

Web-Entwicklung: Übungsblatt 1

Aufgabe 1: Einrichtung von Node.js

Richten Sie eine Node.js-Laufzeitumgebung ein.

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

Aufgabe 2: Funktionen und Objekte

a) Berechnung des arithmetischen Mittelwerts

Schreiben Sie eine Funktion, die den arithmetischen Mittelwert eines Number-Arrays berechnet. Nutzen Sie die Methoden des `console`-Objekts (siehe [Node.js-Dokumentation: Console](#)) zur Ausgabe der Ergebnisse.

Testen Sie Ihre Funktion mit den unten vorgegeben Testdaten.

Beispiel

```
console.log(calculateAverage([1, 2, 3, 4, 5])); // 3
console.log(calculateAverage([2, 2, 3, 3]));    // 2.5
console.log(calculateAverage([6, 5, 3, 9, 2])); // 5
```

b) Berechnung der Standardabweichung

Erweitern Sie Ihre Funktion um die Berechnung der Standardabweichung eines Number-Arrays. 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...
}
*/
```

Aufgabe 3: Ausführungskontext von Funktionen

a) Berechnung der gewichteten Summe

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

```
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 können Sie eine einfache Schleife einsetzen oder Sie versuchen es mal mit der Array-Methode `reduce()`.

b) Festlegen des Ausführungskontextes I

Verändern Sie Ihre Lösung aus Teilaufgabe a, sodass Sie Ihre Funktion auf den Datensatz anwenden können, ohne diesen als Funktionsargument zu übergeben. Benutzen Sie dazu eine der Function-Methoden `call()`, `apply()` oder `bind()`.

c) Festlegen des Ausführungskontextes II

Fügen Sie anschließend die Funktion als Methode zu dem `data`-Objekt hinzu, sodass diese wie folgt aufgerufen werden kann:

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

Aufgabe 4: Array-Methoden

Schreiben Sie eine Funktion, die ein als Argument übergebenes String-Array so verarbeitet, das in jedem Element des zurückgegebenen Arrays nur ein einziges Wort steht. Array-Elemente, die Aufzählungen repräsentieren (durch Kommata oder "und" getrennt), werden auf mehrere Array-Elemente verteilt. Die Array-Elemente sollen lexikografisch absteigend sortiert sein.

Auch hierbei können Sie auf diverse Array-Methoden (z.B. `reduce()`, `forEach()` oder `map()`) sowie auf String-Methoden zurückgreifen.

Beispiel

Aus

```
["Justus, Peter und Bob", "Christian", "Susi und Strolch", "Heinz", "Daniel,Gert",  
"John und George und Paul und Ringo"]
```

wird

```
["Susi", "Strolch", "Ringo", "Peter", "Paul", "Justus", "John", "Heinz", "Gert",  
"George", "Daniel", "Christian", "Bob"]
```

Zusätzliche Informationen

- [Node.js-Dokumentation](#)