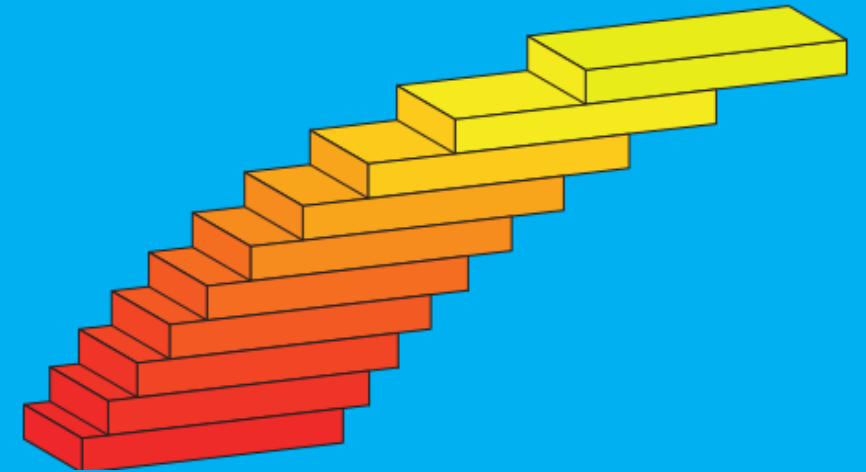


Parametrisierbare Aufgaben mit STACK: Umformung von Gleichungen und Termen mit „Equivalence reasoning“



Stand: 02.10.2026

STACK // Eingabetyp „Equivalence reasoning“

- In dieser Anleitung wird anhand einer Beispielaufgabe erklärt, wie in STACK Aufgaben eine spezielle Art von Eingabefeldern verwendet werden kann, in dem die Studierenden Zeile für Zeile einen Term oder eine Gleichung umformen (Eingabetyp „Equivalence reasoning“).
- In der Beispielaufgabe geht es darum, eine Gleichung nach x aufzulösen:

Lösen Sie die Gleichung $2 \cdot (x + 2) - 6 = 4$ nach x auf, indem Sie für jeden Umformungsschritt eine eigene Zeile im Eingabefeld verwenden.

- Die Anleitung thematisiert unter anderem die Konfiguration eines solchen Eingabefelds (inkl. zusätzlicher Optionen) und die Bewertung von Antworten der Studierenden.

Hinweis: Wir empfehlen, vor dem Anschauen dieser Anleitung zunächst die Klickanleitung „Erstellung einer Aufgabe“ oder das gleichnamige Videotutorial sowie das Videotutorial „Der Rückmeldebaum: Spezifisches Feedback in STACK“ zu schauen.

STACK // Echtzeit-Feedback für Studierende

- Kern eines **1 Eingabefelds des Typs „Equivalence reasoning“** ist die Möglichkeit, dass Studierende beim Umformen von Gleichungen und Termen in Echtzeit ein Feedback darüber erhalten, ob ihre Umformungsschritte korrekt sind. Dazu werden in der Validierung verschiedene Symbole angezeigt:
 - Korrekte Umformung: $\Leftrightarrow$
 - Falsche Umformung: $?$
 - Nur die Hinrichtung gilt: $\Rightarrow$
 - Nur die Rückrichtung gilt: $\Leftarrow$
 - Angabe des Definitionsbereichs, falls dieser eingeschränkt ist, z. B. $x \in (-1, \infty)$
- Die Symbole und die Angabe der Definitionsbereiche können auch deaktiviert werden (siehe Folie [6](#)).

1

Lösen Sie die Gleichung $2 \cdot (x + 2) - 6 = 4$ nach x auf, indem Sie für jeden Umformungsschritt eine eigene Zeile im Eingabefeld verwenden.

```
2*(x+2)-6=4
2*(x+2)=10
2*x+2=10
(2*x+2)^2=100
log((2*x+2)^2)=log(100)
```

$2 \cdot (x + 2) - 6 = 4$
 $\Leftrightarrow 2 \cdot (x + 2) = 10$
 $? 2 \cdot x + 2 = 10$
 $\Rightarrow (2 \cdot x + 2)^2 = 100$
 $\Leftarrow \ln((2 \cdot x + 2)^2) = \ln(100) \quad x \in (-1, \infty)$

In Ihrer Antwort wurden die folgenden Variablen gefunden: $[x]$

STACK // Aufgabenvariablen

- Erstellen Sie in den **1 Aufgabenvariablen** zunächst eine Variable g1 für die Gleichung, die umgeformt werden soll:
g1: $2 \cdot (x+2) - 6 = 4$;
- Erstellen Sie dann eine Variable g2 für die Gleichung, die die Studierenden am Ende ihrer Umformungen herausbekommen sollen:
g2: $x = 3$;
- Zuletzt wird noch eine Variable ta1 für die Musterlösung der Aufgabe benötigt. Diese sollte eine Liste mit allen zu erwartenden Zwischenergebnissen bei der Umformung der Gleichung sein (inkl. g1 und g2):
ta1: [g1, $2 \cdot (x+2) = 10$, $x+2 = 5$, g2];

1 Aufgabenvariablen



```
g1: 2*(x+2)-6=4;  
g2: x=3;  
ta1: [g1, 2*(x+2)=10, x+2=5, g2];
```

Hinweise:

1. Sie können auch für jeden Zwischenschritt eine eigene Variable erstellen und diese dann in ta1 verwenden.
2. Wenn es mehrere gleichwertige Sequenzen von Umformungsschritten gibt, dann entscheiden Sie sich für eine.

STACK // Fragetext und Bereich „Eingabe“

- Schreiben Sie nun den **1 Fragetext**.

1 Fragetext

Bearbeiten Ansicht Einfügen Format Werkzeuge Tabelle Hilfe

↶ ↷ **B** *I*                       

Lösen Sie die Gleichung {@g1@} nach x auf, indem Sie für jeden Umformungsschritt eine eigene Zeile im Eingabefeld verwenden.

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

p 21 Wörter

- Wählen Sie im Bereich **2 Eingabe** den Eingabetyp „Equivalence reasoning“ aus.
- Verwenden Sie im Feld „Musterlösung“ die Variable ta1 (Liste mit allen Umformungsschritten).

2  Eingabe: ans1

Eingabetyp 

Equivalence reasoning 

Musterlösung 

ta1

STACK // Zusätzliche Optionen im Bereich „Eingabe“

- Für Eingabefelder des Typs „Equivalence reasoning“ gibt es folgende zusätzliche Optionen, die optional im entsprechenden **1Feld** ganz unten im Bereich „Eingabe“ eingetragen werden können (mit Kommata getrennt):
 - `firstline` → Bewirkt, dass sofort eine **2Fehlermeldung** in der Validierung angezeigt wird, wenn die Studierenden nicht die ursprüngliche Gleichung (oder den ursprünglichen Term) als erste Zeile des Eingabefelds verwenden. Dies ist vor allem in Kombination mit der Antwortüberprüfungsmethode „EquivFirst“ sinnvoll (siehe Folie [7](#)).
 - `hideequiv` → Bewirkt, dass den Studierenden nicht das Echtzeit-Feedback zur Korrektheit ihrer Umformungsschritte in Form der Symbole $\Leftrightarrow$, $?$ etc. angezeigt wird.
 - `hidedomain` → Bewirkt, dass den Studierenden keine Definitionsbereiche wie $x \in (-1, \infty)$ angezeigt werden.

Hinweis: Eine Verwendung von `hideequiv` und/oder `hidedomain` ist sinnvoll, wenn das „Equivalence reasoning“-Eingabefeld ausschließlich für die Eingabe der Umformungsschritte genutzt wird und keine Unterstützung für die Studierenden liefern soll.

1

Zusätzliche Optionen



$2 \cdot (x + 2) = 10$
 $x + 2 = 5$
 $x = 3$

$2 \cdot (x + 2) = 10$ Sie haben die erste Zeile falsch übernommen.

$x + 2 = 5$

$x = 3$

Diese Antwort ist ungültig.

In Ihrer Antwort wurden die folgenden Variablen gefunden: $[x]$

2

HRW

STACK // Rückmeldebaum (1)

- Im ersten Knoten des Rückmeldebaums können die standardmäßig eingetragenen Werte ans1 (SAns) und ta1 (TAns) beibehalten werden.

1 Knoten 1 ? Beschreibung

Antwortüberprüfung EquivFirst

SAns TAns Test-Optionen Feedback unterdrücken

ans1 ta1 Nein

- Für Eingabefelder des Typs „Equivalence reasoning“ gibt es zwei eigene Antwortüberprüfungsmethoden:
 - „EquivReasoning“:
 - Überprüft nur, ob die einzelnen Zeilen in der Eingabe der Studierenden im Sinne der Antwortüberprüfungsmethode „AlgEquiv“ algebraisch äquivalent zueinander sind
 - Es wird *nicht* überprüft, ob die enthaltenen Gleichungen oder Terme auch mit denen in „TAns“ übereinstimmen
 - STACK würde daher auch dann Punkte vergeben, wenn Studierende algebraisch äquivalente Ausdrücke angeben würden, die gar nicht im Zusammenhang mit der Aufgabenstellung stehen.
 - „EquivFirst“ (hier sinnvoller):
 - Überprüft ebenfalls, ob alle Zeilen in der Eingabe der Studierenden algebraisch äquivalent sind
 - Überprüft zudem, ob die erste Zeile der Studierenden bis auf Kommutativität und Assoziativität mit dem ersten Element der Liste übereinstimmt, die im „TAns“-Feld verwendet wird.

STACK // Rückmeldebaum (2)

- Mit einer Konfiguration wie auf der vorherigen Folie dargestellt würden die Studierenden demnach genau dann die volle Punktzahl für die Aufgabe bekommen, wenn folgende Aspekte erfüllt sind:
 1. Erste Zeile der Studierenden entspricht der ursprünglichen Gleichung
 2. Alle Zeilen sind zueinander algebraisch äquivalent
- Es wird noch nicht überprüft, ob die Studierenden die Umformung bis zum Ende durchgeführt haben!
- Fügen Sie dem Rückmeldebaum dazu einen zweiten Knoten hinzu und konfigurieren Sie diesen wie folgt:
 - SAns: $\text{last}(\text{ans1}) \rightarrow$ entspricht der letzten Zeile der Studierenden im Eingabefeld
 - TAns: $g2 \rightarrow$ Variable für die Gleichung, die herauskommen soll
 - Antwortüberprüfung: „EqualComAss“ $\rightarrow$ SAns und TAns müssen übereinstimmen bis auf Kommutativität und Assoziativität

1

Knoten 2

?

Beschreibung

Antwortüberprüfung

EqualComAss

SAns

last(ans1)

TAns

g2

Feedback unterdrücken

Nein

STACK // Rückmeldebaum (3)

- Konfigurieren Sie den Rückmeldebaum nun so, dass die Studierenden nur dann die volle Punktzahl erhalten, wenn die Überprüfung in beiden Knoten „wahr“ ergibt, zum Beispiel:
 - Wenn die Überprüfung in **1 Knoten 1** „wahr“ ergibt, werden noch keine Punkte vergeben und es wird Knoten 2 nachgeschaltet.

1	Knoten 1 wenn WAHR	?	Mod	=	Score	0	Abzüge	Nächster	Knoten 2	Antworthinweis	prt1-1-T
	Knoten 1 wenn FALSCH	?	Mod	=	Score	0	Abzüge	Nächster	[stop]	Antworthinweis	prt1-1-F

- Wenn die Überprüfung in **2 Knoten 2** „wahr“ ergibt, gibt es die volle Punktzahl. Es ist sinnvoll, für den Fall „falsch“ einen Feedbacktext einzutragen (z. B. „Die Gleichung kann noch weiter umgeformt werden.“).

2	Knoten 2 wenn WAHR	?	Mod	+	Score	1	Abzüge	Nächster	[stop]	Antworthinweis	prt1-2-T
	Knoten 2 wenn FALSCH	?	Mod	-	Score	0	Abzüge	Nächster	[stop]	Antworthinweis	prt1-2-F

STACK // Abschließende Hinweise

- Auch mit dem zweiten Knoten (Folie 8) wird noch nicht überprüft, ob die Studierenden auch die gleichen Zwischenschritte verwendet haben wie in ta1. Fügen Sie daher ggf. noch weitere Knoten hinzu, in denen Sie bspw. überprüfen, ob bestimmte Zwischenschritte von den Studierenden genannt wurden.
- In dieser Anleitung wurde der Eingabetyp „Equivalence reasoning“ anhand einer Beispielaufgabe erklärt, in der die Antwort im „Equivalence reasoning“-Eingabefeld von STACK bewertet wird. Vor allem für formatives Assessment kann es aber auch sinnvoll sein, den Studierenden ein solches Eingabefeld nur zur Unterstützung anzubieten, während die von STACK bewertete Antwort in ein normales algebraisches Eingabefeld eingetragen werden soll. Ein Beispiel für eine solche Aufgabe ist auf dem folgenden 1 **Screenshot** zu sehen.

1

Lösen Sie die Gleichung $2 \cdot (x + 2) - 6 = 4$ nach x auf.

Antwort:

Das folgende Eingabefeld dient Ihnen als Unterstützung. Sie können hier pro Zeile einen Umformungsschritt eintragen und bekommen sofort ein Feedback, ob die Umformungen jeweils korrekt sind. Ihre Eingaben in diesem Feld werden nicht zur Bewertung der Aufgabe herangezogen.

STACK // Moodle-Kurs „Tutorials STACK“

- Dort finden Sie Videotutorials, Beispielaufgaben, weitere Klickanleitung und mehr
- Link: <https://moodle.ruhr-uni-bochum.de/course/view.php?id=22225>
- Passend zu dieser Klickanleitung empfehlen wir Ihnen:
 - Videotutorial „Eingabetyp ‚Equivalence reasoning‘: Gleichungen und Terme umformen“



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 covers various aspects of STACK through video tutorials and example tasks. A final paragraph states that the course is public and accessible without login, but an account is needed for full functionality, with a link provided for account creation.

RUB Dashboard LEARN2Gether OpenRUB Support Teilnehmer/in

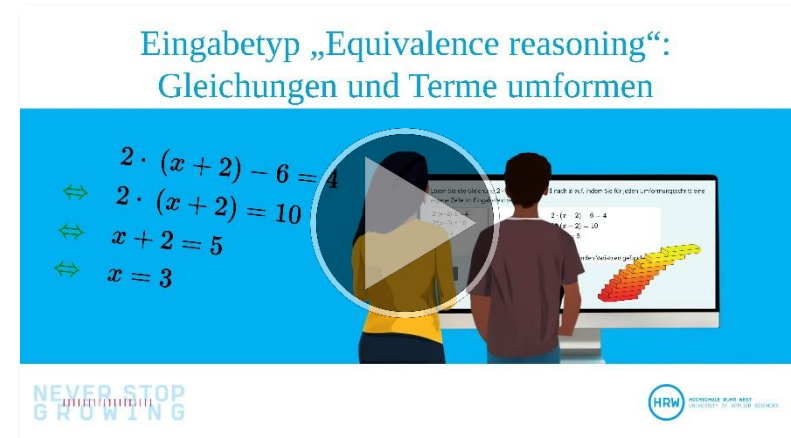
Alles einklappen

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 equations on the left: $2 \cdot (x + 2) - 6 = 4$, $2 \cdot (x + 2) = 10$, $x + 2 = 5$, and $x = 3$. On the right, there is an illustration of two people looking at a screen that displays a similar equation. A large play button icon is overlaid on the center. The title 'Eingabetyp „Equivalence reasoning“: Gleichungen und Terme umformen' is at the top. At the bottom left, it says 'NEVER STOP GROWING' and at the bottom right, there is an HRW logo with the text 'HOCHSCHULE RUHR-WEST UNIVERSITY OF APPLIED SCIENCES'.

Eingabetyp „Equivalence reasoning“:
Gleichungen und Terme umformen

$2 \cdot (x + 2) - 6 = 4$
 $\Leftrightarrow 2 \cdot (x + 2) = 10$
 $\Leftrightarrow x + 2 = 5$
 $\Leftrightarrow x = 3$

NEVER STOP GROWING

HRW HOCHSCHULE RUHR-WEST
UNIVERSITY OF APPLIED SCIENCES

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 1.0.a vom 02.10.2026

Empfohlene Zitation:

„Parametrisierbare Aufgaben mit STACK: Umformung von Gleichungen und Termen mit ‚Equivalence reasoning‘“, 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.

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 1.0.a

Urheber:in(nen):

Jonas Lache, Team E-Learning@Hochschule Ruhr West, Mail: elearning@hs-ruhrwest.de

Bildnachweise (einzeln):

– Logo des Assessment-Systems STACK: stack-assessment.org, CC BY-SA 4.0