

DOM-Zugriff-Beispiel (von Hanser 8.7)

HTML-Gerüst:

```
<!DOCTYPE html>
<html lang="de">
<head>
<meta charset="utf-8" />
<title>DOM</title>
<style type="text/css"/>
  h1 {
    background-color:black;
    color:white;
    text-align:center;
  }
  table{
    width:600px;
    border-style:solid;
    border-width:1pt;
  }
</style>
</head>
<body>
<h2 align="center">Zugriff auf HTML-Attribute über DOM</h2>
<div align="center">
<table>
<tr>
<td>
</td>
<td id="domin" width="200"></td>
</tr>
</table>
</div>
<div align="center">
<button id="dombu1">Groß</button>
<button id="dombu2">Klein</button>
<button id="dombu3">Neues Bild</button>
<button id="dombu4">Altes Bild</button>
</div>
```

Integriertes JavaScript:

```
<script type="text/javascript">// 
with (document) {
getElementById("dombi").onmouseover = function() {
getElementById("domin").innerHTML = getElementById("dombi").alt;
};
getElementById("dombi").onmouseout = function() {
getElementById("domin").innerHTML = "";
};
getElementById("dombi").onclick = function() {
getElementById("domin").innerHTML = getElementById("dombi").src;
};
getElementById("dombu1").onclick = function() {</pre></div>
```

```

getElementById("dombi").width = "400";
};
getElementById("dombu2").onclick = function() {
getElementById("dombi").width = "200";
};
getElementById("dombu3").onclick = function() {
getElementById("dombi").src = "images3/b6.jpg";
getElementById("dombi").alt = "Kater im Korb";
};
getElementById("dombu4").onclick = function() {
getElementById("dombi").src = "images3/b5.jpg";
getElementById("dombi").alt = "Eislaufen in den Weinbergen";
};
}
// ]]></script>

</body>
</html>

```

Bemerkung zu den Knopf-Ereignis-Funktionen

Der dargestellte ID-Zugriff funktioniert richtig, aber die Ereignisfunktionen wie „mouseover, mouseout, onclick, etc“ gehen nur, wenn auf der selben Seite kein window.onload-Befehl existiert.

Ein bis jetzt noch nicht gelöstes Problem ist die am Ende des Scripts verwendete window.onload-Funktion. Diese Funktion bietet bei einer einmaligen Verwendung pro Seite kein Problem, darf aber auf ein und derselben Seite nur ein einziges Mal vorkommen. Existieren auf derselben Seite zwei solche window.onload-Funktionen, so überschreibt die zweite Funktion die erste und macht die Eventhandler des ersten Scripts unwirksam.

In diesem Fall muss der erste onload-Befehl deaktiviert und müssen die Eventfunktionen des ersten onload-Befehls zu den Eventfunktionen des zweiten onload-Befehls verschoben werden. Zugegeben keine saubere Sache, eine bessere Abhilfe konnte aber bis dato nicht gefunden werden.

Es ist erstaunlich, dass die Ereignisaufrufe des unterdrückten onload-Befehls in die andere noch wirkende onload-Funktion übernommen werden können.