

# Aufgaben zum Kickoff-Workshop von WBA2

---

## Grundlegende Informationen

Node.js bietet das Modul `fs` integriert, welches den Zugriff auf Dateien ermöglicht.

```
javascript var fs = require('fs');
```

Die Funktion

```
fs.readFile(__dirname+"/dateiname", function(err, data) { ... });
```

ermöglicht das asynchrone Auslesen von Dateien.

Die Variable `__dirname` enthält den Namen des Verzeichnisses, in dem das aktuelle Programm liegt. Er wird in diesem Beispiel vor den Pfad angehängt, um eine vollständige Pfadangabe zu vermeiden. Der Parameter `data` ist normalerweise ein Buffer mit Binärdaten, der mittels `data.toString()` in einen String umgewandelt werden kann.

Die Funktion

```
fs.writeFile(__dirname+"dateiname", data, function(err) { ... });
```

ermöglicht das asynchrone Schreiben von Dateien.

Weitere Informationen zum `fs`-Modul von node.js finden sie in der [node.js-Dokumentation](#).

Mittels `JSON.stringify( ... )` und `JSON.parse( ... )` können Objekte in einen JSON-String umgewandelt werden und umgekehrt.

**Wichtig: Aus ihrer Git-Historie sollte hervorgehen, dass die Aufgaben in der hier vorgegebenen Reihenfolge gelöst wurden. Erstellen sie also pro Aufgabe mindestens einen separaten Commit.**

## Aufgabe 1

Schreiben sie ein Programm in Node.js, das die Datei [wolkenkratzer.json](#) ausliest und die darin enthaltene Liste von Wolkenkratzern in folgender Form auf der Konsole ausgibt:

```
Name: Burj Khalifa
Stadt: Dubai
Höhe: 828m
-----
Name: Shanghai Tower
Stadt: Shanghai
Höhe: 632m
-----
...
```

Verwenden sie zum Auslesen der Datei die oben genannten **asynchronen** Funktionen des fs-Moduls.

## Aufgabe 2

Erweitern sie das Programm aus Aufgabe 1 so, dass die Namen, Städte und Höhenangaben jeweils **farbig** ausgegeben werden. Hierzu können sie das Modul `chalk` verwenden, welches auch im Workshop vorgestellt wurde. Mehr zum Chalk-Modul finden sie [hier](#). Erstellen zu ihrem Programm auch eine `package.json`-Datei, welche unter anderem die Modulabhängigkeiten definiert. Weitere Informationen zur `package.json`-Datei finden sie in der [NPM-Dokumentation](#).

## Aufgabe 3

Erweitern sie das Programm aus Aufgabe 2 so, dass es die Wolkenkratzer der Höhe nach sortiert, die sortierte Liste in der neuen Datei `wolkenkratzer_sortiert.json` speichert und sie erst danach auf der Konsole ausgibt.

Informationen zum sortieren von Arrays in Javascript finden sie beispielsweise [hier](#).