

Der österreichische Fussballverband (ÖFB) benötigt ein Programm zur Verwaltung der Bundesliga und hat deine Firma damit beauftragt.

Allgemeines

Man hat dir eine Datenbank geschickt (`database.json`), die alle Spiele der aktuellen Saison beinhaltet. Analysiere das File (i.e. dessen Aufbau) und löse folgende Aufgabenstellungen:

1. Liste aller Mannschaften

Schreibe eine Funktion `getAllTeams`, die ein Array aller Mannschaften zurückliefert. Jede Mannschaft darf nur einmal angeführt sein. Die Reihenfolge der Mannschaft spielt keine Rolle.

2. Saison Details

Schreibe eine Funktion `getSaisonDetails`, die ein Objekt aller Mannschaften inklusive deren geschossenen und erhaltenen Tore enthält. Key des Objekts soll der Name der Mannschaft sein - der Value ein Objekt mit den Keys 'shot' (=Anzahl geschossener Tore) und 'got' (=Anzahl bekommener Tore) - siehe Referenzausgabe (*reference.txt*).

3. Torverhältnis

Schreibe eine Funktion `getGoalDifference`, die die Mannschaften samt deren Torverhältnis (*geschossene Tore : bekommene Tore*) zurückliefert.

Die Mannschaften sollen vom besten zum schlechtesten Torverhältnis sortiert sein. Interessant für das Torverhältnis ist nur der Endstand (du kannst den Pausenstand also ignorieren).

Das Torverhältnis berechnet sich - wie der Name schon sagt - aus dem Verhältnis zwischen geschossenen zu bekommenen Toren. Je größer der Wert desto besser. Haben zwei Mannschaften das gleiche Torverhältnis, spielt die Reihenfolge der beiden Teams zueinander keine Rolle.

Hinweis:

Die Werte in der Referenzausgabe (*reference.txt*) wurden bei 'value' zwecks Übersichtlichkeit auf 2 Dezimalstellen gerundet. Du kannst dies mittels `Math.round(num * 100) / 100` erreichen.

4. Punktestand

Schreibe eine Funktion `getScore`, die alle Mannschaften sortiert nach ihren Punkte ausgibt. Eine Mannschaft bekommt für einen Sieg 3 Punkte, für ein Unentschieden einen Punkt. Haben zwei Mannschaften den gleichen Punktestand, spielt die Reihenfolge zueinander keine Rolle.

Aufbau

Schreibe ein Modul `soccer.js` und speichere es im Unterverzeichnis `'src/'`. Implementiere in diesem Modul die auf der ersten Seite geforderten Funktionen.

Importiere dieses Modul in der Datei `index.js`, die im Hauptverzeichnis deiner Solution liegen soll (also nicht im Unterverzeichnis `'src/'`). Dieses Modul soll den Einstiegspunkt in dein Programm darstellen und alle Funktionen des Moduls `soccer.js` der Reihe nach aufrufen und die Ergebnisse auf der Konsole ausgeben.

Wichtige Hinweise

Gestalte dein Programm so, dass du Teile wiederverwenden kannst. Wenn du alles immer von vorne implementierst, wirst du nicht fertig.

Tipps

- die *JavaScript*-Funktion `Object.keys(myObject)` liefert dir ein Array aller *Keys* des Objekts `myObject`
- die *Lodash*-Funktion `uniq` hilft dir doppelte Einträge aus einem Array zu entfernen
- die Funktion `Array.prototype.sort()`, die wir auch im Unterricht verwendet haben, hilft dir Einträge eines Arrays zu sortieren. Du findest die Dokumentation dieser Funktion im Dokumentations-Folder in der Datei `'sort.pdf'`
- falls du nicht weißt wie du einen String in eine Zahl umwandeln kannst, schau dir das Dokument `'String to Number.pdf'` im Dokumentations-Folder an
- erinnere dich an die 2. Klasse: um einen Strings bei einem bekannten Trennzeichen (engl.: *delimiter*) zu zerschneiden, ist die Funktion `split` hilfreich (siehe Dokumentations-Folder, Datei `'split.pdf'`)
- gib dir immer wieder Zwischenstände deiner erzeugten Daten auf der Konsole aus, um festzustellen, ob du auf dem richtigen Weg bist
- verwende den Debugger, um Fehler zu finden
- du kannst nur Arrays sortieren, nicht aber Objekte - konvertiere Objekte also da wo du sortieren musst in entsprechende Arrays
- du kennst aus dem Unterricht bereits die `forEach`-Schleife. Eine weitere Methode eine `foreach`-Schleife zu programmieren geht so: **`for (let element of myArray) { ... }`**
- du kannst auf den *Value* eines Objektes bei gegebenem *Key* auf zwei Arten zugreifen:
 1. `myObject.team`: liefert dir den *Value* für den *Key* `'team'`
 2. `myObject['team']`: liefert dir ebenfalls den *Value* für den *Key* `'team'`

Hast du den *Key* `'team'` in einer Variable gespeichert - sagen wir der Variablen `x` - empfiehlt sich die zweite Schreibweise, also: `myObject[x]`