ARROW FUNCTIONS

Arrow functions wurden mit EcmaScript6 (ES6) 2015 eingeführt.
Es handelt sich hierbei nur um eine andere Schreibweise für Funktionen.

Syntax:

const meineFunktion = (meineParameter) => {Code}

Neu ist hier, dass auf das Schlüsselwort function verzichtet wird, stattdessen wird ein symbolischer Pfeil (engl. arrow = Pfeil), bestehend aus dem Gleichheitszeichen und dem Größer-Zeichen, als Schlüsselwort benutzt.

Der Funktionsaufruf funktioniert wie gewohnt.

Arrow-function ohne Parameter:

```
const hallosagen = () => {
	console.log(`ich sage hallo`);
}
```

Wie bei allen Funktionen sind auch hier die leeren Klammern () nötig, auch wenn keine Parameter übergeben werden!

```
hallosagen();
```
// Gibt auf der Console aus:
ich sage hallo

Arrow-function mit Parameter:

```
const dreieckstest = (alpha, beta, gamma) => {
	if (alpha + beta + gamma === 180) {
		return `Das ist ein Dreieck`;
	} else {
		return `Sorry, kein Dreieck`;
	}
}
```

```
console.log(dreieckstest(90, 30, 60));
```
// Gibt auf der Console aus:
Das ist ein Dreieck

.map() mit Arrow-function

Ebenso ist das “mappen“ durch einen Array mit einer Arrow-function möglich. Dazu können die drei Parameter wert, index und array (in dieser Reihenfolge) übergeben werden. Braucht man nur den Wert, dann wird nur der parameter wert übergeben, braucht man jedoch den Array, so müssen alle drei Parameter angegeben werden.

```
let lichtAn = [`yes`, `Licht ist an`, `switched on`, `ja`, true, `on`];
const lichtTrue = lichtAn.map(wert => true);
```

Hier werden alle “menschlichen“ Antworten (vom Array “lichtAn“) auf die Frage, ob das Licht eingeschaltet ist, in ein boolisches true umgewandelt und im neuen Array lichtTrue gespeichert.

```
console.log(lichtTrue);
// Gibt auf der Console aus:

(6) [true, true, true, true, true, true]
```

```
let pins = ["0687", 7645, "3g99", 465, 7523];
const pinTest = pins.map((einzelnePIN, index, array) => {
        if (einzelnePIN != parseInt(einzelnePIN) ||
        JSON.stringify(einzelnePIN).length !== 4) {
          return `PIN nicht gültig!`
        } else
          return `Diese Pin ist gültig!`
})
```

Hier werden alle Items eines Arrays darauf überprüft, ob sie als Pin-Zahlen fürs Online-Banking gültig sind. Es können sowohl numbers las auch strings gültig sein, wenn sie mindestens vier Stellen haben und nur aus Ziffern bestehen.

Die Parameter index und array werden hier nicht benötigt, sie sind nur angegeben, um zu zeigen, dass man sie so übergeben kann

```
console.log(pinTest);
// Gibt auf der Console aus:

(5) ["PIN nicht gültig!", "Diese Pin ist gültig!", "PIN nicht gültig!", "PIN nicht gültig!", "Diese Pin ist gültig!"]
```

Questions? You‘re welcome!

holger.zerbe@digitalcareerinstitute.org