

Eine Funktion kann man mit einem Unterprogramm im Quellcode vergleichen. Die Funktion wird nicht automatisch ausgeführt, sondern muss durch einen Funktionsaufruf gestartet werden. Sie hat keinen, einen oder mehrere Übernahmeparameter und meist einen oder keinen Rückgabewert.

JS

```
function Funktionsname (Parameter1, Parameter2, ...)
{Anweisungen ...; return;};
```



Eine Funktion beginnt immer mit dem Befehl `function`.

Der Funktionsname soll eindeutig sein und sich von anderen Funktionen unterscheiden. Der Funktionsname ist case-sensitive.

Die Parameter sind lokale Variablen, die in der Funktion abgearbeitet werden.

Der Anweisungsblock steht innerhalb von zwei geschwungenen Klammern `{ }`. Jede Anweisung endet mit einem Semikolon `;`

Mit `return` übergibt man einen Wert oder beendet die Funktion!

z. B. `return ausgabe;` oder `return true;`

```
<script>
function geklickt(welcher) {
    window.alert("Button " + welcher + " wurde geklickt!");
    return;}
</script>
```

```
<button onClick="geklickt('Absenden');" >Absenden</button>
<button onClick="geklickt('Loeschen');" >Löschen</button>
```



Das Attribut `onClick=""` im HTML Element `<button>` führt JavaScript Code aus, wenn man auf das Element klickt. Das `onClick=""` Attribut kann an fast allen HTML Elementen angewendet werden!



Der Funktionsaufruf `geklickt('Absenden');` wurde mit einfachen Anführungszeichen für die Wertübergabe geschrieben, weil das `onClick=""` Attribut die doppelten Anführungszeichen benötigt.



Eine eigene Rundenfunktion mit der Übergabe einer Fließkommazahl und der Anzahl der Stellen.

```
function runden(zahl, stellen) {
    zahl = zahl * Math.pow(10, stellen);
    zahl = Math.round(zahl) / Math.pow(10, stellen);
    return zahl;};
```

```
var ausgabe = runden(34.3862345, 3);
window.alert(ausgabe);
```

← Start der Funktion



Der Funktionsaufruf wurde mit einer Variablen-Vereinbarung aufgerufen. Die lokale Variable `zahl` wurde abgearbeitet und mit `return` an die Variable `ausgabe` übergeben.