Storing data in variables

A “variable” is a container, common to every scripting and programming language, in which data can be stored and retrieved later. Unlike the “strongly typed” variables in most other languages, which must declare a particular data type they may contain, JavaScript variables are much easier to use because they are “loosely typed” – so they may contain any type of data:

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>true</td>
<td>A true (1) or false (0) value</td>
</tr>
<tr>
<td>number</td>
<td>100</td>
<td>An integer or</td>
</tr>
<tr>
<td></td>
<td>3.25</td>
<td>A floating-point number</td>
</tr>
<tr>
<td>string</td>
<td>“M”</td>
<td>A single character or</td>
</tr>
<tr>
<td></td>
<td>“Hello World!”</td>
<td>A string of characters, with spaces</td>
</tr>
<tr>
<td>function</td>
<td>init</td>
<td>A user-defined function or</td>
</tr>
<tr>
<td></td>
<td>fido.bark</td>
<td>A user-defined object method</td>
</tr>
<tr>
<td>object</td>
<td>fido</td>
<td>A user-defined object or</td>
</tr>
<tr>
<td></td>
<td>document</td>
<td>A built-in object</td>
</tr>
</tbody>
</table>

A JavaScript variable is declared using the `var` keyword followed by a space and a name of your choosing, within certain naming conventions. The variable name may comprise letters, numbers, and underscore characters, but may not contain spaces or begin with a number. Additionally you must avoid the JavaScript keywords, reserved words, and object names listed in the tables on page 9. The declaration of a variable in a script may simply create a variable to which a value can be assigned later, or may include an assignation to instantly “initialize” the variable with a value:

```javascript
var myNumber;                 // Declare a variable.
myNumber = 10;                // Initialize a variable.
var myString = “Hello World!”; // Declare and initialize a variable.
```

Multiple variables may be be declared on a single line too:

```javascript
var i, j, k;                  // Declare 3 variables.
var num=10, char=“C”;         // Declare and initialize 2 variables.
```

Upon initialization JavaScript automatically sets the variable type for the value assigned. Subsequent assignation of a different data type later in the script can be made to change the variable type. The current variable type can be revealed by the `typeof` keyword.
Create a HTML document that embeds an external JavaScript file and has a “panel” element

```html
<script type="text/javascript" src="variable.js"></script>
<div id="panel"><noscript>
  <div>
    ! JavaScript is Not Enabled.</div>
</noscript></div>
```

Open a plain text editor, like Windows Notepad, and add a function to execute after the document has loaded

```javascript
function init()
{
}
window.onload=init;
```

In the function block, declare and initialize variables of different data types

```javascript
var str="Text Content in JavaScript";
var num=100;
var bln=true;
var fcn=init;
var obj=document.getElementById( "panel");
```

Now insert statements to write the variable values and data types into the panel

```javascript
obj.innerHTML=str + " : " + typeof str;
obj.innerHTML+="\n"+num+" : " + typeof num;
obj.innerHTML+="\n"+bln+" : " + typeof bln;
obj.innerHTML+="\n"+fcn+" : " + typeof fcn;
obj.innerHTML+="\n"+obj+" : " + typeof obj;
```

Save the script alongside the HTML document then open the page in your browser to see the variable data

**Hot tip**

The `typeof` returns a value of “undefined” for uninitialized variables.

**Hot tip**

Notice how the `+` operator is used here to join (concatenate) parts of a string and with `+=` to append strings onto existing strings.