

PROGRAMMING IN A SCRIPTING LANGUAGE

OBJECT-BASED PROGRAMMING IN JAVASCRIPT

20. Objects

In small programs you can easily manage a few functions and variables in the global space. However when you have thousands of lines with hundreds of pages, data items etc., trying to keep them all in mind is almost impossible.

You can divide this large program into **objects**. Objects exist as a way of organizing **properties** and **methods** into logical groups. An object is a grouping of properties and methods that stay internal to that object and are not “visible” outside that object. For example, in real life, look at two objects, people and trees:

- i) people have properties such as eyes, ears, legs, etc, and methods (action) such as eating, talking, walking, etc.
- ii) trees have properties such as branches, leaves, etc., and methods (action) such as growing, photosynthesis, etc.

21. Object Properties

The **Properties** of an object are its variables. Just as you can assign a value to a variable, you can also assign a value to a property. We have covered the bgColor property as follows:

```
document.bgColor="red" // this will change the background colour of the page to red;
document.bgColor="blue" // now the background colour of the page has been changed to blue;
```

22. Object Methods

Methods are built-in functions of an object. Example of methods so far covered are:

write()	used with it's object	document.write()		
alert()	used with it's object	window.alert()	or	alert()
prompt()	used with it's object	window.prompt()	or	prompt()

23. Predefined Core Objects

JavaScript has built-in objects such as String, Math, Date, Array, etc. These objects come with their methods and properties. We shall look at some of them.

24. The String Object

The **String** object in JavaScript has many **methods and properties** built-in as shown below:

<i>anchor</i>	<i>eval</i>	<i>italics</i>	<i>sub</i>	<i>toString</i>
<i>big</i>	<i>fixed</i>	<i>lastIndexOf</i>	<i>substring</i>	<i>valueOf</i>
<i>blink</i>	<i>fontcolor</i>	<i>small</i>	<i>sup</i>	
<i>bold</i>	<i>fontsize</i>	<i>split</i>	<i>toLowerCase</i>	
<i>charAt</i>	<i>indexOf</i>	<i>strike</i>	<i>toUpperCase</i>	

It has the *length* property which gives the number of characters the string contains.

25. The 'new' Operator

Usually when you declare a new object, the **new** operator is used.

```
var string1 = new String();
string1 = "Hello from Hornsby TAFE";
or
var string1=new String("Hello from Hornsby TAFE");
```

Note - the above code has created a variable called string1 and declared it a String object (now belongs to the String family). The variable, string1, is storing "Hello from Hornsby TAFE".

Exercise 7

1. We will create a new Web page that will use JavaScript to make this string of text: "This text should appear in upper case"; appear in upper case using the String object's **toUpperCase()** method, and also use the **length** property to calculate the number of characters it has. The **ex7-1.js** file will contain the following code:

```
var string1 = new String();
string1 = "This text should appear in upper case";

document.write("Use upper case method: " + string1.toUpperCase() + "<br>");
document.write("Use length property: " + string1.length + "<br>");
document.write("This string:" + string1.toUpperCase()+"has"+string1.length+"characters");
```

2. Now create a web page similar to the one above, but use the **toLowerCase()** method.
3. Create a new Web page, and use the **charAt()** method to return a character at a specified location (index number) in the text string, starting the count from 0. Use the text "john.smith@tafensw.edu.au", and enter the following JavaScript statements in file **ex7-3.js**

```
var stringX = new String("john.smith@tafensw.edu.au");
document.write(stringX.charAt(6)); //this will display the 7th character
```

Try other index numbers

4. Continue with the same Web page, and use the **indexOf()** method to return the index (or position) of a string character at the first occurrence, starting count from 0. Do the following statements in file **ex7-3.js**

```
var stringX = new String("john.smith@tafensw.edu.au");

document.write(stringX.indexOf("john") + "<br>");
document.write(stringX.indexOf(".") + "<br>");
document.write(stringX.indexOf("@") + "<br>");
document.write(stringX.indexOf("tafe") + "<br>");
document.write(stringX.indexOf("n") + "<br>");
document.write(stringX.indexOf("z") + "<br>"); //what value do you get?
```

The above exercise is a very important one. Users fill in forms over the net continuously. JavaScript can check and validate the data (strings) the user has entered through forms. For example, JavaScript can check if a user has entered "@" and "." characters for their email. If not, then JavaScript can give an error message and not allow the form to be submitted to a server.

5. Continue with the same Web page, and use the **lastIndexOf()** method to return the index (or position) of a string object at the last occurrence, starting count from 0. Do the following statements:
stringX.lastIndexOf("n");
stringX.lastIndexOf("a");
6. Continue with the same Web page, and use the **substring()** method to return characters in a string between two indexes. Do the following:
a) stringX.substring(0,4);
b) stringX.substring(5,10);
c) return the substring "tafensw";
d) return the substring "edu.au";

26. The Math Object

The **Math** object provides ample facilities for mathematical functions. This object, like other objects, has properties and methods. We will be looking at the methods. It is a handy object especially for businesses and banks that deal with numerical values over the internet. The **Math** method that produces random numbers can be handy for displaying random graphics on a web page.

Some of the Math methods are listed below:

Method	Description	Example
ceil	Returns an integer value rounded up	X = Math.ceil(2.001) X becomes 3
floor	Returns an integer value rounded down	X = Math.floor(2.99) X becomes 2
max	Returns the greater of two numbers	X = Math.max(4, 5.7) X becomes 5.7
min	Returns the lesser of two numbers	X = Math.min(4, 5.7) X becomes 4
pow	Returns a power value	X = Math.pow(a,b) X becomes a ^b X = Math.pow(10, 2) X becomes 100
random	Returns a random number between 0 and 1	X = Math.random() X becomes say 0.02477046077
round	Returns the value of a number rounded to nearest integer	X = Math.round(3.5) X becomes 4 X = Math.round(3.4) X becomes 3
sqrt	Returns the square root of a number	X = Math.sqrt(16) X becomes 4

Exercise 8

1. Create a Web page that will display the larger value of the pair 4.432 and 4.567. Do the following code in file **ex8-1.js**:

```
var largerValue= Math.max(4.432, 4.567);  
document.write("The larger value between 4.432 and 4.567 is " + largerValue);
```

2. Create a Web page that will display the rounded value of each of these two numbers: 4.432 and 4.567.
3. Create a Web page that will return the integer value of the following division 22/5.

toFixed()

One of the most common tasks is to format a number for currency display- an integer followed by two decimals. You can use number rounding to first shift the number's decimal places (via multiplication), round it, then shift the decimal back (via division). That is

```
round(number*100)/100
```

However this will not display the trailing or end '0's. So, an answer may look like 4.5 or 7 (and not 4.50 or 7.00)

To easily format numbers for a specific number of trailing decimals, JavaScript 1.5 introduced the toFixed() method. For example:

```
var profits=2489.8237  
profits.toFixed(3) //returns 2489.824 (round up)  
profits.toFixed(2) //returns 2489.82  
profits.toFixed(7) //returns 2489.8237000 (padding)
```

Exercise 8 cont.

4. Round the value, 12.9578, to two decimal places, and display it.
5. Generate one random number between 0 and 10, and display it.
6. The following Web page will display a different graphic randomly when ever the web page is refreshed. Place the following code in file **ex8-6.html**:

```
<body>  
    
  <script src="ex8-6.js"></script>  
</body>
```

Enter the following code in file **ex8-6.js**:

```
var randomNumber;

//get a random number 1 to 5;
randomNumber = Math.round(Math.random()*5);

//based on the random number select an image;
switch(randomNumber)
{
    case 1:
        document.bodyImage.src = "happy.jpg";
        break;
    case 2:
        document.bodyImage.src = "sad.jpg";
        break;

    case 3:
        document.bodyImage.src = "angry.jpg";
        break;

    case 4:
        document.bodyImage.src = "Cake.jpg";
        break;
    case 5:
        document.bodyImage.src = "dog_sleeping.bmp";
        break;
    default:
        alert("not one of the numbers!!");
}
```

Note- The reason `<script>` tags were placed in the body area after the `` tag, is so the browser creates the image first then it will read the JavaScript that refers to this image, ie., the image has to exist before JavaScript refers to it, otherwise an error message will occur that states the image object does not exist.

27. The Date Object

The **date** object is used to work with dates, using the internal clock of your PC. Internally, the value of a date object's instance is the time, in milliseconds, from zero o'clock on January 1, 1970, in Greenwich mean time zone.

the following code: `var today = new Date();`
will take a snapshot of the PC's internal clock and place this date value in the variable "today".

Exercise 9

1. Create a Web page that will display the current date and time. Use the following code for file **ex9-1.js**:

```
var currentDate = new Date();
document.write("Now is " + currentDate);
```

28. The get methods

You can use the get methods as follows (assuming you are using the 'currentDate' variable from above):

<code>currentDate.getDate()</code>	gets the day of the month
<code>currentDate.getMonth()</code>	gets the month (0 – 11)
<code>currentDate.getYear()</code>	returns the year in ? digits (you test IE, Netscape, Firefox)
<code>currentDate.getFullYear()</code>	returns the year in 4 digits
<code>currentDate.getDay()</code>	returns day of week starting count from Sunday as "0"
<code>currentDate.getHours()</code>	returns the number of hours (0 – 23)
<code>currentDate.getMinutes()</code>	gets the minutes (0 – 59)
<code>currentDate.getSeconds()</code>	gets the seconds (0 – 59)
<code>currentDate.getTime()</code>	gets the time (and date) as the number of milliseconds since 1 st Jan 1970

Exercise 9 cont.

2. Create a new Web page that will take the current date and time (of the internal PC clock), and display this on the page. It will also display each of the following on separate lines: the year, month, day, hours, minutes and seconds.

29. The Array Object

Arrays are used when you wish to group many quantities under the one variable name, for instance a group of names. Each data is placed in distinct slots and these slots are numbered so that each piece of data has a unique address. In so doing, you automatically have indexed data (data with *slot* numbers). The first index number is 0.

For example, if an array called `num[]` was to contain five numbers, you could visualize it as follows:

		num[]				
index no. or slot no. →		0	1	2	3	4
data →		1	4	9	16	25

To create an Array variable you use the following syntax:

```
var userNames = new Array()  
// browser creates a variable called userNames which is a multi-storied variable
```

or you could use a shortcut which instigates the above code line, ie., creates an Array object

```
var userNames = []; // [] is an array literal, representing an empty array
```

Exercise 10

1. In this exercise you will store the strings “Hello”, “from”, “Australia!” in an array called `message[]`. You will then use a loop to display each element of the array so it displays “Hello from Australia!” in one line. Lets look at different ways to do this:

- 1a. Enter the following code in file **ex10-1a.js**

```
//create variables first  
var i;  
var message = new Array();  
  
//assign values to the Array variable  
message[0] = "Hello";  
message[1] = " from ";  
message[2] = "Australia!";  
  
//display values  
for (i=0; i<3; i++)  
{  
    document.write(message[i]);  
}
```

- 1b. Enter the following code in file **ex10-1b.js**

Note – you can create an array and fill in values in one line as in ex10-1b.js

```
var message = new Array("Hello", " from ", "Australia!");
```

```
//use the 'for' loop as above to display the values
```

- 1c. Enter the following code in file **ex10-1c.js**

Note – you can the use the array literal as in ex10-1c.js

```
var message = [];  
message = ["Hello", " from ", "Australia!"];
```

```
//use the 'for' loop as above to display the values
```

- 1d. Enter the following code in file **ex10-1d.js**

Note – this code can be further reduced as in ex10-1d.js

```
var message = ["Hello", " from ", "Australia!"];
```

```
//use the 'for' loop as above to display the values
```

30. sort() Method

The **sort()** method will sort the elements in an array. Note the results of the following two exercises.

Exercise 10 cont.

2. Create a new Web page and use an array called `firstName[]` which will store four names. Your code is to sort these names before they are displayed on the screen.
Hint - use the `sort()` method as follows: `firstName.sort()` just before the 'for' loop
3. Modify the above exercise so that the user can enter four names. Copy the following code for file **ex10-3.js**

```
//create variables first
var i, j;
var firstName[];

/*let user enter values into the Array variable through the prompt window.
Use a loop that will repeat the prompt window 4 times*/
for (i=0; i<4; i++)
{
    firstName[i] = prompt("Please enter a name");
}

//sort the elements in the Array
firstName.sort();

//display the sorted names
for (j=0; j<4; j++)
{
    document.write(firstName[j] + "<br>");
}
```

4. Create a new Web page and use an array called `digits[]` which will store seven numbers. The code is to sort these numbers and then display them.

What was the result? An ASCII sort!

5. In this exercise you will store seven numbers in an array, then sort them numerically. Note – JavaScript sorts numbers by using differences between the numbers (a little convoluted – but there it is! You have the code if ever needed). Write the following code in file **ex10-5.js**

```
//a function that subtracts 2 numbers
function differences(a,b)
{
    return (a-b);
}

var digits = [7, 53, 100, 26, 93, 242, 19];

//sort the elements in the Array
digits.sort(differences);

//display the elements
for (i=0; i<7; i++)
{
    document.write(digits[i] + "<br>")
}
```

6. Create an array called `marks[]` with any number of marks in it, but make sure the last mark is 999. 999 acts as a trigger for the end. The program is to display each number. However, when it reaches the number 999 it is to stop the display loop and not display that number but rather the message "This is the end".
7. Modify Q6 above so that the user enters the marks. Note, inform the user to enter '999' when they wish to stop.

31. length Property

The **length** property will give the number of elements in an array, hence the length of the array. Sometimes through the process of the program you may change the number of elements in an array. There is no problem in creating an array, without specifying the number of elements. For example

```
var firstName = new Array() // this will allow any number of elements in the array
```

However there is a problem when you wish to display the elements using a loop. The loop needs to adjust itself, if array elements are going to be added or taken off. You can solve this problem by using the **length** property. Try the following examples.

Exercise 10 cont.

8. Create a simple Web Page, and enter a handful of suburbs. The following code will display the number of suburbs which is the length of the array

```
document.write("there are " + arrayName.length + " elements in this array")
```

9. Modify exercise ex10-1c.js, so that the array length is used to stop the 'for' loop, as follows

```
var message = [];  
message = ["Hello", " from ", "Australia!"];  
  
//display values  
for (var i=0; i < message.length; i++)  
{  
    document.write(message[i]);  
}
```

10. (challenge)Modify Q 8 above so that the user can enter as many suburbs as they need. If they wish to stop they are to enter the word "stop". The page is then to display these suburbs.

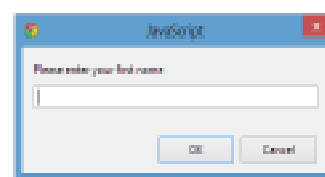
32. Window Object

Let's revisit the window object. Firstly let's look at the 3 methods – alert, confirm, prompt.

window.alert()	window.confirm()	window.prompt()
This is a simple window with a i. message and ii. OK button.	This is a window with a i. question and ii. OK button and iii. Cancel button	This is a window with an iv. input box and v. OK button and vi. Cancel button
	If user clicks 'OK' it will return a TRUE value, otherwise a FALSE value.	User enters a value and this can be saved in a variable.



see Ex1-3



see Ex2-2

Exercise 11

1. Create a simple Web Page where a confirm window asks the user if they wish to see this week's weather forecast. According to the user's response it will write to the screen either "you chose to see this week's weather forecast" or "you didn't choose to see this week's weather forecast". For example

```
var weather = confirm("Do you wish to see this week's weather forecast?")  
if (weather==true)  
{.....}  
else  
{..... }
```

window.location

The location object represents the url area of a window. It can be used as follows:

```
window.location = "http://google.com" //this will go straight to google in the current window  
window.location.assign("http://google.com") //this will do exactly the same
```

Exercise 11 cont.

2. Create a simple Web Page where a confirm window asks the user if they wish to see this week's weather forecast. If the user clicks "OK", then using the location object, the browser's url becomes *http://www.bom.gov.au/nsw/forecasts/sydney.shtml*. If they click "Cancel" then write to the screen *"Another time"*

window.open

this will open a new window with url value as specified in bracket. For example

```
window.open("http://google.com")
```

Exercise 11 cont.

3. Modify Ex11-11, so that *http://www.bom.gov.au/nsw/forecasts/sydney.shtml* is opened in a new window.

window.print

This will print the window. It can be useful for printing a map or a form.

Exercise 11 cont.

4. Create a web page displaying image "Hornsby_Location.jpg". A prompt window is to ask the user if they wish to print it. If user clicks OK, then window.print() will print it.