JavaScript-Grundlagen

1 Einbinden in HTML

Im Quellcode

```
<script> ... </script>
```

In extra Datei

```
<script src='datei.js'></script>
```

2 Ausgabe

Konsole

Firefox: Extras → Web-Entwickler → Web-Konsole
`console.log('Text');`

Popup-Fenster

`alert('Text');`

Im HTML-Dokument

`document.write('Text');`

3 Grundlegende Datentypen

JavaScript verwendet dynamische Typisierung: Der Datentyp einer Variablen wird bei Wertzuweisung festgelegt und kann sich bei Zuweisung eines anderen Wertes ändern.

Datentyp ermitteln

`typeof(variable)` oder `typeof variable` ergibt `undefined`, `string`, `number`, `boolean`, `function` oder `object`

undefined

`var irgendwas; //` Variablen, denen noch kein Wert zugewiesen wurde, haben den Datentyp `undefined`.

string

`var text = 'Hallo'; //` Text kann in Single-Quotes `'Text'` oder Double-Quotes `"Text"` angegeben werden.

number

```
var ganzZahl = 17;
var kommaZahl = -13.5;
var hexZahl = 0xff; //255
```
Es gibt keinen speziellen Datentyp für ganze Zahlen.

boolean

```
var ok = true;
var wahrheit = false;
```

4 Umwandlung von Datentypen

Die Konvertierung von Datentypen erfolgt in vielen Fällen automatisch. Bestimmte Umwandlungen sind aber explizit möglich:

string → number

```
var ganzZahl = parseInt('42');
var kommaZahl = parseFloat('-13.5');
```

number → string

```
var text = ganzZahl.toString();
var hexText = ganzZahl.toString(16);
var dualText = ganzZahl.toString(2);
```

5 String: Methoden und Eigenschaften

```
var zeichen = 'Text'.charAt(1); // 'e'
var laenge = 'Text'.length; // 4
```

6 Math: Methoden und Eigenschaften

```
var zufall = Math.random(); // 0.00 - 0.99
var abgerundet = Math.floor(3.5); // 3
var pi = Math.PI;
```

7 Array []

Ein Array ist ein Object und damit eine Liste aus Schlüssel-Wert-Paaren. Die Schlüssel sind normalerweise die Zahlen 0, 1, 2, ..., die Werte eines Arrays können von verschiedenen Datentypen sein. Es ist möglich, assoziative Arrays mit anderen Schlüsseln als 0, 1, 2, ... zu erstellen. Für assoziative Arrays sollten aber besser reine Objects verwendet werden.

Beispiel

Schlüssel	Wert
0	'Anna'
1	45
2	'Berta'

erstellen

```
var feld = [];
feld[0] = 'Anna';
feld[1] = 45;
feld[2] = 'Berta';
```

oder

```
var feld = [ 'Anna', 45, 'Berta' ];
```

Anzahl der Elemente

```
var anzahl = feld.length; // 3
```

8 Function ()

Eine JavaScript-Funktion ist ein Objekt und kann deshalb in einer Variablen gespeichert werden.

Definition

```
function quadrat(zahl) {
    return zahl * zahl;
}
```

oder

```
var quadrat = function (zahl) {
    return zahl * zahl;
}
```

Aufruf

```
var ergebnis = quadrat(3);
```

9 Object {}

Ein Objekt ist eine Liste aus Schlüssel-Wert-Paaren. Werte können auch Funktionen sein, dann werden sie Methoden genannt. Die Eigenschaften und Methoden von Objekten können auch zur Laufzeit hinzugefügt werden.

Beispiel: Object 'hund'

Schlüssel	Wert
name	'Bello'
alter	5
wirdaelter	function (jahre) { this.alter += jahre; }

einzelnes Objekt erstellen

```
var hund = {};
hund.name = 'Bello';
hund.alter = 5;
hund.wirdaelter = function (jahre) {
    this.alter += jahre;
}
```

oder

```
var hund = {
        name : 'Bello',
        alter : 5,
        wirdaelter : function (jahre) {
                this.alter += jahre;
        }
    };
```

Objekt verwenden

```
hund.wirdaelter(3);
var alter = hund.alter; // 8
```

oder

```
hund['wirdaelter'](3);
var alter = hund['alter'];
```

mehrere gleichartige Objekte erstellen

Konstruktor

Es gibt in JavaScript keine Klassen. Als Schablonen für Objekte dienen Konstruktoren. Die Namen von Konstruktoren sollten mit einem Großbuchstaben beginnen.

```
function Hund(name, alter) {
    this.name = name;
    this.alter = alter;
    this.wirdaelter = function (jahre) {
        this.alter += jahre;
    };
}
```

Objekte erstellen

```
var hund1 = new Hund('Bello', 3);
var hund2 = new Hund('Hasso', 7);
```