log(multiLine);

const multiLineNew = `this is is multi line
this is second line
this is third line
fourth line
`;

// console.log(multiLineNew);
```

```
const friends = ["abul", "babul", "kabul", "sabul"];
console.log(friends);

const count = 5;
const old = '<h3 class="friend-name">Friend-5</h3>';
console.log(old);

const old2 = '<h3 class="friend-name">Friend-' + count + "</h3>";
console.log(old2);

const old1 = `<h3 class="friend-name">Friend-${count}</h3>`;
console.log(old1);

const new1 = `<h3 class="friend-name">Friend-${friends.length}</h3>`;
console.log(new1);

const first = "Mamun";
const last = "Chowdhury";

const fullOld = "This person name is " + first + " " + last;
console.log(fullOld);

const fullNew = `This person name is: ${first} ${last}. Has money ${
  (friends.length + 10) * 500
}. He lives in Dhaka.`;

console.log(fullNew);
```

31-6 Arrow function, multiple parameter, function body

```
// function declaration
function add(num1,num2)
{
    const sum=num1+num2;
    return sum;
}

// function expression
const add2 = function add2(num1, num2) {
  const sum = num1 + num2;
  return sum;
};
```

```
// function expression (anonymous)
const add3=function(num1,num2)
{
    return num1+num2;
}

//arrow function
const add4=(num1,num2)=>num1+num2;

const sum1 = add(15, 17);
const sum2=add2(15,17);
const sum3=add3(15,17);
const sum4=add4(15,17);

console.log(sum1);
console.log(sum2);
console.log(sum3);
console.log(sum4);

// document.getElementById('my-btn').onclick = function handleEvent(){
// }
```

31-7 More Arrow functions and big arrow function

```
const add = (num1, num2) => num1 + num2;
const sum = add(22, 90);
console.log(sum);

const multiply = (num1, num2, num3) => num1 * num2 * num3;
const result = multiply(4, 12, 3);
console.log(result);

const tenTimes = (num) => num * 10;
const output = tenTimes(17);
console.log(output);

const fiveTimes = num => num * 5;
const value = fiveTimes(17);
console.log(value);

const getName = () => "Brandon Sam";
const name = getName();
console.log(name);
```

```
const doMath = (x, y) => {
  const sum = x + y;
  const diff = x - y;
  const result = sum * diff;
  const output = result * 5;
  return output;
};

const total = doMath(12, 5);
console.log(total);
```

31-8 Spread operator, array max, copy array

```
const numbers = [23, 65, 99, 21, 34];
console.log(numbers);
console.log(...numbers);

const max = Math.max(23, 99, 65, 21, 34);
const maxInArray = Math.max(...numbers);
console.log(max, maxInArray, Math.max(numbers));

const numbers2 = [...numbers, 88];
numbers.push(55);
console.log(numbers);
console.log(numbers2);
```

The spread operator is **a new addition to the set of operators in JavaScript** ES6. It takes in an iterable (e.g an array) and expands it into individual elements. The spread operator is commonly used to make shallow copies of JS objects. Using this operator makes the code concise and enhances its readability.

31-9 Module Summary and ES6 Recap

```
let priyoPerson = "Koolsum Begum";
priyoPerson = "Momotaj Begum";
const hubby = "Akbar the great";

// default parameter
function getName(first, last = "Chowdhury") {
```

```
  return first + " " + last;
}

// template string
const myPeople = `My lovely person is are ${hubby} and his fullName is
${getName(
  "Akbar"
)}. My name is ${priyoPerson}.`;

console.log(myPeople);

// arrow function
const getName2 = (first, last) => first + " " + last;
const fiveTimes = (x) => x * 5;
const bigArrowFunc = (x, y, z) => {
  const firstPart = x + y;
  const secondPart = y * z;
  const thirdPart = x / z;
  const result = (firstPart + secondPart) * thirdPart;
  return result;
};

const numbers = [2, 4, 67, 54];
const min = Math.min(...numbers);
```