

# Skill Booster 003 003

## Taschenrechner-Logik in JavaScript implementieren

Implementierung der Funktionalität eines Taschenrechners in einer Webanwendung unter Verwendung von JavaScript. Die vorgegebenen HTML- und CSS-Dateien stellen das Grundgerüst und das Design des Taschenrechners dar. Deine Aufgabe ist es, die Logik für die Durchführung von grundlegenden Rechenoperationen (Addition, Subtraktion, Multiplikation, Division) sowie für erweiterte Funktionen wie das Wechseln des Vorzeichens und Löschen des zuletzt eingegebenen Zeichens, zu entwickeln.

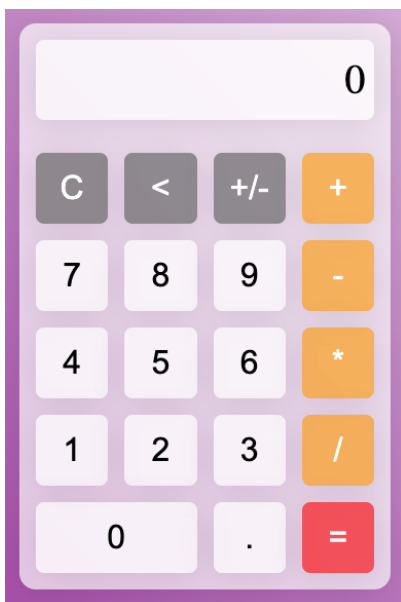
Anforderungen:

### 1. Grundoperationen:

- Addition (+)
- Subtraktion (-)
- Multiplikation (\*)
- Division (/)

### 2. Zusätzliche Funktionen:

- C (Clear): Setze den Taschenrechner zurück und lösche alle Eingaben.
- < (Backspace): Entferne die letzte Ziffer/Eingabe.
- +/- (Vorzeichenwechsel): Wechsle das Vorzeichen der aktuellen Eingabe.
- . (Dezimalpunkt): Ermögliche die Eingabe von Dezimalzahlen.



### 3. Benutzereingabe:

- Die Benutzereingabe muss über die Tastatur des Taschenrechners, per Maus-Klick, erfolgen.
- Jede Taste soll bei Betätigung die entsprechende Zahl oder Operation auf dem Display ausgeben.

#### 4. Display:

- Die Anzeige soll die eingegebenen Zahlen und die Ergebnisse der Berechnungen darstellen.
- Das Überlaufen der Zahlen muss durch die maximal Anzahl von 15 Zeichen, inkl. Minus – Vorzeichen und Dezimalpunkt ., begrenzt sein.



#### 5. Berechnung und Ergebnisanzeige:

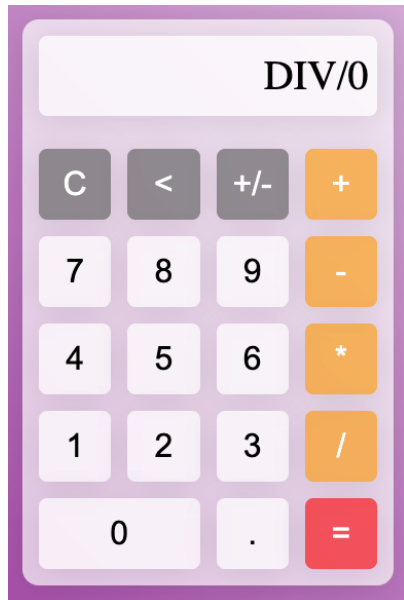
- Die Berechnungen sollen korrekt durchgeführt werden, wenn die "=" Taste gedrückt wird.
- Die Anzeige soll das Ergebnis zeigen, ohne die Seite neu zu laden.

#### Technische Hinweise:

- Implementiere deine Logik innerhalb des window Event-Listener für das **load** Event. Diese Event wird ausgelöst sobald die HTML Seite komplett geladen ist.

```
// Window Event-Listener
window.addEventListener("load", function () {
  // Hier kommt deine Taschenrechner Logik hin
})
```

- Verwende Event-Listener, um auf Benutzereingaben zu reagieren.
- Organisiere den Code in Funktionen für eine bessere Lesbarkeit und Wartbarkeit.
- Implementiere Fehlerbehandlung, für die Division durch Null.



**Beispielcode für eine Event-Listener-Implementierung:**

```
const operandButtons = document.querySelectorAll(".operands-grid button");

operandButtons.forEach((button) => {
  button.addEventListener("click", () => {
    if (button.textContent === "C") {
      // clear all logik
    } else if (button.textContent === "<") {
      // undo logik
    } else if (button.textContent === "+/-") {
      // toggle number negative logik
    } else {
      // add number to first or second operand
    }
  });
});
```

**Abgabe:**

Sende das vollständigen Projekt mit HTML-, CSS- und JavaScript-Code, als ZIP Archiv an deinen Ausbilder.  
Bennen das ZIP Archiv `003_003_skill_booster_firstname_lastname.zip`