3/05/24

Module - 2 : JavaScript

Avantika Khanr

ASSIGNMENT - 2.1

Variables in Javascript :-

```
var a;          // variable
var b = "init"; // String
var c = 1+ 2+ "3"; // typecast to string "33".
var d = [2,3,5,8]; // array
var e = true // boolean
var f = function (){ }; // function object
const PI = 3.14; // constant
let z = 'zzz'; // block scope
```

** let is block-scoped and can be reassigned
const is also block-scoped but cannot be reassigned.

Data Types:- Number, String, Boolean, Object, Array, Null, undefined.

Operators :- Arithematic (+, -, *, /), Comparison (==, ===, !=, !==, >, <, >=, <=)
Logical (&&, ||, !)
BITWISE (&, |, ~, ^, <<, >>, >>> (zero fill)

Output in JS :-

```
console.log(a);   // write to the browser.
document.write(a); // write to HTML.
alert(a); // output in an alert.
confirm ("Hello") // "yes" or "no" dialog.
prompt ("Your age?","0"); // input dialog.
```

Control Flow :-

- **Conditional Statements :-** (if, else-if, else)

```
if (age > 18)
    console.log ("Eligible");
else if (age == 18)
    console.log ("Yes");
else
    console.log ("No");
```

- **Switch Statements :-** (switch, case, break)

```
switch (n) {
    case 1 :
        text = "Monday";
        break;
    case 2 :
        text = "Tuesday";
        break;
    default :
        text = "Sunday";
}
```

- **Loops :-**

① For Loop :-

```
for(var i=0;i<5;i++)
    console.log(i);
```

② While Loop

```
var i=0;
while (i <= 0)
{ console.log (i);
    i--;
}
```

③ Do-while loop :-

```
var i= 10;
do { i = i - 1;
    console.log (i);
} while ( i <= 0);
```

★ Functions in JS :-

- **Function Declaration :**

```
function funcName (parameters)
{ //code ;
}
```

- **Arrow Function :**

```
const funcName = (parameters) => {
    //code ;
}
```

- **Anonymous Function :**

```
const funcName =
    function (parameters) {
    //code;
}
```

- **Function Invocation :**

```
funcName (arguments)
```

Arrays :-

Creation :-

const arrayName = [ele1, ele2, ...]

Accessing Elements :-

arrayName[index]

Array Methods :

push()
pop()
shift()
unshift()
splice()
slice()
map()
forEach()
concat()
join()
indexOf()
find()
filter().

Objects :-

Creation :

const objectName = { key1: "value", key2: "val2" };

Accessing : objectName.key

Adding / Updating :

objectName.newkey = newValue

Remove :-

delete objectName.key

DOM Manipulation :-

Selecting Elements :-

document.getElementById(),
document.getElementByClassName(),
document.getElementByTagName(),

Changing Content :-

element.innerHTML,
element.textContent,
element.value.

** **For Styling :→** element.style.property = value.

Events :-

Event Handlers :-

element.onclick,
element.onmouseover,
element.onsubmit.

Event Listener :-

```
event.addEventListener('event', function () {
    //code;
})
```

Error Handling :-

try... catch... finally

```
try {
    //code that may give error
} catch (error) {
    //code to handle error
} finally {
    //code that always runs.
}
```

Asynchronous JS :-

- Callbacks
- Promises :

new Promise ((resolve, reject) => { //code }), promise.then(), promise.catch(),

- Async/Await :-

async function funcName () { //code }, await promise