

Tag 4 – JavaScript: Kontrollstrukturen

JavaScript bietet verschiedene Möglichkeiten zur **bedingten Ausführung**, darunter:

- `if, else if, else`
- `switch`-Anweisung
- der **ternäre Operator** `? :`

1. Die `if`-Anweisung

```
if (Bedingung) {  
    // Code, der nur ausgeführt wird, wenn die Bedingung true ergibt  
}
```

Beispiel:

```
let isUserReady = confirm("Are you ready?");  
if (isUserReady) {  
    alert("User ready!");  
}
```

- Die Bedingung kann jeder Ausdruck sein, der in einen Boolean konvertiert werden kann.
- Klammern `{}` immer verwenden, auch bei einer einzigen Anweisung – das beugt Fehlern vor.

Typischer Fehler:

```
if (isUserReady)  
    console.log("User ready!");  
    alert("User ready!"); // wird immer ausgeführt
```

2. Gültigkeitsbereich (Scope) innerhalb von Bedingungen

Variablen, die mit `let` oder `const` **innerhalb** eines Blocks `{}` deklariert werden, sind **nur in diesem Block gültig**:

```
let pieces = prompt("Wie viele Stück?", 0);  
if (pieces > 0) {  
    let total = pieces * 10;  
    console.log(total);  
}  
console.log(total); // Fehler: total ist nicht definiert
```

3. Verschachtelte Bedingungen & komplexe Ausdrücke

```
let userAge = 23;  
let points = 703;  
let cartValue = 299;  
let shippingCost = 9.99;  
  
if (userAge > 21 && (cartValue >= 300 || points >= 500)) {  
    shippingCost = 0;  
}
```

Logische Operatoren wie `&&` und `||` lassen sich kombinieren – Klammern sorgen für Übersicht und definieren die Auswertungsreihenfolge.

4. `if...else`-Strukturen

```
if (Bedingung) {  
    // falls true  
} else {  
    // falls false  
}
```

```
let isUserReady = confirm("Are you ready?");  
if (isUserReady) {  
    console.log("User ready!");  
} else {  
    console.log("User not ready!");  
}
```

5. `if...else if...else` – mehrfache Bedingungen

```
let number = prompt("Gib eine Zahl ein", 0);  
  
if (number < 10) {  
    alert("<10");  
} else if (number < 30) {  
    alert("<30");  
} else if (number < 100) {  
    alert("<100");  
} else {  
    alert(">=100");  
}
```

Nur der **erste erfüllte Fall** wird ausgeführt – danach wird der Rest übersprungen.

6. Komplexe Bedingungen in Kombination

```
let userAge = 19;  
let hasParentsApproval = true;  
let cartValue = 310;  
let points = 200;  
let shippingCost = 9.99;  
let addInsurance = true;  
let orderIsValid = true;  
let hasPromoCode = false;  
  
if (userAge < 21 && !hasParentsApproval) {  
    orderIsValid = false;  
} else if (userAge < 21 && hasParentsApproval) {  
    shippingCost -= 5;  
} else if (userAge >= 65) {  
    shippingCost = 0;  
} else if (userAge >= 21 && (hasParentsApproval || cartValue > 300 || points > 500)) {  
    shippingCost = 0;  
}
```

```

if (addInsurance && orderIsValid && !hasPromoCode) {
    shippingCost += 5;
}

if (orderIsValid) {
    alert("Versandkosten: " + shippingCost + " €");
} else {
    alert("Bestellung ungültig");
}

```

7. Der ternäre Operator ? :

```

let preis = 100;
let versand = preis > 50 ? 0 : 5;

```

Kurzform für `if...else`, wenn ein **einzelner Wert** zugewiesen wird:

```

let start = confirm("Start?");
let message = start ? "Here we go!" : "Abgebrochen";
alert(message);

```

Kein Ersatz für lange Verzweigungen – nur für einfache Zuweisungen geeignet.

8. Die `switch`-Anweisung

Alternative zu `if...else if...else`, wenn viele mögliche **konkrete Werte** verglichen werden sollen.

```

let gate = prompt("Choose gate: a, b, or c");
let win = false;

switch (gate) {
    case "a":
        alert("Gate A: empty");
        break;
    case "b":
        alert("Gate B: main prize");
        win = true;
        break;
    case "c":
        alert("Gate C: empty");
        break;
    default:
        alert("No gate " + String(gate));
}

if (win) {
    alert("Winner!");
}

```

- Vergleicht mit `===` (Wert + Typ!)
- `break` ist wichtig, sonst werden nachfolgende `cases` ebenfalls ausgeführt
- `default` wird verwendet, wenn kein `case` zutrifft

9. Übungen zu Kontrollstrukturen

Aufgabe 1: Altersprüfung

Frage per `prompt()` das Alter ab und gib mit `alert()` aus:

- "Minderjährig" bei <18
- "Volljährig" ab 18

Aufgabe 2: Rabattberechnung

- Erstelle eine Variable `warenwert`
- Wenn `warenwert > 100`, setze `rabatt = 10`, sonst `rabatt = 0`
- Gib beides in der Konsole aus

Aufgabe 3: Zahl auswerten

Frage den Benutzer nach einer Zahl und gib abhängig vom Wert folgendes aus:

- kleiner 0 → "Negativ"
- 0 → "Null"
- größer 0 → "Positiv"

Aufgabe 4: `switch` mit Wochentag

- Frage nach einem Wochentag (z. B. "Mo", "Di"...) und gib per `switch` aus:
 - "Wochenstart", "Mitte der Woche", "Wochenende"

Aufgabe 5: Kombination von Bedingungen

- Erstelle Variablen für `alter`, `punkte`, `vipStatus`
 - Wenn `alter >= 18` UND (`punkte > 500` ODER `vipStatus === true`), gib "Zugang erlaubt" aus, sonst "Zugang verweigert"
-