

## Datum- und Zeitberechnung

Hallo du! Ich schreibe dir aus den Tiefen meines Bürostuhls, umgeben von leeren Kaffeetassen und der stillen Verzweiflung eines Bastlers, der dachte: „Datum und Uhrzeit? Pfff, das ist doch einfach.“ **Es war nicht einfach!**

Beginnen wir mit den **Schaltjahren**. Du kennst das sicher: Alle vier Jahre, einmal extra ein Tag — klingt harmlos. Was man mir jedoch verschwiegen hat: Ein Jahr ist nur dann ein Schaltjahr, wenn es durch 4 teilbar ist - außer, wenn es durch 100 teilbar ist - außer wiederum, wenn es durch 400 teilbar ist. Das Jahr 1900? Kein Schaltjahr. Das Jahr 2000? Doch. Das Jahr 2100? Nein.

Dann kamen die **exklusiven**- und **inklusive** Methoden - und damit meine wahre Prüfung. Die Frage ist simpel: Wenn ich die Tage zwischen dem 1. Januar und dem 3. Januar zähle, sind das 2 Tage oder 3? Ich weiß es nicht mehr. Ich wusste es zwischendurch, dann nicht mehr, dann wieder - und danach nie mehr sicher.

Die **inklusive** Methode zählt Anfang UND Ende mit - also quasi: „Ich war Montag, Dienstag UND Mittwoch krank.“ Drei Tage. Logisch für menschliche Gespräche. Die **exklusive** Methode zählt Anfang und Ende nicht! Das Problem? Niemand - wirklich niemand - sagt dir im Voraus, welche Methode gerade gilt. Man entdeckt es, wenn der Buchungskalender behauptet, man habe fünf Nächte im Hotel verbracht, obwohl man Check-in am Montag und Check-out am Freitag hatte. Vier Nächte, sagst du? Fünf, sagt mein Buchungskalender. Wie verhält sich das mit Urlaubstagen? Ach Gott ach Gott!

- Tage zwischen zwei Daten berechnen.
- Zeiteinheiten Umrechner.
- Datum addieren.
- Schaltjahr Rechner.

### Wie geht JavaScript mit Datum und Zeit um?

JavaScript berechnet Datum und Zeit primär über das eingebaute Date-Objekt, welches auf der Anzahl der Millisekunden basiert, die seit dem 1. Januar 1970 (UTC) vergangen sind (der sogenannten Unix-Epoche).

Aktuelles Datum → `let jetzt = new Date();`

Spezifisches Datum → `let spezifisch = new Date(„2023-10-14T10:00:00“);`

Wichtige Methoden zur Datenextraktion: Du kannst einzelne Komponenten des Datumsobjekts abfragen:

`getFullYear();`: Jahr (4-stellig)

`getMonth();`: Monat (0-11, 0 ist Januar!)

`getDay();`: Tag (0-6, 0 ist Sonntag)

`getDate();`: Tag des Monats (1-31)

`getHours();`, `getMinutes();`, `getSeconds();`: Zeitkomponenten

`getTime();`: Zeitstempel in Millisekunden

Umrechnung in Millisekunden:

1 Tag =  $24 * 60 * 60 * 1000$  Millisekunden = 86.400.000 mSek.

1 Stunde = 60 \* 60 \* 1000 Millisekunden = 3.600.000 mSek.

1 Minute = 60 \* 1000 = 60.000 mSek.

1 Sekunde = 1000 Millisekunden

```
8  const tage = ["Sonntag", "Montag", "Dienstag", "Mittwoch", "Donnerstag",
9  |  "Freitag", "Samstag"];

10 const monate = ["Januar", "Februar", "März", "April", "Mai", "Juni",
11 |  "Juli", "August", "September", "Oktober", "November", "Dezember"];

123 const tag1 = tage[date1.getDay()];
124 const monat1 = monate[date1.getMonth()];
```

Zeile 8 & 9: Array für Wochentage. Zeile 123: Ein zuvor erstelltes Date-Objekt „date1“ liefert mit `getDay()` den Wochentag des Datums. Zeile 10 & 11: Array für Monate. Zeile 124: Ein zuvor erstelltes Date-Objekt „date1“ liefert mit `getMonth()` den Monat des Jahres.

#### Plausibilitätsprüfung User-Eingabe.

- Numerische Eingabe, nur Zahlen.
- Eingabe Datum, Jahr: Größer Null.
- Eingabe Datum, Monat: Kleiner 12, größer Null.
- Eingabe Datum, Tag: Prüfung 31, 30, Februar 29 oder 28 Tage, größer Null.
- Eingabe Datum, Differenz Tag: Prüfung 31, 30, Februar 29 oder 28 Tage.
- Eingabe Zeit, Stunden: Kleiner gleich 23.
- Eingabe Zeit, Minuten: Kleiner gleich 59.
- Eingabe Zeit, Sekunden: Kleiner gleich 59.

#### Tage zwischen zwei Daten berechnen.

|                   | Tag                            | Monat                          | Jahr                              |
|-------------------|--------------------------------|--------------------------------|-----------------------------------|
| <b>Datum von:</b> | <input type="text" value="8"/> | <input type="text" value="7"/> | <input type="text" value="1928"/> |

  

|                   | Tag                            | Monat                          | Jahr                              |
|-------------------|--------------------------------|--------------------------------|-----------------------------------|
| <b>Datum bis:</b> | <input type="text" value="8"/> | <input type="text" value="7"/> | <input type="text" value="2026"/> |

  

**Berechnen!**

Vom Sonntag, den 8. Juli 1928 bis Mittwoch, den 8. Juli 2026 ermittelt die Applikation folgende Daten:  
Anzahl Tage: **35793**. Anzahl Jahre: **98**. Anzahl Schaltjahre: **25**.  
**Ohne Gewähr!**

Berechnet die exakte Anzahl der Tage zwischen zwei gegebenen Daten. Dabei werden Kalenderfunktionen wie Schaltjahre und Monatslängen automatisch berücksichtigt.

**Inklusive Berechnung:** Tage(**inkl.**) =  $|D2 - D1| + 1$  → beide Daten eingeschlossen. Üblich bei Urlaubstagen, Ereigniszählungen. **Exklusive Berechnung (Standard):** Tage(**exkl.**) =  $|D2 - D1|$  → ein Datum wird ausgeschlossen. Üblich bei Altersberechnung, Projektlaufzeiten.

#### Detailliertes Rechenbeispiel:

Gegebene Daten:

1. Datum: 5. Februar 2023

2. Datum: 15. Januar 2024

Zeitspanne: 2023 → 2024

Aufschlüsselung:

Verbleibende Tage 2023:  $(28-5) + 31 + 30 + 31 + 30 + 31 + 30 + 31 = 329$  Tage

Tage 2024: 15 Tage

Endresultat:  $329 + 15 = 344$  Tage

Besonderheit: Überspringt das Jahresende 2023/2024

**Interpretation:** Zwischen dem 5. Februar 2023 und dem 15. Januar 2024 liegen exakt 344 Kalendertage.

Das obige Rechenbeispiel in JavaScript (gekürzte Darstellung)

```
const date1 = new Date(Datum_Start);
const date2 = new Date(Datum_Ende);
const diffInMs = Math.abs(date1 - date2);
const diffInTagen = parseInt(diffInMs / (1000 * 60 * 60 * 24));
const diffInJahren = parseInt(diffInTagen / 365); //oder 366
```

#### Zeiteinheiten Umrechner.

Zeitspanne:

Tage

Stunden

Minuten

Sekunden

4

8

30

12

Berechnen!

Tage: 4.35. Stunden: 104.50. Minuten: 6270.20. Sekunden: 376212.  
Ohne Gewähr!

Der Zeiteinheiten Umrechner konvertiert eine eingegebene Zeitspanne in alle verfügbaren Zeiteinheiten. Dabei werden präzise Umrechnungen und mathematische Beziehungen zwischen den Einheiten dargestellt.

| Von       | Nach Sekunden | Nach Minuten | Nach Stunden | Nach Tagen |
|-----------|---------------|--------------|--------------|------------|
| 1 Sekunde | 1             | 0,01667      | 0,000278     | 0,0000116  |
| 1 Minute  | 60            | 1            | 0,01667      | 0,000694   |
| 1 Stunde  | 3.600         | 60           | 1            | 0,04167    |
| 1 Tag     | 86.400        | 1.440        | 24           | 1          |

## Datum addieren.

|               |                                 |                                |                                   |                                |                                |                                |
|---------------|---------------------------------|--------------------------------|-----------------------------------|--------------------------------|--------------------------------|--------------------------------|
|               | Tag                             | Monat                          | Jahr                              | Stunden                        | Minuten                        | Sekunden                       |
| <b>Datum:</b> | <input type="text" value="11"/> | <input type="text" value="2"/> | <input type="text" value="2000"/> | <input type="text" value="0"/> | <input type="text" value="0"/> | <input type="text" value="0"/> |

  

|                  |                                 |                                |                                 |                                 |
|------------------|---------------------------------|--------------------------------|---------------------------------|---------------------------------|
|                  | Tage                            | Stunden                        | Minuten                         | Sekunden                        |
| <b>Zeitraum:</b> | <input type="text" value="60"/> | <input type="text" value="8"/> | <input type="text" value="30"/> | <input type="text" value="12"/> |

  

**Berechnen!**

**Vorheriges Datum:** Freitag, den 11. Februar 2000 0:0:0. **Neues Datum:** Dienstag, den 11. April 2000 9:30:12.  
2000 ist ein Schaltjahr.  
Ohne Gewähr!

## Schaltjahr Rechner.

  

2024 ist ein Schaltjahr. Das Jahr hat 366 Tage – der Februar hat 29 Tage.

Die Erde benötigt für eine komplette Umrundung der Sonne nicht genau 365 Tage, sondern etwa 365 Tage, 5 Stunden, 49 Minuten und 12 Sekunden (ca. 365,2422 Tage).

Das Problem: Würden wir diese ca. 6 Stunden nicht berücksichtigen, würde sich unser Kalender alle vier Jahre um fast einen ganzen Tag verschieben. Nach 100 Jahren wäre der Kalender um etwa 25 Tage verschoben, und Weihnachten würde in den Sommer fallen.

Die Lösung: Alle vier Jahre wird ein Tag hinzugefügt (der 29. Februar), um die angesammelten 24 Stunden (4 Jahre × 6 Stunden) auszugleichen.

**Regel 1:** Ein Jahr ist ein Schaltjahr, wenn es glatt durch 4 teilbar ist.

**Regel 2:** Ist es jedoch auch durch 100 teilbar, ist es kein Schaltjahr (Ausnahme für Jahrhundertwende).

**Regel 3:** Ist es zudem durch 400 teilbar, ist es trotzdem ein Schaltjahr.

```
43
44 //*****
45 // Schaltjahr: Liefert true, wenn das Jahr ein Schaltjahr ist
46 //*****
47 function isLeapYear(year) {
48     return (year % 4 === 0 && year % 100 !== 0) || (year % 400 === 0);
49 }
50
```

Zeile 48: Der Modulo Operator (%) ist in Jeder Programmiersprache vorhanden.

Beispiel:  $10 \% 3$  oder  $10 \bmod(3) = 1 \rightarrow$  da  $3 \times 3 = 9$ , Rest 1.  $12 \% 4$  oder  $12 \bmod(4) = 0 \rightarrow$  da  $4 \times 3 = 12$ , Rest 0. Variable  $i \% 2 = 0 \rightarrow$  Gerade Zahl! Variable  $i \% 2 \neq 0 \rightarrow$  Ungerade Zahl.

**Beispiel Schaltjahr: Jahr 1900** (Jahrhundertregel)  $\rightarrow 1900 \% 4 = 0$  ( $4 \times 475 = 1900$ , Rest 0)  $1900 \% 100 = 0 \rightarrow$  Kein Schaltjahr (**obwohl durch 4 teilbar!**)  $\rightarrow [year \% 100 \neq 0]$ .

**Jahr 2000** (Ausnahme von der Ausnahme)  $\rightarrow 2000 \% 100 = 0 \rightarrow$  Kein Schaltjahr!?  $\rightarrow$  DOCH  $2000 \% 400 = 0 \rightarrow 2000$  ist ein Schaltjahr  $\rightarrow [ (year \% 100 \neq 0) || year \% 400 == 0 ]$ .

Für meine Lernkurve habe ich hier etwas Neues für mich ausprobiert. Vier verschiedene HTML-Seiten mit nur einer JavaScript-Datei.

```
249 //*****
250 // Der Dispatcher-Algorithmus. Ein JavaScript. Mehrere HTML-Seiten!
251 //*****
252 const pageIdentifier = document.body.getAttribute('data-page');
253 switch (pageIdentifier) {
254     case 'schaltjahr':
255         checkLeapYear();
256         break;
257     //case 'contact':
258     //    initContact();
259     //    break;
260     default:
261         console.log("Allgemeine Seite ohne spezifische Logik");
262 }
```

```
21 <body data-page="schaltjahr" class="my1body">
22     <div>
```

Fehlerfrei kann dieses Projekt nicht sein. Ohne Gewähr.

März 2026, Hans Busche