

Functions – Funktionen Teil 1

Funktionen:

Funktionen sind Container für Code / Programme.

Dieser Code wird erst ausgeführt, wenn die Funktion aufgerufen wird (sonst wird aller Code im main.js File ausgeführt, sobald im HTML-Code die Verlinkung zum main.js gelesen wird).

Funktionen werden mit dem Schlüsselwort **function()** definiert, sie haben i.d.R. einen Namen.

Syntax:

```
function() NameDerFunktion {  
    Code  
}
```

Die Funktion wird durch Ihren Namen und die runden Klammern (), welche sie als Funktion kennzeichnen, aufgerufen und ausgeführt.

```
function beispielFunktion_1() {  
    console.log(`Dies ist die beispielFunktion_1`);  
}  
// definiert die Funktion namens beispielFunktion_1
```

```
beispielFunktion_1();  
// ruft die Funktion namens beispielFunktion_1 auf und führt den Code aus
```

Solch eine Definition nennt man **function statement**.

Einer Funktion kann bereits vor ihrer Definition aufgerufen und ausgeführt werden (**hoisting**).

Funktionen mit Parametern

Einer Funktion kann ein oder mehrer Parameter übergeben werden, mit diesen Parametern kann die Funktion dann ihren Code ausführen.

Syntax:

```
function(Parameter_1, Parameter_2) NameDerFunktion {  
    Code  
}
```

```
function beispielFunktion_2(par_1, par_2) {  
  console.log(`Wenn man ${par_1} mit ${par_2} multipliziert, dann erhält man  
folgendes Ergebnis ${par_1*par_2}.`)  
}
```

```
beispielFunktion_2(3, 7);
```

// ruft die Funktion beispielFunktion_2 mit den Parametern par_1 = 3 und par_2 = 7 auf.

Die Funktion gibt auf der Console aus:

Wenn man 3 mit 7 multipliziert, dann erhält man folgendes Ergebnis 21.

Funktionen mit Rückgabewerten (Return)

Werte die innerhalb einer Funktion definiert oder berechnet werden, können (auch nach dem Aufrufen der Funktion) von außen nicht gelesen werden.

```
function beispielFunktion_3() {  
  var beispiel = 1000;  
}
```

```
beispielFunktion_3();
```

// ruft die Funktion beispielFunktion_3 auf und führt sie aus.

```
console.log(beispiel);
```

// gibt auf der Console aus:

Uncaught ReferenceError: beispiel is not defined

Um einen Wert aus einer Funktion zu erhalten, muss dieser Wert returtnt werden und einer Variablen zugewiesen werden.

```
function beispielFunktion_4() {  
  var beispiel2 = 1000;  
  return beispiel2;  
}  
var returnVonBeispielFunktion_4 = beispielFunktion_4();
```

// hier wird der Variablen returnVonBeispielFunktion_4 der returnte Wert durch den Aufruf der Funktion beispielFunktion_4 zugewiesen.

```
console.log(returnVonBeispielFunktion_4);
```

Auf der Console wird der Wert der Variablen returnVonBeispielFunktion_4 ausgewiesen:
1000

Ein Return beendet auch immer eine Funktion!

Wenn einer Variablen der return-Wert einer Funktion zugewiesen wird, so nennt man dies **function expression**.

Function expressions werden sofort ausgewertet, hoisting funktioniert hier nicht.

Questions? You're welcome!

holger.zerbe@digitalcareerinstitute.org