

Web-Entwicklung: Übungsblatt 1

Aufgabe 1: Einrichtung Node.js

Richten Sie eine Node.js-Laufzeitumgebung ein.

Testen Sie Ihre Laufzeitumgebung in einer Konsole, indem Sie mit dem Befehl `node --version` die installierte Version anzeigen.

Aufgabe 2: Funktionen und Objekte

a) Berechnung des Arithmetischen Mittelwerts

Schreiben Sie eine Funktion, die den Mittelwert einer Liste von Zahlen berechnet.

Testen Sie Ihre Funktion mit den unten vorgegeben sowie eigenen Testdaten.

Nutzen Sie die Methoden des `console`-Objekts (siehe [Node.js-Dokumentation: Console](#)) zur Ausgabe der Ergebnisse.

Beispiel

```
calculateAverage([1, 2, 3, 4, 5]) = 3
calculateAverage([2, 2, 3, 3])    = 2.5
calculateAverage([6, 5, 3, 9, 2]) = 5
```

b) Berechnung der Standardabweichung

Erweitern Sie Ihre Funktion um die Berechnung der Standardabweichung einer Liste von Zahlen.

Diese eine Funktion soll sowohl den Mittelwert als auch die Standardabweichung zurückgeben.

Beispiel

```
console.dir(calculateAverageAndDeviation([1, 2, 3, 4, 5]));

/*
{
  average: 3,
  deviation: 1.41...
}
*/
```

Hinweis: In einigen Wochen werden Sie eine Möglichkeit kennen lernen, Methoden mit beliebig vielen Argumenten aufzurufen, ohne diese zuvor in ein Array packen zu müssen.

Aufgabe 3: Ausführungskontext von Funktionen

a) Berechnung der gewichteten Summe

Berechnen Sie mittels einer Funktion die gewichtete Summe des folgenden Datensatz:

```
var data = [  
  { value: 10, weight: 1 },  
  { value: 16, weight: 0.3 },  
  { value: 12, weight: 0.4 },  
  { value: 20, weight: 0.5 },  
  { value: 32, weight: 0.4 },  
  { value: 11, weight: 0.3 },  
  { value: 8, weight: 1 },  
];
```

Dabei kann z.B. die Array-Methode `reduce()` eingesetzt werden. Benutzen Sie die Function-Methode `apply()` oder `bind()`, um Ihre Funktion auf den Datensatz anzuwenden, ohne diesen als Funktionsargument zu übergeben.

b) Festlegen des Ausführungskontextes

Fügen Sie anschließend die Funktion als Methode zu dem Datensatz hinzu, sodass diese wie folgt aufgerufen werden kann:

```
console.log(data.calculateWeightedSum());
```

Aufgabe 4: Array-Methoden

Demonstrieren Sie die Verwendung der Array-Methoden `forEach()` und `map()`, indem Sie ein Programm schreiben, welches String-Arrays so verarbeitet, das am Ende in jedem Feld nur ein einziges Wort steht. Array-Elemente, die Aufzählungen repräsentieren (durch Kommata oder "und" getrennt), werden erkannt und auf mehrere Array-Felder verteilt. Außerdem soll die Liste am Ende in lexikografisch absteigender Sortierung ausgegeben werden.

Beispiel

Aus

```
["Justus, Peter und Bob", "Christian", "Susi und Strolch", "Heinz", "Daniel,Gert"]
```

wird

```
["Susi", "Strolch", "Peter", "Justus", "Heinz", "Gert", "Daniel", "Christian", "Bob"]
```

Zusätzliche Informationen

- [Node.js-Dokumentation](#)