

The background of the slide features a large, faint, circular seal of the University of Cologne. The seal depicts a religious scene with three crowned figures on the left, a kneeling figure in the center, and a seated figure on the right holding a child. The Latin text 'COLONIENSIS' is visible on the left side of the seal, and 'UNIVERSITATIS' is on the right. A shield with three crowns is at the bottom.

Für AutorInnen

# Einführung STACK 3.0

Die Verwendung der STACK-Frage in ILIAS

Eva Mix, Mario Josupeit, Tjark Eilts  
12.10.2020

# Inhaltsverzeichnis

Einleitung .....	4
Beitritt zum Kurs.....	5
Erstellung einer STACK-Frage .....	6
Flowchart.....	7
Die einzelnen Schritte der Fragenerstellung .....	8
STACK-Frage erstellen.....	8
Fragentext eingeben .....	8
Einstellungen der Eingabefelder .....	10
Rückmeldebaum erstellen.....	11
Erster Knoten.....	11
Neuen Knoten erstellen .....	14
Knoten verbinden.....	14
Neuen Rückmeldebaum erstellen .....	15
Abschließende Einstellungen .....	16
Einstellungen der STACK-Frage.....	18
Optionen → Zeige Frageoptionen.....	18
Test-Eingaben → Einstellungen der Eingabe .....	19
Rückmeldebäume (PRTS) .....	21
Eingabetypen .....	25
Algebraische Eingabe .....	25
Wahr/Falsch.....	25
Matrix .....	25
Einzelzeichen .....	25
Mehrzeilig.....	26
Kontrollkästchen (Multiple-Choice Aufgabe) .....	26
Drop-Down-Liste (Single-Choice Aufgabe) .....	27
Radio-Auswahl (Single-Choice Aufgabe).....	28
Gleichungen.....	28
Notizen .....	29
Einheiten .....	29
Zeichenkette .....	30
Numerisch.....	30
Antwortüberprüfungen .....	31
Kommentare einfügen.....	31
Maxima .....	32
Grundlegende Befehle .....	32

Allgemein .....	33
STACK spezifische Maxima-Befehle .....	33
Workarounds .....	34
MathJax .....	35
Grundlegende Befehle .....	36
HTML .....	37
Farbliche Gestaltung von Feedback .....	38
Automatische Gestaltung .....	38
Manuelle Gestaltung .....	38
Weiterführende Links .....	40
Übungsaufgaben .....	41
Übungsaufgaben zur algebraischen Eingabe .....	41
Übungsaufgaben zu anderen Eingabetypen .....	43
Übungsaufgaben für eine ansprechende Rückmeldung .....	45
Schritt-für-Schritt-Musterlösungen .....	46
Musterlösungen zur algebraischen Eingabe .....	46
Musterlösungen zu anderen Eingabetypen .....	54
Musterlösungen für eine ansprechende Rückmeldung .....	61
Kontakt .....	63
STACK-Team .....	63

## Einleitung

Die STACK-Frage ist ein Fragentyp in ILIAS, mit dem es möglich ist zu mathematischen Fragen mit Hilfe von Rückmeldebäumen detaillierte Bewertungen und Rückmeldungen zu geben. STACK steht für **S**ystem for **T**eaching and **A**ssessment using **C**omputer algebra **K**ernel. Dank dem angeschlossenen Computeralgebrasystem (CAS) können durch NutzerInnen eingegebene Antworten beispielsweise auf algebraische Gleichheit geprüft werden. Somit spielt die Schreibweise keine Rolle. Lediglich die getroffene Aussage ist relevant, was die STACK-Frage gut für mathematische Abfragen anwendbar macht. Des Weiteren können mittels verknüpfter Aufgabenstellungen verschiedene Kombinationen aus Teilaufgaben erstellt werden, die einen Einblick in den individuellen Wissensstand der Studierenden geben und die individuellen Rückmeldungen klären beide Seiten über den Lernerfolg auf.

Die STACK-Frage wurde zuerst auf der Lernplattform Moodle verwendet. Über STACK in Moodle gibt es viele gute und ausführliche Dokumentationen. Vieles wurde aus der STACK-Frage in Moodle für die STACK-Frage in ILIAS übernommen. Es gibt jedoch auch Unterschiede. Deswegen kann nicht alles direkt übernommen werden. Doch oft sind die Informationen, die man über STACK in Moodle findet eine Hilfe.

Aktuell gibt es an der Universität zu Köln die STACK-Frage unter:

<https://www.selfassessment.uni-koeln.de/>

<https://www.ilias.uni-koeln.de/>

## Beitritt zum Kurs

Auf ILIAS der Uni Köln finden Sie einen Kurs, der wichtige Informationen zur STACK-Frage zusammenträgt. In diesem Kurs können Sie die im Dokument beschriebenen Beispielfragen finden und eigene Fragen schreiben.

Link zu dem Kurs:

[https://www.ilias.uni-koeln.de/ilias/goto\\_uk\\_crs\\_3001967.html](https://www.ilias.uni-koeln.de/ilias/goto_uk_crs_3001967.html)

Klicken Sie auf die Verlinkung, melden Sie sich mit Ihrem Mitarbeiteraccount an, dann öffnet sich der entsprechende Kurs. Sie können beitreten mit dem **Passwort** STACK.

Alternativ können Sie den Kurs über das Magazin finden:

**Magazin → E-Learning & E-Assessment mit ILIAS → Einführung STACK 3**

Falls Sie versehentlich Beispielfragen verändern, melden Sie uns das bitte zurück, damit wir diese für künftige NutzerInnen wieder in den Ausgangszustand versetzen können.

Sie erreichen uns unter: **ilias-stack@uni-koeln.de**

## Erstellung einer STACK-Frage

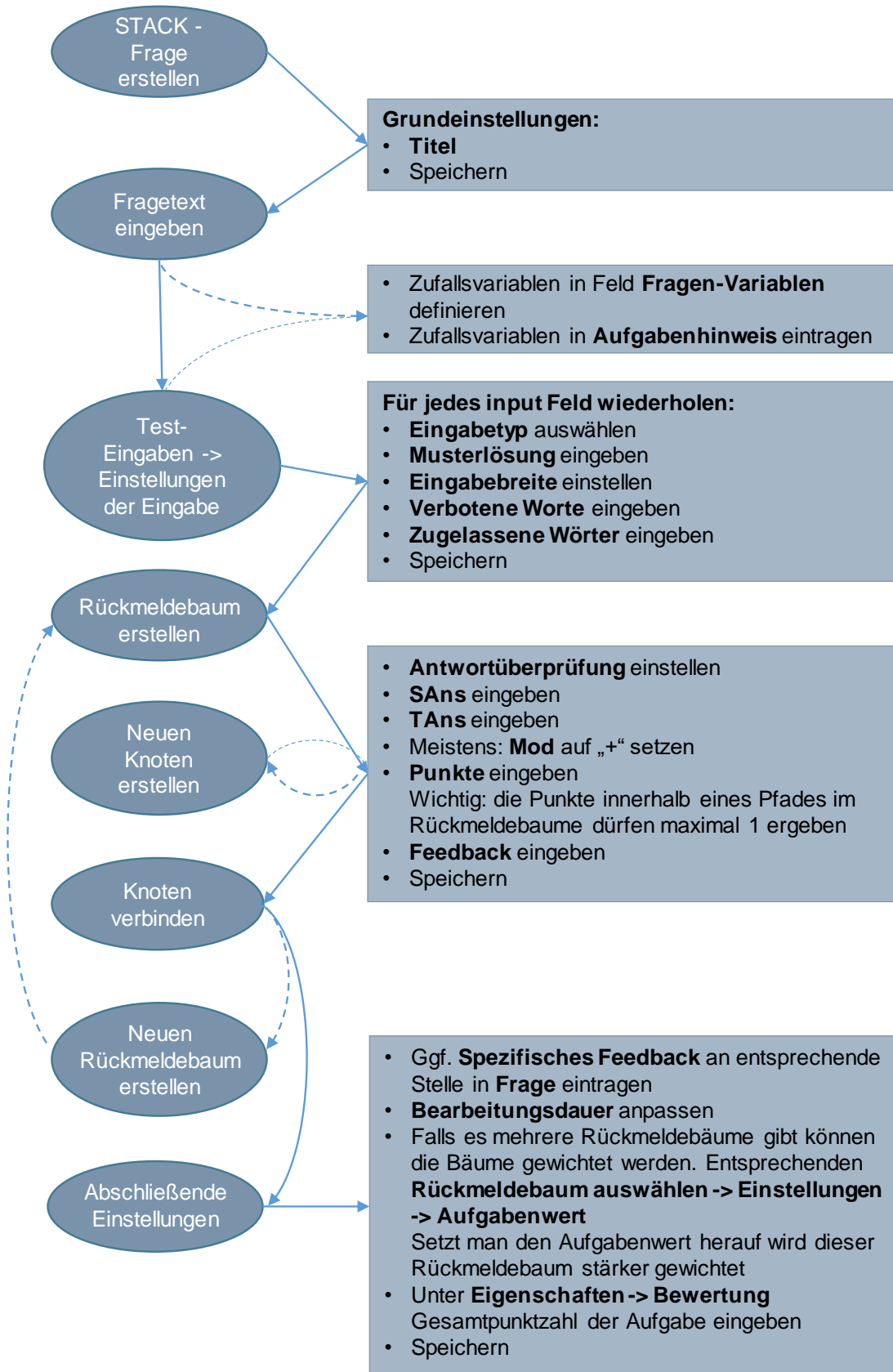
STACK-Fragen bieten viele Möglichkeiten, sind aber dementsprechend kompliziert in der Erstellung. Sofern Sie einige Hinweise beachten, können Sie jedoch schnell Fragen erstellen und erfahren, was STACK so besonders und nützlich macht. Auch hierbei gilt, dass aller Anfang schwer ist. Mit diesem Dokument soll der Einstieg erleichtert werden.

Auf der folgenden Seite finden Sie ein Flowchart, auf dem visualisiert ist, wie man eine STACK-Frage erstellt. Die Grafik ist kurzgehalten, sodass Sie diese auf einem Blatt ausdrucken und bei der Erstellung von Fragen zur Hand nehmen können. Die einzelnen Schritte und Einstellungen werden auf den Seiten danach ausführlicher erklärt.

In dem Einführungskurs haben Sie die Möglichkeit, selbst Fragen zu schreiben und auszuprobieren, was mit STACK möglich ist. Bitte erstellen Sie Ihre Fragen im **Sandkasten zum Ausprobieren**.

[https://www.ilias.uni-koeln.de/ilias/goto\\_uk\\_qpl\\_3001975.html](https://www.ilias.uni-koeln.de/ilias/goto_uk_qpl_3001975.html)

## STACK-Frage erstellen



# Die einzelnen Schritte der Fragerstellung

## STACK-Frage erstellen

1. Fragenpool → Frage erstellen → STACK-Frage



The screenshot shows a navigation bar with tabs: Fragen, Info, Einstellungen, Druckansicht, Metadaten, Export, and Rechte. The 'Fragen' tab is active. Below the tabs, a green button labeled 'Frage erstellen' is visible.

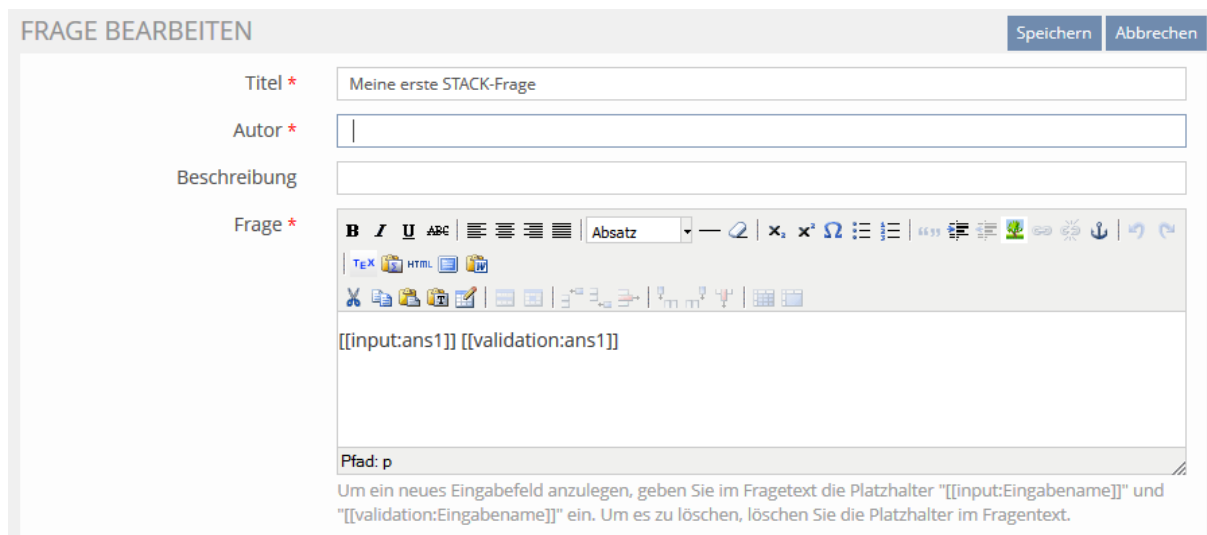


The screenshot shows the 'FRAGE ERSTELLEN' form. It has a navigation bar with tabs: Fragen, Info, Einstellungen, Druckansicht, Metadaten, Export, and Rechte. The 'Fragen' tab is active. Below the tabs, the text 'FRAGE ERSTELLEN' is displayed. There is a label 'Frage typ' and a dropdown menu showing 'STACK Frage'. At the bottom right, there are two buttons: 'Erstellen' and 'Abbrechen'.

2. Titel eingeben (der Titel kann später noch verändert werden)
3. Speichern

## Frage text eingeben

Geben Sie den Fragetext unter Frage bearbeiten → Frage ein. Dies ist der Text, der den Studierenden angezeigt wird, wenn sie die Frage bearbeiten.



The screenshot shows the 'FRAGE BEARBEITEN' form. It has a navigation bar with tabs: Fragen, Info, Einstellungen, Druckansicht, Metadaten, Export, and Rechte. The 'Fragen' tab is active. Below the tabs, the text 'FRAGE BEARBEITEN' is displayed. There are two buttons: 'Speichern' and 'Abbrechen'. The form contains several fields: 'Titel \*' with the value 'Meine erste STACK-Frage', 'Autor \*' with an empty field, 'Beschreibung' with an empty field, and 'Frage \*' with a rich text editor. The rich text editor has a toolbar with various icons for text formatting (bold, italic, underline, link, unlink, list, etc.) and a text area containing the placeholder text '[[input:ans1]] [[validation:ans1]]'. Below the text area, there is a label 'Pfad: p' and a note: 'Um ein neues Eingabefeld anzulegen, geben Sie im Fragetext die Platzhalter "[[input:Eingabename]]" und "[[validation:Eingabename]]" ein. Um es zu löschen, löschen Sie die Platzhalter im Fragetext.'

In diesem Feld können Fragen-Variablen aufgerufen, LaTeX verwendet und der Text mit Hilfe von HTML gestaltet werden.

1. `[[input:Eingabename]]` erstellt ein Eingabefeld und benennt die gegebene Antwort mit *Eingabename*, um später darauf zurückzugreifen (dieses Feld ist notwendig).
2. `[[validation:Eingabename]]` erstellt ein Validierungsfeld, um anzeigen zu lassen, wie Maxima die eingegebene Antwort interpretiert (dieses Feld ist optional).



3. Falls Sie für die Aufgabe mehrere Eingabefelder benötigen, so erstellen Sie diese, indem Sie erneut `[[input:Eingabename]]` eintippen. Die einzelnen Felder benötigen paarweise verschiedene Eingabenamen. Am besten wählen Sie aussagekräftige Benennungen, da Sie die Eingaben später für die Rückmeldungen aufrufen müssen. Alternativ können Sie die Eingabefelder auch durchnummerieren.
4. Sofern Sie (Zufalls-)Variablen verwenden, müssen diese im Feld **Fragen-Variablen** definiert werden ([Maxima-Code](#) verwenden).

Fragen-Variablen

Dieses Feld erlaubt es, CAS-Variablen zu definieren und zu verändern, z.B. um Zufallsversionen zu ermöglichen. Diese Variablen sind in allen anderen Teilen der Aufgabe verfügbar.  
Inhalt: CAS-Ausdruck

- Fragen-Variablen werden mit Hilfe von Doppelpunkten definiert.  
Beispiel: `Var1:5` oder `Var2:rand(9)`  
*Hinweis:* Variablen-Namen können aus Buchstaben und Zahlen bestehen. Wichtig ist dabei jedoch, dass nach Zahlen keine weiteren Buchstaben folgen. So ist beispielsweise `Var1a` kein gültiger Variablen-Name.
- (Zufalls-)Variablen können Sie über `{@Variablennamen@}` (ab STACK 3.0) aufrufen. Somit ist es möglich diese im Fragetext zu verwenden.  
Weiter unten in diesem Dokument finden Sie eine [Erklärung zu den Zufallsfunktionen von Maxima](#). Ebenfalls finden Sie [ein Selbstlernbeispiel](#) weiter unten in diesem Dokument.
- Alle definierten Fragen-Variablen in das Feld **Aufgabenhinweis** eingetragen werden. Das insbesondere bei der Verwendung von Zufallsvariablen wichtig, da der Aufgabenhinweis gemeinsam mit den bearbeiteten Aufgaben gespeichert wird. So können Sie nachvollziehen, welche Zahl die Studierenden in der Aufgabe erhalten haben.

Aufgabenhinweis

Mit dem Aufgabenhinweis soll zwischen den verschiedenen zufälligen Versionen einer Frage unterschieden werden. Zwei Fragen sind gleich, genau dann wenn die Aufgabenhinweise gleich sind. Für die spätere Analyse ist es hilfreich sinnvolle Antworthinweise zu erstellen.  
Inhalt: CAS-Text

## 5. Speichern

*Tipp:* Geben Sie die Formeln in LaTeX-Code ein, dann werden sie den Studierenden schöner angezeigt. Erläuterungen dazu finden Sie in diesem Dokument in dem Kapitel [MathJax](#).

## Einstellungen der Eingabefelder

Nach dem Speichern des Fragentextes erscheint im Abschnitt **Test-Eingaben** → **Einstellungen der Eingabe** für jedes Eingabefeld die Möglichkeit Einstellungen zu treffen.

### TEST-EINGABEN

► Einstellungen der Eingabe ans1 \*

Die Grundeinstellungen, die Sie dort treffen sollten, werden in diesem Abschnitt kurz erklärt. Detaillierte Erklärungen zu den einzelnen Einstellungen finden Sie in diesem Dokument im Abschnitt [Einstellungen der STACK-Frage](#).

Für jedes Eingabefeld müssen folgende Einstellungen getroffen werden:

1. **Eingabetyp** auswählen

Hier wird eingestellt, welche Form die Eingabe der Studierenden haben soll, damit STACK die Eingabe entsprechend auswerten kann. Eine ausführliche Erklärung zu den verschiedenen Eingabetypen finden Sie in diesem Dokument im Kapitel [Eingabetypen](#).

2. **Musterlösung** eingeben

Die Musterlösung zu dem Eingabefeld muss eingegeben werden. Dies ist die Lösung, die im Feld **Die bestmögliche Lösung lautet** angezeigt wird. Dieses Feld erscheint nach der Beantwortung der Frage.

Die Musterlösung wird auch verwendet, um bei Eingabetypen wie Drop-Down und Matrix das Layout der Eingabe zu bestimmen.

*Achtung:* Die Musterlösung ist nicht das, was zur Antwortüberprüfung genutzt wird. Sie können die Musterlösung im Rückmeldebaum in das Feld **TAns** kopieren und somit zu Antwortüberprüfung nutzen, müssen dies jedoch nicht tun.

3. **Eingabebreite** einstellen

Die Eingabebreite gibt an, wie breit das entsprechende Eingabefeld in der fertigen Frage sein wird.

4. **Verbotene Worte** eingeben

In dieses Feld werden Zeichenketten eingetragen, die Studierende nicht in ihrer Antwort verwenden dürfen. Das wird wichtig, sobald die Lösung mit Hilfe von Maxima-Befehlen eingegeben werden könnte.

5. **Gleitkomma verbieten**

Wir empfehlen diese Option zu deaktivieren. Falls Sie diese Option aktivieren, benötigen die Studierenden den Hinweis, dass sie alle Zahlen als Bruch oder als ganze Zahlen schreiben müssen.

6. **Zugelassene Wörter** eingeben

Alle Zeichenketten, die länger als zwei Zeichen sind werden standardmäßig als Eingabe abgelehnt. Sollten Sie eine solche Eingabe fordern, muss die Zeichenkette

hier eingetragen werden, damit Studierende diese für die richtige Antwort verwenden dürfen.

## Rückmeldebaum erstellen

Die Rückmeldebäume ermöglichen es individuelle und detaillierte Rückmeldungen zu geben. Die Rückmeldebäume sind das, was die STACK-Fragen besonders macht. Basierend auf dem, was eingegeben wurde, können passende Rückmeldetexte ausgegeben werden. Ebenso kann die Bewertung entsprechend angepasst werden.

Der Umfang der Rückmeldebäume ist abhängig von dem Detailreichtum, der von Ihnen eingestellten Rückmeldungen. Beispiele zum Aufbau von Bäumen und den damit verbundenen Möglichkeiten finden Sie in diesem Dokument unter [Übungsaufgaben](#).

*Hinweis:* Der Name von Rückmeldebäumen kann ausschließlich vor deren ersten Speicherung geändert werden.

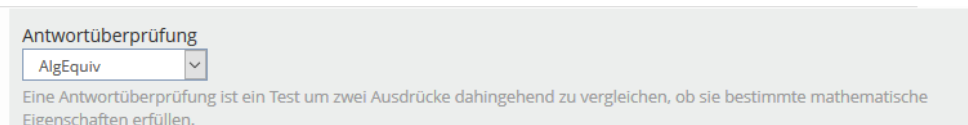


## Erster Knoten

Für den ersten Knoten des Rückmeldebaumes muss Folgendes eingestellt werden:

### 1. Antwortüberprüfung

Mit der Antwortüberprüfung stellen Sie ein mit Hilfe welcher logischer Eigenschaften **SAns** und **TAns** verglichen werden sollen. Weiter Informationen hierzu finden Sie im Abschnitt [Antwortüberprüfungen](#).



## 2. SAns

**SAns** ist das erste Element der Antwortüberprüfung. Meistens ist dies die Studierenden-Antwort, obwohl jeder gültige CAS-Ausdruck möglich ist.

Um hier die Studierenden-Antwort zu überprüfen, tragen Sie *Eingabename* des entsprechenden Eingabefeldes in **SAns** ein. Beispiele hierzu finden Sie in diesem Dokument unter [Übungsaufgaben](#).

**SAns \***  
  
Dies ist das erste Argument der Antwortüberprüfungsfunktion. In asymmetrischen Tests wird dies als die Studierenden-Antwort angesehen, obwohl es jeder gültige CAS Ausdruck sein könnte. Es können darin auch Variablen aus der Aufgabe oder dem Feedback benutzt werden.  
Inhalt: CAS-Ausdruck

## 3. TAns

**TAns** ist das zweite Element der Antwortüberprüfung. Oft kopiert man in **TAns** die Musterlösung des Eingabefeldes. **TAns** und die **Musterlösung** sind allerdings unabhängig voneinander. So können Sie in **TAns** auch etwas anderes als die Musterlösung eingeben. Diese Option kann vor allem für die Einrichtung einer detailreichen Fehlerrückmeldung genutzt werden. Beispiele hierzu finden Sie ebenfalls unter [Übungsaufgaben](#).

**TAns \***  
  
Dies ist das zweite Argument der Antwortüberprüfungsfunktion. In asymmetrischen Tests wird dies als die Dozierenden-Antwort angesehen, obwohl es jeder gültige CAS Ausdruck sein könnte. Es können darin auch Variablen aus der Aufgabe oder dem Feedback benutzt werden.  
Inhalt: CAS-Ausdruck

## 4. Wahr oder falsch (grün oder rot)

**SAns** und **TAns** werden entsprechend der im Feld **Antwortüberprüfung** eingestellten logischen Eigenschaften verglichen. In den üblichsten Fällen gilt: Stimmen **SAns** und **TAns** überein, ist der Knoten wahr. Weitere Einstellungen für diesen Fall werden dann in dem grün unterlegten Feld getroffen. Stimmen **SAns** und **TAns** nicht überein, ist der Knoten falsch. Die weiteren Einstellungen sind für diesen Fall in dem roten Feld zu tätigen.

Ob ein Knoten wahr oder falsch ist, sagt nicht unbedingt etwas darüber aus, ob die Frage richtig beantwortet wurde oder nicht. Beispielsweise können Sie mit einem Knoten prüfen, ob ein bestimmter Fehler in der Aufgabe gemacht wurde. Wurde eben dieser Fehler gemacht, wäre der Knoten wahr. Die Aufgabe selbst wäre allerdings falsch beantwortet.

<b>Mod</b>	<input type="text" value="="/>	<b>Mod</b>	<input type="text" value="="/>
	<small>Wie die Bepunktung angepasst wird. "=" setzt die Punkte auf einen bestimmten Wert. "+/-" addieren oder subtrahieren Punkte von der aktuellen Summe.</small>		<small>Wie die Bepunktung angepasst wird. "=" setzt die Punkte auf einen bestimmten Wert. "+/-" addieren oder subtrahieren Punkte von der aktuellen Summe.</small>
<b>Punkte wenn wahr</b>	<input type="text" value="0"/>	<b>Punkte wenn falsch</b>	<input type="text" value="0"/>
	<small>Bewertung wenn der Antworttest "wahr" ergibt.</small>		<small>Bewertung wenn der Antworttest "falsch" ergibt.</small>

## 5. Mod

Der Modus bestimmt, wie die Punkte berechnet werden. „=“ setzt die Punkte für den Baum auf den Wert, der an diesem Ast des Knotens eingegeben wurde. Die vorher gegebenen Punkte werden überschrieben.

„-“ zieht die in dem Ast des Knotens angegebenen Punkte von den bisher in dem Rückmeldebaum vergebenen Punkten ab.

„+“ addiert die in dem Ast des Knotens angegebenen Punkte zu den bisher in dem Rückmeldebaum vergebenen Punkten.

*Hinweis:* In den meisten Fällen ist es sinnvoll den Modus „+“ zu wählen, damit die bereits in dem Baum vergebenen Punkte nicht überschrieben werden.

## 6. Punkte

*Wichtig:* Die Punkte, die für einen Pfad im Rückmeldebaum vergeben werden, sollten in der Summe maximal eins ergeben. Wobei die Summe 1 bedeutet, dass die volle Punktzahl für diesen Baum vergeben wird und die Summe 0 bedeutet, dass für diesen Baum keine Punkte vergeben werden.

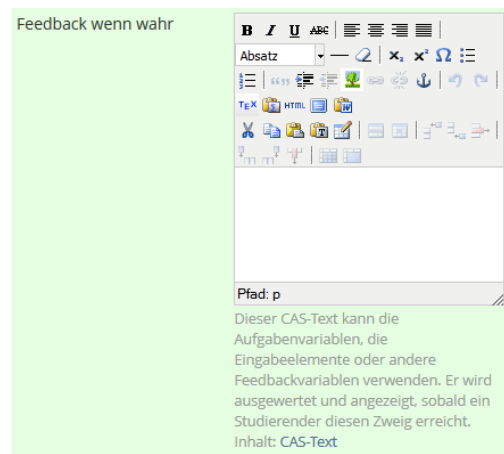
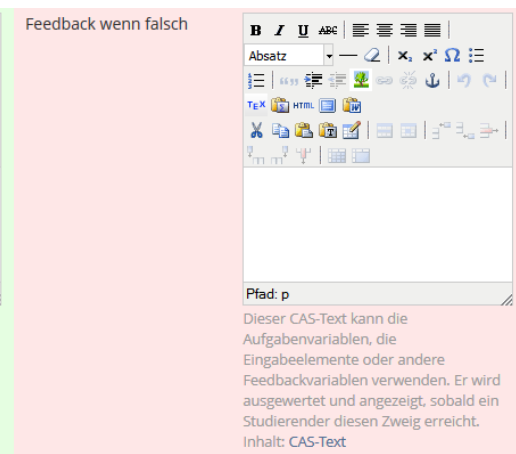
*Tipp:* Erstellen Sie zuerst alle Knoten, verbinden diese und kümmern sich am Ende um die Bepunktung. So ist es leichter zu erkennen, ob die Punkte in einem Pfad wirklich maximal eins ergeben. Eine ausführliche Erklärung dazu, wie Sie bei der Bepunktung vorgehen müssen, finden Sie in diesem Dokument unter [Abschließende Einstellungen](#) und in den [Übungsaufgaben](#).

(Anmerkung: Abzüge funktionieren momentan nicht)

## 7. Feedback eingeben

Unter **Feedback wenn wahr/falsch:** wird das Feedback eingegeben, welches die Studierenden sehen, wenn der entsprechende Pfad des Baumes genommen wurde. Hier können Sie individuelle Rückmeldungen und Hinweise geben.

In diesem Feld können Fragen-Variablen aufgerufen, [LaTeX](#) verwendet und der Text mit Hilfe von [HTML](#) gestaltet werden.

Feedback wenn wahr	Feedback wenn falsch
	

## 8. Speichern

### Neuen Knoten erstellen

Falls der Rückmeldebaum weitere Knoten benötigt, erstellen Sie diese im Abschnitt **Rückmeldebäume (PRTS)** → **Neuer Knoten**. Diese Schaltfläche befindet sich neben den Nummern der bereits erstellten Knoten.

Die Nummerierung der Knoten hängt von der Reihenfolge ab, in der sie erstellt wurden. Sie sagt nichts darüber aus in welcher Reihenfolge die Knoten durchlaufen werden. Die Durchlaufreihenfolge wird durch die [Verbindungen der Knoten](#) bestimmt.

Für jeden weiteren Knoten müssen Sie, wie bei dem ersten Knoten, die entsprechenden Einstellungen treffen. Danach sollten die Knoten miteinander verbunden werden.

### Knoten verbinden

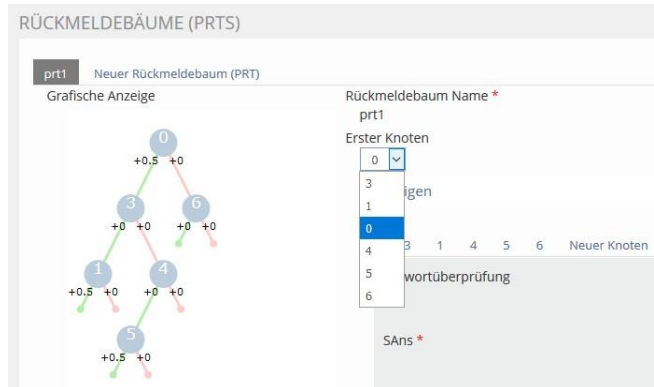
Durch die Verbindung der Knoten stellen Sie ein, in welcher Reihenfolge die Knoten durchlaufen werden. Erst hierdurch gestalten Sie einen sinnvollen Aufbau des Rückmeldebaumes.

*Hinweis:* Neu erstellte Knoten müssen gespeichert werden, bevor Sie diese verbinden können.

1. Wählen Sie den Knoten, der verbunden werden soll, aus.
2. Tragen Sie ein mit welchem Knoten fortgefahren werden soll, nachdem er durchlaufen wurde.
  - **Knoten wenn wahr** gibt an, mit welchem Knoten fortgefahren werden soll, falls die Auswertung des Knotens wahr ist.
  - **Knoten wenn falsch** gibt an, mit welchem Knoten fortgefahren werden soll, falls die Auswertung des Knotens falsch ist.
  - Stellt man hier nichts ein endet der Baum, nachdem der Knoten durchlaufen wurde. Unverbundene Knoten werden nicht durchlaufen.

### 3. Ersten Knoten auswählen

Unter **Rückmeldebäume** → **Erste Knoten** können Sie im Drop-Down Menü auswählen, mit welchem Knoten begonnen werden soll. Standardmäßig wird mit dem Knoten mit der niedrigsten Nummer begonnen.



### 4. Speichern

Sofern Sie die Knoten erfolgreich verbunden haben, werden die Verbindungen zusammen mit der entsprechenden Punktevergabe links in der **Grafischen Anzeige** visualisiert.

## Neuen Rückmeldebaum erstellen

1. Klicken Sie auf **Neuer Rückmeldebaum (PRT)**, um einen neuen Rückmeldebaum zu erstellen.
2. Gehen Sie danach wie beim ersten Rückmeldebaum vor
  1. Knoten einstellen
  2. Neue Knoten erstellen
  3. Knoten verbinden

Mehrere Rückmeldebäume zu erstellen kann sinnvoll sein, wenn verschiedene Aufgabenteile unterschiedlich in der Bewertung gewichtet werden sollen. Wie das geht finden Sie in diesem Dokument unter [Abschließende Einstellungen → Gewichtung der Rückmeldebäume](#).

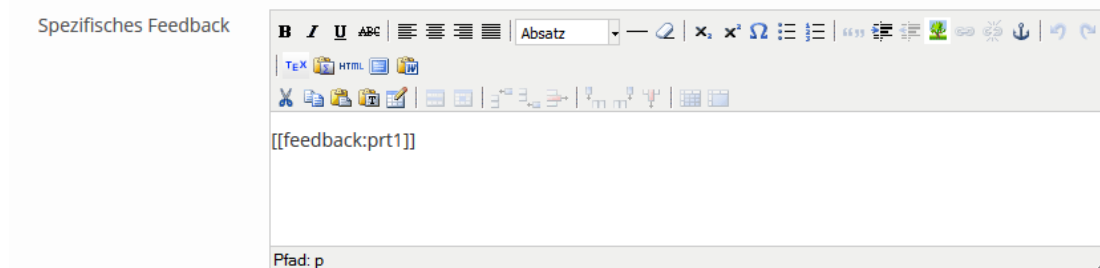
Es ist notwendig mehrere Rückmeldebäume zu erstellen, wenn Sie möchten, dass das Feedback zu einzelnen Aufgabenteilen direkt darunter angezeigt wird. Wie Sie dabei vorgehen finden Sie im folgenden Abschnitt unter [Abschließende Einstellungen → Feedback eintragen](#).

## Abschließende Einstellungen

### 1. Feedback eintragen

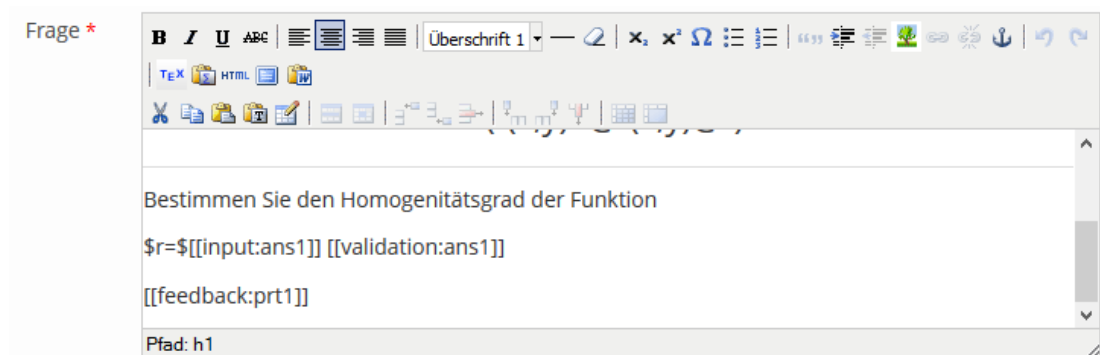
Standardmäßig wird das Feedback nur am Ende der Aufgabe angezeigt. Falls Sie mehrere Rückmeldebäume verwenden, kann das Feedback direkt unter einzelnen Aufgabenteilen angezeigt werden.

Sobald ein neuer Rückmeldeb Baum gespeichert wird, wird das Feedback für den Baum im Feld **Spezifisches Feedback** angezeigt. In diesem Feld steht das Feedback, welches am Ende der Aufgabe angezeigt wird.



Es ist möglich, das Feedback in den Aufgabentext zu verschieben. Somit kann es direkt unter dem entsprechenden Aufgabenteil stehen. Hierzu schneiden Sie `[[feedback:Rückmeldebbaumname]]` aus und fügen es an der entsprechenden Stelle im Fragentext ein.

*Tipp:* Bei Aufgaben mit mehreren Teilaufgaben ist es meistens sinnvoll, das Feedback in den Aufgabentext zu schieben, damit eindeutig ist, auf welchen Aufgabenteil sich das Feedback bezieht.



### 2. Bearbeitungsdauer anpassen

Besonders für komplexe Aufgaben wird die standardmäßig eingestellte Bearbeitungsdauer von einer Minute zu kurz sein. Stattdessen sollte hier die erwartete Bearbeitungsdauer der Studierenden angegeben werden. Mit Hilfe der eingegebenen Bearbeitungsdauern kann die Gesamtbearbeitungsdauer eines Tests abgeschätzt werden. Für die tatsächliche Bearbeitungsdauer hat diese Eingabe aber keine Auswirkungen. Es dient lediglich zur Abschätzung der Bearbeitungsdauer für Tests.

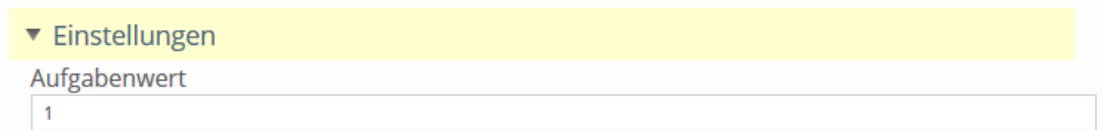
Bearbeitungsdauer    Stunden:     Minuten:     Sekunden:



### 3. Gewichtung der Rückmeldebäume festlegen

Falls es mehrere Rückmeldebäume gibt, können Sie die Bäume gewichten.

Gehen Sie wie folgt vor: **Rückmeldebaum auswählen** → **Einstellungen** → **Aufgabenwert**



Setzen Sie den Aufgabenwert herauf, so wird dieser Rückmeldebaum stärker gewichtet. Setzen Sie also beispielsweise den Aufgabenwert eines Baumes auf 4 so zählt dieser Baum vier Mal so viel wie alle anderen Rückmeldebäume, die einen Aufgabenwert von 1 haben. Setzen Sie den Wert auf 0.5 so zählt der Baum nur halb so viel.

Die Gewichtung können Sie für jeden Baum separat einstellen.

Sie müssen die Einstellung anschließend **speichern**. Unter **Bewertung** ist zu sehen, wie sich die Punktverteilung verändert hat.

### 4. Gesamtpunktzahl einstellen

- Gehen Sie auf den Reiter **Eigenschaften** → **Bewertung**.



- Geben Sie in das Feld **Neue Punktzahl** ein, wie viele Punkte die Aufgabe insgesamt geben soll.
- Der Button **Zeige Vergleich** bewirkt, dass die neue und die alte Punktzahl verglichen werden.
- **Punktzahl speichern** übernimmt die in das Feld **Neue Punktzahl** eingetragenen Punkte.

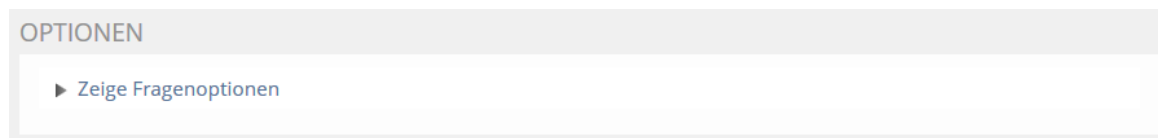
### 5. Speichern

# Einstellungen der STACK-Frage

Hier werden die einzelnen Einstellungen der STACK-Frage mitsamt deren Besonderheiten im Detail erklärt.

## Optionen → Zeige Frageoptionen

Im Abschnitt **Optionen** befindet sich das Drop-Down Menü **Zeige Frageoptionen**. Hier werden Grundeinstellungen für die Aufgabe getroffen.



- Hier befindet sich zuerst die **Aufgabenweite-Vereinfachung**. Aktivieren Sie diese Option, so werden eingegebene Ausdrücke der gesamten STACK-Frage vereinfacht. Dies bezieht sich insbesondere auf die Fragen-Variablen.  
Für die Rückmeldebäume können Sie die Vereinfachung unabhängig davon ein- oder ausstellen. Um die Vereinfachung im Rückmeldebaum auszustellen, setzen Sie im entsprechenden Rückmeldebaum unter **Rückmeldebaum** → **Einstellungen** → **Vereinfachung** die Vereinfachung auf **Nein**.  
Soll die Vereinfachung nur für bestimmte Fragen-Variablen ein- oder ausgestellt werden, können Sie den Befehl *simp:true* (dann ist die Vereinfachung an) oder *simp:false* (dann ist die Vereinfachung aus) in dem Feld Fragen-Variablen vor der entsprechende Definition der Variable nutzen.  
*Hinweis:* Ist **Aufgabenweite-Vereinfachung** aktiviert kann es sein, dass die Fragen-Variablen im Nachhinein dennoch vereinfacht werden.
- **Positiv annehmen** führt dazu, dass ein eingegebener Parameter, dessen Vorzeichen unbekannt ist, stets als positiv angenommen wird. (Für genauere Informationen siehe Kapitel 11.3 “Functions and Variables for Facts” (assume\_pos) der Maxima-Dokumentation.)  
[http://maxima.sourceforge.net/docs/manual/maxima\\_59.html#Functions-and-Variables-for-Facts](http://maxima.sourceforge.net/docs/manual/maxima_59.html#Functions-and-Variables-for-Facts)
- Darunter befinden sich Felder, in die Sie **Rückmeldungen** eingetragen können. Diese Rückmeldungen erscheinen abhängig davon, ob die Aufgabe richtig, teilweise richtig oder falsch beantwortet wurde. Wir empfehlen, diese Felder leer zu lassen, da wir stattdessen eine umfangreiche Rückmeldung durch den Rückmeldebaum generieren werden. Es kann für die Studierenden verwirrend sein, wenn sie mehrere Rückmeldungen bekommen, die scheinbar nicht zusammenpassen oder sich doppeln.

*Hinweis:* Sie werden diese Felder in der ILIAS Versionen der UzK in der Regel leer vorfinden. Dies ist jedoch nicht notwendigerweise in allen STACK-Versionen so geregelt.

- Anschließend folgen einige grundsätzliche Einstellungen zu der Darstellung der Frage. Hier können Sie die Anzeige der **Multiplikations-Zeichen**, der **Wurzelzeichen**, von  **$\sqrt{-1}$** , der **inversen trigonometrischen Funktionen** sowie der **Matrix-Klammern** einstellen.

Wie die Einstellungen vorzunehmen sind und was die Unterschiede sind erklärt sich von selbst, wenn Sie die Auswahlmöglichkeiten ausprobieren.

- In das Eingabefeld **Allgemeines Feedback** am Ende des Abschnittes **Optionen** können Sie Feedback eingetragen, das immer, unabhängig von den gegebenen Antworten, unter der bestmöglichen Lösung angezeigt wird. Hier können Sie beispielsweise eine ausführliche Erklärung für die Lösung angeben.

## Test-Eingaben → Einstellungen der Eingabe

Im Abschnitt **Test-Eingaben** haben Sie die Möglichkeit für jedes Eingabefeld im Fragentext, Einstellungen für das entsprechende Feld zu treffen. Klappen Sie die **Einstellungen der Eingabe** auf, um die Optionen für das entsprechende Eingabefeld festzulegen.

### TEST-EINGABEN

► Einstellungen der Eingabe ans1 \*

**Eingabetyp** gibt an in welcher Form die Studierenden ihre Antworten eingeben sollen, damit STACK die Eingabe entsprechend auswerten kann. Die einzelnen Optionen werden im Kapitel [Eingabetypen](#) erklärt.

- Die **Musterlösung** muss für jedes Eingabefeld eingegeben werden. Sie wird den Studierenden, nachdem die Rückmeldung angefordert wurde unter **Die bestmögliche Lösung lautet** angezeigt. Die Musterlösung wird in Maxima-Code angegeben und kann vorher definierte Fragen-Variablen enthalten.  
Das Musterlösungsfeld besitzt je nach Eingabetyp auch noch andere Aufgaben. Dies ist im Kapitel [Eingabetypen](#) genauer erklärt.
- Die **Eingabebreite** stellt ein, wie groß das entsprechende Eingabefeld angezeigt werden soll.

- Ist **Strikte Syntax** aktiviert, wird strikte Maxima-Syntax bei der Eingabe erfordert, sodass Maxima das Eingebene eindeutig interpretieren kann.
- Die Option **Sterne einfügen** bewirkt, dass Sterne automatisch als Multiplikationszeichen in die Eingabe eingefügt werden, sofern die Studierenden das vergessen haben sollten. Hierbei kann es allerdings zu Fehlinterpretationen kommen.
- Unter **Syntax-Hinweis** können Sie einen Hinweis einfügen, der erscheint, sofern das Antwortfeld von Studierenden leer gelassen wird.
- **Verbotene Worte** sind Zeichenketten, die in der Eingabe verboten sind. Dies wird beispielsweise wichtig, sobald Maxima-Befehle bei der Lösung helfen könnten. Denn es ist möglich, Maxima-Ausdrücke in das Eingabefeld zu schreiben, die für die Antwort interpretiert werden. Damit die Antwort nicht mit Hilfe von Befehlen erzielt werden kann, sollten eben diese Befehle/Wörter/Zeichen unter **Test-Eingaben** → **Einstellungen der Eingabe** → **Verbotene Worte** eingegeben werden. Mehrere Wörter werden mit Kommas voneinander getrennt.
- Ist **Gleitkomma verbieten** aktiviert, wird jede Gleitkommazahl in der eingegebenen Antwort als falsch ausgewiesen. Falls Sie diese Option aktivieren, benötigen die Studierenden den Hinweis, dass sie Zahlen als Bruch bzw. Integer schreiben müssen.
- **Wörter zulassen** ist eine Funktion aus älteren STACK-Versionen. Sie wird nicht mehr benötigt.
- Ist **Verlange vollständige Kürzung** aktiviert, muss die Eingabe als kleinstmöglicher Term geschrieben werden. Ansonsten wird sie als falsch abgewiesen. Diese Option funktioniert jedoch ausschließlich bei Brüchen.
- Ist **Antwort-Typ überprüfen** aktiviert werden Antworten mit einem anderen Typ als erwartet direkt als ungültig abgewiesen, bevor der Rückmeldebaum versucht diese auszuwerten.
- **Verifizierung erforderlich** funktioniert momentan nur in Kombination mit **Validierung anzeigen**.
- **Validierung anzeigen** funktioniert momentan nur in Kombination mit **Verifizierung erforderlich**. Diese Einstellung bewirkt, dass validierte Ausdrücke in zweidimensionaler

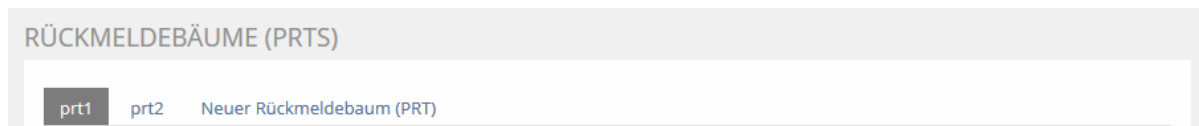
Schreibweise angezeigt werden. Zweidimensionale Schreibweise bedeutet, dass es so geschrieben ist, wie man es auch auf ein Blatt Papier schreiben würde.

- **Zusätzliche Optionen** werden für einige Eingabetypen benötigt. Für gewöhnlich muss hier aber nichts eingetragen werden.

## Rückmeldebäume (PRTS)

Im Abschnitt Rückmeldebäume (engl. potential response trees) werden die Rückmeldebäume der Aufgabe bearbeitet.

Mittels der Reiter können Sie zwischen den verschiedenen Bäumen der Aufgabe umschalten. Ein Klick auf **Neuer Rückmeldebaum (PRT)** erstellt einen neuen Rückmeldebaum.



Sobald Sie einen Namen eingegeben *und* gespeichert haben, erscheint der neue Baum als Reiter in der Auswahl. Die Bäume dienen dazu Feedback zu geben und verschiedene Aufgabenteile auszuwerten. Jeder Rückmeldebaum bekommt sein eigenes Feedback im Abschnitt **Spezifisches Feedback**. Dieses Feedback kann auch in den Abschnitt **Frage** verschoben werden. Wie dies geht finden Sie unter [Abschließende Einstellungen → Feedback eintragen](#).

- **Erster Knoten:** Legt den ersten Knoten des Baumes fest. Alle gespeicherten Knoten können der erste Knoten sein. Achten Sie darauf, dass alle Knoten vom ersten aus erreichbar sind, so dass der Baum vollständig durchlaufen werden kann.

Durch das Aufklappen von **Einstellungen** erscheinen folgende Optionen:

- **Aufgabenwert:** Gibt die Gewichtung des Rückmeldebaumes an. Eine ausführliche Erklärung finden Sie in diesem Dokument unter [Abschließende Einstellungen → Gewichtung der Rückmeldebäume](#).
- **Vereinfachung:** Falls Sie Vereinfachung auf **Ja** setzen, werden alle Variablen und Ausdrücke, die im Rückmeldebaum definiert und aufgerufen werden, vereinfacht. Die Einstellung hier ist unabhängig von der Einstellung unter **Optionen → Zeige Frageoptionen → Aufgabenweitevereinfachung**.

*Hinweis:* Ist **Aufgabenweite-Vereinfachung** aktiviert kann es sein, dass die Variablen im Rückmeldebaum im Nachhinein dennoch vereinfacht werden.

- **Feedback-Variablen:** Wie im Feld **Fragen-Variablen** werden hier Variablen oder Funktionen definiert. Hierbei können Variablen und Funktionen aus den **Fragen-Variablen** verwendet werden. Die Verwendung hier definierter Variablen ist auf den jeweiligen Rückmeldebaum beschränkt.
- **Lösche Rückmeldebaum** löscht den Baum mitsamt aller darin enthaltenen Knoten.

Die Knoten des aktuell ausgewählten Baumes können ebenfalls über Reiter ausgewählt werden.

Der letzte Reiter ist dazu da einen neuen Knoten zu erstellen.



Um einen **Neuen Knoten** zu erstellen geben Sie etwas in **SAns** und **TAns** ein und klicken **Speichern**.

Danach kann der Knoten verbunden werden.

Folgende Einstellungen können für die einzelnen Knoten vorgenommen werden:

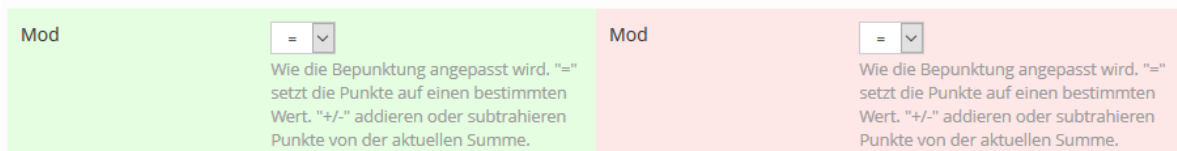
- **Antwortüberprüfung:** Hier wird die Antwortüberprüfungsoption für den Knoten ausgewählt. Mit der Antwortüberprüfung stellen Sie ein durch welche (mathematischen) Eigenschaften **SAns** und **TAns** verglichen werden sollen. Entsprechend dem, was überprüft werden soll, müssen Sie den passenden Antwortüberprüfungstypen auswählen. Die wichtigsten Optionen finden Sie im Kapitel [Antwortüberprüfungen](#).
- **SAns** und **TAns** sind (inkl. Test-Option) die Argumente, die an die Funktion der ausgewählten Antwortüberprüfung übergeben werden. In den üblichsten Fällen gilt: Stimmen **SAns** und **TAns** überein, ist der Knoten wahr. Weitere Einstellungen für diesen Fall werden dann in dem grün unterlegten Feld getroffen. Stimmen **SAns** und **TAns** nicht überein, ist der Knoten falsch. Die weiteren Einstellungen sind für diesen Fall in dem roten Feld zu tätigen.

In **SAns** kann beispielweise der Namen des Eingabefeldes der Frage geschrieben werden (z.B. *ans1*). In **TAns** sollten Sie dementsprechend die Antwort eingeben, mit der die Antwort der Studierenden verglichen werden soll. Die Funktion gibt dann aus, ob der Knoten wahr oder falsch ist.

Ob ein Knoten wahr oder falsch ist, sagt nicht unbedingt etwas darüber aus, ob die Frage richtig beantwortet wurde oder nicht. Beispielsweise können Sie mit einem Knoten prüfen, ob ein bestimmter Fehler in der Aufgabe gemacht wurde. Wurde ebendieser Fehler gemacht, ist der Knoten wahr. Die Aufgabe selbst wurde allerdings falsch beantwortet.

- **Test-Optionen:** Hier können Sie je nach eingestellter Antwortüberprüfung Werte eingeben, die von der Überprüfungsfunktion verwendet werden. Ein Beispiel dafür sieht man in [Runden und Funktionen](#).
- **Feedback unterdrücken:** Dies sollte auf **Ja** eingestellt bleiben, um mehrfaches Anzeigen zu vermeiden.
- **Lösche Knoten:** Löscht den aktuellen Knoten, sofern er nicht mit einem anderen Knoten verbunden ist. Um einen Knoten zu löschen, ist es also notwendig zuerst all seine Verbindungen zu entfernen.

Die Überprüfungsfunktion kann wahr oder falsch ausgeben. Dies bedeutet nicht, dass die Antwort der Studierenden wahr oder falsch sind, sondern nur ob der Vergleich von **SAns** und **TAns** übereinstimmt (wahr) oder nicht (falsch). Abhängig von der Ausgabe der Überprüfungsfunktion können verschiedenen Feedbackausgaben in dem grünen (wahr) und dem roten (falsch) Feld eingegeben werden.



- **Mod:** Gibt an, wie die gegebenen Punkte verrechnet werden. Hier empfiehlt es sich „+“ anzugeben. Dies verhindert, dass in vorherigen Knoten bereits gesammelten Punkte verworfen werden.
- **Punkte/Abzüge wenn wahr/falsch::** Hier können Sie die Punktzahl für den Knoten eingeben, welche Sie abhängig von der Rückmeldung der Überprüfungsfunktion geben möchten.  
(Anmerkung: Abzüge funktionieren momentan nicht)
- **Knoten wenn wahr/falsch:** Hier können Sie den Knoten angeben, der als nächstes durchlaufen werden soll, nachdem der ausgewählte Knoten als wahr bzw. falsch bewertet wurde.
- **Antworthinweis wenn wahr/falsch::** Gibt den Pfad an, der bei einer Rückmeldung im Baum genommen wurde. Dieser Antworthinweis wird lediglich intern benötigt.
- **Feedback wenn wahr/falsch:** Hier können Sie das entsprechende Feedback eingetragen, welches vom Rückmeldebaum erzeugt und ausgegeben werden soll. Sie können dabei außerdem die zuvor definierten Variablen verwenden.

*Hinweis:* Bei der Verteilung der Punkte ist es wichtig, dass die Punkte so verteilt sind, dass es einen Pfad gibt, der maximal 1 ergibt. Dies ist wichtig, denn die Studierenden erhalten die volle

Punktzahl, wenn sie im Rückmeldebaum einen Punkt (100%) erhalten haben. Außerdem ist es wichtig, dass die Pfade (in der Summe) maximal 1 ergeben. Falls Sie den Studierenden mehr Punkte für die gesamte Aufgabe geben möchten, stellen Sie unter „Bewertung“ die Punkte für die Aufgabe ein. Soll außerdem ein Rückmeldebaum mehr gewichtet werden als ein anderer, so sollten Sie den Aufgabenwert dementsprechend einstellen.



## Eingabetypen

Dieses Kapitel bietet jeweils eine kurze Erklärung zu den einzelnen Eingabetypen.

▼ Einstellungen der Eingabe ans1 \*

Eingabetyp *	Algebraische Eingabe ▼
Musterlösung *	Algebraische Eingabe
Eingabebreite	Wahr/Falsch
Strikte Syntax	Matrix
Sterne einfügen	Einzelzeichen
	Mehrzeilig
	Kontrollkästchen
	Drop-Down-Liste
	Gleichungen
	Notizen
	Radio-Auswahl
	Einheiten
	Zeichenkette
	Numerisch

### Algebraische Eingabe

Bei diesem Typ können Antworten in Form von Formeln und Zahlen gegeben werden. Als **Antwortüberprüfung** ist **AlgEquiv** in den meisten Fällen sinnvoll.

### Wahr/Falsch

Ist dies ausgewählt erscheint an der Stelle des Eingabefeldes ein Drop-Down Menü, in welchem Sie **Wahr** oder **Falsch** auswählen können. Die Musterlösung lautet dann entweder **Wahr** oder **Falsch**. Um dieses Feld auswerten zu können, wählen Sie **Antwortüberprüfung** → **AlgEquiv** aus. In **SAns** schreiben Sie den Namen des *Eingabefeldes* und in **TAns** entsprechend *true* oder *false*.

### Matrix

Dieser Typ kann verwendet werden, um eine Matrix als Eingabe der Antwort zu ermöglichen. Einzelne Einträge werden mit  $M[i,j]$  aufgerufen, wenn  $M$  eine  $J \times I$  Matrix ist. Einzelne Zeilen rufen Sie mit  $M[i]$  auf, einzelne Spalten mit  $transpose(M)[j]$ .

Die Musterlösung wird bei diesem Eingabetyp verwendet, um die Größe des Eingabefeldes festzulegen. Wenn die Musterlösung eine  $J \times I$  Matrix ist, wird das Eingabefeld  $J$  Zeilen und  $I$  Spalten haben.

### Einzelzeichen

Falls Sie diesen Eingabetypen auswählen, kann in das Feld nur ein Zeichen angegeben werden. Die Eingabebreite lässt sich nicht verändern. Dies ließe sich im Prinzip auch über die algebraische Eingabe realisieren. Bei diesem Eingabetyp ist jedoch der Vorteil, dass es gar nicht erst möglich ist, mehrere Zeichen einzugeben.

## Mehrzeilig

Dieser Eingabetyp stellt eine Textbox zur Verfügung, in deren Zeilen algebraische Ausdrücke geschrieben werden können. Der Wert, den die Studierendenantwort annimmt, ist dann eine Liste der eingegebenen Ausdrücke. ans1 sieht dann zum Beispiel folgendermaßen aus: [ersteZeile, zweiteZeile, ...]

## Kontrollkästchen (Multiple-Choice Aufgabe)

Hiermit können Sie Aufgaben erstellen, die Kontrollkästchen verwenden. Die Aufgabe soll den Bearbeitenden eine Reihe von Antwortmöglichkeiten im Multiple Choice Stil bieten, von denen ein Teil richtig und ein Teil falsch ist.

### 1 - Kontrollkästchen - Einstieg

Bitte geben Sie alle Lösungen der folgenden Gleichung an:  $\text{abs}(x) = 3$

- ☐  $x=0$
- ☐  $x=1$
- ☒  $x=3$
- ☒  $x=-3$
- ☐ Keine der Antworten ist korrekt.

Rückmeldung anfordern

Syntaxhinweise:

Sie sollten in den **Fragen-Variablen** Listen mit Hilfe von Maxima definieren. Die Syntax der einzelnen Zeilen der Multiple Choice Aufgabe ist:

[Antwort1, true/false, „Anzeigetext“]

- *Antwort1* ist der Wert, der beim Ankreuzen in die Liste der Antwort <ans1> geschrieben wird. Sollte das Kästchen nicht angekreuzt sein wird nichts in die Liste geschrieben.
- *true/false* gibt an ob die entsprechende Antwortmöglichkeit als richtig (true) oder falsch (false) in der Musterlösung angezeigt wird.
- „*Anzeigetext*“ wird in der Bearbeitungsoberfläche den Beantwortenden angezeigt. Dies ist optional, wird jedoch zur besseren Darstellung empfohlen.
- Mittels `mcq_correct(ta)` und `mcq_incorrect(ta)` kann die Liste and korrekten/inkorrekten internen Namen ausgegeben werden.

Die **Musterlösung** ist wiederum eine Liste dieser Einträge:

[[Antwort1, true, Anzeigetext], ..., [AntwortN, false, Anzeigetext]]

Für Antwortoptionen, die **LaTeX-Eingaben** beinhalten und mit Hilfe von MathJax gerendert werden sollen, verwenden Sie " $\backslash$ ( LaTeX-Formel  $\backslash$ )" als „Anzeigetext“.

**Bitte verwenden Sie hierbei keine Leerzeichen außerhalb von wörtlicher Rede.**

Wenn Sie Leerzeichen verwenden möchten ist es notwendig, dass Sie in den Einstellungen des Eingabefeldes den Haken hinter **Strikte Syntax** entfernen.

### Drop-Down-Liste (Single-Choice Aufgabe)

Mit diesem Eingabetyp können Sie eine Drop-Down-Liste mit verschiedenen Antwortmöglichkeiten erstellen. Die Vorgehensweise ist ähnlich wie bei dem Eingabetyp Kontrollkästchen. Wichtig ist hierbei, dass nur eine Antwortmöglichkeit als *true* ausgewiesen werden kann.

#### 2 - Drop-Down - Einstieg

Die Funktion  $f(x) = x^3 - 2 \cdot x^2 - 5 \cdot x + 6$  ist eine .

Die Maximale Anzahl der Nullstellen dieser Funktion kann man  abschätzen.

Die Anzahl der Nullstellen von der Funktion  $f(x)$  lautet: .

Die Funktion hat den Schnittpunkt mit der y-Achse bei .

Nicht ausgewählt  
6  
**-6**  
-2  
-5

Syntaxhinweise:

Definieren Sie in den **Fragen-Variablen** eine Liste der angezeigten Antwortmöglichkeiten. Sie sollte folgendes Format haben:

[Antwort1, true, „Anzeigetext“]

Die **Musterlösung** ist wiederum eine Liste dieser Einträge:

[[Antwort1, true, Anzeigetext], ..., [AntwortN, false, Anzeigetext]]

Bitte verwenden Sie auch hierbei keine Leerzeichen außerhalb von wörtlicher Rede.

Der Befehl *notanswered* in **Test-Eingaben** → **Einstellung der Eingabe ansN** → **Zusätzliche Optionen** verhindert die Darstellung der Antwortmöglichkeit „Nicht ausgewählt“

## Radio-Auswahl (Single-Choice Aufgabe)

Hiermit können Sie Single-Choice-Fragen erstellen.

Die Syntax entspricht genau der im Abschnitt Drop-Down-Liste beschriebenen Syntax. Auch hierbei sollte nur eine Antwortmöglichkeit als *true* ausgewiesen werden.

Im Gegensatz zu dem Eingabetyp Drop-Down-Liste sind bei diesem Eingabetyp jedoch Eingaben in LaTeX möglich. Für Antwortoptionen, die **LaTeX-Eingaben** beinhalten und mit Hilfe von MathJax gerendert werden sollen, verwenden Sie "`\\( LaTeX-Formel \\)`" als „Anzeigetext“.

### 3 - Radio-Button - Einstieg

Welche der Antworten ist richtig?

- ☐ eins plus vier ist fünf
- ☒ zwei plus fünf ist zwölf
- ☐ drei plus sechs ist einundzwanzig
- ☐ fünf plus acht ist vierunddreißig

## Gleichungen

Dieser Eingabetyp ermöglicht es einzelne Äquivalenzumformungen zu überprüfen. Der Eingabetyp ist besonders für Aufgaben, in denen Gleichungen oder Ungleichungen gelöst werden sollen, geeignet. Den Studierenden wird direkt zurückgemeldet, welche Äquivalenzumformungen fehlerhaft sind.

### 4 - Gleichungen Einstieg

Berechnen Sie die Nullstellen von  $x^2 - x = 30$

$x^2 - x = 30$   
 $x^2 - x - 30 = 0$   
 $(x+5)(x-6) = 0$

Validieren

$$x^2 - x = 30$$

$$\Leftrightarrow -30 - x + x^2 = 0$$

$$\Leftrightarrow (x+5) \cdot (x-6) = 0$$

Die folgenden Variablen wurden gefunden:  $[x]$

#### Syntaxhinweise:

Definieren Sie in den **Fragen-Variablen** eine Liste mit Äquivalenzumformungen

loesung: [Gleichung1,...,GleichungN]

*Gleichung1* sollte die in der Frage gestellte Gleichung sein. *GleichungN* sollte die Lösung sein. Achten Sie darauf, dass die Befehle für die logischen Operationen und/oder *nounand* und *nounor* sind.

In die **Musterlösung** sollte die Fragen-Variable *loesung* eingetragen werden.

Es macht außerdem Sinn die **Eingabebreite** etwas breiter als die Defaultwert von 15 einzustellen.

Tragen Sie bei **Syntax-Hinweis** den Befehl *firstline*. Dies schreibt den ersten Eintrag der Musterlösung in das Eingabefeld der Frage.

Nutzen Sie zur Antwortüberprüfung im **Rückmeldebaum** den Befehl *last(...)* , um auf den letzten Eintrag der Liste mit Äquivalenzumformungen zuzugreifen,

#### Notizen

Mit Hilfe dieses Typs können die Studierenden Freitext als Antwort einfügen. Dieser Fragetyp eignet sich beispielsweise dazu, eine Rückmeldung der Studierenden zu erfragen.

#### Syntaxhinweise:

Bei einer beliebigen Antwort wird **ans1** der Wert *true* zugeordnet. Es empfiehlt sich außerdem die Eingabebreite größer als den Default-Wert von 15 einzustellen.

#### Bewertung:

Über den Reiter **Manuelle Bewertung** innerhalb eines Tests ist es für Dozierende bei jedem Eingabetyp möglich, die entsprechende Frage für alle Testteilnehmenden einzusehen und im Nachhinein zu bewerten. Dies kann für den Eingabetyp Notizen von praktischem Nutzen sein. Sie sollten im Rückmeldebaum allerdings darauf hinweisen, dass eine Bewertung im Nachhinein geplant ist.

#### Einheiten

Mit Hilfe dieser Funktion können physikalische Einheiten in der Antwort überprüft werden. Sie sollten in der Musterlösung Standard-Einheiten verwenden oder die verlangte Einheit im Fragetext erwähnen. Da der Eingabetyp etwas kniffliger zu benutzen ist, melden Sie sich bei Interesse am besten einfach bei uns: **ilias-stack@uni-koeln.de**

Geben Sie die Erdbeschleunigung in den entsprechenden SI-Einheiten an.

Ihre Antwort wurde wie folgt interpretiert:

**9.81 m/s<sup>2</sup>**

Syntaxhinweise:

Die Eingabe der Musterlösung wird wie bei der algebraischen Eingabe vorgenommen. Stellen Sie unter **Rückmeldebäume** die **Antwortüberprüfung** auf **Units\***(-SigFig,-StrictSigFigs,-Absolute,-StrictAbsolute,-Relative oder -StrictRelative).

### Zeichenkette

Dieser Eingabetyp ermöglicht die Eingabe von Zeichenketten.

Syntaxhinweise:

Die Musterlösung muss in Anführungszeichen angegeben werden. Stellen Sie unter **Rückmeldebäume** die **Antwortüberprüfung** auf **StringSloppy** oder **String**.

### Numerisch

Mit diesem Eingabetyp sind als Eingabe ausschließlich Zahlen erlaubt. Abhängig von der von Ihnen gewählten Antwortüberprüfung können Sie beispielsweise einstellen, um wie viel eine Antwort von der Musterlösung abweichen darf. Dies ist bei erwartbaren Rundungsungenauigkeiten beispielsweise sinnvoll. Es ist allerdings zu beachten, dass viele Antwortüberprüfungsfunktionen fehlerbehaftet sind. Oft ist es dementsprechend leichter die algebraische Eingabe zu verwenden und mit Hilfe von Maxima die gewünschte Funktion zu definieren.

Weitere Ausführungen zum Zusammenspiel von dem Eingabetyp Numerisch und verschiedenen Antwortüberprüfungen finden Sie unter folgendem Link:

[https://github.com/mathsmoodle-gtype\\_stack/blob/master/doc/en/Authoring/Answer\\_tests\\_numerical.md](https://github.com/mathsmoodle-gtype_stack/blob/master/doc/en/Authoring/Answer_tests_numerical.md)

## Antwortüberprüfungen

Die Einstellung der Antwortüberprüfung dient dazu einzustellen welche (mathematischen) Eigenschaften verwendet werden, um **SAns** und **TAns** zu vergleichen.

In den meisten Fällen empfiehlt es sich die Standardeinstellung **AlgEquiv** beizubehalten. Im Folgenden finden Sie die wichtigsten Optionen und deren Funktion:

- **AlgEquiv**: Vereinfacht **SAns** und **TAns** und vergleicht anschließend. Somit spielt die Schreibweise keine Rolle. Lediglich die getroffene Aussage ist relevant.
- **EqualComAss**: Eingaben werden bis auf Kommutativität (Vertauschung der Ausdrücke) und Assoziativität (Klammersetzung) verglichen. Diese Antwortüberprüfung bietet weniger Spielraum für die Eingabe, als **AlgEquiv**.
- **NumAbsolute**:  $|\text{SAns} - \text{TAns}| < \text{Test-Option}$  wird geprüft. In die Test-Option muss eingetragen werden, wie weit **SAns** von **TAns** abweichen darf. Gilt  $|\text{SAns} - \text{TAns}| < \text{Test-Option}$  wird die Überprüfung als wahr ausgewiesen.
- **String**: Wortwörtliche Zeichenketten werden verglichen.  
Eine **Algebraische Eingabe** darf keine Leerzeichen enthalten darf. Somit ist es dabei nicht möglich, dass Studierende Eingaben machen, die aus mehr als einem Wort bestehen. Verwenden Sie hierfür den Eingabetyp **Zeichenkette**.
- **StringSloppy**: Die Überprüfung funktioniert wie **String**. Jedoch werden Zeichen automatisch klein geschrieben und Leerzeichen gelöscht. Es kommt bei der Eingabe also nicht auf Groß-/Kleinschreibung an.

Eine vollständige Liste mit den Antwortüberprüfungstypen und deren Funktionsweisen finden Sie unter: [https://github.com/mathsmoodle-gtype\\_stack/blob/master/doc/en/Authoring/Answer\\_tests.md](https://github.com/mathsmoodle/gtype_stack/blob/master/doc/en/Authoring/Answer_tests.md)

## Kommentare einfügen

Kommentare können folgendermaßen in STACK-Fragen eingefügt werden: */\* Kommentar \*/*

Dies ist unter anderem sinnvoll, um zu erklären, wie Fragen aufgebaut sind. Einerseits damit andere die Frage verstehen können, jedoch auch um nach einiger Zeit selbst die Frage gut verstehen zu können. Besonders bei der Definition von Fragen-Variablen kann dies hilfreich sein.

# Maxima

Maxima ist die Programmiersprache, bzw. genauer das Computer Algebra System (CAS), das in STACK steckt.

Um STACK erfolgreich zu benutzen ist es also hilfreich grundlegende Befehle aus Maxima zu kennen. Wie bei jeder anderen Programmiersprache auch ist das Internet hierbei ein sehr guter Lehrmeister.

Hilfreiche Links hierzu sind:

<http://maxima.sourceforge.net/docs/manual/maxima.html>

[https://github.com/mathsmoodle-qtype\\_stack/tree/master/doc/en/CAS](https://github.com/mathsmoodle-qtype_stack/tree/master/doc/en/CAS)

(Hinweis: Dieser Link bezieht sich auf STACK in Moodle. Vieles davon kann so übernommen werden, da es bei STACK in ILIAS genauso funktioniert, doch das trifft nicht auf alles zu.)

[https://de.wikipedia.org/wiki/Maxima\\_\(Computeralgebrasystem\)](https://de.wikipedia.org/wiki/Maxima_(Computeralgebrasystem))

In STACK müssen Maxima-Befehle einzellig geschrieben werden, weil am Ende jeder Zeile automatisch ein „;“ eingefügt wird. Dieses Zeichen gibt normalerweise in Maxima das Ende einer Eingabe an und muss nach jeder Eingabe einzeln eingegeben werden. Dies ist bei STACK nicht notwendig.

Nicht jede Maxima-Version funktioniert gleich. Die Maxima Version hinter STACK ist: 5.32.1

Unter <http://maxima.cesga.es/> können Ausdrücke zur Probe eingegeben werden.

Hinweis: Hier müssen Sie nach jeder Zeile ein Semikolon einfügen, um einzelne Befehle voneinander zu trennen.

## Grundlegende Befehle

Im Folgenden sind die Maxima-Befehle aufgeführt, welche in den Musterlösungen der Übungsaufgaben verwendet wurden. Für eine umfangreichere Dokumentation können Sie den folgenden Link nutzen:

Maxima-Einführung für STACK-Nutzer:

[https://github.com/mathsmoodle-qtype\\_stack/tree/master/doc/en/CAS](https://github.com/mathsmoodle-qtype_stack/tree/master/doc/en/CAS)



## Allgemein

- Der Doppelpunkt „:“ wird genutzt, um einer Variablen einen Wert zuzuweisen. Zum Beispiel um der Variablen  $n$  den Wert 5 zuzuweisen, nutzen wir die Syntax „ $n:5$ “.
- **diff(<expr>, <x>)** differenziert die Funktion <expr> nach <x>.
- **integrate(<expr>, <x>)** integriert die Funktion <expr> nach <x>.
- **is (<expr>)** prüft, ob eine Aussage <expr> wahr oder falsch ist und gibt entsprechend true oder false aus.
- **and** ist der logische Operator und. Hiermit können beispielsweise durch **is (<expr>)** erzeugte Aussagen miteinander verknüpft werden.
- **simp** ist das Systemattribut für Vereinfachung. Der Standardwert ist true.
- Die **genmatrix** Funktion generiert mit Hilfe, einer angegebenen Funktion, eine Matrix. **gematrix(f,n,m)** bedeutet, dass eine Matrix mit  $n$  Zeilen und  $m$  Spalten generiert wird. Die Funktion  $f$  entscheidet darüber, wie die einzelnen Einträge belegt werden. Damit dies möglich ist muss  $f$  von  $n$  und  $m$  abhängen, denn diese beiden Werte werden an die Funktion  $f$  weitergegeben.  $f$  könnte beispielsweise so aussehen:  $f(m,n):=n^2-m^4$ . Dies bedeutet auch, dass Einträgen mit gleichen Indizes immer die gleichen Werte zugewiesen werden. *Hinweis:* Dies gilt auch wenn  $f(m,n):=\text{rand}(9)$ . Dies kommt dadurch zustande, dass der Seed für die Zufallszahlen zum Zeitpunkt der Definition der Funktion festgelegt wird. Dieser Seed wird ab dann verwendet und führt bei gleichen Argumenten zu gleichen Ergebnissen. Die Funktion muss also neu definiert werden, wenn eine weitere Matrix mit neuen Zufallszahlen generiert werden soll. Eine Anwendung dieser Funktion finden Sie in den Beispielen unter [9. Matrizen](#).

## STACK spezifische Maxima-Befehle

- **rand(n)**
  - erzeugt eine zufällige ganze Zahl zwischen 0 und  $n$ , sofern  $n$  eine ganze Zahl ist.
  - erzeugt eine zufällige Gleitkommazahl zwischen 0 und  $n$ , sofern  $n$  eine Gleitkommazahl ist.
- **rand\_with\_step(a,b,c)** erzeugt eine zufällige Zahl zwischen  $a$  und  $b$  mit der Schrittweite  $c$ .
- **rand\_with\_prohib(a,b,[c,d,e,...])** gibt eine zufällige Zahl aus der Liste  $[a, a+1, \dots, b]$  ohne die Liste  $[c,d,e,\dots]$  aus.
- **plot([f1,f2,...],[x,x0,x1],[y,y0,y1])** gibt einen Plot der Funktionen  $f1, f2, \dots$  aus, welcher zwischen  $x0$  und  $x1$  bzw.  $y0$  und  $y1$  dargestellt wird.  
Sie können auch **plot(f1,[x,x0,x1])** verwenden. In den Beispielen gibt es in der Frage *13\_Plot-Optionen* weitere Optionen zum Anschauen.
- **solve(<expr>)** löst die Gleichung <expr>

- **float(<expr>)** wandelt <expr> in eine Fließkommazahl um
- **round(<expr>\*10^n)/10^n** <expr> auf n Nachkommastellen runden.  
 Z.B.:  
 x:%pi;  
 n:3;  
 u:float(round(x\*10^n)/10^n);
- Um Schleifen in STACK verwenden zu können müssen sie immer auf eine (Dummy-) Variable geschrieben werden.  
 for i:1 thru 6 do z:z\*2; geht nicht, aber  
**x:for i:1 thru 6 do z:z\*2;** geht.

## Workarounds

In diesem Abschnitt finden Sie Funktionen, die kopiert und verwendet werden können. Steht eine Funktion in der entsprechenden Frage im Fragen-Variablentext, kann sie in der gesamten Aufgabe aufgerufen und verwendet werden.

All diese Funktionen sind zur Anschauung in Aufgaben eingebaut. Diese Aufgaben finden Sie im Fragenpool für besondere Maxima-Befehle unter:

[https://www.ilias.uni-koeln.de/ilias/goto\\_uk\\_qpl\\_3001984.html](https://www.ilias.uni-koeln.de/ilias/goto_uk_qpl_3001984.html)

- **round2(ZTV,ZTN):=(if ZTV\*10^ZTN-floor(ZTV\*10^ZTN)<0.5 then float(floor(ZTV\*10^ZTN)\*10^-ZTN) else float(floor(ZTV\*10^ZTN+1)\*10^-ZTN))**  
 Diese Funktion dient dem kaufmännischen Runden. Sie bewirkt, dass die Zahl ZTV auf ZTN Nachkommastellen gerundet wird. bewirkt, dass die Variable ZTV auf ZTN Nachkommastellen gerundet wird.
- **trennzeichen(zahl):=(if numberp(zahl)=true then zahl:string(zahl),if ssearch(".",zahl)=false then (l:length(zahl), while l>3 do (l:(l-3),(zahl:sinsert(" ",zahl,l+1)))) else (nachkommastellen:sconcat(".",part(split(zahl,"."),2)),vorkommastellen:part(split(zahl,"."),1),l:length(vorkommastellen),while l>3 do (l:(l-3),(vorkommastellen:sinsert(" ",vorkommastellen,l+1))),zahl:sconcat(vorkommastellen,nachkommastellen),zahl);**

Diese Funktion fügt Leerzeichen als Trennzeichen in große Zahlen ein. So wird die Ansicht für Studierende übersichtlicher.

## MathJax

Eine begrenzte Menge an LaTeX, MathML oder ASCIIMath-Befehlen kann in STACK verwendet werden, um beispielsweise Formeln ästhetisch ansprechender auszugeben.

LaTeX lernen Sie am besten, indem Sie sich bereits existierende Dokumente mit Erläuterungen anschauen oder indem Sie die Befehle mit Hilfe einer Online-Suchmaschine herauszufinden. Für eine umfangreiche Nutzung von STACK sind bereits Basis-Kenntnis in LaTeX ausreichend.

LaTeX-Befehle, die verwendet werden, um Formeln darzustellen, sind in STACK nutzbar. LaTeX-Befehle, die darüber hinaus die Ansicht gestalten (wie z.B. `\vspace`) können nicht verwendet werden. Um den Fragentext zu gestalten klicken Sie auf das HTML-Symbol im Feld **Frage** und gestaltet den Text mit Hilfe von [HTML](#).

Der TeX-Knopf im Kopf des Eingabefeldes **Frage** fügt an der entsprechenden Stelle den Formelbefehl ein. Eine Vorschau der Formel wird angezeigt. Dort sehen Sie, wie die Formel den Studierenden angezeigt werden wird. So können Sie verschiedene Eingaben ausprobieren, bis Ihnen die Anzeige gefällt.

Neben dem TeX-Knopf ist es auch möglich LaTeX-Befehle einzugeben, indem man in dem Fragetext eine Zeichenkombination für Beginn und Ende des LaTeX-Befehls (Mathmode) verwendet. Die sogenannten Parser. Davon gibt es verschiedene. Wir empfehlen: `\( \)`  
Angewendet könnte das beispielsweise so aussehen: `\(3+\frac{4}{2}\)`

Falls Sie sich unsicher sind, wie man Eingaben mit LaTeX macht, finden Sie außerdem viele wichtige Befehle sind unter:

[https://de.wikibooks.org/wiki/LaTeX-Kompodium: F%C3%BCr Mathematiker](https://de.wikibooks.org/wiki/LaTeX-Kompodium:_F%C3%BCr_Mathematiker)

<http://www.physki.de/PhysKi/index.php?title=MathJax-Formelsatz>

## Grundlegende Befehle

Im Folgenden sind die MathJax-Befehle angeführt, welche in den Musterlösungen der Übungsaufgaben verwendet wurden. Für eine umfangreichere Dokumentation können Sie den folgenden Link nutzen:

<https://www.mathjax.org/#gettingstarted>

MathJax ermöglicht eine anschauliche Darstellung der Frage- und Feedbacktexte mit Hilfe ausgewählter LaTeX-Befehle.

Inline-Eingaben in LaTeX beginnt man mit „\(" und beendet sie mit „\)“.  
Statt  $\left( \dots \right)$  kann man auch  $\$ \dots \$$  schreiben.

Newline-Eingaben in LaTeX beginnt man mit „\[“ und beendet sie mit „\]“.  
man kann alternativ auch  $\$ \$ \dots \$ \$$  schreiben.

- Brüche:  $\frac{\text{Zähler}}{\text{Nenner}}$
- Produkte:  $a \cdot b$  oder  $a \times b$
- Integrale:  $\int_{\text{Untergrenze}}^{\text{Obergrenze}} \text{Funktion} \, dx$

Für Doppelintegrale setzt man einfach ein weiteres i vor das int. Für drei noch ein i und so weiter.

- Wurzeln:  $\sqrt{x}$  oder  $\sqrt[n]{x}$

Potenzen:  $a^x$

# HTML

HTML können Sie verwenden, um die Frage ansprechender zu gestalten oder um beispielsweise die Hintergrundfarben für die Rückmeldungen zu ändern. Hierzu klicken Sie auf den HTML-Knopf über dem Eingabefeld.



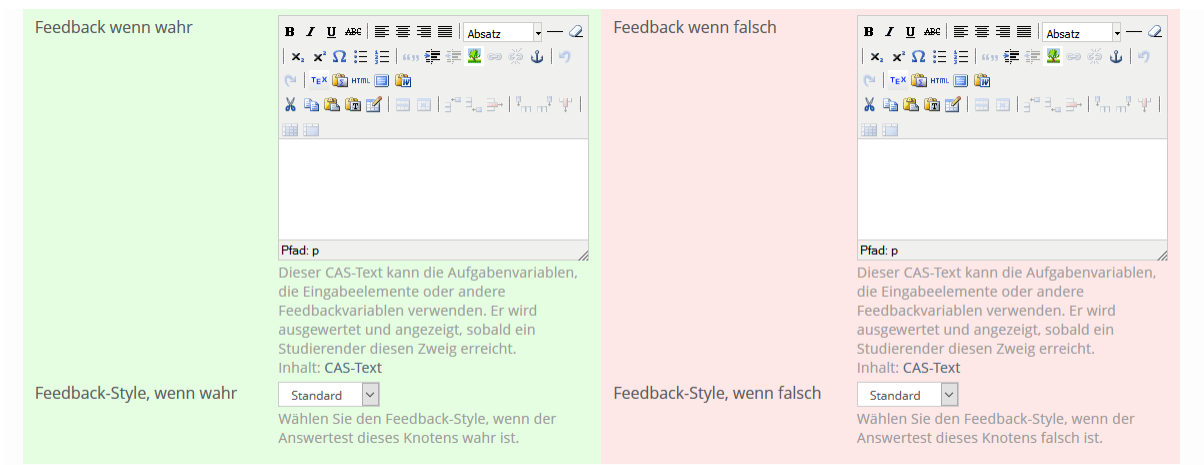
Beispiele hierzu finden Sie im Abschnitt [Manuelle Gestaltung](#).

Wichtige Befehle finden Sie im Internet beispielsweise auf [www.w3schools.com](http://www.w3schools.com) oder Sie kopieren sich den HTML-Text von anderen Fragen in ihre eigenen.

## Farbliche Gestaltung von Feedback

### Automatische Gestaltung

Unter jeder Verzweigung der Knoten kann ein vordefinierter Feedback-Style ausgewählt werden.



Zur Wahl stehen sechs verschiedene Optionen.

Die Benennungen geben einen Hinweis darauf, in welchen Situationen der entsprechende Feedback-Style verwendet werden kann. Dies ist lediglich eine Empfehlung. Abgesehen von dem Erscheinungsbild der Rückmeldung ändert diese Einstellung nichts. Die Bewertung wird dadurch nicht beeinflusst.

### Manuelle Gestaltung

Eine manuelle Gestaltung des Feedbacks über HTML-Befehle ist möglich und kann eine gute Alternative zu den voreingestellten Optionen sein, wenn Sie spezielle Wünsche haben.

Um das Feedback individuell zu gestalten klicken Sie, wie oben beschrieben, auf den HTML-Knopf und fügen die entsprechenden HTML-Befehle ein.

Für ein simples farbles Feedback geben Sie folgendes in das Feld ein:

```
<div style="margin: -16px -16px -0px; padding: 16px 16px 0px; opacity: 1; background-color: Farbcode; color: black;"></div>
```

```
<div style="margin: -0px -16px -0px; padding: 0px 16px 0px; opacity: 1; background-color: Farbcode; color: black;">Text der im Feedback angezeigt werden soll.</div>
```

```
<div style="margin: -0px -16px -36px; padding: 0px 16px 36px; opacity: 1; background-color: Farbcode; color: black;"></div>
```

### Farbvorschläge

Typ: background-color (Farbcode):

---

Richtig	#e7ffe7
Falsch	#ffe7e7
Lösungshinweis	#fcf8e3
Zusätzliche Information	#f2f6ff
Plot	#fefcf2

Sie können ebenso jede andere Farbe mit Hilfe von HTML Code eingeben.

### Ableitung 1

Gegeben seien die Funktionen  $f_1(x) = 2 \cdot x^5 - 6 \cdot x^3 + 7$  und  $f_2(x) = 3 \cdot (9 - 4 \cdot x)^3 - 2 \cdot x$

Leiten Sie  $f_1(x)$  nach  $x$  ab.

$f'_1(x) =$

Ihre Antwort wurde wie folgt interpretiert:

$10 \cdot x^4 - 18 \cdot x^2$

Die Antwort  $f'_1(x) = 10 \cdot x^4 - 18 \cdot x^2$  ist richtig.

Leiten Sie  $f_2(x)$  nach  $x$  ab.

$f'_2(x) =$

Ihre Antwort wurde wie folgt interpretiert:

3

Die Antwort  $f'_2(x) = 3$  ist falsch.

## Weiterführende Links

Es gibt darüber hinaus noch viele weitere Seiten auf denen man nützliche Infos findet.  
Eine **Special Interest Group** auf **Ilias.de** beschäftigt sich zudem mit STACK.

STACK-Dokumentation für Moodle (englisch):

[https://github.com/mathsmoodle-qtype\\_stack/tree/master/doc](https://github.com/mathsmoodle-qtype_stack/tree/master/doc)

MathJax:

<https://www.mathjax.org/#gettingstarted>

Ausführliche Behandlung der Antwortüberprüfungsmöglichkeiten:

[https://github.com/mathsmoodle-qtype\\_stack/blob/master/doc/en/Authoring/Answer\\_tests.md](https://github.com/mathsmoodle-qtype_stack/blob/master/doc/en/Authoring/Answer_tests.md)

Maxima-Einführung für STACK-Nutzer:

[https://github.com/mathsmoodle-qtype\\_stack/blob/master/doc/en/CAS/Maxima.md](https://github.com/mathsmoodle-qtype_stack/blob/master/doc/en/CAS/Maxima.md)

Maxima-Dokumentation:

<http://maxima.sourceforge.net/docs/manual/maxima.html>

Hier kann unter [Index] nach bestimmten Maxima-Funktionen gesucht werden:

[http://maxima.sourceforge.net/docs/manual/maxima\\_89.html#SEC422](http://maxima.sourceforge.net/docs/manual/maxima_89.html#SEC422)

MAXIMA-Onlinekurs des ACDCA:

<http://www.austromath.at/daten/maxima/>

Maxima Online Shell:

<http://maxima.cesga.es/>



# Übungsaufgaben

Im Folgenden wird eine STACK-Frage Schritt für Schritt erstellt. Versuchen Sie zuerst selbstständig die STACK-Aufgaben zu erstellen. Orientieren Sie sich dabei an dem [Flowchart](#) zu Beginn dieses Dokuments. Bei Unklarheiten, können Sie auf die verlinkten Ausführungen sowie die Musterlösung zurückgreifen. Die Musterlösung finden Sie in dem Kurs „Einstieg STACK 3“ unter

[Einstieg STACK 3 -> Einstieg für AutorInnen -> Musterlösungen zur „Erklärung STACK3“<sup>1</sup>](#)

Die Übungsaufgaben sind unterteilt in drei Abschnitte. Im Abschnitt zur algebraischen Eingabe, lernen Sie die Grundlagen der STACK-Frage, Im zweiten Abschnitt lernen Sie verschiedene Eingabetypen kennen, die für bestimmte Aufgabentypen äußerst nützlich sein können. Der letzte Abschnitt beschäftigt sich mit einer ansprechenden Darstellung der Frage und der Rückmeldung. Hier lernen Sie, wie Sie Lerninhalte beispielsweise mittels einer graphischen Darstellung besser vermitteln können.

Falls Sie versehentlich Beispielfragen verändern, melden Sie uns das bitte zurück, damit wir diese für künftige NutzerInnen wieder in den Ausgangszustand versetzen können.

Sie erreichen uns unter: **ilias-stack@uni-koeln.de**

In dem Kurs der Self-Assessment-Instanz haben Sie auch die Möglichkeit, selbst Fragen zu schreiben. Bitte erstellen Sie Ihre Fragen im **Sandkasten zum Ausprobieren**.

## Übungsaufgaben zur algebraischen Eingabe

1. Erstellen Sie die erste Stack-Frage:

„Rechnen Sie  $3 + \frac{4}{2}$  aus.“

[Musterlösung – Schritt für Schritt](#)

2. Fügen Sie der Aufgabe einen zweiten Rückmeldeknoten hinzu, der überprüft, ob die Punkt-vor-Strich-Regel missachtet wurde.

[Musterlösung – Schritt für Schritt](#)

3. a) Fügen Sie der Stack-Frage eine weitere Frage hinzu:

„Rechnen Sie  $6 - 2 * 2$  aus“

Fügen Sie entsprechend weitere Rückmeldeknoten hinzu.

- b) Überprüfen Sie mit Hilfe des Rückmeldebaums, ob ein Wiederholungsfehler vorliegt.

[Musterlösung – Schritt für Schritt](#)

---

<sup>1</sup> Eine Anleitung, wie Sie dem Kurs beitreten finden Sie in diesem Dokument unter [Beitritt zum Kurs](#).

4. Überprüfen Sie nun alternativ mit Maxima-Befehlen, ob ein Wiederholungsfehler vorliegt. Definieren Sie hierfür eine Feedbackvariable „WdhF“ mithilfe der Maxima-Befehle **is(<expr>)** und **<expr>and<expr>**. Nutzen Sie die Variable zum Abgleich mit **TAns** und **SAns**.

[Musterlösung – Schritt für Schritt](#)

5. Überprüfen Sie die Antworteingabe nun alternativ in drei verschiedenen Rückmeldebäumen. Die Rückmeldebäume sollen folgendes überprüfen:
1. Ob Antwort 1 richtig oder falsch ist,
  2. ob Antwort 2 richtig oder falsch ist,
  3. ob ein Wiederholungsfehler vorliegt (*Tipp*: Nutzen Sie hierfür die zuvor definierte Variable WdhF).

Schreiben Sie zusätzlich aussagekräftige und hilfreiche Rückmeldungen zu jedem einzelnen Rückmeldebaum.

Verschieben Sie anschließend die Platzhalter der Rückmeldebäume aus *Spezifisches Feedback* an die entsprechende Stelle in das *Frage*-Feld.

[Musterlösung – Schritt für Schritt](#)

6. Ersetzen Sie die Zahlen in der Aufgabenstellung durch Zufallsvariablen. Nutzen Sie dafür die Befehle **rand**, **rand\_with\_step** und **rand\_with\_prohib**.

*Hinweis*: Eine Erklärung zu der [Verwendung von Variablen](#) und eine [Erklärung der Befehle](#) finden Sie in diesem Dokument.

[Musterlösung – Schritt für Schritt](#)

7. Erstellen Sie folgende Frage:

„Differenzieren Sie  $x^{a1} * e^{x*a2}$  nach  $x$ . Berechnen Sie den Bruch  $\frac{a1}{a2}$ .“

$a1, a2$  sollen Zufallsvariablen sein. Achten Sie darauf entsprechende Maxima-Befehle zu verbieten, sodass sich die Lösung nicht durch Eingabe des Befehls „erschummeln“ lässt.

[Musterlösung – Schritt für Schritt](#)

8. Erstellen Sie folgende Frage:

„Geben Sie  $\pi$  auf 4 Nachkommastellen genau an.“

Überprüfen Sie ob die angegebene Lösung korrekt und auf exakt 4 Stellen gerundet wurde. Benutzen Sie **round2()** aus den [Maxima Workarounds](#).

*Hinweis*: Die Maxima-Funktion **round()** ermöglicht das Runden ebenfalls. Jedoch verwendet diese Funktion das symmetrische Runden. Die Funktion **round2()**, die wir selbst geschrieben haben, ermöglicht kaufmännisches Runden, wie man es in den meisten Fällen verwenden möchte.

[Musterlösung – Schritt für Schritt](#)

## Übungsaufgaben zu anderen Eingabetypen

In den bisherigen Aufgaben wurde unter *Einstellungen der Eingabe* immer der Eingabetyp *Algebraische Eingabe* verwendet. In den folgenden Aufgaben wird es darum gehen einige der anderen *Eingabetypen* innerhalb von STACK kennenzulernen. Wählen Sie diese entsprechend der Aufgabenstellungen unter *Einstellungen der Eingabe* aus.

9. Erstellen Sie die Matrix

$$C = \begin{pmatrix} -2 & 4 & 1 \\ 1 & 0 & 2 \\ -2 & -5 & 3 \end{pmatrix}$$

sowie zwei Zufallsmatrizen A und B und stellen Sie folgende Aufgabe:

„Berechnen Sie  $D = A * B$

Berechnen Sie  $E = A - C$ “

Verwenden Sie dazu den Eingabetyp **Matrix**.

[Musterlösung – Schritt für Schritt](#)

10. Erstellen Sie die folgende Frage als Multiple Choice-Aufgabe mit Hilfe des Eingabetyps

**Kontrollkästchen:**

„Bitte geben Sie alle Lösungen der folgenden Gleichung an:  $|x| = 3$ “

[Musterlösung – Schritt für Schritt](#)

11. Erstellen Sie die folgende Aufgabe mit Hilfe des Eingabetyps **Drop-Down-Liste:**

„Die Funktion  $f(x)=x^3-2\cdot x^2-5\cdot x+6$  ist eine

- a) Ganzrationale Funktion
- b) Exponential Funktion
- c) Gebrochenrationale Funktion
- d) Trigonometrische Funktion

Die Maximale Anzahl der Nullstellen dieser Funktion kann man abschätzen

- a) nur durch eine genaue Rechnung
- b) mit Hilfe des niedrigsten Exponenten
- c) mit Hilfe des höchsten Exponenten
- d) erst wenn man Mathe studiert hat

Die Anzahl der Nullstellen von der Funktion  $f(x)$  lautet:

- a) 3
- b) 1
- c) 0
- d) 2

Die Funktion hat den Schnittpunkt mit der y-Achse bei

- a) 6
- b) -6
- c) -2
- d) 5“

12. Erstellen Sie außerdem die folgende Aufgabe mit Hilfe des Eingabetyps **Radio-Auswahl**:

„Welche Aussage ist richtig:

- a) eins plus vier ist fünf
- b) zwei plus fünf ist zwölf
- c) drei plus sechs ist einundzwanzig
- d) fünf plus acht ist vierunddreißig“

[Musterlösung – Schritt für Schritt](#)

13. Erstellen Sie folgende Aufgabe mit Hilfe des Eingabetyps **Gleichungen**:

„Berechnen Sie die Nullstellen von:  $x^2 - x = 30$ “

[Musterlösung – Schritt für Schritt](#)

14. Erstellen Sie mit Hilfe des Eingabetyps **Notizen** eine Frage, die es den Studierenden am Ende eines Tests ermöglicht eine Rückmeldung zum Test zu geben.

[Musterlösung – Schritt für Schritt](#)

15. Erstellen Sie mit Hilfe des Eingabetyps **Einheiten** die folgende Aufgabe:

„Geben Sie die Erdbeschleunigung in SI-Einheiten an.“

[Musterlösung – Schritt für Schritt](#)

16. Erstellen Sie mit Hilfe des Eingabetyps **Zeichenkette** die folgende Aufgabe:

„Welche Methode würden Sie nutzen, um  $\int x \sin(x) dx$  zu berechnen?“

[Musterlösung – Schritt für Schritt](#)

17. Erstellen Sie mit Hilfe des Eingabetyps **Numerisch** die folgende Aufgabe: „Geben Sie die ersten vier Stellen von  $\pi$  an.“

[Musterlösung – Schritt für Schritt](#)

## Übungsaufgaben für eine ansprechende Rückmeldung

18. Versuchen Sie ihr Frage ansprechend zu gestalten, in dem Sie aus der [manuellen Gestaltung mit HTML](#) Teile des Codes rauskopieren und für Ihre eigene Frage nutzen.  
[Musterlösung – Schritt für Schritt](#)

19. Erstellen Sie eine Frage, in der die Studierenden eine Funktion mit Zufallsvariablen ableiten sollen. Als Rückmeldung sollen die richtig und die falsch abgeleitete Funktion in einem Plot verglichen werden.  
[Musterlösung – Schritt für Schritt](#)

20. Nutze Sie verschiedene Optionen, um Plots zu erstellen. Lassen Sie sich von den Darstellungsbeispielen aus der Frage [20 Plot Optionen](#) in dem ILIAS-Kurs inspirieren.  
[Musterlösung – Schritt für Schritt](#)

# Schritt-für-Schritt-Musterlösungen

## Musterlösungen zur algebraischen Eingabe

### 1. Erstellen der ersten STACK-Frage (01\_Erste Frage)

Inhalt: Frage erstellen

Zu beachtende neue Einstellungen:

- **Titel**
- **Frage**
- **Test-Eingaben**
  - **Einstellungen der Eingabe ans1**
    - **Musterlösung**
- **Rückmeldebäume**
  - **SAns**
  - **TAns**
  - **Mod**
  - **Punkte wenn wahr**
  - **Feedback wenn wahr**
  - **Feedback wenn falsch**

[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 01 Erste Frage](#)

### 2. Einfügen von weiteren Knoten (02\_Zweiter Knoten - Punkt vor Strich)

Inhalt: Knoten erstellen und verknüpfen

Man kann sich vorstellen, dass Studierende die Punkt-vor-Strich Regel missachten. Daher macht es Sinn einen zweiten Knoten zu erstellen welcher, falls das Ergebnis falsch ist, prüft ob die Punkt-vor-Strich Regel missachtet wurde.

In dem Beispiel *02\_Zweiter Knoten* wird zuerst (in Knoten 1) abgefragt, ob die Punkt-vor-Strich missachtet wurde. Danach (in Knoten 2) wird erst geprüft, ob das Ergebnis richtig ist.

Die Reihenfolge der beiden Knoten ist allerdings egal, da beide Abfragen sich gegenseitig ausschließen.

Zu beachtende neue Einstellungen:

- **Rückmeldebäume**
  - **Neuer Knoten**
  - **Knoten wenn falsch**

[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 02 Zweiter Knoten – Punkt vor Strich](#)

### 3. Einfügen eines weiteren Fragenteils (03\_Zweites Eingabefeld a) & 03\_Zweites Eingabefeld b) )

Inhalt: Mehrere Input-Felder, Bewertung anpassen

Fügen Sie weiteres Eingabefeld ein, in welchem nach dem Ergebnis von  $6 \cdot 2 \cdot 2$  gefragt wird. Falls sich ein Flüchtigkeitsfehler eingeschlichen hat, ist das für die Lehrenden irrelevant und die Studierenden brauchen lediglich die Information, dass sie einen Fehler gemacht haben. Hierfür nehmen Sie folgende Schritte vor:

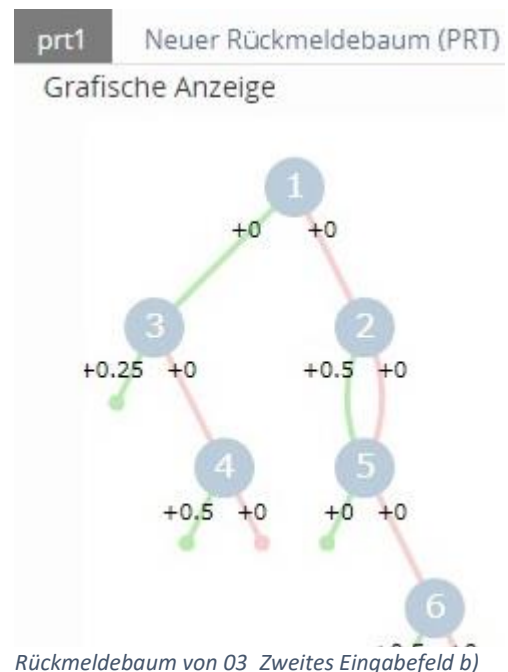
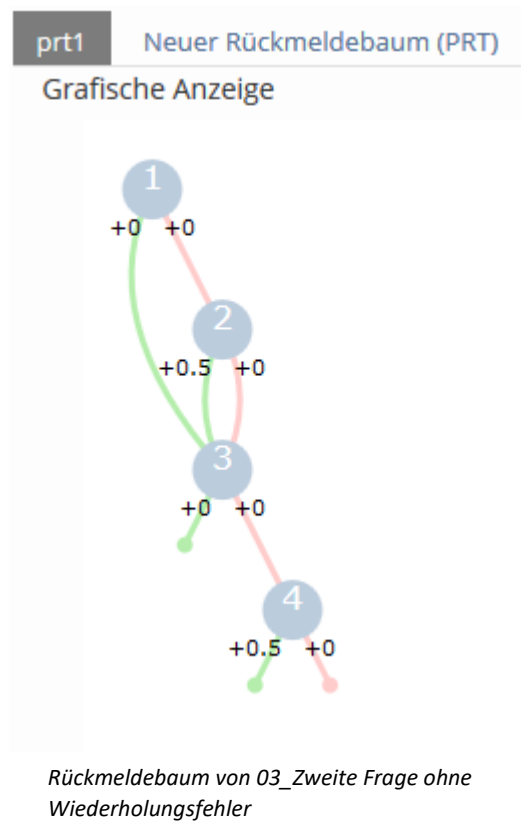
1. Erstellen Sie zwei neue Knoten für die zweite Aufgabenzeile wie schon im vorherigen Beispiel. In der Beispielaufgabe sind das Knoten 3 und 4.
2. Dann verbinden Sie die Knoten so, dass, nachdem herausgefunden wurde, was die erste Antwort war, die Zweite überprüft wird. Mit entsprechenden Bepunktungen (0.5 pro Aufgabe und richtiger Antwort) sähe der Rückmelde-Baum so aus wie in **03\_Zweites Eingabefeld a).**

Zu beachtende neue Einstellungen:

- **Frage**
- **Rückmeldebäume**
  - **Neue Knoten**
  - **Knoten wenn wahr**

Hiermit kann allerdings nicht festgestellt werden, ob ein Wiederholungsfehler aufgetreten ist. Dafür müssen die Fälle, dass zwei Mal die Regel missachtet wurde (Knoten 3 – wahr), und, dass die Regel mindestens einmal beachtet wurde, (Knoten 1 oder 3 – falsch) unterschieden werden. Hierfür erstellen Sie eine Frage wie **03\_Zweites Eingabefeld b).**

Die Knoten 5 und 6 werden zusätzlich erstellt (als Kopien von 3 und 4) mit dem Unterschied, dass bei Knoten 5 – falsch keine Punkte für den hier nicht vorhandenen Wiederholungsfehler gegeben werden. Nach dem Verbinden mit Knoten 2 sind Sie fertig.



Zu beachtende neue Einstellungen:

- **Rückmeldebäume**
  - **Punkte wenn wahr**
  - **Neue Knoten**

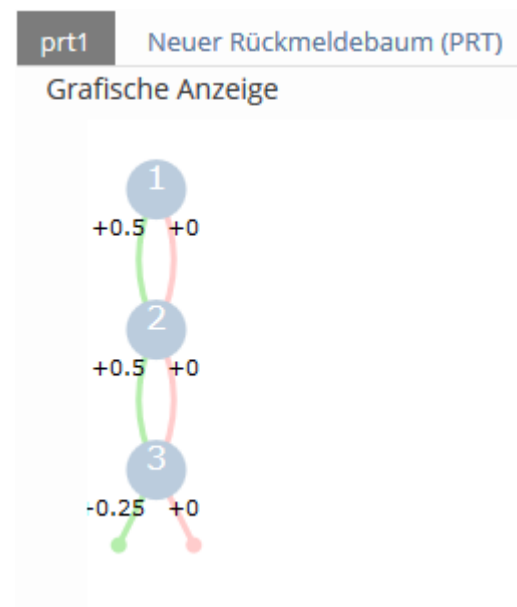
[Einstieg STACK-Frage](#) → [Musterlösungen zur „Erklärung STACK 3“](#) → [03 Zweites Eingabefeld](#)

#### 4. Nutzen von Feedbackvariablen (04\_Maxima)

Inhalt: Maxima, Logische Verknüpfungen, Antwortüberprüfungen, Feedbackvariablen

Bisher wurde die Programmiersprache Maxima, mit der man Variablen global und lokal in STACK definieren kann, in den Beispielen noch nicht benutzt. Diese ist jedoch hilfreich, um logische Verknüpfungen, Berechnungen und Zufallswerte einzufügen. Im Folgenden werden diese Feedbackvariablen benutzt, um die Aufgabe umzuschreiben.

1. Starten Sie mit dem gleichen Aufgabentext wie zuvor.
2. Klicken Sie im Rückmeldebäum **Einstellungen**, im Feld Feedback-Variablen können Sie nun in Maxima programmieren. Hierbei fragt **is(<expr>)** ab, ob der Inhalt der Klammern wahr ist. Der Befehl **<expr>and<expr>** verknüpft die beiden logischen Aussagen zu einem **true** oder **false**.
3. Die so erhaltene Variable kann nun in einem Knoten verwendet werden. Hierfür stellen Sie die Antwortüberprüfung auf **String** und schreiben den Variablennamen in **TAns** und **true** in **SAns**.



Rückmeldebäum von 03\_Zweite Frage 1

4. Nach dem Anpassen der Bepunktung und der Rückmeldungen sind Sie fertig.  
Da sich Knoten 3 – wahr und Knoten 1 oder 2 – wahr ausschließen, ist die maximale Punktzahl auch hier nicht größer als 1.

Beachten Sie, dass das Abfragen von mehr Aufgabenteilen (statt wie hier nur zwei) ohne die Nutzung von Maxima-Befehlen sehr viel aufwendiger ist. Bei drei Abfragen bräuchte man bereits 14 Knoten. Mithilfe des Programmierens über Feedback-Variablen allerdings nur vier.



Zu beachtende neue Einstellungen:

- **Rückmeldebäume**
  - **Einstellungen**
    - **Feedback-Variablen**
  - **Knoten 3**
    - **Antwortüberprüfung**
    - **SAns & TAns**

[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 04\\_Maxima](#)

## 5. Feedback in Fragentext (05\_Feedback)

Inhalte: Mehrere Rückmeldebäume, Feedback im Fragentext

Bei langen Aufgaben ist es übersichtlicher das Feedback passend unter die entsprechenden Aufgabenteile zu schreiben. Hierfür müssen Sie für jedes Feedback einen eigenen Rückmeldebaum (**prt0, 1, ...**) anlegen. Das durch den Baum generierte Feedback wird dann mittels **[[feedback:Rückmeldebaumname]]** im Fragentext aufgerufen. Den Befehl sollten Sie, um Dopplung zu vermeiden, im Feld **Spezifisches Feedback** löschen.

Definieren Sie die Feedback-Variable im Rückmeldebaum wie folgt:

Feedback-Variablen

WdhF: is(ans1=3.5) and is(ans2=8)

Die Feedback-Variablen erlauben es die Eingabe zusammen mit den Aufgabenvariablen zu manipulieren, bevor der Rückmeldebaum durchlaufen wird. Variablen, die hier definiert werden, können überall im Rückmeldebaum benutzt werden.  
Inhalt: CAS-Ausdruck

Zu beachtende neue Einstellungen:

- **Frage**
- **Spezifisches Feedback**
- **Rückmeldebäume**
  - **Neuer Rückmeldebaum (PRT)**
  - **prt1, prt2 & prt3**
    - **Feedback wenn wahr, Feedback wenn falsch**
  - **prt3**
    - **Einstellungen**
      - **Feedback-Variablen**

[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 05\\_Feedback](#)

## 6. Verwendung zufälliger Variablen (06\_Zufallszahlen)

Inhalt: Zufallszahlen, simp, individuelle Aufgaben

Um Aufgaben verschieden zu gestalten oder um bspw. Abschreiben vorzubeugen, sollten Sie Zufallszahlen oder Zufallsausdrücke verwenden. Diese werden durch Maxima mittels Zufallsgeneratoren ausgewürfelt. Der Seed des Zufallsgenerators ist an die Session gebunden, das heißt, dass ein User immer dieselben Zufallszahlen in derselben Aufgabe angezeigt bekommt. Auch bei späteren Logins ist diese gleichbleibend, solange der Test nicht beendet wird. Zufallszahlen können theoretisch jede feste Zahl und jeden Operator zufällig festlegen.

In unserem Beispiel wird die vermutlich häufigste Verwendung von Zufallsgeneratoren in Aufgaben gezeigt. Wir ersetzen feste Zahlenwerte durch Zufallszahlen. Eine Liste mit Erklärungen der Zufallszahlenfunktionen in Maxima gibt es unter: [https://github.com/mathsmoodle-qtype\\_stack/blob/master/doc/en/CAS/Random.md](https://github.com/mathsmoodle-qtype_stack/blob/master/doc/en/CAS/Random.md)

(Hinweis: Dieser Link bezieht sich auf STACK in Moodle. Vieles davon kann so übernommen werden, da es bei STACK in ILIAS genauso funktioniert, doch das trifft nicht auf alles zu.)

Sie müssen die Zufallszahlen müssen im Feld **Aufgabenhinweis** notieren. Dies dient dazu, dass Sie später für einzelne Studierende nachvollziehen können, welche Zufallszahlen in der Aufgabe verwendet wurden. Stellen Sie außerdem **simp:false** ein, um zu verhindern, dass der Bruch automatisch gekürzt wird.

Achten Sie darauf, dass Sie sowohl die Musterlösung als auch den Rückmeldebaum an die neuen Einstellungen anpassen. Dementsprechend sollten Sie die Feedback-Variable WdhF und die Fragen-Variablen anpassen. Sie können die zuvor definierten Funktionen und Fragen-Variablen in den entsprechenden Feldern verwenden. Hierfür können Sie außerdem vorbereitete Zufallswerte nutzen, um für den Test Zufallswerte auszugeben, die man passend findet.

- Den Reiter **Eigenschaften** auswählen
- Darunter wählen Sie **Vorbereitete Zufallswerte** aus.
- **Neuen Wert erzeugen** klicken bewirkt, dass der erste Satz an Zufallswerten und damit die erste Version der Aufgabe erzeugt wird.
- Weitere Werte erzeugen Sie, indem Sie erneut **Neuen Wert erzeugen** klicken.
- Werte, die Ihnen nicht passend erscheinen, können Sie löschen.

Fragenpool für Tests   Seite bearbeiten   Vorschau   **Eigenschaften**   Rückmeldung   Statistik  
 Frage bearbeiten   Bewertung   Vorbereitete Zufallswerte   Testfälle   Fragen aus MoodleXML importieren   Export nach MoodleXML

Der gewählte Wert wurde erfolgreich gelöscht.

NEUEN ZUFALLSWERT ERZEUGEN

Zufallswert

Neuen Wert erzeugen

Dies bewirkt, dass in einem Test nur Zufallswerte gegeben werden, die Sie für sinnvoll erachtet. Diese Einstellung muss nicht zwingend zur Nutzung von Zufallswerten getroffen werden. Alle erzeugten Werte werden schließlich in einer Tabelle angezeigt. Unter **Fragen-Notiz für den Zufallswert** können Sie den Aufgabenhinweis einsehen.

Zu beachtende neue Einstellungen:

- **Frage**
- **Fragen-Variablen**
- **Aufgabenhinweis**
  - **Einstellung der Eingabe**
    - **Musterlösung**
- **Rückmeldebäume**
  - **TAns**
  - **Einstellungen**
    - **Feedback-Variablen**

[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 06 Zufallszahlen](#)

## 7. Vorbeugende Maßnahmen (07\_Verbotene Wörter)

Inhalt: Verbotene Wörter, Zufällige Operatoren

Um Wörter oder Maxima-Befehle (z.B. differenzieren, integrieren, ggT ausrechnen, etc.), zu verbieten, müssen Sie die entsprechenden Befehle oder Operatoren in **Test-Eingaben** → **Einstellungen der Eingabe** → **Verbotene Wörter** eintragen. Dies verhindert, dass Antworten abgegeben werden, welche nicht in der Form, in der Sie diese erwarten.

In der Beispielaufgabe sind dies der Maxima-Begriff „diff“ und die Rechenzeichen „\*“ und „/“. Letztere Maßnahme verhindert, dass die Aufgabenstellung in das Antwortfeld eingetippt wird, statt selbstständig zu rechnen.

Für diese Beispielaufgabe wurden außerdem Zufallsvariablen erstellt. Weiter wurde darauf geachtet, dass eine spezifische Rückmeldung gegeben wird, falls die Produktregel missachtet wird. Der Aufbau der Rückmeldung ist analog zu den in Aufgabe 4 und Aufgabe 5 erläuterten Einstellungen.

Zu beachtende neue Einstellungen:

- **Einstellungen der Eingabe**
- **Verbotene Worte**

[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 07 Verbotene Wörter](#)

## 8. Runden und Funktionen (08\_Runden)

Inhalt: Runden, Funktionen (selbst Definieren)

Um irrationale Zahlenwerte abzufragen ist es nötig, dass die Antwort gerundet wird. Man kann dafür die **Antwortüberprüfung** → **NumAbsolute** verwenden. Dies ist für die meisten Anwendungen ausreichend. Es kann hierbei allerdings zu Problemen führen: Wenn Sie auf zu viele Nachkommastellen runden lassen, wird das Ergebnis ungenau und die Frage sieht falsch gerundete Antworten für richtig an. In der Antwort sind angeblich 3.1415 und 3.1416 „richtig“, in Wahrheit ist jedoch nur 3.1416 richtig gerundet.

Es ist aber möglich eine selbstgemachte Rundungsfunktion zu nutzen. Aus Gründen der Einfachheit ist diese schon geschrieben und Sie benötigen lediglich den Befehl **round2()**, welcher die erste Zeile des **Fragenvariablen**-Text sein sollte. Sich diese oder ähnliche Funktionen anzuschauen ist hilfreich, wenn man selber Funktionen schreiben möchte. Es ist aber auch hilfreich einfach eine passende Funktion zu kopieren und direkt zu verwenden. Die Funktion, die in der Beispielfrage definiert wurde, (mittels „:=“ ) heißt **round2()**.

```
round2(ZTV,ZTN):=(if ZTV*10^ZTN-floor(ZTV*10^ZTN)<0.5 then  
float(floor(ZTV*10^ZTN)*10^-ZTN) else float(floor(ZTV*10^ZTN+1)*10^-ZTN));
```

Sie rundet einen Wert **ZTV** auf **ZTN** Nachkommastellen.

Diese Funktion wird auf die Studierendenantwort in den **Feedback-Variablen** und auf die Musterantwort angewendet, bevor diese verglichen werden. Wenn diese nicht zusammenstimmen wird nachgeschaut ob die Antwort falsch gerundet wurde. Hierfür kann man **NumAbsolute** nehmen und als Intervall (in den **Test-Optionen**) **10^-ZTN** als Dezimalzahl) eingeben.

Zu beachtende neue Einstellungen:

### 09\_Kaufmännisches Runden

- **Fragenvariablen**
  - **round2()**
- **Feedback-Variablen**
  - **TeaAns**
  - **StuAns**

[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 08\\_Runden](#)

## Musterlösungen zu anderen Eingabetypen

### 9. Matrizen (09\_Matrizen)

Inhalt: Matrizen erstellen, Rechnen mit Matrizen

Um Matrizen eingeben zu lassen, stellen Sie den Eingabetyp auf **Matrix**. Die Rechenoperationen in Maxima Syntax zur Verwendung innerhalb der Aufgabe können unter folgendem Link gefunden werden:

[http://maxima.sourceforge.net/docs/manual/maxima\\_118.html#Functions-and-Variables-for-Matrices-and-Linear-Algebra](http://maxima.sourceforge.net/docs/manual/maxima_118.html#Functions-and-Variables-for-Matrices-and-Linear-Algebra)

Herauszustellen ist:

- Das nicht kommutative Matrixprodukt wird durch  $\langle \text{Matrix1} \rangle . \langle \text{Matrix2} \rangle$  beschrieben, die anderen Grundrechenarten sind durch die gewohnten Zeichen aufzurufen.
- Die **genmatrix** Funktion generiert mit Hilfe, einer angegebenen Funktion, eine Matrix. Die im Beispiel benutzte **genmatrix** Funktion generiert eine Matrix aus Integer-Zufallszahlen. Eine ausführliche Erklärung zu dieser Funktion finden Sie in diesem Dokument unter [Maxima → Grundlegende Befehle](#).

Zu beachtende neue Einstellungen:

#### 10\_Zusatzbeispiel: Matrizen

- **Frage**
- **Fragen-Variablen**
- **Einstellungen der Eingabe**
  - **Eingabetyp**
- **Zeige Fragenoptionen**
  - **Matrix Klammern**

[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 09\\_Matrizen](#)

### 10. Kontrollkästchen Einstieg (10\_Kontrollkästchen)

Inhalt: Multiple-Choice Fragen mit Kontrollkästchen, Maxima Syntax Liste

Beachten Sie die [Syntaxhinweise](#) aus den Erläuterungen zum Eingabetyp **Kontrollkästchen**

Arbeitsschritte:

1. Erstellen Sie eine STACK Aufgabe und schreiben Sie in das Feld *Frage*:  
„Bitte geben Sie alle Lösungen der folgenden Gleichung an:  $\backslash ( \{ @ \text{Funk} @ \} \backslash )$   
[[input:ans1]] [[validation:ans1]]“
2. Geben Sie in *Fragen-Variablen* **Funk** einen Wert und geben sie die Lösungsmenge zu Funk in der Variable **Sol** an.

3. Schreiben Sie eine Schleife die die Lösungsmenge syntaktisch in die richtige Form bringt.
  - Schauen Sie sich dazu die Syntaxhinweise für den Kontrollkästchen Eingabetyp an.
  - Schauen Sie sich die Hinweise zu Schleifen an.
  - Schauen Sie sich Maxima Befehle zum Bearbeiten von Listen an.
4. Geben Sie falsche Lösungen an. Schreiben Sie diese syntaktisch korrekt in eine andere Liste.
5. Fügen Sie beide Listen zusammen und hängen Sie „Keine der Antworten ist korrekt.“ an.
6. Werten Sie die Eingabe im PRT mit Hilfe des Feldes Feedback-Variablen aus.
  - Schauen Sie sich die Hinweise zu Schleifen an.
  - Schauen Sie sich Maxima Befehle zum Bearbeiten von Listen an.
7. Geben Sie sinnvolle Rückmeldungen im PRT ein und legen Sie eine Bewertung fest.

Zu beachtende neue Einstellungen:

- **Frage**
- **Fragen-Variablen**
- **Test-Eingabe**
  - **Einstellung der Eingabe**
    - **Eingabetyp**
      - **Kontrollkästchen**
    - **Musterlösung**
- **Rückmeldebäume**
  - **SAns**
  - **TAns**
  - **Einstellungen**
    - **Feedback-Variablen**
  - **Feedback, wenn wahr/falsch**

[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 10 Kontrollkästchen](#)

## 11. Drop-Down-Liste und Radio-Auswahl (11\_Drop-Down-Liste)

Inhalt: Drop-Down-Listen

Das Vorgehen ist ähnlich wie bei der Erstellung einer Multiple-Choice-Aufgabe mit Hilfe des Eingabetyps **Kontrollkästchen**. Beachten Sie dementsprechend die [Syntaxhinweise](#) aus den Erläuterungen den Eingabetypen. Es ist allerdings zu beachten, dass beim Eingabetyp **Drop-Down-Liste** nur eine Antwortmöglichkeit als "true" ausgewiesen werden kann.

Zu beachtende neue Einstellungen:

- **Frage**
- **Fragen-Variablen**
- **Test-Eingabe**
  - **Einstellung der Eingabe**
    - **Eingabetyp**
      - **Drop-Down-Liste**
    - **Musterlösung**
- **Rückmeldebäume**
  - **SAns**
  - **TAns**
  - **Einstellungen**
    - **Feedback-Variablen**
  - **Feedback, wenn wahr/falsch**

[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 11\\_Drop-Down-Liste](#)

## 12. Radio-Auswahl (12\_Radio-Auswahl)

Inhalt: Radio-Auswahl

Das Vorgehen ist ähnlich wie bei der Erstellung einer Multiple-Choice-Aufgabe mit Hilfe des Eingabetyps **Kontrollkästchen**. Beachten Sie dementsprechend die [Syntaxhinweise](#) aus den Erläuterungen den Eingabetypen. Es ist allerdings zu beachten, dass beim Eingabetyp **Radio-Auswahl** nur eine Antwortmöglichkeit als "true" ausgewiesen werden kann.

Zu beachtende neue Einstellungen:

- **Frage**
- **Fragen-Variablen**
- **Test-Eingabe**
  - **Einstellung der Eingabe**
    - **Eingabetyp**
      - **Radio-Auswahl**
    - **Musterlösung**



- **Rückmeldebäume**
  - **SAns**
  - **TAns**
  - **Einstellungen**
    - **Feedback-Variablen**
  - **Feedback, wenn wahr/falsch**

[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 12 Radio-Auswahl](#)

### 13. Gleichungen Einstieg (13\_Gleichungen)

Inhalt: Gleichungen

Beachten Sie die Syntaxhinweise im Abschnitt Eingabetypen. Außerdem müssen folgende Einstellungen getroffen werden:

- **Frage**
- **Fragen-Variablen**
- **Test-Eingabe**
  - **Einstellung der Eingabe**
    - **Eingabetyp**
      - **Gleichungen**
    - **Musterlösung**
    - **Eingabebreite**
    - **Syntax-Hinweis**
      - *firstline*
- **Rückmeldebäume**
  - **SAns**
  - **TAns**
  - **Einstellungen**
    - **Feedback-Variablen**
  - **Feedback, wenn wahr/falsch**

[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 13 Gleichungen](#)

### 14. Notizen (14\_Notizen)

Inhalt: Notizen, Feedback, Rückmeldungen

Zu beachtende Einstellungen:

- **Frage**
- **Test-Eingabe**
  - **Einstellung der Eingabe**
    - **Eingabetyp**

- **Notizen**
  - **Musterlösung**
    - *true*
  - **Eingabebreite**
- **Rückmeldebäume**
  - **SAns**
  - **TAns**
  - **Feedback, wenn wahr/falsch**

[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 14. Notizen](#)

## 15. Einheiten (15\_Einheiten)

Inhalt: SI-Einheiten

Beachten Sie bei der Eingabe der Musterlösung, dass Sie die Maßeinheiten mit Multiplikationszeichen verbinden müssen.

Zu beachtende Einstellungen:

- **Frage**
- **Fragen-Variablen**
- **Test-Eingabe**
  - **Einstellung der Eingabe**
    - **Eingabetyp**
      - **Einheiten**
    - **Musterlösung**
- **Rückmeldebäume**
  - **SAns**
  - **TAns**
  - **Feedback, wenn wahr/falsch**

[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 15. Einheiten](#)

## 16. Zeichenkette (16\_Zeichenkette)

Inhalt: Zeichenkette, Begriffe

Beachten Sie die [Syntaxhinweise](#) zu dem Eingabetyp **Zeichenkette**.

Zu beachtende Einstellungen:

- **Frage**
- **Fragen-Variablen**
- **Test-Eingabe**
  - **Einstellung der Eingabe**
    - **Eingabetyp**
      - **Zeichenkette**
    - **Musterlösung**
- **Rückmeldebäume**
  - **1**
    - **SAns**
    - **Tans**
    - **Knoten, wenn falsch**
    - **Feedback, wenn wahr**
  - **Neuer Knoten**
    - **SAns**
    - **TAns**
    - **Feedback, wenn wahr/falsch**

[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 16 Zeichenkette](#)

## 17. Numerisch (17\_Numerisch)

Inhalt: Numerische Eingaben, Runden, signifikante Stellen, Rundungsfehler

Sie können die Antwort auf die richtige Anzahl der signifikanten Stellen mithilfe der Einstellung **NumSigFigs** unter **Antwortüberprüfung** im **Rückmeldebaum** untersuchen. Unter **Test-Optionen** können Sie die gewünschte Anzahl eingeben. Beachten Sie, dass hier nicht die Anzahl der Nachkomma-Stellen, sondern aller signifikanteren Stellen überprüft wird.

Falls Sie abweichende Ergebnisse aufgrund von Rundungsungenauigkeiten zulassen möchten, können Sie unter **Test-Optionen** eine Liste der Form  $[n, n-1]$  angeben. Hierdurch wird die Antwort auf  $n$  Stellen geprüft. Die Übereinstimmung mit der Musterlösung wird allerdings nur bis zur  $n-1$ -ten Stellen abgeglichen. Die Nutzung von **NumSigFigs** bei einer Eingabe von  $[4, 3]$  unter **Testoptionen** bedeutet in diesem Fall, dass alle Angaben von  $\pi$  der Form  $3.14x$  als richtig bewertet werden.

Des Weiteren werden mit **NumSigFig** bei Abweichungen der signifikanten Stellen oder der Rundungsgenauigkeit automatische Fehlermeldungen generiert:

„Ihre Antwort hat die falsche Anzahl an Dezimalstellen. Die Genauigkeit Ihrer Antwort ist nicht korrekt. Entweder haben Sie das Endergebnis oder einen Zwischenwert falsch gerundet.“

„Die Genauigkeit Ihrer Antwort ist nicht korrekt. Entweder haben Sie das Endergebnis oder einen Zwischenwert falsch gerundet.“

Zu beachtende Einstellungen:

- **Frage**
- **Fragen-Variablen**
- **Test-Eingabe**
  - **Einstellung der Eingabe**
    - **Eingabetyp**
      - **Numerisch**
    - **Musterlösung**
- **Rückmeldebäume**
  - **Antwortüberprüfungs**
    - **NumSigFigs**
  - **Test-Optionen**
  - **SAns**
  - **TAns**
  - **Feedback, wenn wahr**

[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 17 Numerisch](#)

## Musterlösungen für eine ansprechende Rückmeldung

### 18. HTML zur Darstellung verwenden. (18\_HTML Attribute)

Inhalt: HTML verwenden

Die Gestaltung der Felder mit HTML kann die STACK-Frage für Studierende übersichtlicher machen. Dies gilt vor allem für die Feedback Felder. Wenn Sie hier Farben verwenden, können Studierende auf den ersten Blick sehen, ob ihre Antwort richtig oder falsch ist.

Um HTML zu verwenden benutzen sie den HTML Knopf über den Eingabefeldern (entweder Frage oder Feedback)



Es öffnet sich ein neues Fenster, in welches Sie nun HTML Befehle eingeben können. Nach einem Klick auf **Aktualisieren** wird das Fenster geschlossen und die Änderungen werden übernommen.

Wenn Sie sich selbst mit den HTML-Befehlen auseinandersetzen wollen, geht dies beispielsweise unter [www.w3schools.com](http://www.w3schools.com). Alternativ können Sie auch aus den Beispielfragen den HTML Code herauskopieren und in Ihre eigenen Fragen einbauen.

Der Code, den wir zur Darstellung von farbigem Feedback vorschlagen finden Sie in diesem Dokument unter HTML -> Darstellung.

In unserem Anschauungsbeispiel wird gezeigt wie man:

- Ein Feedback-Feld versteckt.
- Hintergrundfarben von Feedback-Feldern ändert.
- Anker für HTML-Bereiche setzt.
- Ränder an HTML-Bereiche anfügt.
- Transparenz für HTML-Bereiche einstellt.
- Schriftfarben ändert.
- Mehrere oder einzelne Feedbacks nahtlos darstellt.
- Hyperlinks einfügt.
- Kommentare einfügt.

*Hinweis:* Die von uns verwendeten Standardfarben sind: rot (#ffe7e7), grün (#e7ffe7) und gelb (#fcf8e3). Andere Farben finden Sie, wenn Sie HTML Farben im Internet suchen.

[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 18\\_HTML Attribute](#)

## 19. Plots für Rückmeldungen verwenden (19\_Plot Ableitung)

Inhalt: Plots verwenden

Plots können Sie verwenden, indem Sie im „Frage“-Feld oder im „Feedback wenn...“ Feld @plt@ aufrufen, wobei auf p in den Frage-Variablen oder den Feedback-Variablen vorher ein Plot-Befehl geschrieben wurden muss.

Feedback-Variablen

```
plt:plot([ans1,f],[x,-5,5],[color,red,blue])  
plt2:plot([ans1,fdx],[x,-5,5],[color,red,green])
```

Die Feedback-Variablen erlauben es die Eingabe zusammen mit den Aufgabenvariablen zu manipulieren,

Herauszustellen ist:

- Der Maxima-Befehl „plot2d“ oder „plot3d“ funktioniert in STACK aus Sicherheitsgründen nicht, jedoch können viele der „plot2d“ Attribute in **plot** verwendet werden.
- **plot([f1,f2,...],[x,x0,x1],[y,y0,y1])** gibt einen Plot der Funktionen f1, f2,... aus, welcher zwischen x0 und x1 bzw. y0 und y1 dargestellt wird.

Sie können auch **plot(f1,[x,x0,x1])** verwenden. In den Beispielen gibt es in der Frage *14\_Plot-Optionen* weitere Optionen zum Anschauen.

Wenn Sie @plt@ in der Rückmeldung aufrufen, sollten Sie folgenden HTML-Befehl verwenden:

```
<div style="margin: -0px -16px -0px; padding: -0px 16px 0px; opacity: 1; background-color: ffcccc70;"><div style="opacity: 1; background-color: white;">@plt@</div></div>
```

Dieser macht die Hintergrundfarbe weiß und entfernt die Ränder. Auf dem gelben Standard-Hintergrund ist der Plot schwieriger zu erkennen.

[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 19\\_Plot Ableitung](#)

## 20. Plots Attribute Beispielsammlung (20\_Plot-Optionen)

Inhalt: Beispielplots

Herauszustellen ist:

- Die Beispiele stammen von [https://github.com/mathsmoodle-qtype\\_stack/blob/master/doc/en/CAS/Plots.md](https://github.com/mathsmoodle-qtype_stack/blob/master/doc/en/CAS/Plots.md). Die auf dieser Seite verwendete Syntax ist ab STACK 3.0.x auch in ILIAS verwendbar. Die meisten Befehle funktionieren bereits in STACK 2.
- Zur einfacheren Darstellung wurde diese Frage erstellt. Sie prüft nichts ab, soll aber zeigen wie Plots in STACK aussehen können.
- Die Plots sind alle im Fragefeld erstellt worden. Wir empfehlen zur besseren Übersicht den Plot im Frage-Variablen oder Feedback-Variablen Feld zu erstellen.

[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 20\\_Plot-Optionen](#)

## Kontakt

Falls Sie Fragen und Anregungen zu diesem Dokument oder zu STACK haben können Sie sich gerne jederzeit an uns wenden.

Sie erreichen uns per E-Mail:

**[llias-stack@uni-koeln.de](mailto:llias-stack@uni-koeln.de)**

## STACK-Team

Eva Mix

Mario Josupeit

Tjark Eilts