

# Javascript Grundlagen Cheat Sheet by Huluvu424242 (FunThomas424242) via cheatography.com/19991/cs/19484/

### Javascript - Currying

#### Was ist das?

Beim Currying wird eine Funktion der Ordnung n+1 (also n+1 Parameter) auf eine Funktion der Ordnung n (also n Parameter) reduziert.

Das Konzept dabei ist, die Funktion um einen Parameter zu kürzen und statt des ursprü ngl ichen, berech neten Wertes eine Funktion mit dem gekürzten Parameter zur weiteren Berechnung zurück zu geben. Damit wird die Funktion nun mit einem Parameter weniger aufger ufen. Nun muss aber das Ergebnis noch mit dem reduzi erten Parameter aufgerufen werden. Also in Summe wurden wieder alle Parameter zur Berechnung übergeben, nur verteilt auf 2 Aufrufe oder mehr.

#### Beispiel für Currying und Uncurrying

```
currye dMu ltiply = (n) => (m) => n * m currye dMu lti ply (3)(4) === 12 // true multiply = (n, m) => currye dMu lti ply (n)(m) multip ly(3, 4) === 12 // true
```

**Quelle:** https://blog.benestudio.co/currying-in-javascript-es6-540d2a-d09400

### Javascript - Ich finde Lambdas verkürzt doof

### Javascript - Ich finde Lambdas verkürzt doof (cont)

```
> // vollständige Schreibweise mit Selbstausführung am Ende
const berechneWortlaenge = function(){
 const map = processArray(woerter, function( wort ){
  return { wort: wort, laenge: wort.length};
 console.log('1. Ausgabe: '+JSON.stringify(map));
}();
// Umgewandelt in lambda Ausdruck
const berechneWortlaenge = function(){
 const map = processArray(woerter, ( wort ) =>{
   return { wort: wort, laenge: wort.length};
 console.log('2. Ausgabe: ' + JSON.stringify(map));
}();
// obsolete Klammern entfernt: Parameterliste. Body und
// return Schlüsselwort entfernt, da an letzter Stelle
// => compile Fehler weil das Ergebnisobjekt als Body
// interpretiert wird !!!
//const berechneWortlaenge = function(){
// const map = processArray(woerter, wort =>
// { wort: wort, laenge: wort.length}
//);
// console.log('3. Ausgabe: ' + JSON.stringify(map));
// Obsolete Klammern zur Umwandlung in Expression hinzugefügt
const berechneWortlaenge = function(){
 const map = processArray(woerter, wort =>
  ({ wort: wort, laenge: wort.length})
 console.log('4. Ausgabe: ' + JSON.stringify(map));
}();
```

```
// Lambdas sind so eine schöne Sache aber diese
ständige Verkürzung
// soweit wie geht ist einfach nervig und birgt
// Zwei Klammern für die Parame ter liste und zwei
für den Body kann man
// doch schreiben oder?
// Sie sind anderer Meinung? - is ok.
// Ich nehme an Sie wissen, dass Pfeilf unk tionen
rechts assoziativ sind.
// Trotzdem glaube ich nicht, dass Sie das hier
lesen wollen:
curry = f \Rightarrow a \Rightarrow b \Rightarrow f(a, b)
uncurry = f \Rightarrow (a, b) \Rightarrow f(a)(b)
papply = (f, a) \Rightarrow b \Rightarrow f(a, b)
Quelle: https://b log.be nes tud io.c o/ cur ryi -
ng- in- jav asc rip t-e s6- 540 d2a d09400
// Anderes und hoffen tlich besseres Beispiel
// Nur zur Vorber eitung
const woerter = ['Hallo', 'heute', 'geste rn'];
const proces sArray = function( items, fn) {
    const result = [];
    ite ms.f or Each( (item) => result.pu sh( fn( -
ite m)));
    return result;
```



By **Huluvu424242** (FunThomas424242)

cheatography.com/funthomas424242/ github.com/Huluvu424242 Published 28th April, 2019. Last updated 2nd July, 2019. Page 1 of 4. Sponsored by CrosswordCheats.com Learn to solve cryptic crosswords! http://crosswordcheats.com



# Javascript Grundlagen Cheat Sheet by Huluvu424242 (FunThomas424242) via cheatography.com/19991/cs/19484/

Javascript - Op	peratoren
a + b	Addition von a und b
a - b	Subtraktion: subtrahiert b von a
a * b	Multiplikation von a und b
x ** y	Exponentialfunktion: x hoch y (wobei sowohl x als auch y variabel sein können - leider keinen korrekten mathematischen Namen gefunden: Potenzfunktion wäre fester Exponent, also auch nicht korrekt.)
a/b	Division: a durch b
a % b	Modulus: a modulo b
a++	Inkrement: erhöht a um 1
a	Dekrement: verringert a um 1
==	Gleichheitsprüfung nur auf Wertebasis (verschiedene aber wertgleiche Datentypen ergeben true)
===	Gleichheitsprüfung inklusive Werte- und Typgle- ichheit
!=	Ungleich ohne Typberücksichtigung
!==	Ungleich mit Typberücksichtigung
>	größer als
<	kleiner als
>=	größer gleich
<=	kleiner gleich
?	Ternärer Operator: (bedingung) ? ifAnweisung : elseAnweisung;
&&	logisches UND
	logisches ODER
!	logisches NOT
typeof <a></a>	Bestimmt den Typ der Variablen a
o instanceof t	true falls o ein Objekt vom Typ t ist
&	bitweises UND
	bitweises ODER
~	bitweises NOT
٨	bitweises XOR

Javascript - Operatoren (cont)		
<<	bitweises links Schieben; mit 0 auffüllend	
>>	bitweises rechts Schieben; vorzeichenerhaltend (so die Theorie)	
>>>	bitweises rechts Schieben; mit 0 auffüllend	

Quelle: https://www.w3schools.com/js/js\_operators.asp

# Javascript - Datentypen

In Javascript existieren 5 primitive Datentypen:

- \* String = Zeichenkette
- \* Number = Ziffernfolge
- \* Boolean = Boolscher Datentyp
- \* Null = leere Referenz
- \* Undefined = Nicht vorhandener Wert

Quelle: https://www.ab-heute-programmieren.de/js-teil-3-datentype-n-und-hoisting/

### Javascript Konzept: Hoisting

```
Beim Hoisting werden später definierte Variablen
bereits am Scopebeginn (für den Entwickler
unsichtbar) automatisch deklariert. Dadurch können
Wertzuweisungen auf eine Variable erfolgen, welche
laut Quellkode noch gar nicht existiert.
*Beispiel
num = 6;
consol e.l og( num); // Gibt 6 zurück
var num = 7;
consol e.l og( num); // Gibt 7 zurück
Hoisting gibt es auch für Funktionen
Beispiel
catNam e("C hlo e");
function catNam e(name) {
    con sol e.l og( "Der Name meiner Katze ist " \pm
name);
}
/*
Das Ergebnis des Codes oben ist: "Der Name meiner
Katze ist Chloe"
*/
```



By **Huluvu424242** (FunThomas424242)

cheatography.com/funthomas424242/

github.com/Huluvu424242

Published 28th April, 2019. Last updated 2nd July, 2019. Page 2 of 4. Sponsored by CrosswordCheats.com Learn to solve cryptic crosswords! http://crosswordcheats.com



# Javascript Grundlagen Cheat Sheet by Huluvu424242 (FunThomas424242) via cheatography.com/19991/cs/19484/

### Javascript Konzept: Hoisting (cont)

> Wichtig: Es werden nur die Deklarationen gehoisted, nicht die Zuweisungen.

#### Beispiele

```
console.log(num); // Gibt undefined zurück
var num;
num = 6;
var x = 1; // Initialisiere x
console.log(x + " " + y); // '1 undefined'
var y = 2; // Initialisiere y
// Das obige Beispiel wird implizit als das folgende verstanden:
var x; // Deklariere x
var y; // Deklariere y
// Hoisting beendet.
x = 1; // Initialisiere x
console.log(x + " " + y); // '1 undefined'
y = 2; // Initialisiere y
```

Quelle: https://developer.mozilla.org/de/docs/Glossary/Hoisting

#### Javascript - Typische Datentypkonvertierungen

25 + " Zahl -> String, Die Zahl 25 wird zu '25'

'25' - 0 String -> Zahl, Der Text '25' wird zur Zahl 25

!!a any -> boolean, Der Wert der Variablen a wird zum
Wahrheitswert: true falls a einen Wert enthält und false falls

# Umwandlung einer Promise Callback Hölle

# Umwandlung einer Promise Callback Hölle (cont)

```
.then(function(newResult) {
    return doThirdThing(newResult);
})
.then(function(finalResult) {
    console.log('Got the final result: ' + finalResult);
})
.catch(failureCallback);
Der alte Stil kann in den neuen Stil überführt werden, wenn es nur darum geht, dass die inneren Promises nach den äußeren ausgeführt werden und die äußeren Variablen in den inneren Promises keine Rolle mehr spielen.
```

https://developer.mozilla.org/de/docs/Web/JavaScript/Guide/Usin-g\_promises

# Overloading von Funktionssignaturen

```
export class Test{
        public add( a: Number, b:Numb er) :void;
        public add( a: String, b:Stri ng) :void;
        public add( a: boolean, b:bool -
ean ):void;
        public add( a, b):void{
               con sol e.l og( a+b);
        public static run():any{
               let t:Test = new Test();
               t.a dd( 'a' ,'b');
                t.a dd(3,5);
                t.a dd( tru e,t rue);
Test.r un();
Erzeugt die Ausgaben:
8
2
```



By **Huluvu424242** (FunThomas424242)

Published 28th April, 2019. Last updated 2nd July, 2019. Page 3 of 4. Sponsored by CrosswordCheats.com Learn to solve cryptic crosswords! http://crosswordcheats.com

cheatography.com/funthomas424242/ github.com/Huluvu424242