

Chapter 11 - JavaScript: Arrays

Outline

- 11.1 Introduction
- 11.2 Arrays
- 11.3 Declaring and Allocating Arrays
- 11.4 Examples Using Arrays
- 11.5 References and Reference Parameters
- 11.6 Passing Arrays to Functions
- 11.7 Sorting Arrays
- 11.8 Searching Arrays: Linear Search and Binary Search
- 11.9 Multiple-Subscripted Arrays
- 11.10 JavaScript Internet and World Wide Web Resources



11.2 Arrays

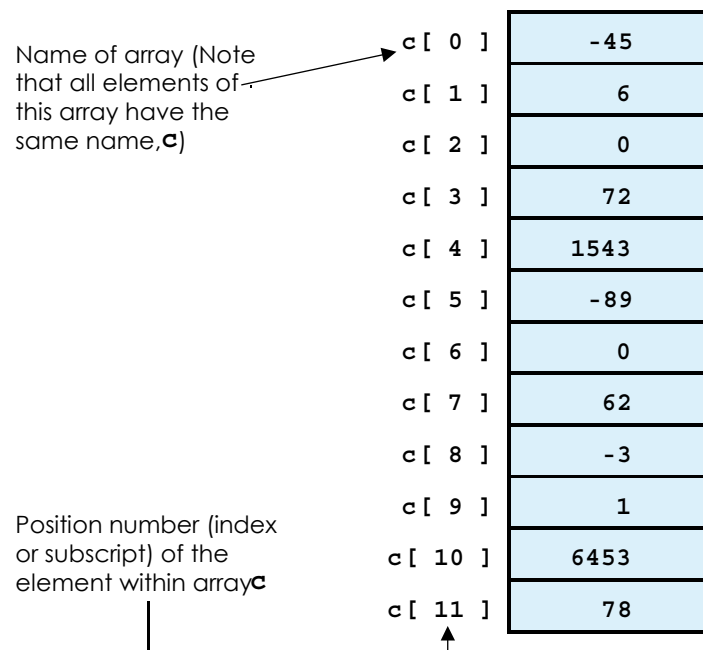


Fig. 11.1 A 12-element array.



11.3 Declaring and Allocating Arrays

Operators	Associativity	Type
() [] .	left to right	highest
++ -- !	right to left	unary
* / %	left to right	multiplicative
+ -	left to right	additive
< <= > >=	left to right	relational
== !=	left to right	equality
&&	left to right	logical AND
	left to right	logical OR
? :	right to left	conditional
= += -= *= /= %=	right to left	assignment

Fig. 11.2 Precedence and associativity of the operators discussed so far.



Outline

InitArray.html

```

1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 11.3: InitArray.html -->
6  <!-- Initializing an Array -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9    <head>
10     <title>Initializing an Array</title>
11
12     <script type = "text/javascript">
13       <!--
14       // this function is called when
15       // onload event occurs
16       function initializeArrays()
17       {
18         var n1 = new Array( 5 );
19         var n2 = new Array();
20
21         // assign values to elements of array n1
22         for ( var i = 0; i < n1.length; ++i )
23           n1[ i ] = i;
24
25         // create and initialize array n2
26         for ( i = 0; i < 5; ++i )
27           n2[ i ] = i;
28
29         outputArray( "Array n1 contains", n1 );
30         outputArray( "Array n2 contains", n2 );
31       }
32     </script>

```

Array n1 has five elements.

Array n2 is an empty array.

The **for** loop initializes the elements in **n1** to their subscript numbers (0 to 4).

The **for** loop adds five elements to **Array n2** and initialize each element to its subscript number (0 to 4).

Each function displays the contents of its respective Array in an XHTML table.



```

33 // output "header" followed by a two-column table
34 // containing subscripts and elements of "theArray"
35 function outputArray( header, theArray )
36 {
37     document.writeln( "<h2>" + header + "</h2>" );
38     document.writeln( "<table border = \"1\" width = \"100%\">" );
39     document.writeln( "<tr>" );
40     document.writeln( "<td>" );
41     document.writeln( "<td>" );
42     document.writeln( "</tr>" );
43     document.writeln( "<tr>" );
44     document.writeln( "<td>" );
45     document.writeln( "<td>" );
46     document.writeln( "</tr>" );
47     document.writeln( "</tbody></table>" );
48 }
49 // -->
50 </script>
51
52 </head><body onload = "initializeArrays()"></body>
53
54 </html>

```

The second time function `outputArray` is called, variable `header` gets the value of "Array n2 contains" and variable `theArray` gets the value of `n2`.



Array n1 contains

Subscript	Value
0	0
1	1
2	2
3	3
4	4

Array n2 contains

Subscript	Value
0	0
1	1
2	2
3	3
4	4



```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 11.4: InitArray2.html -->
6  <!-- Initializing an Array with a Declaration -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9    <head>
10     <title>Initializing an
11
12     <script type = "text/javascript">
13       <!--
14       function start()
15       {
16         // Initializer list specifies number of elements and
17         // value for each element.
18         var colors = new Array( "cyan", "magenta",
19           "yellow", "black" );
20         var integers1 = [ 2, 4, 6, 8 ];
21         var integers2 = [ 2, , , 8 ];
22
23         outputArray( "Array colors contains", colors );
24         outputArray( "Array integers1 contains", integers1 );
25         outputArray( "Array integers2 contains", integers2 );
26       }
27
```

Array integers1 is initialized using an initializer list.

Two values are not supplied for **integer2**, which will be displayed as **undefined**.



```
28     // output "header" followed by a two-column table
29     // containing subscripts and elements of "theArray"
30     function outputArray( header, theArray )
31     {
32       document.writeln( "<h2>" + header + "</h2>" );
33       document.writeln( "<table border = \"1\" \" +
34         \"width = \"100%\">" );
35       document.writeln( "<thead><th width = \"100\" \" +
36         \"align = \"left\">Subscript</th>\" +
37         "<th align = \"left\">Value</th></thead><tbody>" );
38
39       for ( var i = 0; i < theArray.length; i++ )
40         document.writeln( "<tr><td>" + i + "</td><td>" +
41           theArray[ i ] + "</td></tr>" );
42
43       document.writeln( "</tbody></table>" );
44     }
45     // -->
46   </script>
47
48   </head><body onload = "start()"></body>
49 </html>
```



Program Output

C:\jw3http2\ch11examples\InitArray2.html - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search History

Address C:\jw3http2\ch11examples\InitArray2.html Go

Array colors contains

Subscript	Value
0	cyan
1	magenta
2	yellow
3	black

Array integers1 contains

Subscript	Value
0	2
1	4
2	6
3	8

Array integers2 contains

Subscript	Value
0	2
1	undefined
2	undefined
3	8

My Computer

© 2001 Prentice Hall, Inc.
All rights reserved.



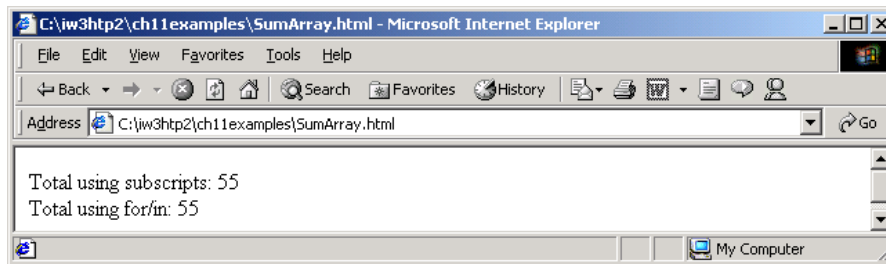
SumArray.html

```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 11.5: SumArray.html      -->
6  <!-- Summing Elements of an Array -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9    <head>
10     <title>Sum the Elements of an Array</title>
11
12     <script type = "text/javascript">
13       <!--
14       function start()
15       {
16         var theArray = [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ];
17         var total1 = 0, total2 = 0;
18
19         for ( var i = 0; i < theArray.length; i++ )
20           total1 += theArray[ i ];
21
22         document.writeln( "Total using for loop: " + total1 );
23
24         for ( var element in theArray )
25           total2 += theArray[ element ];
26
27         document.writeln( "<br />Total using for/in: " + total2 );
28       }
29     // -->
30   </script>
31
32   </head><body onload = "start()"></body>
33
34 </html>
```

The **for** loop sums the values contained in the 10-element integer array called **theArray**.

Variable **element** is assigned a subscript in the range of 0 up to, but not including, **theArray.length**.

© 2001 Prentice Hall, Inc.
All rights reserved.



Program Output



```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 11.6: RollDie.html      -->
6  <!-- Roll a Six-Sided Die 6000 Times -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9    <head>
10     <title>Roll a Six-Sided Die 6000 Times</title>
11
12     <script type = "text/javascript">
13       <!--
14       var face, frequency = [ , 0, 0, 0, 0, 0 ];
15
16       // summarize results
17       for ( var roll = 1; roll <= 6000; ++roll ) {
18         face = Math.floor( 1 + Math.random() * 6 );
19         ++frequency[ face ];
20       }
21
22       document.writeln( "<table border = \"1\" +
23         \"width = \"100%\"> );
24       document.writeln( "<thead><th width = \"100\" +
25         \" align = \"left\">Face<th align = \"left\"> +
26         \"Frequency</th></thead></tbody>\" );
27
28       for ( face = 1; face < frequency.length; ++face )
29         document.writeln( "<tr><td>\" + face + "</td><td>\" +
30           frequency[ face ] + "</td></tr>\" );
31
32       document.writeln( "</tbody></table>\" );
33       <!--
34     </script>
35
```

Referencing **Array frequency** replaces the **switch** statement used in Chapter 10's example.

RollDie.html

```

36     </head>
37     <body>
38         <p>Click Refresh (or Reload) to run the script again</p>
39     </body>
40 </html>

```



Outline

13

RollDie.html

Roll a Six-Sided Die 6000 Times - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Reload Home Search Favorites History Print View Source

Address C:\jw3http2\ch11examples\RollDie.html Go

Face	Frequency
1	1023
2	952
3	1028
4	994
5	991
6	1012

Click Refresh (or Reload) to run the script again

Done My Computer

Program Output

© 2001 Prentice Hall, Inc.
All rights reserved.

```

1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 11.7: PassArray.html -->
6  <!-- Passing Arrays -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9    <head>
10      <title>Passing Arrays and Individual Array
11        Elements to Functions</title>
12
13      <script type = "text/javascript">
14        <!--
15        function start()
16        {
17          var a = [ 1, 2, 3, 4, 5 ];
18
19          document.writeln( "<h2>Function modifyArray multiplies each element by 2.
20            "array call-by-reference</h2>" );
21
22          // The value of a [ 3 ] is output to show its
23          // contents before it is modified.
24          modifyArray( a );
25
26          // Function outputArray is called to show
27          // The values of the modified array are: ", a );
28          outputArray(
29            "The values of the modified array are: ", a );
30
31          document.writeln( "<h2>Effects of passing array " +
32            "element call-by-value</h2>" +
33            "a[3] before modifyElement: " + a[ 3 ] );
34          modifyElement( a[ 3 ] );

```



Outline

14

PassArray.html

© 2001 Prentice Hall, Inc.
All rights reserved.



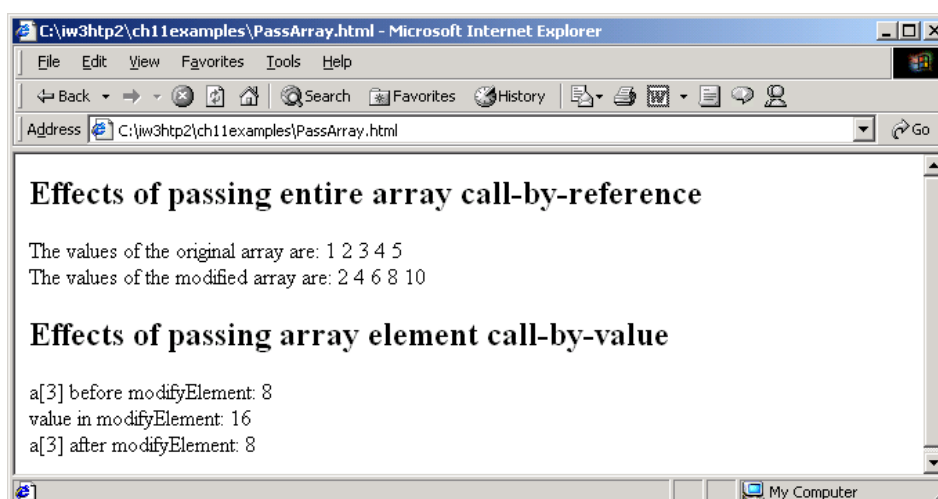
```

35     document.writeln(
36         "<br />a[3] after modifyElement: " + a[ 3 ] );
37     }
38
39     // outputs "header" followed by the contents of "theArray"
40     function outputArray( header, theArray )
41     {
42         document.writeln(
43             header + theArray.join( " " ) + "<br />" );
44     }
45
46     // function that modifies the elements of the array
47     function modifyArray( theArray )
48     {
49         for ( var j in theArray )
50             theArray[ j ] *= 2;
51     }
52
53     // function that attempts to modify the value passed
54     function modifyElement( e )
55     {
56         e *= 2;
57         document.writeln( "<br />value in modifyElement: " + e );
58     }
59     // -->
60 </script>
61
62 </head><body onload = "start()"></body>
63 </html>

```

Method **join** takes as its argument a string containing a separator that should be used to separate the elements of the array in the string that is returned.

Multiply each element in **theArray** by 2.





```

1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 11.8: sort.html -->
6  <!-- Sorting an Array -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9    <head>
10     <title>Sorting an Array</title>
11
12     <script type = "text/javascript">
13       <!--
14       function start()
15       {
16         var a = [ 10, 1, 9, 2, 8, 3, 7, 4, 6, 5 ];
17
18         document.writeln( "<h1>Sorting an Array</h1>" );
19         outputArray( "Data items in original order: ", a );
20         a.sort( compareIntegers ); // sort the array
21         outputArray( "Data items in ascending order: ", a );
22       }
23
24       // outputs "header" followed by the array
25       function outputArray( header, theArray )
26       {
27         document.writeln( "<p>" + header +
28                           theArray.join( " " ) + "</p>" );
29       }
30

```

Method `sort` takes as its optional argument the name of a function that compares two arguments and returns a value of `-1`, `0` or `1`.

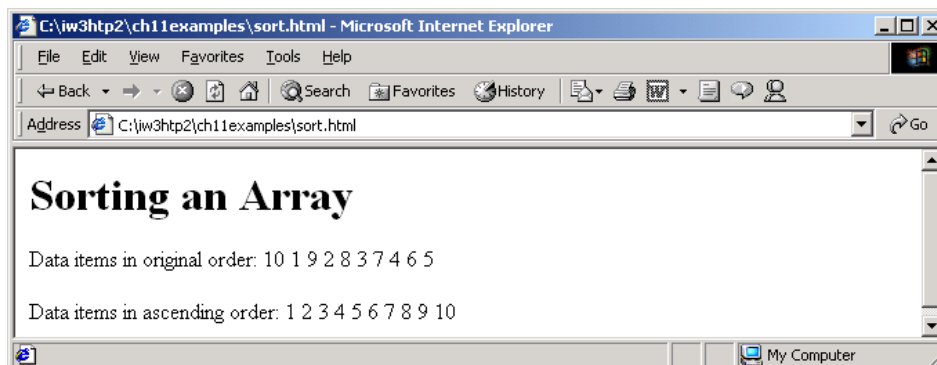
Function `compareIntegers` calculates the difference between the integer values of its arguments.



```

31     // comparison function for use with sort
32     function compareIntegers( value1, value2 )
33     {
34       return parseInt( value1 ) - parseInt( value2 );
35     }
36     // -->
37   </script>
38
39   </head><body onload = "start()"></body>
40 </html>

```



Program Output



```

1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 11.9: LinearSearch.html -->
6  <!-- Linear Search of an Array -->
7
8  <html xmlns = "http://www.w3.org
9    <head>
10     <title>Linear Search of an Array</title>
11
12     <script type = "text/javascript"
13       <!--
14         var a = new Array( 100 ); // create an Array
15
16         // fill Array with even integer values from 0 to 198
17         for ( var i = 0; i < a.length; ++i )
18           a[ i ] = 2 * i;
19
20         // function called when "Search" button is pressed
21         function buttonPressed()
22         {
23           var searchKey = searchForm.inputVal.value;
24
25           // Array a is passed to linearSearch even though it
26           // is a global variable. Normally
27           // be passed to a method for search
28           var element = linearSearch( a, par
29
30           if ( element != -1 )
31             searchForm.result.value
32             "Found value in element"
33           else
34             searchForm.result.value = "value not found";
35         }

```

Array a is initiated with 100 elements.

Array a is populated with the integers 0 to 198.

Get value of search key from the input field in the XHTML form.

Calling function **linearSearch** and passing it the **Array a** and the value of variable **searchKey** as an integer.

© 2001 Prentice Hall, Inc.
All rights reserved.



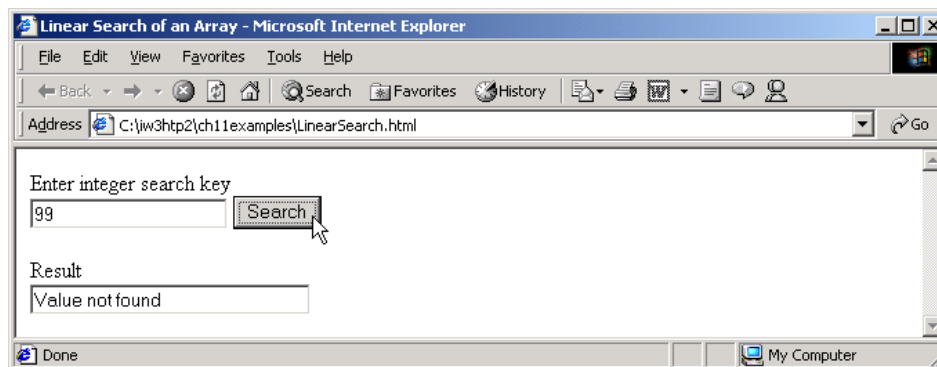
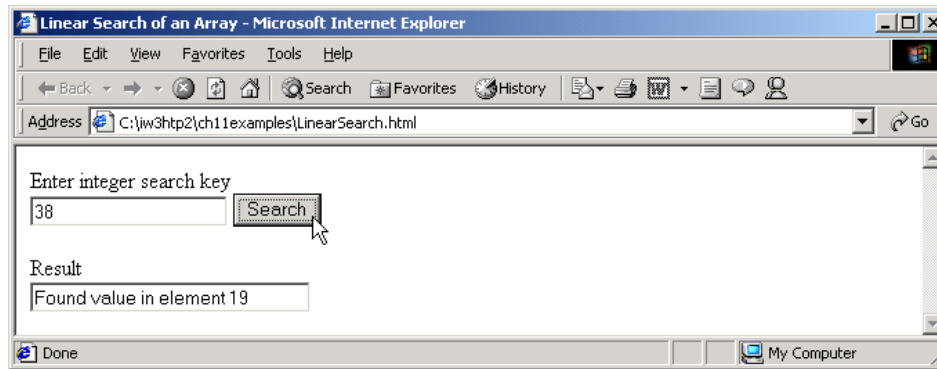
```

36
37     // Search "theArray" for the specified "key" value
38     function linearSearch( theArray, key )
39     {
40       for ( var n = 0; n < theArray.length; ++n )
41         if ( theArray[ n ] == key )
42           return n;
43
44       return -1;
45     }
46     // -->
47   </script>
48
49 </head>
50
51 <body>
52   <form name = "searchForm" action = "">
53     <p>Enter integer search key<br />
54     <input name = "inputVal" type = "text" />
55     <input name = "search" type = "button" value = "Search"
56       onclick = "buttonPressed()" /><br /></p>
57
58     <p>Result<br />
59     <input name = "result" type = "text" size = "30" /></p>
60   </form>
61 </body>
62 </html>

```

Variable **theArray** gets the value of **Array a** and variable **key** gets the value of variable **searchKey**.

Function **linearSearch** compares each element of **theArray** with the search key.



BinarySearch.htm
1

```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5  <!-- Fig. 11.10 : BinarySearch.html -->
6  <!-- binary search -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9    <head>
10     <title>Binary Search</title>
11
12     <script type = "text/javascript">
13       <!--
14       var a = new Array( 15 );
15
16       for ( var i = 0; i < a.length; ++i )
17         a[ i ] = 2 * i;
18
19       // function called when "Search" button is pressed
20       function buttonPressed()
21       {
22         var searchKey = searchForm.inputVal.value;
23
24         searchForm.result.value =
25           "Portions of array searched";
26
27         // Array a is passed to binarySearch
28         // is a global variable. This is done because
29         // normally an array is passed to a method
30         // for searching.
31         var element =
32           binarySearch( a, parseInt( searchKey ) );
33
```

Array a is initiated with 15 elements.

Function **binarySearch** receives two arguments: the **Array a** and the search key, **searchKey**.



```

34     if ( element != -1 )
35         searchForm.result.value +=
36             "\nFound value in element " + element;
37     else
38         searchForm.result.value += "\nValue not found";
39 }
40
41 // Binary search
42 function binarySearch( theArray, key )
43 {
44     var low = 0;                // low subscript
45     var high = theArray.length - 1; // high subscript
46     var middle;                // middle subscript
47
48     while ( low <= high ) {
49         middle = ( low + high ) / 2;
50
51         // The following line is the key to the binary search.
52         // part of theArray current element is compared to key
53         // during each iteration of the search loop.
54         buildOutput( theArray, low, middle, high );
55
56         if ( key == theArray[ middle ] ) // match
57             return middle;
58         else if ( key < theArray[ middle ] )
59             high = middle - 1;
60         else
61             low = middle + 1; // search high end of array
62     }
63
64     return -1; // searchKey not found
65 }
66
67

```

If the **key** matches the **middle** element of a subarray, the subscript of the current element is

If **key** is less than the **middle** element, the high subscript is set to **middle - 1**.

If **key** is greater than the **middle** elements, the high subscript is set to **middle + 1**.



```

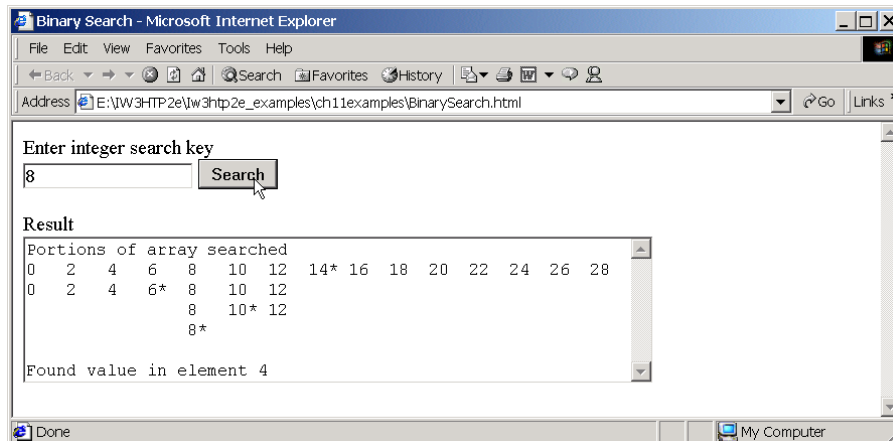
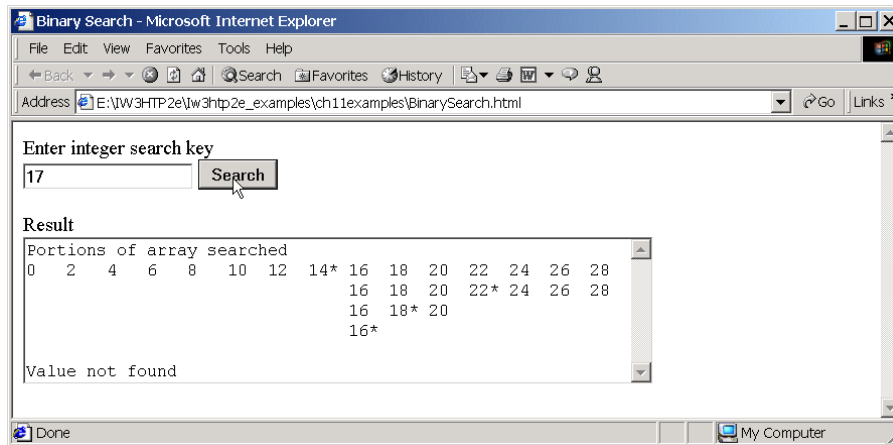
68 // Build one row of output showing the current
69 // part of the array being processed.
70 function buildOutput( theArray, low, mid, high )
71 {
72     for ( var i = 0; i < theArray.length; i++ ) {
73         if ( i < low )
74             searchForm.result.value += " ";
75         // mark mid
76         else if ( i == mid )
77             searchForm.result.value += a[ i ] +
78                 ( theArray[ i ] < 10 ? " " : " * " );
79         else
80             searchForm.result.value += a[ i ] +
81                 ( theArray[ i ] < 10 ? " " : " " );
82     }
83
84     searchForm.result.value += "\n";
85 }
86 // -->
87 </script>
88 </head>
89
90 <body>
91     <form name = "searchForm" action = "">
92         <p>Enter integer search key<br />
93         <input name = "inputVal" type = "text" />
94         <input name = "search" type = "button" value =
95             "Search" onclick = "buttonPressed()" /><br />
96         <p>Result<br />
97         <textarea name = "result" rows = "7" cols = "60">
98         </textarea></p>
99     </form>
100 </body>
101 </html>

```

Function **buildOutput** creates the markup that displays the results of the search.



Program Output



© 2001 Prentice Hall, Inc.
All rights reserved.

11.9 Multiple-Subscripted Arrays

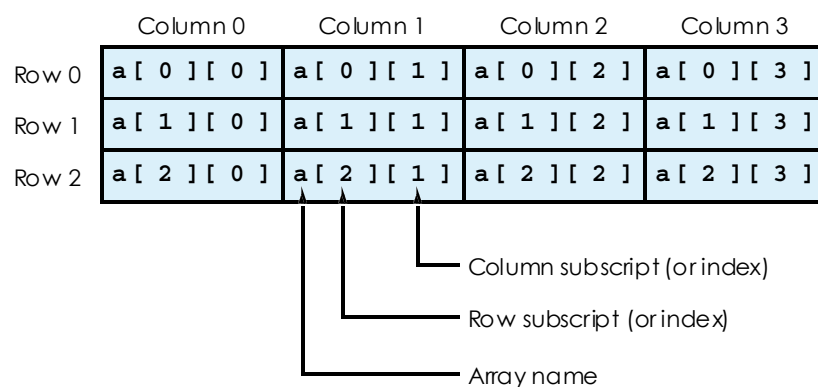


Fig. 11.11 Double-subscripted array with three rows and four columns.



```

1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 11.12: InitArray3.html
6  <!-- Initializing Multidimensional Arrays
7
8  <html xmlns = "http://www.w3.org/1999/xhtml"
9    <head>
10     <title>Initializing Multidimensional Arrays
11
12     <script type = "text/javascript">
13       <!--
14       function start()
15       {
16         var array1 = [ [ 1, 2, 3 ], // first row
17                       [ 4, 5, 6 ] ]; // second row
18         var array2 = [ [ 1, 2 ], // first row
19                       [ 3 ], // second row
20                       [ 4 ] ];
21
22         outputArray( "Values in array1 by row", array1 );
23         outputArray( "Values in array2 by row", array2 );
24       }
25
26       function outputArray( header, theArray )
27       {
28         document.writeln( "<h2>" + header + "</h2><tt>" );
29

```

Array array1 provides six initializers in two sublists.

Array array2 provides six initializers in three sublists.

Function outputArray displays each array's elements in a Web page.

© 2001 Prentice Hall, Inc.
All rights reserved.

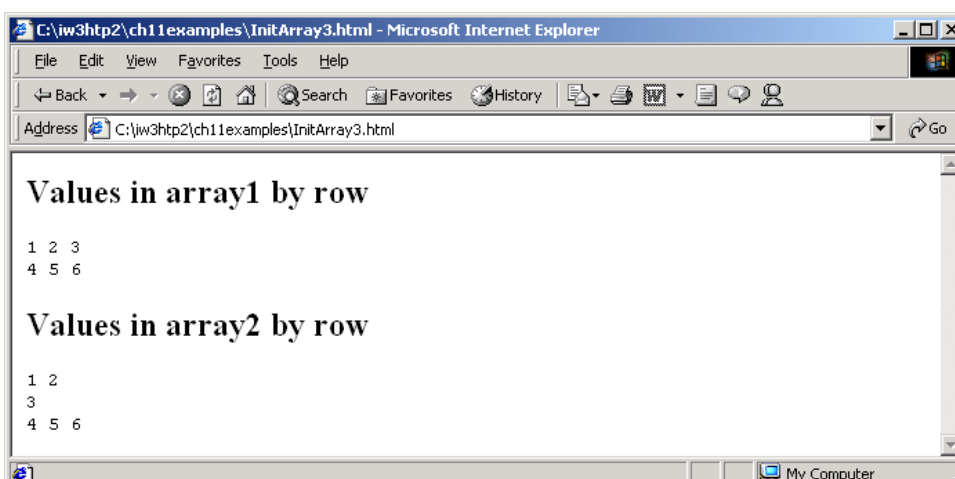
```

30     for ( var i in theArray ) {
31
32         for ( var j in theArray[ i ] )
33             document.write( theArray[ i ][ j ] + " " );
34
35         document.writeln( "<br />" );
36     }
37
38     document.writeln( "</tt>" );
39 }
40 // -->
41 </script>
42
43 </head><body onload = "start()"></body>
44 </html>

```

Referencing the multidimensional array **theArray**.

InitArray3.html



Program Output

© 2001 Prentice Hall, Inc.
All rights reserved.