

JavaScript – Objekte

Inhalt:

- Theorie
- Fertige Objekte z.B. Math.Pi und Math.round
- eigenes Objekt erstellen
- Date-Objekt

Ein Objekt ist ein Container für Werte in Form von **Schlüssel-Wert-Paaren** (key-value pairs) und somit eine Datenstruktur zum Speichern vieler Eigenschaften in einer Variable

- Der **Schlüssel** (key) ist immer ein String (oder Symbol),
- der **Wert** kann alles sein: eine Zahl, ein String, ein anderes Objekt, ein Array oder eine Funktion.

Warum sind Objekte wichtig?

- Alles in JavaScript ist ein Objekt oder kann wie eines behandelt werden.
- Sie bilden die Grundlage für komplexe Datenstrukturen.
- Sie ermöglichen die Erstellung von Instanzen mit gemeinsamem Verhalten (z. B. in OOP). OOP = objektorientiertes Programmieren
- JSON (JavaScript Object Notation) basiert auf Objekt-Syntax.

Beispiel für ein Objekt: Datenstruktur zum Speichern vieler Eigenschaften in einer Variable

```
11 let person = {
12     firstname: "Patrick",
13     lastname: "Spongebob",
14     hometown: "Digbiztown",
15     birth: "2017",
16 };
```

Objekte besitzen normalerweise Eigenschaften und Methoden.

Objekte haben eine eigene Notation, also Schreibweise, beim Aufruf:

- Eine Methode wird immer mit dem Namen des Objekts,
- einem Punkt als Trennzeichen und
- dem Namen der Funktion (bzw. der Methode) aufgerufen.

Beispiel: Math.PI
 Math.round()

Math.PI ist in JavaScript eine **vordefinierte Konstante** des eingebauten Math-Objekts und stellt den **Wert der Kreiszahl π (Pi)** dar – also ungefähr:

```
Math.PI ≈ 3.141592653589793
```

- `Math` ist ein **globales Objekt** in JavaScript, du brauchst also nichts extra zu importieren.
- `PI` ist eine **Konstante**, keine Funktion – also kein `Math.PI()`, sondern einfach `Math.PI`.

Fertige Objekte

Es gibt feste, statische, fertige Objekte, auf die man zugreifen kann, z.B.

- „Math“.
Math hat viele unterschiedliche Methoden (Funktionen), z.B. `Math.PI` oder `Math.round`

Achtung: das M bei Math ist immer groß!

Beispiele des vorhandenen und fertigen Objektes „Math“:

```
8 <body>
9   <script>
10     let pi = Math.PI
11     pi = Math.round(pi);
12
13     document.write(pi);
14
15   </script>
16 </body>
17 </html>
```

```
let pi = Math.PI
pi = Math.round(pi);
document.write(pi);
```

Ergebnis: 3

- In diesen drei Zeilen wird die Eigenschaft „PI“ des Math-Objektes abgerufen und in einer Variablen abgespeichert.
- Anschließend wird diese mit der Funktion „Math.round()“ gerundet und
- per „document.write()“ ausgegeben.

Folgende Objekte mit ihren Eigenschaften und Methoden sind ein fester Bestandteil der JavaScript-Syntax. Es gib, je nach JavaScript-Implementierung, 30 bis 40 feste Objekte, auf die man zurückgreifen kann.

Die wichtigsten sind:

- `document`: enthält Infos über das HTML-Dokument

- Date: kann die aktuelle Zeit und Datum zurückgeben
- Math: kann gut rechnen
- Array: eine Kette von mehreren Variablen gleichen Typs

Erstellen: Ein Objekt kann man erstellen mit

- var
- let
- const

eigenes Objekt erstellen

Erstelle ein neues Dokument namens „objekt.html“.

1. Variante 1 – wird kaum genutzt: Ein neues Objekt wird mit „newObject();“ deklariert. Man muss jedoch auf Groß- und Kleinschreibung achten. Leerzeichen und Sonderzeichen sind in Namen und Eigenschaften nicht erlaubt.

2. Variante 2: erstellen mit {}

innerhalb der geschwungenen Klammern können Eigenschaften definiert werden. Dies erfolgt mit „key – value“ Paaren. Gibt es mehr als eine, muss ein Komma dazwischenstehen.

z.B.:

```
1 let Max = {
2   gröÙe: 180
3 };
```

```
11 let person = {
12   firstname: "Patrick",
13   lastname: "Spongebob",
14   hometown: "Digbiztown",
15   birth: "2017",
16 };
```

Übung Variante 2: Hier wird die 2. Art der Schreibweise genutzt, nämlich {}:

```
let person = {

  firstname: "Patrick",
  lastname: "Spongebob",
  hometown: "Digbiztown",
  birth: "2017"

};
```

Häufige Anwendung – es soll nur eine Eigenschaft ausgegeben werden

obwohl mehrere Eigenschaften vorhanden sind.

Dafür schreibt den Namen der Variable (hier: person) und mit einem Punkt verknüpft die Eigenschaft danach.

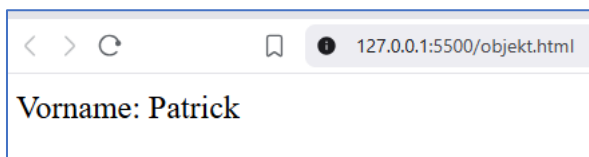
Lösung: `document.write(person.firstname);`

Beispiel:

```
<script>
let person = {
  firstname: "Patrick",
  lastname: "Spongebob",
  hometown: "Digbiztown",
  birth: "2017"
};
document.write("Vorname: " + person.firstname + "<br>");
</script>
```

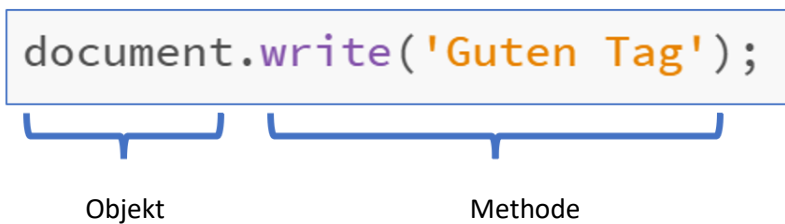
```
8 <body>
9   <script>
10     let person = {
11       firstname: "Patrick",
12       lastname: "Spongebob",
13       hometown: "Digbiztown",
14       birth: "2017",
15     };
16
17     document.write("Vorname: " + person.firstname + "<br>");
18
19   </script>
```

Ergebnis:



Auch das ist ein Objekt:

Beispiel mit `document.write()`: zeigt beispielhaft einen Aufruf einer Methode für ein Objekt.



Bei **`document.write()`** handelt es sich um eine Methode des `document`-Objekts.

Die Funktion `document.write()` gibt den in Klammern angegebenen Wert in diesem Dokument aus.

- Eine Eigenschaft ist wie eine Variable, die einem Objekt zugewiesen wird. Sie beschreibt das Objekt.
- Eine Methode ist wie eine Funktion, die aber zu einem Objekt gehört. Sie beschreibt Dinge, die das Objekt machen kann.

Date-Objekt

Der Konstruktor „`Date()`“ teilt dem JavaScript-Interpreter mit, dass die Variable ein Datum ist. Dadurch wird es möglich, die Methode des `Date`-Objektes zu verwenden, um Datum und Uhrzeit abzurufen.

```
var today = new Date();
```

BEACHTEN: `Date` mit einem großen „D“

Date und Time

Wenn man ein `Date`-Objekt erstellt, enthält es **standardmäßig** das aktuelle Datum und die aktuelle Uhrzeit. Soll es eine andere Zeitangabe speichern, muss man es ausdrücklich angeben.

Die Methode „`toString()`“ zeigt das Datum im Format „Wed Apr 16 2016“ an. Wenn man ein anderes Format wünscht, kann man es mit folgenden Methoden aus Tag, Monat und Jahr zusammenstellen:

- `getDate()` gibt den Tag im Monat (1-31) zurück
- `getDay()` gibt den Wochentag (0-6) zurück
- `getFullYear()` gibt das Jahr (4 Stellen) zurück
- `getHours()` gibt die Stunden (0-23) zurück
- `getMinutes()`

- `getMonth()` gibt den Monat (0-11) zurück, Januar ist 0
- `getTime()` Anzahl der Millisekunden seit 1. Januar 1970

lesbare Strings:

- `toDateString()` gibt das Datum als lesbaren String zurück
- `toTimeString()` gibt die Uhrzeit als lesbaren String zurück

Beispiel: Man muss die Details mit Punktnotation anhängen, z.B. `today.getFullYear()`;

Da die Tages- und Monatsnamen in den einzelnen Sprachen anders heißen, werden sie nicht im Date-Objekt gespeichert. Stattdessen werden für die Wochentage die Zahlen 0-6 und für die Monate die Zahlen 0-11 verwendet. Um die Namen anzuzeigen, muss man sie in ein Array aufnehmen.

Beispiel:

Schreibe in der Datei „1.datum2.html“ folgenden Code

BEACHTET: Date mit einem großen „D“

Die Erstellung von Variablen wird anschließend genauer erklärt.

```

16 <script>
17 var today = new Date(); //Zwischenspeicher in eine neue Variable
18 var year = today.getFullYear();
19 document.write("<p>Copyright &copy; " + year + "</p>");
20 </script>
21

```

Ergebnis:

```

Die Webseite wurde zuletzt am
Fri Feb 10 2017 11:51:10 GMT+0100Josef
aktualisiert.
Copyright © 2017

```

Übungen:

a) Erstelle eine Ausgabe eines lesbaren Strings, nämlich dem Datum:

```
22 <h2>Übung: Datum </h2>
23 <script>
24 var today2 = new Date();
25 var datum = today2.toDateString();
26 document.write("<p>Das Datum lautet: " + datum + "</p>");
27 </script>
28 </body>
```

Ergebnis:

Übung: Datum

Das Datum lautet: Sun Feb 12 2017

b) Tagesausgabe: Tag ausgeben

```
29 <h2>Übung: Methode - Tagesausgabe </h2>
30 <script>
31 var today3 = new Date();
32 var tag = today3.getDay();
33 document.write("<p>Das Datum lautet: " + tag + "</p>");
34 </script>
35
```

Ergebnis:

Übung: Methode - Tagesausgabe

Das Datum lautet: 0

c) Um die Namen ausgeben zu können, muss man ein Array anlegen.

Statt „5“ soll also „Freitag“ stehen.

```
29
30 <h2>Übung: Methode - Tagesausgabe mit Wochentag </h2>
31 <script>
32 //Array erstellen
33 var wochentage = ['Sonntag', 'Montag', 'Dienstag', 'Mittwoch',
34 |'Donnerstag', 'Freitag', 'Samstag'];
35 var today4 = new Date();
36 var tag = wochentage[today4.getDay()];
37 document.write("<p>Das Datum lautet: " + tag + "</p>");
38 </script>
39
```

```
//Array erstellen
```

```
var wochentage = ['Sonntag', 'Montag', 'Dienstag', 'Mittwoch', 'Donnerstag', 'Freitag', 'Samstag'];
```

Ergebnis:

Übung: Methode - Tagausgabe mit Wochentag

Das Datum lautet: Sonntag

d)sechs Tage addieren

```
41 <h2>Übung: sechs Tage addieren </h2>
42 <script>
43 //Array erstellen
44 var wochentage = ['Sonntag', 'Montag', 'Dienstag', 'Mittwoch',
45 'Donnerstag', 'Freitag', 'Samstag'];
46 //sieben Tage dazu angegeben in Millisekunden
47 var wocheDazu = new Date(today.getTime()+ 6*24*60*60*1000);
48 var tag = wochentage[wocheDazu.getDay()];
49 document.write("<p>Das Datum lautet: " + tag + "</p>");
50 </script>
51
```

Ergebnis:

Übung: sechs Tage addieren

Das Datum lautet: Samstag

e) Tagesnamen, Tag, Monat und Jahr

```
52
53 <h2>Übung: inkl. Tag, Monat und Jahr </h2>
54 <script>
55 //Array erstellen
56 var wochentage = ['Sonntag', 'Montag', 'Dienstag', 'Mittwoch',
57 'Donnerstag', 'Freitag', 'Samstag'];
58 var monate = ['Jänner', 'Feber', 'März', 'April', 'Mai', 'Juni',
59 'Juli', 'August', 'September', 'Oktober', 'November', 'Dezember'];
60 //sieben Tage dazu angegeben in Millisekunden
61 var today5 = new Date(today.getTime());
62 var tag = wochentage[today5.getDay()];
63 var day = today5.getDate();
64 var monatAusgabe = monate[today5.getMonth()];
65 var jahr = today5.getFullYear();
66 document.write("<p>Das Datum lautet: " + tag + ', ' + day + ' '
67 + monatAusgabe + ' ' + jahr + "</p>");
68 </script>
69
```

<h2>Übung: inkl. Tag, Monat und Jahr </h2>

```
<script>
//Array erstellen
var wochentage = ['Sonntag', 'Montag', 'Dienstag', 'Mittwoch', 'Donnerstag', 'Freitag', 'Samstag'];
var monate = ['Jänner', 'Feber', 'März', 'April', 'Mai', 'Juni', 'Juli', 'August', 'September', 'Oktober',
'November', 'Dezember'];
//sieben Tage dazu angegeben in Millisekunden
var today5 = new Date(today.getTime());
var tag = wochentage[today5.getDay()];
var day = today5.getDate();
var monatAusgabe = monate[today5.getMonth()];
var jahr = today5.getFullYear();
document.write("<p>Das Datum lautet: " + tag + ', ' + day + ' ' + monatAusgabe + ' ' + jahr + "</p>");
</script>
```

Ergebnis:

Übung: inkl. Tag, Monat und Jahr

Das Datum lautet: Montag, 13 Feber 2017

Quelle:

<https://www.youtube.com/watch?v=EenvvRCcVI4>

<https://www.youtube.com/watch?v=cAiSbkWvBQU> ab ca. 56:30 Minuten (Mario,2022)

Florian Franke, Johannes Ippen; in: Apps mit HTML5, CSS3 und JavaScript;
Rheinwerk Verlag, Bonn 2015; S. 106-107

Andy Harris, in: JavaScript für dummies; Wiley-Vch Verlag, 2017, Weinheim; S.128-143