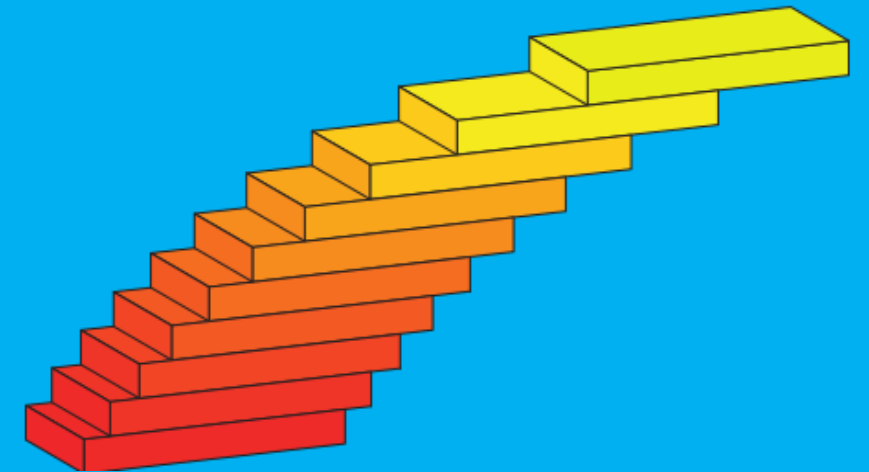


# Parametrisierbare Aufgaben mit STACK: Aufgaben mit numerischer Eingabe



*Stand: 02.10.2026*

# STACK // Erstellung einer Beispielaufgabe

- In dieser Anleitung wird erklärt, wie STACK-Aufgaben erstellt werden können, in denen die Antwort der Studierenden eine Zahl ist.

Dies geschieht anhand der folgenden Beispielaufgabe zur Addition von Dezimalzahlen:

Bitte addieren Sie:  $0.1111 + 1.0005$

Antwort:

- In dieser Anleitung werden dazu unter anderem der Eingabetyp „Numerisch“ und verschiedene Antwortüberprüfungsmethoden gezeigt.
- Erstellen Sie zunächst eine neue STACK-Aufgabe und füllen Sie das Feld „Fragetitel“ aus.

*Hinweis: Wir empfehlen, vor dem Anschauen dieser Anleitung zunächst die Klickanleitung „Erstellung einer Aufgabe“ durchzuarbeiten oder das gleichnamige Videotutorial zu schauen.*

# STACK // Aufgabenvariablen und Fragetext

- Erstellen Sie im Feld **1 Aufgabenvariablen** zunächst je eine Variable für die beiden Zahlen, die addiert werden sollen.
- Nennen Sie diese Variablen bspw. x1 und x2 und setzen sie sie auf die Werte 0.1111 und 1.0005.
- Erstellen Sie zuletzt noch eine Variable für die korrekte Lösung der Aufgabe, die Sie bspw. ta nennen. Die korrekte Antwort ist hier die Summe der beiden Zahlen, also  $x1+x2$ , was einem Wert von 1,1116 entspricht.
- Schreiben Sie dann den **2 Fragetext** der Aufgabe.

1

Aufgabenvariablen



```
x1: 0.1111;  
x2: 1.0005;  
ta: x1+x2;
```

2

Bearbeiten Ansicht Einfügen Format Werkzeuge Tabelle Hilfe



Bitte addieren Sie: \({@x1@}+{@x2@}\)

Antwort: [[input:ans1]] [[validation:ans1]]

p

9 Wörter

HRW

## STACK // Bereich „Eingabe“

- Wählen Sie im Bereich „Eingabe“ als **1 Eingabetyp** nun die Option „Numerisch“ aus und verwenden als **2 Musterlösung** die Variable, die Sie in den Aufgabenvariablen für die korrekte Antwort der Aufgabe angelegt haben.
- Achten Sie zudem darauf, dass die Option **3 Verbiete Fließkommazahlen** auf „Nein“ gesetzt ist.

▼ Eingabe: ans1

1 Eingabetyp ? Numerisch ▾

2 Musterlösung ? ta

3 Verbiete Fließkommazahlen ? Nein ▾

- Der Eingabetyp „Numerisch“ hat die Besonderheit, dass alle Ausdrücke, die Variablen enthalten, direkt in der Validierung als ungültig markiert werden.

# STACK // Bereich „Eingabe“: Zusätzliche Optionen (1)

- Für Eingabefelder des Typs „Numerisch“ gibt es einige **1zusätzliche Optionen**, mit denen Sie festlegen können, in welcher Form die Antwort vorliegen muss. Tut sie dies nicht, gibt es eine Fehlermeldung in der Validierung.
- Beispiel: Mit dem Schlüsselwort `floatnum` kann festgelegt werden, dass nur Fließkommazahlen akzeptiert werden. Ist die Antwort dann beispielsweise eine ganze Zahl, gibt es eine **2Fehlermeldung**.
- Eine vollständige Liste der verfügbaren zusätzlichen Optionen folgt auf der nächsten Folie.

**1**

Zusätzliche Optionen

?

`floatnum`

Bitte addieren Sie:  $0.1111 + 1.0005$

Antwort:

Ihre letzte Antwort wurde folgendermaßen interpretiert:

1

**2**

Diese Antwort ist ungültig. Für diese Eingabe wird eine Fließkommazahl erwartet.

## STACK // Bereich „Eingabe“: Zusätzliche Optionen (2)

Für Eingabefelder des Typs „Numerisch“ sind folgende Optionen verfügbar. Wenn mehrere von ihnen gleichzeitig verwendet werden sollen, dann können sie mit Kommata getrennt in das Feld „Zusätzliche Optionen“ eingetragen werden.

| Option              | Erläuterung                                                                | Beispiele                                                                                                   |
|---------------------|----------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| floatnum            | Nur Dezimalzahlen sind gültige Eingaben                                    | 1.5 wird akzeptiert, aber 1, 1/2 und pi nicht                                                               |
| intnum              | Nur ganze Zahlen sind gültige Eingaben                                     | 1 wird akzeptiert, aber 1.5, 1/2 und pi nicht                                                               |
| rationalnum         | Nur Brüche sind gültige Eingaben                                           | 1/2 wird akzeptiert, aber 1, 1.5 und pi nicht                                                               |
| rationalized        | Nenner von vorkommenden Brüchen dürfen keine irrationalen Zahlen enthalten | $\sqrt{2}$ wird akzeptiert, aber $1/\sqrt{2}$ nicht                                                         |
| mindp:n und maxdp:m | Festlegung einer Mindest- und/oder Höchstzahl an Nachkommastellen          | mindp:2 = mindestens zwei Stellen<br>maxdp:4 = maximal vier Stellen<br>mindp:3,maxdp:3 = genau drei Stellen |
| minsf:n und maxsf:m | Festlegung einer Mindest- und/oder Höchstzahl an signifikanten Stellen     | s. o.                                                                                                       |

# STACK // Rückmeldebaum

- Im Rückmeldebaum kann als **1 SAns-Wert** die Antwort der Studierenden (ans1) und als **2 TAns-Wert** die korrekte Antwort (z. B. Variable ta) verwendet werden.

**Knoten 1** ? Beschreibung

Antwortüberprüfung  SAns ans1 **1**

TAns ta **2** Test-Optionen

Feedback unterdrücken Nein

- Für Aufgaben mit Einheiten stehen mehrere eigene **3 Antwortüberprüfungsmethoden** zur Verfügung, nämlich „NumAbsolute“, „NumRelative“, „Num-GT“, „Num-GTE“, „NumDecPlaces“, „NumDecPlacesWrong“, „NumSigFigs“ und „SigFigsStrict“.
- Auf den nachfolgenden Folien werden sie genauer erklärt.

**3**

- Num-GT
- Num-GTE
- NumAbsolute**
- NumDecPlaces
- NumDecPlacesWrong
- NumRelative
- NumSigFigs

# STACK // Antwortüberprüfungen (1)

## „NumAbsolute“:

- Als **1 Test-Option** muss eine absolute Toleranz festgelegt werden.
- Die Antwort wird als richtig gewertet, wenn sie um höchstens die angegebene Toleranz von TAns abweicht.
- Beispiel: Bei einer Test-Option von 0.01 und einer korrekten Antwort von 1.1116 werden alle Antworten aus dem Intervall  $[1.1116 - 0.01, 1.1116 + 0.01] = [1.1016, 1.1216]$  akzeptiert.

## „NumRelative“:

- Als Test-Option muss eine relative Toleranz festgelegt werden.
- Die Antwort wird als richtig gewertet, wenn sie um höchstens das Produkt Toleranz\*TAns von TAns abweicht.
- Beispiel: Bei einer relativen Toleranz von 0.1 und einer korrekten Antwort von 1.1116 werden alle Antworten aus dem Intervall  $[1.1116 - 1.1116 \cdot 0.1, 1.1116 + 1.1116 \cdot 0.1] = [1.00044, 1.122276]$  akzeptiert.

*Tipp: Sie können im Feld „Test-Optionen“ jeweils auch 0 eintragen, wenn die Antwort der Studierenden genau mit der Musterlösung übereinstimmen soll.*

|                    |             |               |               |
|--------------------|-------------|---------------|---------------|
| Beschreibung       |             |               |               |
| Antwortüberprüfung | NumAbsolute | SAns          | ans1          |
| TAns               | ta          | Test-Optionen | 0.01 <b>1</b> |

## STACK // Antwortüberprüfungen (2)

### „Num-GT“ und „Num-GTE“:

- Die Antwort wird als richtig gewertet, wenn sie größer ist als TAns („Num-GT“; „GT“ steht für „greater than“) bzw. größer als oder gleich TAns („Num-GTE“; „GTE“ steht für „greater than or equal“).
- Beispiel: Bei einer korrekten Antwort von 1.1116 werden alle Antworten akzeptiert, die größer sind als 1.1116 („Num-GT“) bzw. größer als oder gleich 1.1116.
- Eine Test-Option wird hier nicht benötigt.

### „NumDecPlaces“:

- Als Test-Option muss eingetragen werden, auf wie viele Nachkommastellen genau die Antwort der Studierenden sein muss.
- Die Antwort wird nur dann als richtig gewertet, wenn sie genau so viele Nachkommastellen hat, wie es als Test-Option festgelegt wurde – unabhängig davon, wie viele Nachkommastellen TAns besitzt.
- Beispiel: Bei einer korrekten Antwort von 1.1116 und Testoption 3 wird nur 1.112 als korrekte Antwort akzeptiert (TAns auf drei Nachkommastellen gerundet).
- Hinweis: Wenn als Test-Option ein größerer Wert verwendet wird als TAns Nachkommastellen hat, dann muss die Antwort mit Nullen aufgefüllt werden (z. B. 1.5000, wenn TAns 1.5 und Test-Option 4 ist).

## STACK // Antwortüberprüfungen (3)

### „*NumDecPlacesWrong*“:

- Die Antwort wird als richtig gewertet, wenn sie bis auf die Position des Dezimaltrennzeichens mit TAns übereinstimmt.
- Als Test-Option muss die gewünschte Anzahl an signifikanten Stellen eingegeben werden, auf die die Antwort ansonsten genau eingegeben werden muss.
- Beispiel: Wenn TAns einen Wert von 1.1116 hat, dann werden auch 111.16 und 0.011116 als richtige Antworten akzeptiert, weil diese durch das Verschieben des Kommas in TAns erreicht werden können. Wenn bspw. eine Test-Option von 3 verwendet wird, überprüft STACK nur die ersten drei signifikanten Stellen der Antwort, d. h. es werden bspw. auch 1.11, 111.444 oder 0.01113 akzeptiert.

### „*SigFigsStrict*“:

- Als Test-Option muss eine gewünschte Anzahl an signifikanten Stellen eingegeben werden, die die Antwort besitzen muss.
- STACK überprüft nur, ob die eingegebene Zahl so viele signifikante Stellen hat, wie es als Testoption festgelegt wurde. Insbesondere wird der konkrete Wert nicht überprüft und nicht mit TAns verglichen.
- In der Regel wird „*SigFigsStrict*“ daher nicht als einzige Überprüfung einer Antwort genutzt, sondern in Kombination mit anderen Tests in separaten Knoten im Rückmeldebaum.

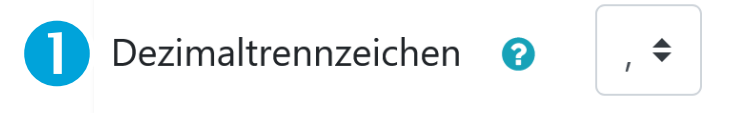
# STACK // Antwortüberprüfungen (4)

## „NumSigFigs“:

- Für die Test-Option gibt es hier zwei Möglichkeiten:
  - a) Anzahl der signifikanten Stellen, auf die die Lösung genau angegeben werden muss (z. B. 3)
  - b) Liste mit zwei Zahlen  $[m, n]$ , wobei die Antwort der Studierenden  $m$  signifikante Stellen enthalten muss, aber nur die ersten  $n$  von ihnen auf Korrektheit überprüft werden (z. B.  $[4, 3]$ )
- Beispiel zu a): Wenn TAns einen Wert von 1.1116 hat und die Test-Option 3 verwendet wird, dann wird ausschließlich 1.11 als korrekte Antwort akzeptiert (TAns gerundet auf drei signifikante Stellen). 1.112 und 1.1116 (d. h. TAns selbst) hingegen besitzen zu viele signifikante Stellen und werden daher nicht akzeptiert.
- Beispiel zu b): Wenn TAns einen Wert von 1.1116 hat und die Test-Option  $[4, 3]$  verwendet wird, dann muss die Antwort der Studierenden vier signifikante Stellen enthalten, aber nur auf drei signifikante Stellen genau sein. Beispielsweise wird so 1.115 akzeptiert: Die Zahl enthält vier signifikante Stellen und die ersten drei von ihnen sind korrekt.

# STACK // Ändern des Dezimaltrennzeichens

- In STACK-Aufgaben müssen Studierende bei Dezimalzahlen standardmäßig einen Punkt als Dezimaltrennzeichen verwenden, z. B. 1.5.
- Im Bereich „Optionen“ kann über die Option **1 Dezimaltrennzeichen** eingestellt werden, dass Studierende stattdessen ein Komma verwenden müssen, z. B. 1, 5.
- Studierende *müssen* dann alle Dezimalzahlen mit einem Dezimalkomma eingeben, sonst gibt es eine Fehlermeldung in der Validierung.
- Zudem muss überall, wo sonst ein Komma verwendet werden würde, ein Semikolon verwendet werden. Dies betrifft Mengen (z. B. {1;2;3}), Listen (z. B. [1;2;3]) und Funktionen mit mehreren Argumenten (z. B. Binomialkoeffizient: `binomial(n;k)`).
- Die Änderung der Option „Dezimaltrennzeichen“ führt nur zu einer Änderung der Anzeige und Eingabe von Ausdrücken für Studierende. Bei der Erstellung von STACK-Aufgaben müssen alle CAS-Ausdrücke weiterhin wie üblich notiert werden.



*Hinweis: Auch wenn die Notation mit einem Dezimalkomma hierzulande üblich ist, sollte das Dezimaltrennzeichen nicht um jeden Preis geändert werden. Wenn Ihre Studierenden es bereits gewöhnt sind, in STACK-Aufgaben einen Dezimalpunkt zu verwenden, könnte diese Änderung sie verwirren. In jedem Fall sollte das verwendete Dezimaltrennzeichen innerhalb eines Kurses einheitlich gehandhabt werden.*

# STACK // Moodle-Kurs „Tutorials STACK“

- Dort finden Sie Videotutorials, Beispielaufgaben, weitere Klickanleitungen und mehr
- Link: <https://moodle.ruhr-uni-bochum.de/course/view.php?id=22225>
- Passend zu dieser Klickanleitung empfehlen wir Ihnen:
  - Videotutorial „STACK-Aufgaben mit numerischer Eingabe“



The screenshot shows the Moodle course page for 'Tutorials zu STACK' at RUB. The header includes the RUB logo, navigation links (Dashboard, LEARN2Gether, OpenRUB, Support), and a user profile icon. The main content area features the title 'Tutorials zu STACK' with a colorful staircase graphic. Below the title, there is a paragraph explaining that STACK is a Moodle question type for digital mathematical tasks. Another paragraph mentions that the course provides tutorials and examples. A final paragraph states that the course is public and accessible without login, but an account is needed for full functionality.

**RUB** Dashboard LEARN2Gether OpenRUB Support Teilnehmer/in

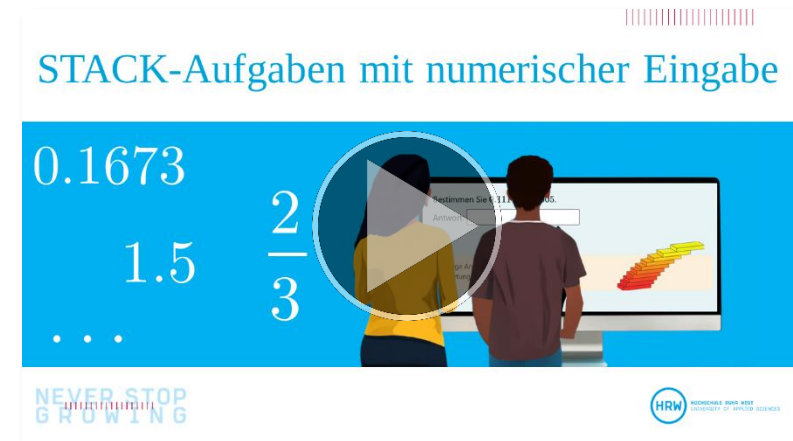
Alles einklappen

## Tutorials zu STACK

STACK ist ein Moodle-Fragetyp, mit dem sich geschlossene und offene digitale mathematische Aufgaben umsetzen lassen, die automatisch ausgewertet werden. Dabei ist es möglich, dass STACK Aufgabenstellungen zufällig generiert und den Lernenden ein differenziertes und adaptives Feedback bereitstellt.

In den Abschnitten des Kurses werden unterschiedliche Aspekte des Fragetyps STACK in Form von Videotutorials und Beispielaufgaben vermittelt. Zum Start empfehlen wir Ihnen die Inhalte des Abschnitts "Einführung in STACK" und "Eine STACK-Aufgabe erstellen", die Grundlagen vermitteln. Die weiteren Abschnitte können in beliebiger Reihenfolge bearbeitet werden.

Dieser Kurs ist komplett öffentlich und ohne Login zugänglich. Um alle Funktionen uneingeschränkt nutzen zu können, ist allerdings ein Account für das Moodle der Ruhr-Universität Bochum notwendig. Falls Sie noch keinen Account besitzen, können Sie sich über diesen Link kostenlos und unkompliziert einen Account erstellen. Danach müssen Sie sich noch in den Kurs einschreiben. Dies funktioniert unter diesem Link.



The thumbnail features a blue background with mathematical examples: the decimal 0.1673, the decimal 1.5, and the fraction  $\frac{2}{3}$ . On the right, two people are shown from behind, looking at a computer screen that displays a STACK question interface with a staircase graphic. The text 'NEVER STOP GROWING' is at the bottom left, and the HRW logo is at the bottom right.

### STACK-Aufgaben mit numerischer Eingabe

0.1673  
1.5  
...

$\frac{2}{3}$

NEVER STOP GROWING

HRW

Dieses Dokument steht – ausgenommen verwendeter Logos bzw. Wort-/Bildmarken, Screenshots und anders gekennzeichnete Elemente – unter einer CC BY-SA 4.0 Lizenz. Sie dürfen die Inhalte **vervielfältigen, verwenden, verarbeiten, vermischen und verbreiten** unter der **Bedingung**, dass Sie die Originalautor:innen nennen und Beiträge, die durch Remix, Veränderung oder Aufbau auf diesem Video entstehen, nur unter derselben Lizenz wie das Original verbreiten.



#### *Anmerkung für Nachnutzende | Version 1.1.a vom 02.10.2026*

##### **Empfohlene Zitation:**

„Parametrisierbare Aufgaben mit STACK: Aufgaben mit numerischer Eingabe“, Team E-Learning@Hochschule Ruhr West von Jonas Lache, ist lizenziert unter einer CC BY-SA 4.0 International Lizenz (<https://creativecommons.org/licenses/by-sa/4.0/>).

Ausgenommen von der Lizenz sind die verwendeten Wort-/Bildmarken und alle anders gekennzeichneten Elemente.

Dieses Video basiert auf dem Video „STACK-Aufgaben mit numerischer Eingabe“, Team E-Learning@Hochschule Ruhr West von Jonas Lache, das unter CC BY-SA 4.0 (<https://creativecommons.org/licenses/by-sa/4.0/>) im Moodle-Kurs „E-Learning@HRW – Selbstlernkurs STACK“ (<https://elearning.hs-ruhrwest.de/course/view.php?id=24999>) veröffentlicht wurde.

#### *Zusätzliche Informationen für Nachnutzende:*

##### **Ursprungsort:**

Moodle-Kurs „E-Learning@HRW – Selbstlernkurs STACK“ (<https://elearning.hs-ruhrwest.de/course/view.php?id=24999>), Stand 02.10.2026, Version 1.1.a

##### **Urheber:in(nen):**

Jonas Lache, Team E-Learning@Hochschule Ruhr West, Mail: [elearning@hs-ruhrwest.de](mailto:elearning@hs-ruhrwest.de)

##### **Bildnachweise (einzeln):**

– Logo des Assessment-Systems STACK: [stack-assessment.org](https://stack-assessment.org), CC BY-SA 4.0