

Aufgabe 2: Praxis

1. Cat clone

- In dieser Aufgabe geht es darum einen einfachen Clone des `cat` core utils mit Javascript und Node Js zu erstellen
- Erstelle eine Datei `cat.js`
- Erstelle eine Datei `text.txt` mit folgendem Inhalt: `Hallo Welt`
- Beim Aufruf von `node cat.js text.txt` soll in der Console der Inhalt der Datei (also `Hallo Welt` wieder gegeben werden)
- Der Name der zu öffnenden Datei soll also per Command Line Argument übergeben werden

1.1 Error Handling

- Nun geht es darum, das Script durch effektives Error Handling robuster zu machen
- Ein wichtiger Fehler, der häufig auftreten kann, ist dass die gewünschte Datei nicht existiert
- Erstelle also bevor du versuchst die Datei auszulesen einen Check, der überprüft ob die Datei existiert:
 - Falls Ja, gebe den Inhalt wie zuvor in der Console aus
 - Falls Nein, Schreibe eine Error Nachricht in die Console (`console.error`), die darauf hindeutet, dass die Datei nicht existiert. **Danach schließe das Programm mit `process.exit(1)`

1.2 Concatenate

- Das `cat` Command leitet sich ursprünglich von dem Englischen Begriff `concatenate` ab, was *verketten* bedeutet
- Wenn man `cat` mit mehreren Dateien aufruft, werden diese nacheinander in einem Text ausgegeben
- Die einzelnen Dateien, werden jedoch mit einem Newline Symbol (`\n`) getrennt (Neue Zeile / Enter auf der Tastatur drücken)
- Erstelle eine neue Datei `text2.txt` mit folgendem Inhalt: `Hallo von TechStarter`
- Schreibe dein Programm so um, dass wenn mehr als eine Datei als Commandline Argument übergeben wird, alle Dateien nacheinander in der Console ausgegeben werden

```
node cat.js text.txt text2.txt
# Hallo Welt
# Hallo von TechStarter
```

Aufgabe 2 (Optional): Find clone

- Diese Aufgabe ist freiwillig!
- Jetzt geht es darum, die grundlegende Funktion des `find` commands mit Javascript zu replizieren
- Ähnlich wie beim `ls` command, werden alle Dateien im aktuellen Ordner angezeigt
- Allerdings werden auch alle Ordner in Sub-Ordnern (usw.) angezeigt -> Rekursiv
- Erstelle eine neue Javascript Datei `find.js`

Beispiel

- Stelle dir einen Ordner mit folgender Struktur vor:

```
find.js
my-files/
```

```
README.md
src/
  index.html
  script.js
  assets/
    img.png
```

- Dein Programm soll nun wie folgt funktionieren:

```
node find.js my-files/
# README.md
# src/index.html
# src/script.js
# src/assets/img.png
```

Tipps

- In dieser Aufgabe musst du von **Recursive Functions** gebrauch machen:
 - Das bedeutet, man ruft eine Funktion von der selben Funktion auf
 - Um keinen *Infinite Loop* zu erstellen, ist es wichtig, dass der Aufruf an eine If Bedingung gekoppelt ist
 - Schaue dir hier mehr Informationen an <https://www.javascripttutorial.net/javascript-recursive-function/>