

JavaScript 101: Grundlagen

Object

```
var map = {  
  feld1: 'Huhu',  
  zweites$Feld: "Auch sowas geht!"  
};  
console.log(typeof map === "object"); // true  
console.log(map.feld1); // Huhu  
console.log(map["zweites$Feld"]); // Auch sowas geht!  
  
map.hund = "Ganz neu geht auch";  
map.f = function() { return "Aha!" };  
  
delete map.hund; // löschen  
console.log(map.hund); // undefined
```

Array

```
var array = ["a", "b", "c"];
var el = array[2];
array[1] = 20;
typeof array === "object";
// fügt die 4 am Ende hinzu
array.push(4);
// An Position 1 werden 2 Elemente entfernt und zurück gegeben
array.splice(1, 2);
// An Position 1 werden 0 Elemente entfernt und zurück gegeben
// Zudem wird an Position 1 "x" hinzugefügt
array.splice(1, 0, "x");
```

Typen

```
var string = "String";  
typeof string === "string";
```

```
var int = 1;  
typeof int === "number";
```

```
var float = 1.0;  
typeof float === "number";
```

```
var bool = true;  
typeof bool === "boolean";
```

```
var func = function() {};  
typeof func === "function";
```

```
typeof michGibtEsNicht === "undefined";
```

string

```
var s1 = 'Hallo, ';  
var s2 = "Olli's Oma";  
  
var s3 = s1 + s2;  
s3 === "Hallo, Olli's Oma";  
  
s3[1] === "a";  
s3.charAt(1) === s3[1];  
  
// kann viel schneller sein als +  
var builder = ["a", "b", "c"];  
var s4 = builder.join("");  
s4 === 'abc';
```

Kontrollstrukturen

Iterieren über Array-Elemente

```
for (var i=0; i < array.length; i++) {  
    console.log(i + ": " + array[i]);  
}
```

```
// Achtung: Iteriert über Indizes, nicht über die Elemente!  
for (var i in array) {  
    console.log(i + ": " + array[i]);  
}
```

Kontrollstrukturen ansonsten wie in Java

- if / else
- while / do
- switch
- break / continue
- [Referenz](#)

Übung 1: Basiswissen

- Erzeuge ein Array mit Personen-Objekten mit mindestens den Eigenschaften
 - Name
 - Alter
 - Geschlecht
- Gib alle Personen nacheinander aus
 - Erzeuge aus den Daten einen string, der dann mit `console.log` ausgegeben wird
 - Benutze dafür beide Varianten der `for`-Schleife

Funktionen

```
var f1 = function(p1, p2) {  
    return p1 + p2;  
};  
var result1 = f1(1,2);  
result1 === 3;  
  
function f2() {  
    console.log("Called!");  
}  
var result2 = f2();  
result2 === undefined;
```

Arrays revisited

```
var array = [1,2,3];  
  
array.forEach(function(e) { console.log(e); });  
  
array.filter(function(e) { return e > 2; });  
  
array.map(function(e) { return e + 100; });  
  
array.reduce(function(reduced, current) {  
    return reduced + current;  
}, 100);
```

Optionale Parameter

```
function f1(p1) {  
    if (typeof p1 === 'undefined') {  
        return null;  
    } else {  
        return p1;  
    }  
}
```

```
var result1 = f1(1);  
console.log(result1 === 1);
```

```
var result2 = f1();  
console.log(result2 === null);
```

Varargs #1

```
function f2() {  
    // "arguments" enthält immer alle Argumente der Funktion  
    if (typeof arguments[0] === 'undefined') {  
        return null;  
    } else {  
        return arguments[0];  
    }  
}  
  
var result3 = f2(1);  
console.log(result3 === 1);  
  
var result4 = f2();  
console.log(result4 === null);
```

Varargs #2

```
function summe() {  
    var sum = 0;  
    for (var a in arguments) {  
        sum += arguments[a];  
    }  
    return sum;  
}
```

```
var result5 = summe(1,2,3);  
console.log(result5 === 6);
```

Scopes

So nicht!

```
{  
    var huch = "Ich bin noch da";  
}  
  
console.log(huch); // Ich bin noch da
```


So!

```
(function () {  
    var achso = "Ich bin weg";  
})();  
  
console.log(achso); // ReferenceError
```

Immediately-Invoked Function Expression (IIFE)

Exception Handling

```
try {  
    throw "Error";  
    // or  
    throw new Error("Error");  
} catch (e) {  
    console.log("Caught: " + e)  
} finally {  
    console.log("Always")  
}
```

Übung 2: Mehr JavaScript Grundlagen

Funktionen und Exceptions kennen lernen

Schreibe den Code der vorherigen Übung auf die Verwendung von Funktionen und Exceptions um

Schritt 1: Funktionen

1. Erstelle eine Funktion, die eine Person formatiert auf der Console ausgibt
2. Benutze `forEach` auf dem Array der Personen um dieselben Aufgaben wie in Übung 1 zu lösen
3. Du solltest hier keine `for`-Schleife mehr haben

Optionaler Schritt 2: Exceptions

1. Erweitere das Programm um eine Funktion zum überprüfen von Personen. Kann jemand über 200 Jahre alt sein?
2. Die Funktion soll im Fehlerfall eine Exception schmeißen
3. Erweitere dein Array um Methusalem oder eine andere geeignete Person
4. Rufe deine Check-Funktion nacheinander auf allen Elementen des Arrays auf
5. Umschließe den Aufruf mit `try...catch...finally` und gib eine gefangene Exception aus

Optionaler Schritt 3: Map/Reduce

1. Bilde das Array von Personen mittels `map` auf ihr Alter ab
2. Nutze `reduce`, um daraus das Durchschnittsalter aller Personen zu berechnen