Universität zu Köln
Historisch-Kulturwissenschaftliche Informationsverarbeitung
IT-Zertifikat der Phil.-Fak.: Advanced Web Basics
Dr. Jan G. Wieners - jan.wieners@uni-koeln.de

Kurzreferenz JavaScript

1. JavaScript im Webbrowser: Skripten einbetten

Clientseite – JavaScript-Anweisungen lassen sich auf unterschiedliche Arten in HTML-Dokumente einbetten:

- Unter Verwendung des <script>-Tags z.B.(!) im Kopfbereich des HTML-Codes :

```
<html><head><title>JavaScript</title>
      <script type="text/javascript">
        document.write("Hello World");
      </script>
</head> […]
```

…oder an anderer Stelle, z.B. im <body> des HTML-Dokumentes:

```
<div id="inhaltsbereich">
      <div class="textbox">
      <script type="text/javascript">
           document.write("Hello World");
      </script>
      </div>
</div>
```

- Auslagern: Über eine externe Datei, die durch das src-Attribut des <script>-Tags angegeben wird:

```
<script src="scripts/meinskript.js"></script>
```

Über das <noscript> Tag lässt sich – bei älteren Browsern oder Browsern, in denen JavaScript deaktiviert ist – darauf hinweisen, dass JavaScript zur vollen Nutzung der Funktionalität einer Seite notwendig ist:

```
<script type="text/javascript">[…]</script>
<noscript>Ihr Browser muss JavaScript unterstützen</noscript>
```

2. Der Sprachkern von JavaScript

Kommentare

```
// Ein einzeiliger Kommentar

/* Ein
mehrzeiliger
Kommentar */
```

Semikolon

Anweisungen werden in JavaScript immer mit einem Semikolon abgeschlossen: `eineVariable = 23;`

Variablen

Variablen lassen sich über das Schlüsselwort `var` deklarieren. Eine Variable ist ein Bezeichner, der sich mit einem Wert verbinden lässt: Die Variable speichert bzw. enthält den Wert. Variablen ermöglichen es, Daten

in JavaScript-Programmen zu speichern und zu bearbeiten. Die folgenden beiden Zeilen deklarieren die Variable summe und weisen ihr (Zeile 2) das Ergebnis der Addition von 37 und 49 zu:

```
var summe;
summe = 37 + 49;
```

Die Deklaration einer Variable lässt sich mit Ihrer Initialisierung (Zuweisung eines Wertes) verbinden:

```
var nachricht = "Hallo";
var x=5, y=2, z=0;
```

Zahlen

JavaScript unterscheidet nicht zwischen Integer- und Gleitkommawerten: In JavaScript werden alle Zahlen intern als Gleitkommawerte dargestellt.

- Integer (ganze Zahlen): `zaehlVariable = 23 + 15;`
- Gleitkommazahlen (Anstatt Komma: Punkt ('.')): `zaehlVariable = 23 + 15.23;`

Zeichenketten (Strings)

Ein **String** ist der JavaScript-Datentyp für Text. Zeichenketten lassen sich in JavaScript-Anwendungen einfügen, indem sie in Paare **einfacher** oder **doppelter Anführungszeichen** eingeschlossen werden:

```
var eineStringVariable = 'Hallo Welt';

document.write(eineStringVariable);
```

Strings lassen sich mit dem Operator '+' **verketten**, d.h. aneinander anfügen:

```
var ersteStringVariable = 'Hallo';

var zweiteStringVariable = ' Welt';

var dritteStringVariable = ersteStringVariable + zweiteStringVariable;

document.write(dritteStringVariable);
```

Wird eine Zahl in einem Stringverkettungsausdruck verwendet, so wird die Zahl in einen String **umgewandelt**:

```
var n = 12;
var eineStringVariable = n + " Monate hat das Jahr";

document.write(eineStringVariable);
```

Die **Länge** eines Strings lässt sich mit der Funktion length ermitteln:

```
var stringLaenge = eineStringVariable.length;
```

Auf **einzelne Zeichen** der Zeichenketten lässt sich über die folgende Syntax zugreifen:

```
letztesZeichen = eineStringVariable.charAt(eineStringVariable.length-1);
```

Zeichenketten lassen sich über die Funktion `substring` in **Teilzeichenketten** zerlegen:

```
ausschnitt = eineStringVariable.substring(1, 4);
```

Arrays

Ein Array ist eine **Sammlung** von Datenwerten. Jeder Wert in einem Array wird über eine Nummer, den Index adressiert. Arrays lassen sich dem Ausdruck *[ ]* erzeugen:

```
var einArray = [];
```

Der Zugriff auf die Elemente eines Arrays funktioniert über die Angabe der Indices zwischen eckigen Klammern; gezählt wird hierbei ab 0:

```
einArray[0] = 1.2;
einArray[1] = 'Hallo Welt';
```

Funktionen

Eine Funktion ist ausführbarer Code, der von einem JavaScript-Programm definiert wird oder in der JavaScript-Implementierung vordefiniert ist. Obwohl eine Funktion nur ein einziges Mal definiert wird, kann ein JavaScript Programm jede Funktion beliebig oft aufrufen oder ausführen. Einer Funktion lassen sich Argumente oder Parameter übergeben. Darüber hinaus kann eine Funktion einen Wert zurückgeben (return).

Eigene Funktionen werden über das Schlüsselwort `function` definiert:

```
function quadrat(x){ // Die Funktion quadrat erwartet das Eingabeargument x
     return x*x;     // Die Funktion quadriert das Eingabeargument
}                    // und gibt das Ergebnis der Rechenoperation zurück
```

Anweisungen und Kontrollstrukturen: if, else if, while, for

if, else if und else: Über die if-Anweisung lassen sich Entscheidungen treffen bzw. Anweisungen je nach gegebenen Umständen auszuführen:

```
if (Ausdruck) {

     Anweisung
} else if (Ausdruck) {
} else {
     Anweisung
}
```

while: Ist die if-Anweisung eine grundlegende Steueranweisung, mit deren Hilfe sich Anweisungen je nach gegebenen Umständen ausführen lassen, so ist die while-Anweisung eine Anweisung, mit der sich wiederholte Aktionen ausführen lassen:

```
while (Ausdruck){
     Anweisung
}
```

Beispiel:

```
var zaehlVariable=0;
while (zaehlVariable < 10){
     document.write(zaehlVariable);

     zaehlVariable += 1;
}
```

Zu Beginn wird die Variable `zaehlVariable` angelegt und auf 0 gesetzt. Anschließend wird der Wert der Variablen so oft ausgegeben (`document.write(zaehlVariable);`) und der Wert der Variablen um eins erhöht (`zaehlVariable += 1;`), bis die Bedingung der while-Schleife nicht mehr wahr ist (`zaehlVariable < 10`), d.h. die Variable den Wert 10 erreicht hat.

for: Die for-Schleife fasst die notwendigen Gegebenheiten der while-Schleife kompakt zusammen:

```
for (Initialisierung; Bedingung; Veränderung der Variable) {
     Anweisung
}
```

Das folgende Beispiel ist analog dem zuvor betrachteten Beispiel der while-Schleife: Eine Variable wird initialisiert, in jedem Schritt der Schleife wird geprüft, ob eine Bedingung erfüllt ist („ist der Wert der Variablen noch immer kleiner als 10?“) und der Inhalt der Variablen inkrementiert.

```
for (var zaehlVariable=0; zaehlVariable < 10; zaehlVariable += 1){
     document.write(zaehlVariable);
}
```

Operatoren

In den Schleifen haben wir Operatoren verwendet, um Bedingungen zu prüfen. Wir unterscheiden (u.a.) folgende Operatoren:

Operator	Operation	Beispiel
===	Auf Gleichheit prüfen	variable1 === variable2
!==	Auf Ungleichheit prüfen	variable1 !== variable2
&&	Logisches UND: Zwei Bedingungen prüfen	if ((a===b) && (b===c))
||	Logisches ODER	if ((a===b) || (b===c))
>	Größer als („ist Wert1 größer als Wert2?“)	Wert1 > Wert2
<	Kleiner als („ist Wert 1 kleiner als Wert2?“)	Wert1 < Wert2
>=	Größer gleich („ist Wert1 kleiner oder gleich Wert2?“)	Wert1 >= Wert2
<=	Kleiner gleich („ist Wert1 kleiner oder gleich Wert2?“)	Wert1 <= Wert2