



JavaScript Funktionen



- Funktionen sind Objekte:
 - Sie können über Variablen referenziert werden.
 - Sie können an andere Funktionen übergeben werden.
- Die Deklaration einer Funktion erfolgt:
 - mit dem Schlüsselwort `function` oder
 - mit dem speziellen Konstruktor `Function`.
- Der Aufruf einer Funktion erfolgt:
 - über ihren Namen oder
 - über eine Variable, die auf die Funktion zeigt oder
 - über die `apply`-Methode, die jede Funktion hat.



- Die Parameterübergabe erfolgt wie in Java:
 - call-by-value: eine Kopie der Parameter wird angelegt.
 - Formale Parameter sind lokale Variablen.
 - Bei Objekten wird die Referenz kopiert, nicht das Objekt.
- Rückgabe von Werten:
 - Der Rückgabeparameter wird in JavaScript nicht deklariert.
 - Über Rückgabe erfolgt dann über das Schlüsselwort `return`.
- Funktionen können Funktionen enthalten.



- einfache Deklaration:
`function f() { }`
- Benannter Funktionsausdruck (named function expression):
`var f = function f() { };`
- Anonyme Funktion (anonymous function):
`var f = function() { };`
- Bei der Deklaration und dem benannten Funktionsausdruck kann über die Property `name` der Name der Funktion ausgelesen werden.
 - Dies ist jedoch nicht gemäß ECMA-Standard.



Deklaration einer Funktion: Beispiele



```
function f1() { }  
var f2 = function() { };  
var f3 = function f3() { };  
document.write(f1.name+"<br/>");  
document.write(f2.name+"<br/>");  
document.write(f3.name+"<br/>");
```

f1
f2
f3



Deklaration einer Funktion: Beispielhafte println-Funktion



```
function println(x) {  
    document.write(x + "<br/>");  
}
```

```
println("Hallo");  
println("Welt");
```

Hallo
Welt



Deklaration einer Funktion: feine Unterschiede



- Deklarierte Funktionen sind auch vor der Deklaration schon sichtbar.
- Anonyme und benannte Funktionen sind erst ab der Zuweisung sichtbar.

```
// println() ist vorher definiert
```

```
println(typeof sumDek);
```

```
function sumDek(a, b) { return a + b; }
```

```
println(typeof sumAnon); // nicht verfügbar
```

```
var sumAnon = function(a, b) { return a + b; };
```

```
println(typeof sumAnon);
```

```
function  
undefined  
function
```



Deklaration einer Funktion: verschiedene Versionen



```
function add(a, b) {  
    var result = a + b;  
    return result;  
}
```

```
var sub = function(a, b) {  
    return a - b;  
};
```

// Funktionskonstruktor:

```
var mul = new Function("a", "b", "return a * b;");
```



```
// println() ist vorher definiert
function add(a, b) { return a + b; }
var sub = function(a, b) { return a - b; };
```

```
// Zuweisung des Funktionsobjektes:
```

```
var subtract = sub;
var addiere = add;
```

```
// Aufruf einer Funktion
```

```
var i = add(3, 5);      println(i);      8
var j = sub(7, 4);     println(j);      3
var k = subtract(12, 6); println(k);      6
var l = addiere(12, 6); println(l);      18
var m = add.apply(null, new Array(7, 9)); println(m); 16
```



```
// println() ist vorher definiert
```

```
function aussen(a, b) {  
    function innen(c) {  
        return a + c;  
    }  
    return innen(b);  
}  
  
var m = aussen(3, 4);  
println(m);
```

a = 3

b = 4

c = b = 4

innen(4) = 3 + 4 = 7



```
// println() ist vorher definiert
function varag() {
  var summe = 0;
  for (var i = 0; i < arguments.length; i++) {
    summe += arguments[i];
  }
  return summe;
}
var n = varag(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
println(n);
```



- **eval (string)**
interpretiert übergebenen String als Code
- **isFinite (wert)**
prüft auf gültigen Wertebereich
- **isNaN (wert)**
prüft auf ungültigen Wertebereich
- **parseFloat (string)**
in Fließkommazahl umwandeln
- **parseInt (string)**
String in Ganzzahl umwandeln



- **Number (wert)**
in eine Zahl umwandeln
- **String (wert)**
in eine Zeichenkette umwandeln
- **encodeURIComponent (string)**
in das URI-Format umwandeln
- **decodeURI (string)**
aus dem URI-Format umwandeln
- **Zahl.toFixed (n)**
n Nachkommastellen erzwingen



```
// println() ist vorher definiert
var a = eval("2 + 5");
println(a);
var b = 0.0 / 0.0;
println(isFinite(b));
println(isNaN(b));
var c = parseFloat("24.42") - 1.0;
println(c);
var d = String(5) + "0";
println(d);
var e = encodeURI("Ein String / Strung");
println(e);
var f = 42;
var g = f.toFixed(5);
println(g);
```

7
false
true
23.42
50
Ein%20String%20/%20Strung
42.00000