

Aufgaben

1. Ist die folgende Funktion eine Pure Function?

```
function add10 (x) {  
    return x + 10;  
}
```



Ja, die Funktion bekommt eine direkte Eingabe.

2. Nutze die Datei „aufgabe_02.js“. Gegeben sind die Funktionen map und der Array data. Nutz beides um ein neues Array zu erzeugen, dass nur die Namen der Personen enthält.



3. Beschreibe, welche Eigenschaften aus einer normalen Funktion eine Higher Order Function machen.



Higher Order Functions bekommen ihrerseits Funktionen als Input, oder liefern Funktionen als Ergebnis.

4. Betrachte die Datei „aufgabe_04.js“. Verwendet das Programm nur immutable data?

Nein, let color = "red" ist eine veränderbare Variable.



5. Erläutere, was ist ein Side Effect?

Ein Side Effect tritt auf wenn die Funktion ihren auf Inputs außerhalb sich selbst bezieht, sie z.B. durch Variablen gespeist wird.



Aufgaben

6. Betrachte die folgende Funktion. Welche Parameter hat die Funktion und was gibt sie als Return-Wert zurück?

```
const square = value => value * value;
```

Eingabeparameter Das Ergebnis von value*value wird returned



7. Betrachte die Funktion aus Aufgabe 6. Schreibe eine Zeile, welche die Funktion aufruft. Als Parameter kannst du einen beliebigen Wert eingeben. Welchen Output erzeugt die Funktion mit dem gewählten Input?



8. Gegeben ist die Datei „aufgabe_08.js“. Welche Personen verbleiben im gefilterten Array, dass durch den console.log-Aufruf ausgegeben wird?



9. Notiere die folgende Funktion in Form einer Arrow-Funktion

```
function (x) {  
    return x * 2;  
}
```



x => x*2