Recognizing variable scope

The extent to which a variable is accessible is called its “scope” and is determined by where the variable is declared:

- A variable declared inside a function block is only accessible to code within that same function block. This variable has “local” scope – it is only accessible locally within that function, so is known as a “local variable”
- A variable declared outside all function blocks is accessible to code within any function block. This variable has “global” scope – it is accessible globally within any function in that script so is known as a “global variable”

Local variables are generally preferable to global variables as their limited scope prevents possible accidental conflict with other variables. Global variable names must be unique throughout the entire script but local variable names only need be unique throughout their own function block – so the same variable name can be used in different functions without conflict.

scope.html

scope.js

1. Create a HTML document that embeds an external JavaScript file and has a “panel” element

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

2. Open a plain text editor then declare and initialize a global variable

```
var global="This is Worldwide Global news<hr>";
```

3. Add a function to execute after the document has loaded

```
function init()
{

}
window.onload=init;
```

4. In the function block, declare and initialize a local variable

```
var obj=document.getElementById( "panel" );
```

5. Next in the function block, write the value of the global variable into the panel

```
obj.innerHTML=global;
```

6. Now in the function block call two other functions, passing the value of the local variable to each one

```
us(obj);
eu(obj);
```

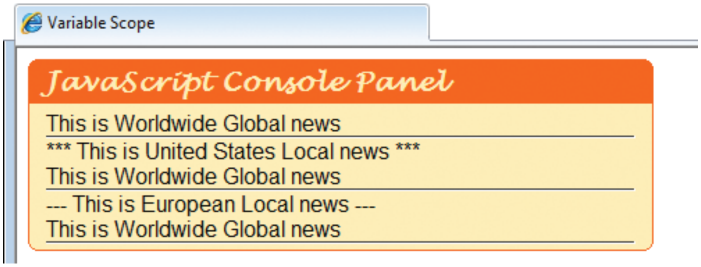
7. Before the "init" function block, insert a function with one argument that initializes a local variable, then appends its value and that of the global variable into the panel

```
function us(obj)
{
  var local="***This is United States Local news***<br>";
  obj.innerHTML+=local;
  obj.innerHTML+=global;
}
```

8. Before the init function block, insert another function with one argument that initializes a local variable, then appends its value and that of the global variable into the panel

```
function eu(obj)
{
  var local="---This is European Local news---<br>";
  obj.innerHTML+=local;
  obj.innerHTML+=global;
}
```

9. Save the script alongside the HTML document then open the page in your browser to see the values of the global and local variables written by the functions



```
This is Worldwide Global news
*** This is United States Local news ***
This is Worldwide Global news
--- This is European Local news ---
This is Worldwide Global news
```

Hot tip

Notice that the "local" variable names do not conflict because they are only visible within their respective function block.

Don't forget

A variable can be declared without initialization, then assigned a value later in the script to initialize it.