**4.3 Screen Output and Keyboard Input**

* The JavaScript model for the HTML document is the Document object
* The model for the browser display window is the Window object
  + The Window object has two properties, document and window, which refer to the document and window objects, respectively
* The Document object has a method, write, which dynamically creates content
  + The parameter is a string, often catenated from parts, some of which are variables  
    e.g., document.write("Answer: " + result + "<br />");
* The Window object has three methods for creating dialog boxes, alert, confirm, and prompt  
  The default object for JavaScript is the Window object currently being displayed, so calls to these methods need not include an object reference.
  + alert("Hej! \n");
    - Parameter is plain text, not HTML
    - Opens a dialog box which displays the parameter string and an OK button
  + confirm("Do you want to continue?");
    - Opens a dialog box and displays the parameter and two buttons, OK and Cancel
  + prompt("What is your name?", "");
    - Opens a dialog box and displays its string parameter, along with a text box and two buttons, OK and Cancel
    - The second parameter is for a default response if the user presses OK without typing a response in the text box (waits for OK)

**PROGRAM:**

<!DOCTYPE html>  
<html>  
<body>  
  
<h1>My First Web Page</h1>  
<p>My First Paragraph</p>  
  
<p id="demo"></p>  
  
<script>  
document.getElementById("demo").innerHTML = 5 + 6;  
</script>  
  
</body>  
</html>

**Output:**

**My First Web Page**

My First Paragraph.

11

**PROGRAM:**

<!DOCTYPE html>

<html>

<body>

<h2>My First Web Page</h2>

<p>My first paragraph.</p>

<script>

window.alert(5 + 6);

</script>

</body>

</html>

**OUTPUT:**

**My First Web Page**

My first paragraph.

**PROGRAM:**

<!DOCTYPE html>

<html>

<body>

<h2>My First Web Page</h2>

<p>My first paragraph.</p>

<p>Never call document.write after the document has finished loading.

It will overwrite the whole document.</p>

<script>

document.write(5 + 6);

</script>

</body>

</html>

**OUTPUT:**

**My First Web Page**

My first paragraph.

Never call document.write after the document has finished loading. It will overwrite the whole document.

11

**4.4 Object Creation and Modification, Arrays, Functions**

Javascript Object Creation:

**PROGRAM:**

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Objects</h2>

<p id="demo"></p>

<script>

// Create an object:

const person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

// Display some data from the object:

document.getElementById("demo").innerHTML =

person.firstName + " is " + person.age + " years old.";

</script>

</body>

</html>

**OUTPUT:**

**JavaScript Objects**

John is 50 years old.

**PROGRAM:**

<html>

<body>

<h2>objects</h2>

<p id="demo"></p>

<p id="demo1"></p>

<p id="demo2"></p>

<script>

const p=new Object();

p.fname="john";

p.lname="doe";

p.age=19;

p.eyecolor="black";

document.getElementById("demo").innerHTML=p.fname+" is"+ p["age"]+" years old.";

document.getElementById("demo1").innerHTML=p.fname+ " "+p.lname;

document.getElementById("demo2").innerHTML=p.fname + " has"+ p.eyecolor+" eyes";

</script>

</body>

</html>

**OUTPUT:**

**objects**

john is19 years old.

john doe

john hasblack eyes

**PROGRAM:**

<html>

<body>

<h2>getting a property value</h2>

<p id="demo"></p>

<script>

const person={fname:"john",lname:"doe",age:50,eyecolor:"blue",language:"english",

get lang()

{

return this.language;

}

}

document.getElementById("demo").innerHTML=person.lang;

</script>

</body>

</html>

**OUTPUT**:

**getting a property value**

english

**PROGRAM**:

<html>

<body>

<h2>setting property </h2>

<p id="demo"></p>

<script>

const person={fname:"john",lname:"doe",age:50,eyecolor:"blue",language:"no",

set lang(value)

{

this.language=value;

}

}

person.lang="english";

document.getElementById("demo").innerHTML=person.language;

</script>

</body>

</html>

**OUTPUT:**

**setting property**

english

**PROGRAM:**

<html>

<body>

<h2>javascript prototype</h2>

<p id="demo"></p>

<p id="demo1"></p>

<script>

function person(first,last,age,eye)

{this.fname=first;

this.lname=last;

this.age=age;

this.eyecolor=eye;

}

person.prototype.nationality="english";

const p=new person("john","doe",50,"blue");

document.getElementById("demo1").innerHTML=p.nationality;

</script>

</body>

</html>

**OUTPUT**:

**javascript Prototype**

English

**Javascript Arrays:**

**Creating an Array**

Using an array literal is the easiest way to create a JavaScript Array.

**Syntax:**

const *array\_name* = [*item1*, *item2*, ...];

**PROGRAM:**

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Arrays</h2>

<p id="demo"></p>

<p id="demo1"></p>

<p id="demo2"></p>

<p id="demo3">we can create array with new array()</p>

<script>

const cars = ["Fiat", "Volvo", "BMW"];

document.getElementById("demo").innerHTML = cars;

document.getElementById("demo1").innerHTML = cars[0];

const fruits = ["Banana", "Orange", "Apple", "Mango"];

document.getElementById("demo2").innerHTML = fruits.length;

var points = new Array(40, 100, 1);

document.getElementById("demo3").innerHTML = points;

</script>

</body>

</html>

**OUTPUT:**

**JavaScript Arrays**

Fiat,Volvo,BMW

Fiat

4

40,100,1

**PROGRAM:**

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Array Methods</h2>

<p>The toString() method returns an array as a comma separated string:</p>

<p>The shift() method returns the element that was shifted out.</p>

<p>The unshift() method adds a new element to an array (at the beginning), and "unshifts" older elements:

The unshift() method returns the new array length.</p>

<p id="demo"></p>

<p id="demo1"></p>

<p id="demo2"></p>

<p id="demo3"></p>

<p id="demo4"></p>

<p id="demo5"></p>

<p id="demo6"></p>

<p id="demo7"></p>

<script>

const fruits = ["Banana", "Orange", "Apple", "Mango"];

document.getElementById("demo").innerHTML = fruits.toString();

document.getElementById("demo1").innerHTML = fruits.join(" \* ");

fruits.pop();

document.getElementById("demo3").innerHTML = fruits;

fruits.push("Kiwi");

document.getElementById("demo4").innerHTML = fruits;

document.getElementById("demo5").innerHTML = fruits.shift();

document.getElementById("demo6").innerHTML = fruits.unshift("Lemon");

document.getElementById("demo7").innerHTML = fruits;

</script>

</body>

</html>

**OUTPUT:**

**JavaScript Array Methods**

The toString() method returns an array as a comma separated string:

The shift() method returns the element that was shifted out.

The unshift() method adds a new element to an array (at the beginning), and "unshifts" older elements: The unshift() method returns the new array length.

Banana,Orange,Apple,Mango

Banana \* Orange \* Apple \* Mango

Banana,Orange,Apple

Banana,Orange,Apple,Kiwi

Banana

4

Lemon,Orange,Apple,Kiwi

**PROGRAM**:

!DOCTYPE html>

<html>

<body>

<h2>JavaScript Array Methods</h2>

<p>The concat() method merges (concatenates) arrays:</p>

<p>The splice() method adds new items to an array.

Example

const fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits.splice(2, 0, "Lemon", "Kiwi");

The first parameter (2) defines the position where new elements should be added (spliced in).

The second parameter (0) defines how many elements should be removed.

The rest of the parameters ("Lemon" , "Kiwi") define the new elements to be added.</p>

<p>The slice() method slices out a piece of an array into a new array.</p>

<p id="demo"></p>

<p id="demo1"></p>

<p id="demo2"></p>

<p id="demo3"></p>

<p id="demo4"></p>

<script>

const array1 = ["Cecilie", "Lone"];

const array2 = ["Emil", "Tobias", "Linus"];

const array3 = ["Robin", "Morgan"];

constmyChildren = array1.concat(array2, array3);

document.getElementById("demo").innerHTML = myChildren;

const fruits = ["Banana", "Orange", "Apple", "Mango"];

document.getElementById("demo1").innerHTML = "Original Array:<br> " + fruits;

let removed = fruits.splice(2, 2, "Lemon", "Kiwi");

document.getElementById("demo2").innerHTML = "New Array:<br>" + fruits;

document.getElementById("demo3").innerHTML = "Removed Items:<br> " + removed;

const citrus = fruits.slice(3);

document.getElementById("demo4").innerHTML = fruits + "<br><br>" + citrus;

</script>

</body>

</html>

**OUTPUT:**

**JAVASCRIPT ARRAY METHODS**

**The concat() method merges (concatenates) arrays:**

The splice() method adds new items to an array. Example const fruits = ["Banana", "Orange", "Apple", "Mango"]; fruits.splice(2, 0, "Lemon", "Kiwi"); The first parameter (2) defines the position where new elements should be added (spliced in). The second parameter (0) defines how many elements should be removed. The rest of the parameters ("Lemon" , "Kiwi") define the new elements to be added.

The slice() method slices out a piece of an array into a new array.

Cecilie,Lone,Emil,Tobias,Linus,Robin,Morgan

Original Array:  
Banana,Orange,Apple,Mango

New Array:  
Banana,Orange,Lemon,Kiwi

Removed Items:  
Apple,Mango

Banana,Orange,Lemon,Kiwi  
  
Kiwi

**Program:**

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Array Sort and reverse</h2>

<p id="demo1"></p>

<p id="demo2"></p>

<script>

const fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits.sort();

document.getElementById("demo1").innerHTML = fruits;

fruits.reverse();

document.getElementById("demo2").innerHTML = fruits;

</script>

</body>

</html>

**Output:**

**JavaScript Array Sort and reverse**

Apple,Banana,Mango,Orange

Orange,Mango,Banana,Apple