

aus: H. Schröder, R. Steyer: JavaScript 1.8, Herdt 2014

3.4 Bezeichner

Bezeichner benennen unter anderem Konstanten, Variablen und Funktionen in JavaScript-Programmen. Bei der Programmierung können Sie über einen Bezeichner auf diese Elemente zugreifen. Ein Bezeichner wird vom Programmierer festgelegt. Dabei gelten folgende Regeln:

- ✓ Die Bezeichner müssen mit einem Buchstaben, dem Zeichen `$` oder einem Unterstrich `_` beginnen.
- ✓ Bezeichner dürfen nur Buchstaben, Ziffern und die beiden Sonderzeichen `$` und `_` enthalten. Deutsche Umlaute wie ä, ö, ü sind zwar möglich, sollten jedoch nicht eingesetzt werden.
- ✓ Die Groß- und Kleinschreibung ist für den späteren Aufruf von Bedeutung, da JavaScript **case-sensitive** ist, d. h., zwischen Groß- und Kleinschreibung unterscheidet.
- ✓ Namen dürfen keine Leerzeichen enthalten.
- ✓ Reservierte Wörter dürfen nicht als Bezeichner verwendet werden.

Beispiel

```
var KundenNr = 123; // Richtig, enthält nur Buchstaben
var $Kunden_Nr = 123; // Richtig, enthält $, Buchstaben und den Unterstrich
var Kunden Nr = 123; // Falsch, enthält Leerzeichen
var -KundenNr = 123; // Falsch, beginnt mit einem Minuszeichen
var 0KundenNr = 123; // Falsch, beginnt mit einer Zahl
var case = 123; // Falsch, ist ein reserviertes Wort
```

3.5 Variablen

Variablen können während der Programmausführung unterschiedliche, veränderbare Werte (**Literale** - ein Wert wie 5 oder true) annehmen, wie z. B. Zwischen- oder Endergebnisse aus Berechnungen. Für jede Variable wird ein Speicherplatz im Arbeitsspeicher Ihres Computers reserviert. Im Programm greifen Sie auf diesen Bereich über den Variablennamen zu. Variablen haben die folgenden Eigenschaften:

- ✓ Variablen können an einer beliebigen Stelle in einem JavaScript-Programm durch die Angabe des Variablennamens und der Zuweisung eines Wertes definiert werden. Obwohl es nicht zwingend notwendig ist, (globale) Variablen mit dem einleitenden Schlüsselwort `var` (variable) zu definieren, dient es der besseren Lesbarkeit.
- ✓ Variablen, die Sie über die Angabe `var` definieren und denen Sie keinen Wert zuweisen, haben den Zustand undefiniert (`undefined`).
- ✓ Variablen, die weder über `var` noch durch eine Wertzuweisung definiert wurden, erzeugen bei der Verwendung einen Laufzeitfehler. Die Interpretation des JavaScript-Codes wird abgebrochen.
- ✓ Haben Sie den Wert für eine Variable nicht explizit definiert, erfolgt dies automatisch bei ihrer ersten Verwendung.
- ✓ Mehrere Variablen können durch Kommata getrennt definiert werden.
- ✓ Einer Variablen können in JavaScript im Programmverlauf Werte verschiedener Datentypen zugewiesen werden (**lose Typisierung**).

Beispiel für die Definition von Variablen

Im Folgenden werden die unterschiedlichen Möglichkeiten der Definition einer Variablen gezeigt. Ein Programm wird verständlicher, wenn Variablen schon vor ihrer Verwendung definiert und initialisiert werden. Initialisieren bedeutet, dass eine Variable mit einem Anfangswert belegt wird.

```
k = 10;           // die Variable k besitzt den Wert 10
var k;           // die Variable k ist noch undefiniert
var i, k = 10;    // die Variable i ist undefiniert und k hat den Wert 10
var k = 10, i = "Text"; // mehrere Definitionen durch Kommata trennen
```

Da die Namen von Variablen, Funktionen, Objekten usw. case-sensitive sind (d. h., es wird zwischen groß- und kleingeschriebenen Buchstaben unterschieden), verweisen z. B. die Variablen `Auto` und `auto` auf unterschiedliche Speicherbereiche.

Beispiel: *kap03/variablen.html*

Da der Variablen `k` kein Wert zugewiesen wurde, wird im Browser als Wert `undefined` ausgegeben.

```
<script type="text/javascript">
  j = 10;
  var k;
  var l = 11;
  document.write("j hat den Wert: ", j, "<br>");
  document.write("k hat den Wert: ", k, "<br>");
  document.write("l hat den Wert: ", l, "<br>");
</script>
```

Wertzuweisungen

Variablen müssen vor ihrer ersten Verwendung mit einem Wert belegt werden. Diese Wertzuweisung erfolgt über das Gleichheitszeichen (der sogenannte **Zuweisungsoperator**). Diese Werte gelten lediglich zu Anfang eines Programms und können im Programmverlauf jederzeit geändert werden. Sie erreichen dadurch, dass alle Variablen vor der ersten Verwendung gültige Werte besitzen.

Beispiel

Für eine Operation werden drei Variablen definiert und mit Werten belegt. Dadurch vermeiden Sie Fehler, die durch eine Verwendung der Variablen auftreten, bevor ihnen über einen Ausdruck ein Wert zugewiesen wird.

```
var summand1 = 1;
var summand2 = 8;
var summe = 0;
document.write(summe); // Diese Anweisung löst jetzt keinen Fehler aus!
summe = summand1 + summand2;
```