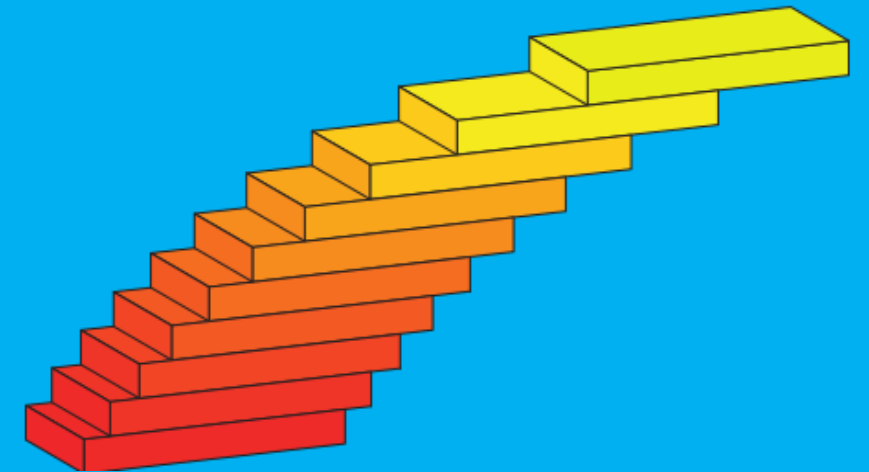


Parametrisierbare Aufgaben mit STACK: Aufgaben mit verschiedenen Zahlensystemen



Hinweis: Hierbei handelt es sich um ein Feature, das erst in STACK Version 4.12.0 (April 2026) hinzugefügt wurde. Wenn Sie eine ältere Version von STACK nutzen, können die im Video gezeigten Features noch nicht genutzt werden.

Stand: 09.10.2026

STACK // Aufgaben mit verschiedenen Zahlensystemen

- In dieser Anleitung wird anhand zweier Beispielaufgaben erklärt, wie Aufgaben mit STACK erstellt werden können, in denen die Studierenden Zahlen in verschiedenen Zahlenbasen eingeben müssen (z. B. Binär- oder Hexadezimalsystem):
 1. Es ist eine Zahl im Binärsystem gegeben und diese soll in das Dezimalsystem umgerechnet werden.
 2. Es ist eine Zahl im Dezimalsystem gegeben und diese soll in das Hexadezimalsystem umgerechnet werden.

Wandeln Sie 10100_2 vom Binärsystem in das Dezimalsystem um.

Antwort:

Wandeln Sie die Zahl 64 vom Dezimalsystem in das Hexadezimalsystem um.

Antwort:

Hinweis: Wir empfehlen, vor dem Anschauen dieser Anleitung zunächst die Klickanleitungen „Erstellung einer Aufgabe“ und „Variablen und Randomisierung“ durchzuarbeiten oder die gleichnamigen Videotutorials zu schauen.



Aufgabe 1

Umwandlung vom Binär- in das Dezimalsystem



STACK // Aufgabe 1: Fragetitel und Aufgabenvariablen (1)

- Erstellen Sie eine neue STACK-Aufgabe und geben Sie ihr einen geeigneten Fragetitel, z. B. „Basis 2 zu Basis 10“.
- Im Feld „Aufgabenvariablen“ wird zunächst eine Variable für die Zahl im Binärsystem benötigt, die den Studierenden im Fragetext erscheint und die in das Dezimalsystem umgerechnet werden soll. Ziel ist also, ein sogenanntes „STACK-Base-n-Objekt“ zu erzeugen. Dazu gibt es im Wesentlichen zwei Möglichkeiten:
 1. Funktion `stackbasen`:

Mit dieser Funktion kann auf direktem Wege ein STACK-Base-n-Objekt erstellt werden. Die Funktion benötigt drei Argumente:

 - Zahl in der gewünschten Basis (hier also Basis 2) in Anführungszeichen, z. B. `"10001_2"` (wobei mit `_2` gekennzeichnet wird, dass es sich um eine Zahl im Binärsystem handelt)
 - Entweder `"C"` oder `"S"`. Diese Option steuert, wie genau die Zahl angezeigt wird (z. B. im Fragetext) und wie Studierende sie eingeben müssen. Wird `"C"` verwendet, wird die Binärzahl 10001 als `0b10001` angezeigt und muss auch so eingegeben werden. Mit der Option `"S"` wird diese Zahl als `100012` angezeigt und als `10001_2` eingegeben. Letztere Option wird in diesem Video verwendet.
 - Basis der Zahl, in diesem Fall also 2.

→ Insgesamt also so: `stackbasen("10001_2", "S", 2)`

*Hinweis: Es können die Basen
2 bis 36 verwendet werden.*

STACK // Aufgabe 1: Aufgabenvariablen (2)

2. Funktion `sbasen_construct_from_base10`:

Mit dieser Funktion kann eine (ggf. randomisierte) Zahl im Dezimalsystem in ein STACK-Base-n-Objekt konvertiert werden. Dies ist in der Regel die sinnvollere Option. Auch sie benötigt drei Argumente:

- Zahl im Dezimalsystem. Dies kann auch eine randomisierte Variable sein, zum Beispiel:
`x: rand_with_step(10, 20, 1);`
- Entweder "C" oder "S" (siehe vorherige Folie).
- Gewünschte Basis der Zahl, in diesem Fall also 2.

→ Insgesamt also so: `x2: sbasen_construct_from_base10(x, "S", 2);`

- Zuletzt wird noch eine Variable für die Musterlösung `ta1` der Aufgabe benötigt. Wenn gewünscht ist, dass Studierende die Dezimalzahl mit Basisangabe eingeben (d. h. mit `_10`), wird als Musterlösung ebenfalls ein Base-n-Objekt benötigt. Dazu kann erneut `sbasen_construct_from_base10` verwendet werden, zum Beispiel so:
`ta1: sbasen_construct_from_base10(x, "S", 10);`

Hinweis: Möchten Sie einen STACK-Base-n-Ausdruck in eine Zahl im Dezimalsystem umwandeln, dann ist dies mit der Funktion `sbasen_convert_to_base10` möglich. Sie erwartet einen Base-n-Ausdruck als Argument und gibt eine ganze Zahl zurück.

Aufgabenvariablen ?

```
x: rand_with_step(10,20,1);  
x2: sbasen_construct_from_base10(x,"S",2);  
  
ta1: sbasen_construct_from_base10(x,"S",10);
```

STACK // Aufgabe 1: Fragetext, Aufgabenhinweis und Bereich „Eingabe“

- Tragen Sie nun den Fragetext der Aufgabe ein, zum Beispiel „Wandeln Sie $\{ @x2@ \}$ vom Binärsystem in das Dezimalsystem um.“
- Im Feld „Aufgabenhinweis“ kann der bereits vorhandene Wert $\{ @ta1@ \}$ beibehalten werden.
- Verwenden Sie im Bereich **1 Eingabe** den Eingabetyp „Algebraische Eingabe“ und ta1 als Musterlösung.
- Da die Studierenden intuitiv nicht wissen, wie ein Base-n-Ausdruck in STACK eingegeben werden kann, ist es sinnvoll, einen **2 Syntaxhinweis** zu geben (vgl. Screenshots auf Folie 2). Tragen Sie dazu im Feld „Syntax-Hinweis“ . . . _10 ein und setzen die Option „Hinweis-Attribut“ auf „Platzhalter“.
- Damit Base-n-Objekte als Eingabe akzeptiert werden, muss im Feld **3 Zusätzliche Optionen** ganz unten das Schlüsselwort basen eingetragen werden.

1  Eingabe: ans1

Eingabetyp		Algebraische Eingabe 
Musterlösung		ta1

2 Syntax-Hinweis  || Hinweis-Attribut | Platzhalter |

3 Zusätzliche Optionen  |

STACK // Aufgabe 1: Rückmeldebaum

- In **1 Knoten 1** des Rückmeldebaums können Sie die voreingestellte Konfiguration (SAns: ans1, TAns: ta1, Antwortüberprüfung: AlgEquiv) beibehalten. Dann akzeptiert STACK Antworten als korrekt, in denen die Studierenden via `_10` die Basis mit angegeben haben.

1 Knoten 1

Beschreibung:

Antwortüberprüfung: AlgEquiv

SAns: ans1

TAns: ta1

Feedback unterdrücken: Nein

- Wenn zusätzlich auch Antworten ohne `_10` akzeptieren werden sollen:
 - Fügen Sie einen **2 zweiten Knoten** hinzu.
 - Wenn im ersten Knoten der Fall „Falsch“ eintritt, soll der neue **3 Knoten 2 nachgeschaltet** werden.
 - Konfigurieren Sie **4 Knoten 2** wie folgt: SAns: ans1, TAns: x (x wurde auf [Folie 5](#) definiert). Legen Sie zudem über das Feld „Score“ fest, dass die Studierenden im Fall „Wahr“ die volle Punktzahl erhalten.

2 Weitere(n) Knoten hinzufügen

Knoten 1 wenn FALSCH

Mod: =

Score: 0

Abzüge:

Nächster: Knoten 2 **3**

4 Knoten 2

Beschreibung:

Antwortüberprüfung: AlgEquiv

SAns: ans1

TAns: x

Feedback unterdrücken: Nein

Knoten 2 wenn WAHR

Mod: +

Score: 1

Abzüge:

Nächster: [stop]

Antworthinweis: prt1-2-T

Aufgabe 2

Umwandlung vom Dezimal- in das Hexadezimalsystem

STACK // Aufgabe 2: Fragetitel und Aufgabenvariablen

- Erstellen Sie eine neue STACK-Aufgabe und geben Sie ihr einen geeigneten Fragetitel, z. B. „Basis 10 zu Basis 16“.
- Im Feld **1 Aufgabenvariablen** gibt es Ähnlichkeiten zur vorherigen Beispielaufgabe, allerdings wird hier mit einer Zahl im Dezimalsystem gestartet und als Zielbasis 16 verwendet.
- Starten Sie dazu mit einer zufälligen Variable, zum Beispiel: `x: rand_with_step(50, 70, 1);` Diese Zahl erscheint den Studierenden später im Fragetext. Optional können Sie auf `x` noch die Funktion `sbasen_construct_from_base10` anwenden, damit die Zahl den Studierenden als Base-n-Ausdruck angezeigt wird (wird in dieser Anleitung nicht gemacht).
- Als Musterlösung `ta1` der Aufgabe wird ein STACK-Base-n-Objekt benötigt (Basis 16):
`ta1: sbasen_construct_from_base10(x, "S", 16);`

1

Aufgabenvariablen ?

```
x: rand_with_step(50,70,1);
```

```
ta1: sbasen_construct_from_base10(x, "S", 16);
```

STACK // Aufgabe 2: Fragetext, Aufgabenhinweis, Eingabe und PRT

- Schreiben Sie als nächstes den Fragetext der Aufgabe, zum Beispiel: „Wandeln Sie die Zahl {@x@} vom Dezimalsystem in das Hexadezimalsystem um.“
- Das Feld „Aufgabenhinweis“ und der Bereich „Eingabe“ sind bei dieser Aufgabe analog zur vorherigen Beispielaufgabe (siehe Folie 6).
- Behalten Sie die voreingestellte Konfiguration des Rückmeldebaums erneut bei (SAns: ans1, TAns: ta1, Antwortüberprüfung: AlgEquiv).

Hinweis: Ein zweiter Knoten wie in der ersten Beispielaufgabe wird hier nicht benötigt, denn dies war nur für den Spezialfall relevant, dass eine Zahl im Dezimalsystem angegeben werden soll: Eine Zahl im Hexadezimalsystem kann auch die Buchstaben A bis F enthalten und daher nicht ohne Angabe der Basis 16 eingegeben werden.

- Die Erstellung der Aufgabe ist so nun erst einmal abgeschlossen. Allerdings können optional noch zwei Dinge angepasst werden:
 - Bis jetzt akzeptiert STACK nur Hexadezimalzahlen, in denen Großbuchstaben verwendet werden (Beispiel: Wenn 783 im Hexadezimalsystem angegeben werden soll, dann wird 30F_16 akzeptiert, aber 30f_16 nicht).
 - Bis jetzt akzeptiert STACK keine Antworten mit führenden Nullen (Beispiel: Wenn 783 als Hexadezimalzahl angegeben werden soll, dann wird 30F_16 akzeptiert, aber 0030F_16 nicht).

STACK // Aufgabe 2: Groß-/Kleinschreibung und führende Nullen

- Erstellen Sie dazu im Rückmeldebaum im Feld **1 Feedback-Variablen** eine Variable sans. Mithilfe der Funktion `sbasen_convert_to_suffix_notation` kann die Antwort der Studierenden `ans1` in ein Base-n-Objekt umgewandelt werden, in dem nur Großbuchstaben verwendet werden:
`sans: sbasen_convert_to_suffix_notation(ans1);`
- Mithilfe der Funktion `sbasen_trim_padding` können anschließend noch alle führenden Nullen aus `sans` entfernt werden: `sans: sbasen_trim_padding(sans);`
- Nutzen Sie in Knoten 1 des Rückmeldebaums im Feld `SAns` nun `sans` anstelle von `ans1`.

1 Feedback-Variablen ?

```
sans: sbasen_convert_to_suffix_notation(ans1);  
sans: sbasen_trim_padding(sans);
```

Hinweis: Die Studierenden können mit dieser Konfiguration des Rückmeldebaums nicht nur Kleinbuchstaben und führende Nullen verwenden, sondern ihre Antwort auch in der Schreibeise angeben, in der die Basis 16 nicht im Index, sondern durch Voranstellung von 0x angegeben wird (z. B. 0x17F2).

STACK // Moodle-Kurs „Tutorials STACK“

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



The screenshot shows the Moodle course page for 'Tutorials zu STACK' at RUB. The header includes the RUB logo and navigation links: Dashboard, LEARN2Gether, OpenRUB, and Support. The course title 'Tutorials zu STACK' is prominently displayed next to a colorful staircase graphic. Below the title, a paragraph explains that STACK is a Moodle question type for digital mathematical tasks. Another paragraph mentions that the course includes video tutorials and example tasks. A final paragraph states that the course is public and accessible without login, but an account is needed for full functionality.

RUB Dashboard LEARN2Gether OpenRUB Support Teilnehmer/in

Alles einklappen

Tutorials zu STACK

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

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

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



The thumbnail features a blue background with various number systems: hexadecimal (0x1F47B), decimal (9F96E₁₆), binary (101101₂), and octal (0b1111011). It includes an illustration of two people at a computer screen displaying a conversion task. The text on the screen asks to convert a number from hexadecimal to decimal. The HRW logo is in the bottom right corner.

Aufgaben mit verschiedenen Zahlensystemen

0x1F47B

9F96E₁₆

101101₂

0b1111011

NEVER STOP GROWING

HRW

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



Anmerkung für Nachnutzende | Version 1.0.a vom 09.10.2026

Empfohlene Zitation:

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

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

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

Zusätzliche Informationen für Nachnutzende:

Ursprungsort:

Moodle-Kurs „E-Learning@HRW – Selbstlernkurs STACK“ (<https://elearning.hs-ruhrwest.de/course/view.php?id=24999>), Stand 09.10.2026, Version 1.0.a

Urheber:in(nen):

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

Bildnachweise (einzeln):

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