

# JavaScript

---

Luca Berres

Allgemeines zu JavaScript

Syntax

Einbindung JavaScript

Allgemeine Programmierkonzepte

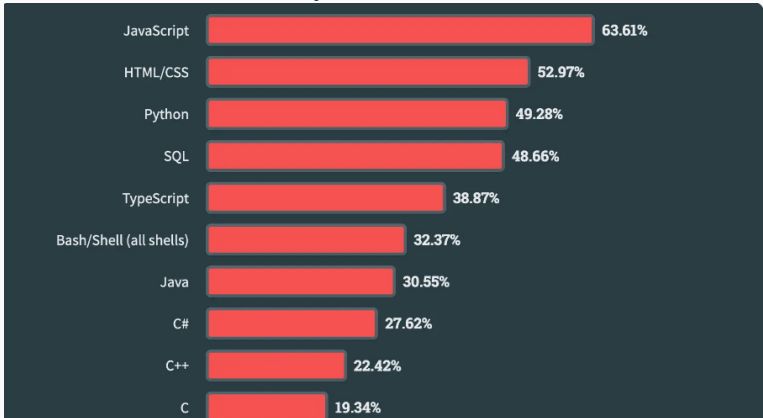
JavaScript im Browser

# Allgemeines zu JavaScript

---

# Allgemeines

- entwickelt 1995 von Brendan Eich um Webseiten mit Interaktion auszustatten
- Eine der beliebtesten Programmiersprachen
- Trotz Namensähnlichkeit nicht mit JAVA verwandt, aber beide orientieren sich von der Syntax an C



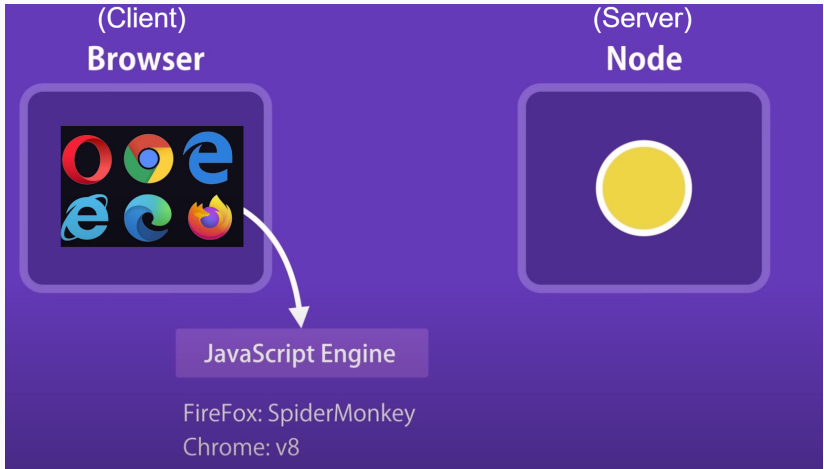
# Wer steht hinter JavaScript

ECMA International (früher: European Computer Manufacturers Association)



**Figure 1:** ECMA

# Wo läuft JavaScript?



**Figure 2:** JavaScript Runtime

# Auf welchen Plattformen läuft JavaScript?

- Server Applikationen-> Node.js
- Desktop Applikationen -> Electron
- Mobile Applikationen -> React Native oder Ionic

# Syntax

---



# Einbindung JavaScript

---

# Eingebettetes im HTML

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="UTF-8" />
    <title>Meine Webseite</title>
  </head>
  <body>
    <h1>Willkommen auf meiner Webseite</h1>
    <script>
      console.log("Hallo Welt");
    </script>
  </body>
</html>
```

## Extern referenziert im HTML

1. Erstelle eine Datei namens script.js mit folgendem Inhalt:

```
alert("Hallo, Welt!");
```

2. Binde die externe Datei in dein HTML-Dokument ein

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="UTF-8" />
    <title>Meine Webseite</title>
  </head>
  <body>
    <h1>Willkommen auf meiner Webseite</h1>
    <script src="script.js"></script>
  </body>
</html>
```

1. Öffne die Entwicklertools in deinem Browser: In Chrome:  
Rechtsklick -> "Untersuchen" -> Tab "Konsole" oder F12
2. Führe JavaScript-Code direkt in der Konsole aus:

```
console.log("Hallo, Welt!");
```

1. Erstelle eine Datei mit folgendem Inhalt und speicher sie als Test.js ab

```
console.log("Hallo, Welt!");
```

2. Öffne sie mit einem Browser
3. Öffne die Konsole wie zuvor gezeigt



# Allgemeine Programmierkonzepte

---





## Übersicht

- speichern Daten temporär
- Analogie: beschrifteter Karton mit Inhalt

<b>Kom- ponente (DE)</b>	<b>Kom- ponente (EN)</b>	<b>Beschreibung</b>	<b>Beispiel</b>
Bezeichner	Identifier	Name der Variable, die ihren Wert bezeichnet.	test
Literal	Literal	Wert, der der Variable zugewiesen wird.	42, "Text"
Schlüssel- wort	Keyword	Reserviertes Wort in der Programmiersprache für Deklaration oder Steuerung.	let, const, var

## Keyword in JavaScript

- var (veraltet, weil globaler scope(Geltungsbereich))
- let (block-scoped -> Geltungsbereich ist eine nächste von geschweiften Klammern umschlossenen Syntaxen, z.B. if statement)
- const (block-scoped, kann nicht nochmals zugewiesen werden)

## Numerische und Boolesche Literals

Typ	Beispiel
Hexadezimale Konstanten	<code>var test = 0x12f</code>
Binäre Konstanten	<code>var test = 0b011101</code>
Oktale Konstanten	<code>var test = 0o767</code>
Ganzzahlenkonstanten	<code>var test = 123456</code>
Gleitkommazahlen	<code>var test = 12.34</code> <code>var test = 12.34e2</code>
Boolesche Konstanten	<code>var test = true</code> <code>var test = false</code>

## Zeichenketten/Strings Literals

```
var jsString = `Das ist ein String`; // Backticks
```

```
var jsString = "Das ist ein String"; // einfache Anführungszeichen
```

```
var jsString = "Das ist ein String"; // doppelte Anführungszeichen
```

```
// Vorteil von Backticks:
```

```
var jsString = `half of 100 is ${100 / 2}`;
```

```
console.log(jsString);
```

```
// -> half of 100 is 50
```

# Operatoren

Operator	Bedeutung	Beispiel
+, +=	Addition	x+=3
-, -=	Subtraktion	x=x-5
, =	Multiplikation	a=b*c
/, /=	Division	z=e/5
%	Modulus	m=5 % 3
++, -	Inkrement, Dekrement	x++ oder y--
«, «=	Bitweise Linksschieben	x « 4
», »=	Bitweise Rechtsschieben	y » 5
»>	Bitweise Linksschieben mit Nullfüllung	a »> b
&	Bitweise UND	a & b
	Bitweise ODER	a   b
^	Bitweise Negieren	a ^ b

- Dynamisch typisiert -> bedeutet nicht, dass JS eine untypisierte Sprache ist. Vielmehr werden die Typen automatisiert bei der Wertzuweisung vergeben
- Typen:
  - Number: Zahlen
  - String: Zeichenketten
  - Boolean: logische Werte
  - Object: alles andere
- Spezielle Zustände von Variablen
  - undefined bedeutet, dass einer Variable kein Wert zugewiesen wurde.
  - null ist ein absichtlich zugewiesener Wert, der "kein Wert" oder "leerer Wert" bedeutet

# Automatische Typumwandlung

Wird ein Operator auf einen Wert eines unpassenden Typs angewandt, wandelt JS diesen Wert stillschweigend in den erforderlichen Wert um => implizierte Typumwandlung

```
console.log(8 * null); // -> 0
```

8 \* null ergibt 0, da null bei arithmetischen Operationen zu 0 konvertiert wird.

```
console.log("5" - 1); // -> 4
```

"5" - 1 ergibt 4, weil der String "5" bei Subtraktion zu einer Zahl konvertiert wird.

```
console.log("5" + 1); // -> 51
```

"five" kann nicht in eine Zahl umgewandelt werden. Der +-Operator führt hier zur Zeichenkettenverknüpfung



## Erklärung

Short Circuit Evaluation ist eine Programmiertechnik, bei der der Auswertungsprozess eines logischen Ausdrucks frühzeitig beendet wird, sobald das Ergebnis feststeht.

## Beispiele in JavaScript

### Logisches UND (&&)

```
const a = false;  
const b = true;  
const result = a && b; // result ist false,
```

# JavaScript im Browser

---