

Object / Array Destructuring, Spread Operator für Arrays und Objects, Shorthand Property

MI-Proj-B/M – Gruppe 1
Wintersemester 2020/2021
Fakultät Wirtschaftsinformatik und Angewandte Informatik



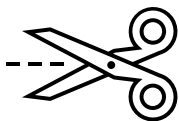
Object / Array Destructuring - Überblick



Einfacher Zugriff auf einzelne Elemente eines Arrays oder Objektes



Standardwerte für bestimmte Elemente innerhalb eines Arrays oder Objektes setzen



Initialisierung von Variablen können von ihrer Deklaration separiert werden



Treffer eines regulären Ausdrucks behandeln



Object / Array
Destructuring

Spread
Operator

Shorthand
Property

Zugriff auf Objekte im Array

```
1  const animals = ['Dog', 'Cat', 'Turtle', 'Bunny', 'Hamster'];
2
3  const first = animals[0];
4  const second = animals[1];
5
6  console.log(first);
7  console.log(second);
```

Dog

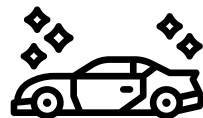
Cat



```
1  const animals = ['Dog', 'Cat', 'Turtle', 'Bunny', 'Hamster'];
2
3  const [first, second] = animals;
4  console.log(first);
5  console.log(second);
```

Dog

Cat



Object / Array
Destructuring

Spread
Operator

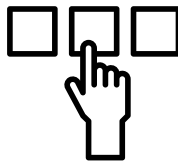
Shorthand
Property

Zugreifen auf einzelne Elemente eines Arrays bequem in einer Zeile möglich

Auswahl und Auslassen von Elementen

```
1  const animals = ['Dog', 'Cat', 'Turtle', 'Bunny', 'Hamster'];
2
3  const [first, , third] = animals;
4  console.log(first);
5  console.log(third);
```

Dog
Turtle



Den nicht-gewollten Elementen wird kein Variablenname zugewiesen.

Durch Kommata wird klar, wie viele Elemente ausgelassen werden sollen.

Object / Array
Destructuring

Spread
Operator

Shorthand
Property

Separate Deklaration und Initialisierung

```
1  let dog, cat;
2
3  [dog, cat] = ['Dog', 'Cat'];
4
5  console.log(dog);
6  console.log(cat)
```

Deklaration

Initialisierung

Dog
Cat

Durch Destructuring kann einer Variable ihr Wert getrennt von ihrer Deklaration zugewiesen werden.

Object / Array
Destructuring

Spread
Operator

Shorthand
Property

Destructuring in einer Methode

```
1 function returnName(firstName, lastName) {  
2   |   return [firstName, lastName, firstName + " " + lastName];  
3 }
```

```
4 const fullName = returnName ('Max', 'Mustermann');  
5 console.log(fullName);
```

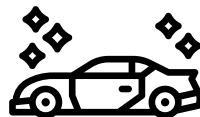
```
► (3) ["Max", "Mustermann", "Max Mustermann"]
```

```
4 const [first, last, both] = returnName('Max', 'Mustermann');  
5 console.log(both);  
6 console.log(first);  
7 console.log(last);
```

Max Mustermann

Max

Mustermann



Object / Array
Destructuring

Spread
Operator

Shorthand
Property

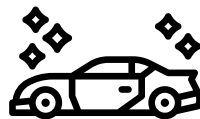
Destructuring in einer Methode (cont'd)

```
1 function returnName(firstName, lastName) {  
2   |   return [firstName, lastName, firstName + " " + lastName];  
3 }  
4  
5 const [first, last, both] = returnName('Max', 'Mustermann');  
6 console.log(both);  
7 console.log(first);  
8 console.log(last);
```

Max Mustermann

Max

Mustermann



Object / Array
Destructuring

Spread
Operator

Shorthand
Property

- Funktion, die einen Array aus den Input-Parametern erstellen soll
- Beim Aufrufen der Methode können Parameter wieder entweder alle oder nur einzelne Elemente zugegriffen werden.

Defaults setzen

```
1 function personInfo(name, hobby) {  
2   |   return [name, hobby];  
3 }  
4  
5 const [name, hobby = 'Nichts tun'] = personInfo('Max');  
6  
7 console.log(name);  
8 console.log(hobby);
```

```
Max  
Nichts tun
```

```
1 function personInfo(name, hobby) {  
2   |   return [name, hobby];  
3 }  
4  
5 const [name, hobby = 'Nichts tun'] = personInfo('Max', 'Netflix');  
6  
7 console.log(name);  
8 console.log(hobby);
```

```
Max  
Netflix
```

- Methode bekommt nur einen Input-Parameter statt zwei
- Der zweite Parameter, der ausgegeben werden soll, wird mit dem Default-Wert belegt

Object / Array
Destructuring

Spread
Operator

Shorthand
Property

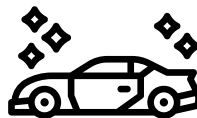
Attribute im Objekt zugreifen

```
1  const cat = {  
2    type: 'Cat',  
3    sound: 'Meow'  
4  }  
5  
6  const type = cat.type;  
7  const sound = cat.sound;  
8  console.log(`${type} says ${sound}.`);
```

```
1  const cat = {  
2    type: 'Cat',  
3    sound: 'Meow'  
4  }  
5  
6  const { type, sound } = cat;  
7  console.log(`${type} says ${sound}.`);
```



Cat says Meow.



- Nutzung von geschweiften Klammern für Objekte
- Zugriff auf Attribute eines Objektes in einer Zeile möglich
- Variablen sollten so heißen, wie Attribute, die man zugreifen will

Object / Array
Destructuring

Spread
Operator

Shorthand
Property

Attribute im Objekt zugreifen (cont'd)

```
1  const cat = {  
2    type: 'Cat',  
3    sound: 'Meow'  
4  }  
5  
6  const { type: animalType, sound} = cat;  
7  console.log(`${animalType} says ${sound}.`);
```

Max is Cat

Falls die Variable nicht so heißen soll, wie das Objekt-Attribut, ist ein Zugriff unter einem anderen Namen mit Doppelpunkt möglich.

Object / Array
Destructuring

Spread
Operator

Shorthand
Property

Defaults bei Object Destructuring

```
1  const cat = {  
2    |    type: 'Cat'  
3  };  
4  
5  const {type, toy = 'wool'} = cat;  
6  console.log(`${type} plays with ${toy}.`);
```

Cat plays with wool.

```
1  const dog = {  
2    |    type: 'Dog',  
3    |    toy: 'ball'  
4  };  
5  
6  const {type, toy = 'wool'} = dog;  
7  console.log(`${type} plays with ${toy}.`);
```

Dog plays with ball.

Kein Attribut mit dem Namen *toy* im Objekt
-> Ersetzen durch default Value

Attribut mit dem Namen *toy* im Objekt
-> Wert des Objektes wird ausgegeben

Object / Array
Destructuring

Spread
Operator

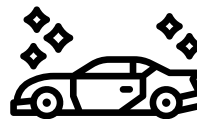
Shorthand
Property

Object Destructuring in Funktionen

```
14 function printAnimal(animal) {  
15   console.log(`The ${animal.type} makes ${animal.sound}.`);  
16 }  
17  
18 printAnimal(cat);
```



```
14 function printAnimal({type, sound}) {  
15   console.log(`The ${type} makes ${sound}.`);  
16 }  
17  
18 printAnimal(cat);
```



The Cat makes Meow.

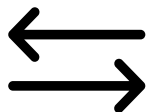
- gezielter Zugriff auf einzelne Attribute des Objektes möglich
- Attribute müssen nicht mehr mit Punktnotation referenziert werden

Object / Array
Destructuring

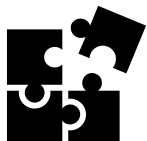
Spread
Operator

Shorthand
Property

Spread Operator - Überblick



Arrays oder Objekte vollständig kopieren



Erweitern von Strings um die Werte eines Objektes oder Arrays



Verknüpfen von mehreren Arrays oder Objekten



! Iterierbarkeit ist Voraussetzung !



Object / Array
Destructuring

Spread
Operator

Shorthand
Property

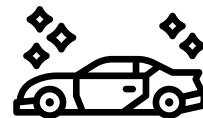
Ersetzen von .apply() durch Funktionsaufrufe

```
1 function myFunction(x, y, z) { }  
2 var args = [0, 1, 2];  
3 myFunction.apply(null, args);|
```



```
1 function myFunction(x, y, z) { }  
2 var args = [0, 1, 2];  
3 myFunction(...args);|
```

```
1 function myFunction(v, w, x, y, z) { }  
2 var args = [0, 1];  
3 myFunction(-1, ...args, 2, ...[3]);|
```



Object / Array
Destructuring

Spread
Operator

Shorthand
Property

Erweitern und Zusammenfügen von Arrays

```
1 var parts = ['Schultern', 'Knie'];  
2 var lyrics = ['Kopf', ...parts, 'und', 'Zehen'];  
3 console.log(lyrics);
```

► (5) ["Kopf", "Schultern", "Knie", "und", "Zehen"]



```
1 var arr1 = [0, 1, 2];  
2 var arr2 = [3, 4, 5];  
3 var both = [...arr1, ...arr2];  
4 console.log(both);
```

► (6) [0, 1, 2, 3, 4, 5]



Object / Array
Destructuring

Spread
Operator

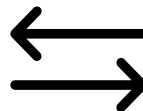
Shorthand
Property

Kopieren von Arrays

Durch die Kopie wird mit `.push()` nicht das Ursprungs-Array verändert.

```
1 var arr1 = [0, 1, 2];
2 var arr2 = [...arr1];
3 arr2.push(4);
4 console.log(arr1);
5 console.log(arr2);
```

```
► (3) [0, 1, 2]
► (4) [0, 1, 2, 4]
```



Object / Array
Destructuring

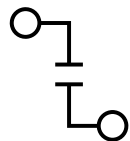
Spread
Operator

Shorthand
Property

Shorthand Property



Syntactic Sugar, um das Anlegen von Objects kompakter zu machen



Prinzip: Wenn die Keys gleich heißen wie die Properties, reicht eine einfache Erwähnung



Auch Funktionen innerhalb von Objekten können kürzer geschrieben werden



Object / Array
Destructuring

Spread
Operator

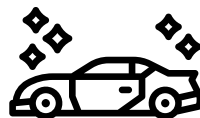
Shorthand
Property

Beispiel für Shorthand Property

```
1 function addPerson(surname, lastname, birthyear) {  
2     return {  
3         surname: surname,  
4         lastname: lastname,  
5         birthyear: birthyear,  
6         getAge: function() {  
7             return 2020 - birthyear;  
8         }  
9     }  
10 }
```



```
1 function addPerson(surname, lastname, birthyear) {  
2     return {  
3         surname,  
4         lastname,  
5         birthyear,  
6         getAge () {  
7             return 2020 - birthyear;  
8         }  
9     }  
10 }
```



Object / Array
Destructuring

Spread
Operator

Shorthand
Property

Quellen

- <https://www.youtube.com/watch?v=FtLRx14wl9s>
- <https://ui.dev/shorthand-properties/>
- <https://www.freecodecamp.org/news/array-and-object-destructuring-in-javascript/>
- <https://www.freecodecamp.org/news/array-destructuring-in-es6-30e398f21d10/>
- <https://www.youtube.com/watch?v=NIq3qLaHCIs>

Object / Array
Destructuring

Spread
Operator

Shorthand
Property

Quellen (cont'd)

- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Spread_syntax
- <https://www.geeksforgeeks.org/javascript-spread-operator/>
- <https://ui.dev/shorthand-properties/>
- <https://eslint.org/docs/rules/object-shorthand>

Object / Array
Destructuring

Spread
Operator

Shorthand
Property

Icons

- Old Car Icon: https://www.flaticon.com/free-icon/old-car_1819538?term=old%20car&page=1&position=1&related_item_id=1819538
- New Car Icon: https://www.flaticon.com/free-icon/new-car_3366177?term=new%20car&page=1&position=2&related_item_id=3366177
- Youtube Icon: https://www.flaticon.com/free-icon/youtube_1384060?term=youtube&page=1&position=3&related_item_id=1384060
- Rack-wheel Icon: https://www.flaticon.com/free-icon/settings_2092297?term=default&page=1&position=9&related_item_id=2092297
- Code Icon: https://www.flaticon.com/free-icon/coding_711284?term=code&page=1&position=4&related_item_id=711284
- Selection Icon: https://www.flaticon.com/free-icon/choose_3860283?term=select&page=1&position=11&related_item_id=3860283

Object / Array
Destructuring

Spread
Operator

Shorthand
Property