

# Einführung in JavaScript

# Was ist JavaScript?

- **JavaScript** ist eine der drei Kerntechnologien des Webs (neben HTML und CSS).

Verwendet für:

- Interaktive Webseiten (Formulare, Animationen)
- Serverseitige Anwendungen (Node.js)
- Mobile/ Desktop-Apps (React Native, Electron)
- Spielentwicklung & Machine Learning

## Was kann JavaScript **nicht** gut?

- **Rechenintensive Aufgaben:** Andere Sprachen (C, Python) sind effizienter.
- **Low-Level-Systemprogrammierung:** Kein direkter Hardware-Zugriff.
- **Multi-Threading:** Unterstützt nur begrenzte Parallelität (Web Workers).
- **Hochsichere Anwendungen:** Sprachen wie Java oder Rust bieten mehr Sicherheit.

# Interessante Fakten über JavaScript

- **1995** entwickelt von Brendan Eich in nur 10 Tagen.
- Ursprünglich "Mocha" genannt, dann "LiveScript".
- **Nicht** mit Java verwandt, trotz des Namens.
- Eine der meistgenutzten Programmiersprachen weltweit.

# JavaScript einbinden

- JavaScript kann direkt im HTML-Dokument eingebunden werden.
- Beispiel:

```
<!DOCTYPE html>
<html lang="de">
<head>
  <title>Meine Webseite</title>
</head>
<body>
  <h1>Hallo, Welt!</h1>
  <script>
    console.log('JavaScript ist aktiv!');
  </script>
</body>
</html>
```

# Konsolenausgabe

Die Konsole wird verwendet, um Ausgaben während der Entwicklung zu sehen.

```
console.log('Dies ist eine Nachricht in der Konsole.');
```

# Variablen und Datentypen

- Variablen speichern Daten, die später im Code verwendet werden.
- JavaScript unterstützt drei Arten von Variablen:
  - `var` (ältere Methode, wird heute selten genutzt)
  - `let` (variabel, kann geändert werden)
  - `const` (konstant, kann nicht geändert werden)

```
let name = 'Max'; // String
const age = 30;   // Number
let isStudent = true; // Boolean
```

# Datentypen in JavaScript

Wichtige Datentypen:

- **Number:** `42` , `3.14`
- **String:** `'Hallo'` , `"Welt"`
- **Boolean:** `true` , `false`
- **Array:** `[1, 2, 3]`
- **Object:** `{name: 'Max', age: 30}`



# Operatoren

- Arithmetische Operatoren: `+`, `-`, `*`, `/`, `%`
- Vergleichsoperatoren:
  - `==` (lose Gleichheit)
  - `===` (strikte Gleichheit)
  - `<`, `>`, `<=`, `>=`

```
let a = 5;  
let b = 10;  
console.log(a + b); // 15  
console.log(a === 5); // true
```

# Lose vs. Strikte Gleichheit

- `==` (lose Gleichheit):
  - Vergleicht nach Typkonvertierung.
  - Beispiel: `5 == "5"` ergibt `true`.
- `===` (strikte Gleichheit):
  - Vergleicht ohne Typkonvertierung.
  - Beispiel: `5 === "5"` ergibt `false`.

# Bedingungen (If-Else)

Mit `if-else` kontrollierst du den Programmablauf basierend auf Bedingungen.

```
let score = 75;

if (score >= 90) {
  console.log('Ausgezeichnet!');
} else if (score >= 75) {
  console.log('Gut gemacht!');
} else {
  console.log('Mehr Übung erforderlich.');
```

# Schleifen

- **For-Schleife:** Wird verwendet, um eine Aktion mehrfach auszuführen.

```
for (let i = 1; i <= 10; i++) {  
  console.log(i); // Gibt 1 bis 10 aus  
}
```

- **While-Schleife:** Läuft, solange die Bedingung wahr ist.

```
let i = 0;  
while (i < 5) {  
  console.log(i);  
  i++;  
}
```

# Funktionen

Funktionen sind Codeblöcke, die Aufgaben ausführen. Sie können Parameter annehmen und einen Wert zurückgeben.

```
function greet(name) {  
    return 'Hallo, ' + name;  
}  
  
console.log(greet('Max')); // Hallo, Max
```

# Arrays und Objekte

- **Array:** Eine Liste von Werten.

```
let fruits = ['Apfel', 'Banane', 'Kirsche'];  
console.log(fruits[0]); // Apfel
```

- **Objekt:** Sammlung von Schlüssel-Wert-Paaren.

```
let person = {  
  name: 'Max',  
  age: 30,  
  isStudent: true  
};  
  
console.log(person.name); // Max
```