

LU01.A01 - MyAlgebraCalculator

Rahmenbedingungen

- Sozialform: Einzelarbeit
- Hilfsmittel: Openbooks
- Zeit: 30 Minuten
- Erwartetes Resultat: JavaScript-File, dass die unten genannten Taschenrechner-Berechnungen durchführen kann.

Hinweise

Wichtig bei der Umsetzung Ihrer Lösung ist, dass sie nach best-practise programmieren. Konkret heisst das:

- Ihr Script, sowie die verwendeten Subroutinen (Funktionen, Methoden) müssen ei-nen Header haben: Autor, Datum, Angaben zur Input-, und Output-Parameter, eine kurze Beschreibung zum Verhalten der Funktion.
- Bei fehlenden Parametern wird die eingebende Person entsprechend über das HTML-Formular oder der Kommandozeile informiert.
- Verwenden Sie die unbedingt die passenden Datentypen (primitive oder komplexe).
- Verwenden Sie keine JS-Bibliotheken, sondern programmieren Sie alle relevanten Funktionen selbst. Letztendlich ist das das eigentliche Ziel dieser Übung.
- Aus Gründen der Datenkapselung und der umsichtigen Programmierung realisieren sie Ihre Funktionen ausschliesslich mit Inputparameter und Return-Values. Es wird also möglichst nicht auf globale Werte innerhalb der Methoden zugegriffen.
- Die für die Berechnung benötigten Werte können über Commandline, als Variable oder über ein simples HTML-Formular eingegeben werden. ↗Vergewischen Sie keine Zeit in eine „schöne“ Oberflächengestaltung, weil es aktuell um Programmierung und nicht um Gestaltung geht.
- Listenpunkt

Auftrag

Programmieren Sie einen Rechner *myAlgebraCalculator.js*, der nicht nur die Grundrechenarten durchführt, sondern auch quadrieren, potenzieren und die Wurzeln ziehen kann. Testen Sie anschliessend Ihre Lösung auf Funktion, indem Sie die Funktionen ausführen und das Ergebnis auf der Kommandozeile ausgeben lassen.

Addition

$$\begin{array}{r} 33 \\ + \quad 44 \\ \hline 77 \end{array}$$
$$5 \quad + \quad x \quad = \quad 5 + x$$

1. Summand + 2. Summand = Summe

Die Summanden sind vertauschbar.

Subtraktion

$$\begin{array}{r} 330 \\ - \quad 45 \\ \hline 285 \end{array}$$
$$5 \quad - \quad x \quad = \quad 5 - x$$

Minuend - Subtrahend = Differenz

Minuend und Subtrahend sind nicht vertauschbar.

Multiplikation

$$\begin{array}{r} 33 \\ \cdot \quad 44 \\ \hline 1452 \end{array}$$
$$5 \quad \cdot \quad x \quad = \quad 5x$$

1. Faktor · 2. Faktor = Produkt

Die Faktoren sind vertauschbar.

Division

$$\begin{array}{r} 330 \\ : \quad 33 \\ \hline 10 \end{array}$$
$$10 \quad : \quad x \quad = \quad 10 : x$$

Dividend : Divisor = Quotient

Dividend und Divisor sind nicht vertauschbar.

Der Rechner soll die nachfolgenden Funktionen realisieren:

Teilauftrag 1: Addition

```
// Autor:  
// Datum:  
// Beschreibung: Zwei eingegebene summand1 und summand2 werden als Summe zurückgegeben.  
// Hinweis: Die beiden Summanden sind vertrauschbar.  
function addiere(summand1, summand2) {  
    ....  
    return summe  
}
```

===== Teilauftrag 2: Subtraktion =====

```
// Autor:  
// Datum:  
// Beschreibung: Der eingegebene minuend wird von subrahend abgezogen und die differenz zurückgegeben  
// Hinweis: Minuend und Subrahend sind nicht vertauschbar.  
function subrahiere(minuend, subrahend ){  
    ....  
    return differenz  
}
```

Teilauftrag 3: Division

```
// Autor:  
// Datum:  
// Beschreibung: Der eingegebene dividend wird durch den divisor geteilt und  
als quotient zurückgegeben.  
// Hinweis: dividend und divisor sind nicht vertauschbar.  
function teile(dividend, divisor) {  
    ....  
    return quotient  
}
```

Teilauftrag 4: Multiplikation

```
// Autor:  
// Datum:  
// Beschreibung: Zwei eingegebene Zahlen faktor1 und faktor2 werden  
multipliziert und als produkt zurückgegeben.  
// Hinweis: Faktor1 und faktor2 sind vertauschbar.  
function multipliziere(faktor1, faktor2) {  
    ....  
    return produkt  
}
```

Teilauftrag 5: Quadrat

```
// Autor:  
// Datum:  
// Beschreibung: Eine Zahl basis soll mit sich selbst multipliziert und das  
Ergebnis als produkt zurückgegeben werden.  
// Hinweis: basis und exponent sind nicht vertauschbar.  
function multipliziere(basis){  
    ....  
    return produkt  
}
```

Teilauftrag 6: Wurzel

```
// Autor:  
// Datum:  
// Beschreibung: Von einer Zahl basis soll die Wurzel gezogen und das  
Ergebnis als produkt zurückgegeben werden.  
// Hinweis: Wurzel-Ziehen ist das gleiche wie basis ^1/2  
function wurzel(basis) {  
    ....  
    return produkt
```

}

Teilauftrag 6: Exponent

```
// Autor:  
// Datum:  
// Beschreibung: Eine Zahl basis soll mittels dem exponenten die mit sich  
multipliziert, und das Ergebnis als produkt zurückgegeben werden.  
// Hinweis: basis und exponent sind nicht vertauschbar.  
function multipliziere(basis, exponent{  
    ....  
    return produkt  
}
```

Lösungen

[LU01.L01](#)



Volkan Demir

From:
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:
<https://wiki.bzz.ch/de/modul/m307/learningunits/lu01/aufgaben/01?rev=1770644323>

Last update: **2026/02/09 14:38**

