

Primitives VS Objects

In Javascript gibt es mehrer primitive und einen komplexen Datentypen.

Primitives

Primitive Daten sind Strings, Numbers, Booleans, Null, undefined sowie Symbols (seit 2015, sehr selten).

NaN (not a number) gehört zum Datentyp numbers

Objects

Komplexe Datentypen sind Objekte.

typeof

Mit typeof kann der Datentyp einer Variablen ausgegeben werden.

```
let numberVar = 44;
let stringVar = `ich bin ein String`;
let booleanVar = true;
let nullVar = null;
let undefinedVar;
let objectVar = {
  vorname: `Ich`,
  alter: 111
}
```

```
console.log(typeof numberVar);
// gibt auf der Console aus:
number
```

```
console.log(typeof stringVar);
// gibt auf der Console aus:
string
```

```
console.log(typeof booleanVar);
// gibt auf der Console aus:
boolean
```

```
console.log(typeof nullVar);
// gibt auf der Console aus:
object
```

```
console.log(typeof undefinedVar);
// gibt auf der Console aus:
undefined
```

```
console.log(typeof objVar);
// gibt auf der Console aus:
object
```

Das = Zeichen bei Primitives als Assignment (Zuweisung)

Bei primitiven Daten ist das = Zeichen ein Zuweisung, steht zwischen einer Variableninitialisierung und einer anderen Variablen ein = Zeichen, so wird eine Kopie (Call by Value) erstellt.

```
let zahl = 100;  
let kopieVonZahl = zahl;
```

```
console.log(zahl);  
gibt auf der Console aus:  
100
```

```
console.log(kopieVonZahl);  
gibt auf der Console aus:  
100
```

Wird nun die ursprüngliche Variable mit einem Wert überschrieben, so bleibt die Kopie erhalten!

```
zahl = 999;  
  
console.log(zahl);  
gibt auf der Console aus:  
999  
  
console.log(kopieVonZahl);  
gibt auf der Console aus:  
100
```

Das = Zeichen bei Objects als Reference (Referenz)

Bei Objekten ist das = Zeichen eine Referenz, keine Kopie sondern eine Referenz (Call by Reference)
Es wird kein neues Objekt kreiert, sondern eine Referenz die auf das alte Objekt zeigt wie Links im Internet

```
let zahlObject1 = {  
  a: 1,  
  b: 2,  
  c: 3  
}  
  
let zahlObject2 = zahlObject1
```

```
console.log(zahlObject1.c);
```

gibt auf der Console aus:

3

```
console.log(zahlObject2.c);
```

gibt auf der Console aus:

3

```
zahlObject1.c = 555;
```

```
console.log(zahlObject1.c);
```

gibt auf der Console aus:

555

```
console.log(zahlObject2.c);
```

gibt auf der Console aus:

555

Questions? You're welcome!

holger.zerbe@web.de