

ARROW FUNCTIONS

HOISTING UND THIS-KEYWORD

Arrow functions unterscheiden sich zu “normalen” Funktionen in zwei Punkten. Erstens werden Arrow functions immer als function expression deklariert

Syntax:

```
const myArrowFunction = () => {Code}
```

Dadurch funktioniert hoisting hier nicht, d.h. die Funktion muss deklariert sein, bevor sie aufgerufen werden.

Zweitens ändert sich die Bedeutung des this keywords.

Bei einer Arrow function referenziert das this-keyword auf die Umgebung, in der eine Methode (also eine Funktion des Objekts) ausgeführt wird und nicht auf das Objekt selbst.

Dies hat vor allem Auswirkungen auf Function Constructors. Damit das this-keyword weiter funktioniert, müssen im Function Constructor Klassen (classes) benutzt werden.

Wenn in einer Klasse (classes) eine Arrow Function benutzen, können wir später bei dem instanziierten Objekt mit dem this keyword auf die Klasse zugreifen.

Function Constructor mit “normaler Funktion”:

```
function Nameconstructor(vorname, nachname) {  
  this.vorname = vorname;  
  this.nachname = nachname;  
  this.namensZusammenSetzer = function () {  
    this.kompletterName = this.vorname + " " + this.nachname  
  }  
}
```

```
let bundeskanzlerin = new Nameconstructor("Angela", "Merkel");  
bundeskanzlerin.namensZusammenSetzer();
```

```
console.log(bundeskanzlerin);
```

// Gibt auf der Console aus:

```
Nameconstructor {vorname: "Angela", nachname: "Merkel",  
  namensZusammenSetzer: f, kompletterName: "Angela Merkel"}
```

```
1.kompletterName: "Angela Merkel"  
2.nachname: "Merkel"  
3.namensZusammenSetzer: f ()  
4.vorname: "Angela"  
5.__proto__: Object
```

Function Constructor mit Arrow Function:

```
const Nameconstructor2 = (vorname, nachname) => {  
  class NeuerName {  
    vorname = "Dies ist der Vorname";  
    nachname = "und das ist der Nachname";  
    constructor() {}  
    namenszusammensetzer2 = () => {  
      this.kompletterNameMitThis = this.vorname + " " +  
        this.nachname;  
      this.kompletterNameOhneThis = vorname + " " + nachname;  
    };  
  }  
  const namensObjekt = new NeuerName(vorname, nachname);  
  return namensObjekt;  
}
```

```
let aussenminister = Nameconstructor2("Heiko", "Maas");  
ausсенminister.namenszusammensetzer2();  
console.log(aussenminister);
```

// Gibt auf der Console aus:

```
NeuerName {vorname: "Dies ist der Vorname", nachname: "und das ist  
der Nachname", namenszusammensetzer2: f, kompletterNameMitThis:  
"Dies ist der Vorname und das ist der Nachname",  
kompletterNameOhneThis: "Heiko Maas"}
```

```
kompletterNameMitThis: "Dies ist der Vorname und das ist der  
Nachname"  
kompletterNameOhneThis: "Heiko Maas"  
nachname: "und das ist der Nachname"
```

```
namenszusammensetzer2: () => {...}  
vorname: "Dies ist der Vorname"  
__proto__: Object
```

Questions? You're welcome!

holger.zerbe@digitalcareerinstitute.org