

DIDAKTISCHER KOMMENTAR

MATHEMODUL MZ

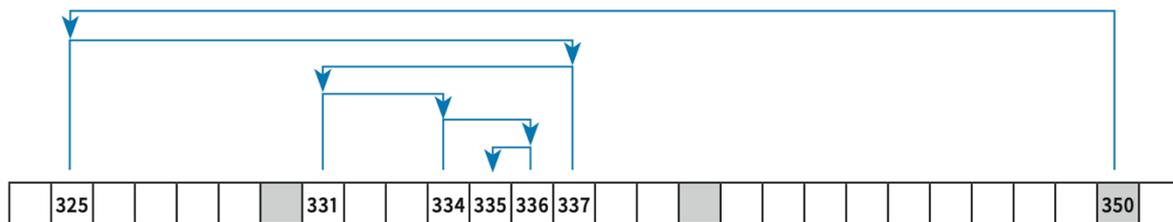
ZAHLEN RATEN

DIDAKTISCHE GRUNDSÄTZE

In diesem Modul wird das in der Grundschule bekannte Spiel **Zahlen raten** zum Gegenstand der Vertiefung algorithmischen Denkens. *Zahlen raten* kann dabei aus zwei Perspektiven betrachtet werden:

- Aus der Rolle der *Spielleitung* - dieses Kind überlegt sich die *gedachte Zahl* in einem definierten Zahlenraum, lässt andere Kinder raten, und gibt die Rückmeldungen „Zu groß!“, „Zu klein!“ oder „Stimmt!“.
- Aus der Rolle des *Ratekindes*, welches versucht, mehr oder weniger geschickt aus den Rückmeldungen der Spielleitung eine *Ratezahl* zu nennen und damit die *gedachte Zahl* zu finden.

Zum einen ist der Spielablauf erkennbar algorithmisch, beide Rollen folgen einem festgelegten Ablauf im Spiel, zum anderen aber ist es möglich, die *gedachte Zahl* in einer vorhersagbaren möglichst kleinen maximalen Zahl von Rateschritten zu raten. Dahinter steckt die Idee, nicht beliebig zu raten, sondern den **Algorithmus der Suchraumhalbierung** anzuwenden. Dieser funktioniert wie folgt:



1

In der Abbildung wird eine *gedachte Zahl* zwischen 325 und 350 gesucht, es ist die Zahl 335. Im ersten Schritt besteht der Fall einer ungeraden Differenz, hier 25, zwischen 325 und 350. Die halbe Differenz beträgt damit 12,5, und aus $325 + 12,5$ ergibt sich, dass die Mitte von 325 und 350 zwischen 337 und 338 liegt. In diesem Fall (ungerade Differenz) wählt man eine der beiden „fast“-Mitten-Zahlen als *Ratezahl*. Hier wird die Zahl 337 gewählt, und die Antwort ist „Zu groß!“. Damit kann die obere Hälfte des Suchraums zwischen 337 und 350 ausgeschlossen werden, die *Ratezahl* muss sich in der unteren Hälfte des vorherigen Zahlenbandes befinden. Damit wurde der Suchraum halbiert (was dem Algorithmus seinen Namen gibt). In diesem neuen Suchraum zwischen 325 und 337 ist die Differenz nun 12, die halbe Differenz 6, und somit kann über $325 + 6$ die Mittelzahl 331 ermittelt werden. In diesem Fall ist die Differenz gerade, und die Mitte liegt nicht zwischen zwei Zahlen, sondern ist eine Zahl, hier die 331. Nun heißt die Antwort „Zu klein!“. Das bedeutet, die untere Hälfte des Zahlenbandes des Suchraums wird gestrichen, und die obere Hälfte davon ist der neue wiederum halb so große Suchraum für die Position der *gedachten Zahl* und des Probierens der neuen *Ratezahl* mit einer Differenz von nur noch 6. Die neue (zu kleine) Mittelzahl = *Ratezahl* 334 lässt dann nur noch die Zahlen 335 und 336 als

¹ Abbildung entnommen aus: Schwätzer, U. (2023). Algorithmisches Denken im Mathematikunterricht – Was unter der Förderung algorithmischen Denkens verstanden wird und warum das Fach Mathematik sich dazu anbietet. *Mathematik differenziert: Zeitschrift für die Grundschule*, 2, S. 7

mögliche *gedachte Zahl* zu. Spätestens nach 5 Rateschritten lässt sich also die *gedachte Zahl* finden bei einer Suchraumdifferenz 25 finden ($25 \rightarrow 12 \rightarrow 6 \rightarrow 3 \rightarrow 2 \rightarrow \text{Treffer}$), ggf. früher, wenn die *gedachte Zahl* zufällig schon auf einer der gewählten *Ratezahlen* liegt.

Weil dieser Algorithmus die Fähigkeit voraussetzt, **die Mitte zwischen zwei Zahlen finden** zu können, beginnen die Aktivitäten des Moduls MZ eben in diesem Themenbereich. Zunächst spielerisch, dann rechnerisch (mit den beiden Fällen gerade und ungerade Differenz) wird das Finden der Mittelzahl erarbeitet und dabei durch ikonische Darstellungen unterstützt. Der Zahlenraum wurde auf 1 bis 60 festgelegt, weil in diesem Zahlenraum noch gut ikonische Darstellungen in den Scratch-Modulen möglich sind, und die kopfrechnerischen Anforderungen von allen Schülerinnen und Schülern meisterbar sein sollten. Durch ein bereitgehaltenes physisches 60er-Feld kann zudem noch die enaktive Ebene hinzugenommen werden (bspw. für gegensinniges Abstreichen von der unteren und der oberen Zahl aus).

Dann wird das **Spiel Zahlen raten** gespielt – zunächst gegeneinander, dann gegen Dee (so heißt der gewählte Protagonist in Scratch) bzw. das Programm, das hinter Dees Verhalten steht. Zunächst wird in der Variante „Dee als Spielleiter“ gespielt – Dee legt die *gedachte Zahl* fest, die Kinder geben die *Ratezahlen* ein. Dann wird in der Variante „Dee als Rate-Kind“ gespielt. Nun legen die Kinder die *gedachte Zahl* fest, und Dee gibt *Ratezahlen* an. Dabei scheint Dee eine Art künstlicher Intelligenz zu benutzen – er gibt schon vorab an, dass er maximal 6 Versuche brauchen wird (und auch schafft). Über den Impuls, was Dees Trick mit der Mitte zwischen zwei Zahlen zu tun hat, sollen die Kinder auf die Spur des dahinterliegenden Algorithmus der Suchraumhalbierung (hier aus dem Moduleinstieg als *Mitte zwischen zwei Zahlen* bekannt) gebracht werden. Das heißt, algorithmisches Denken erfolgt hier rekursiv-analytisch – aus der Vorgehensweise von Dee wird die algorithmische Struktur abgeleitet, wobei die verwendeten Programm-Ablauf-Pläne und die halb fertige Lernumgebung in Scratch, in der die Kinder dann selbst Dees Algorithmus wiederherstellen müssen, diesen Prozess unterstützen. Abschließend erfolgt ein Transfer des Verständnisses, wenn Dees Algorithmus im Zahlenraum bis eine Million getestet wird.

Um dieses Modul im Unterricht einsetzen zu können (Empfehlung: Klasse 4), müssen die Kinder das **Basismodul** bearbeitet haben oder adäquate Vorkenntnisse besitzen. Die Inhalte des Basismoduls und die darin erlernten Fähigkeiten und Umgehensweisen mit der Programmierumgebung Scratch werden vorausgesetzt.

Ebenso werden im didaktischen Kommentar des Basismoduls alle Hinweise zur Arbeit mit den Projekt-Materialien gegeben (Partnerarbeit – technische Voraussetzungen – Umgang mit Aufgabenkarten – 3yp – Hinweise zur Unterrichtsplanung und -durchführung – vor allem sei hier an die prinzipielle Adaptionmöglichkeit einzelner Aufgaben in gemeinsame Besprechungen oder Frontalunterricht erinnert). Diese gelten für das vorliegende Mathemodul analog und werden daher an dieser Stelle nicht wiederholt. Es empfiehlt sich also, vorab einen Blick in den didaktischen Kommentar des Basismoduls zu werfen.

Es ist sinnvoll, parallel zur Arbeit am Modul einen Sprachspeicher gemeinsam mit den Kindern zu entwickeln, der die wichtigsten Begriffe aus dem hier vorliegenden Themenfeld enthält: *die untere/ mittlere/ obere Zahl* (siehe MZ2) | *die Differenz* und *die Summe* | *Zahlen raten* (so heißt das Spiel) | *die gedachte Zahl* (die Zahl, die vor Spielbeginn von der Spielleitung festgelegt wird) und *die Ratezahl* (die Zahl, die in jedem Versuch gesagt wird, um die gedachte Zahl zu raten) | *die Spielleitung* (gegenderte Form von Spielleiterin/ Spielleiter, legt die gedachte Zahl fest) und *das Ratekind* (rät via *Ratezahlen* die *gedachte Zahl*).

ÜBERSICHT ÜBER DIE MODUL-INHALTE

Das Modul MZ – *Zahlen raten* – lässt sich in drei Teile untergliedern: Die Aufgaben MZ2 bis MZ4 thematisieren die Idee „die Mitte zwischen zwei Zahlen finden“ und damit die zugrundeliegende Mathematik. Die Aufgaben MZ5 bis MZ10 fokussieren dann auf das Spiel *Zahlen raten* und die den Algorithmus der Suchraumhalbierung – auf enaktiver Ebene (60er-Feld), auf ikonischer Ebene (Bildschirm), auf symbolischer Ebene (Kopfrechnen), auf

konzeptueller Ebene (Programm-Ablauf-Pläne) und auf digitaler Ebene (in der Programmierumgebung Scratch). Die Aufgaben MZ11 und MZ12 stellen dann den reflektiven Transfer in andere Zahlenräume und ggf. der dahinterliegenden Struktur dar. Aus der Erprobung lässt sich als Erfahrungswert angeben, dass man für jeden dieser drei Teile etwa mit etwa 2-5 Unterrichtsstunden an Arbeitszeit einplanen kann (abhängig von der konkreten Ausgestaltung und der Bearbeitungstiefe, wie im didaktischen Kommentar zum Basismodul beschrieben).

Vorzubereitendes Material aus *MZ1.pdf* (verfügbar unter *promaprim.de* - Projektmaterialien oder unter *3yp.de*):

- *Aufgabenkarten 1-4*: 1x pro Partnerteam.
- Arbeitsblatt *MZ6* (Anhang zur Aufgabenkarte): 1x pro Partnerteam.
- Arbeitsblatt *60er-Feld* (Anhang zur Aufgabenkarte): in geeignet großer Zahl bereithalten.
- Dazu ein digitales Arbeitsmittel mit Internetzugang pro Partnerteam.

Kommentar zu den einzelnen Aufgaben:

MZ1 Die Kinder tragen ihre Namen ein und haken im Folgenden die erledigten Aufgaben auf der Karte ab.

MZ2 Zunächst sollen die Kinder in Teilaufgabe 1 (Aufgabenkarte in 3yp) mehrfach gegeneinander in der Scratch-Animation das Spiel „Die Mitte zwischen zwei Zahlen finden“ spielen. Um die Spielanleitung zu präzisieren: Wenn beide Kinder die untere und die obere eingekreiste Zahl sehen, dann solle jedes für sich versuchen, die Mitte zu finden, und wer von beiden meint, diese gefunden zu haben, soll sie laut sagen. Das andere Kind hat dann die Aufgabe, die Angabe zu überprüfen. An dieser Stelle sind zur Einführung und Durchführung auch anderen Unterrichtsszenarien denkbar und ggf. auch sinnvoll. Ebenso hat sich in der Erprobung gezeigt, dass eine Zwischenreflexion sinnvoll sein kann, wenn die Kinder ein bis zwei erste Aufgabe gelöst haben. Die Strategien sind sehr unterschiedlich – einige Kinder schätzen oder raten auch nur.

Vielleicht ist es wichtig, die Kinder daran zu erinnern, dass auf der Aufgabekarte MZ2 noch weitere Teilaufgabe notiert sind, die ggf. durch den Tab mit der Scratch-Animation überdeckt sind. In Teilaufgabe 2 sollen die Kinder eine konkrete Aufgabe am 60er-Feld (passend bereithalten) einzeichnen, lösen, und ihr Vorgehen besprechen. Hier bietet es sich im Unterricht an, über die Strategien zu reflektieren (bspw. gegenseitiges Abstreichen, „sehen“ der Mittelzahl, Rechnen, ...). Ein Impuls, die Mittelzahl zu berechnen, wird in Teilaufgabe 3 gegeben, aber es wird nicht erwartet, dass alle Kinder diesen Impuls aufgreifen. Er dient zur Anregung der Zone der nächsten Entwicklung.

MZ3 In dieser Aufgabe erfolgt die Einführung in das Finden der Mitte zwischen zwei Zahlen durch Rechnen. Auch sind alternative Unterrichtsszenarien denkbar. In Teilaufgabe 1 wird dies durch eine Scratch-Animation angeregt, dabei werden beide Fälle (gerade und ungerade Differenz) unterschieden. In Teilaufgabe 2 wird das erworbene Wissen angewendet. Im Unterricht kann das Berechnen der Mittelzahl reflektiert werden. Dazu kann die Lehrkraft die Scratch-Animation MZ2 per Beamer/Digitaltafel frontal benutzen. Es ist ein Easteregg eingebaut: Wenn zwei Zahlen eingegeben sind und die Lösung angezeigt wird, dann kann durch dreimaliges Tappen auf „Lösung“ der Differenz-Rechenweg und durch erneutes Tappen der Additionsrechenweg (für MZ4) angezeigt und wieder ausgeblendet werden. Wichtig: In der Scratch-Animation wird die Differenz durch Ergänzen ermittelt, weil damit der Rechenweg anschaulicher ist. Gerechnet werden darf natürlich auch Minus, das ist bei komplexeren Zahlen einfacher. Gerade bei großen Differenzen haben die Kinder sich in der Erprobung bei ergänzendem Rechnen oft verrechnet.

MZ4 Diese freiwillige Aufgabe dient zur Differenzierung als Zusatzangebot. Hier wird eine zweite Berechnungsart der Mitte zwischen zwei Zahlen vorgestellt, indem diese addiert und durch 2 geteilt

werden. Dieser Rechenweg ist einfacher zu rechnen, aber schwerer zu verstehen, warum man so rechnen darf.

MZ5 In Teilaufgabe 1 wird das Spiel *Zahlen raten* an einem Beispiel erklärt. Dann sollen die Kinder in Teilaufgabe 2 selbst *Zahlen raten* spielen. Vorab kopierte 60er-Feldern können dabei helfen, sowohl beim Spielen als auch zur Protokollierung der Lösungsversuche und der Lösung (Zahlen einkreisen). Zudem sollen die Kinder die Anzahlen ihrer Rateversuche auf der Aufgabenkarte notieren. Diese Anzahlen können auch nachträglich ermittelt werden, wenn die Kinder die 60er-Feld-Protokolle benutzen. Dieses Anzahlen-Protokoll soll später mit den Rateversuch-Anzahlen von Dee verglichen werden.

MZ6 Mit dem Aufgabenblatt MZ6 wird die konzeptuelle Repräsentationsebene durch die von einem Kind erstellten Programm-Ablauf-Pläne angeregt. Zu den Kompetenzbereichen algorithmischen Denkens gehört es auch, Programmabläufe analysieren und einer realen Vorgehensweise zuordnen zu können. Das soll hier geschehen, indem argumentiert wird, wie die Ablauf-Pläne zu den Rollen im Spiel passen könnten. Aus der Erprobung wissen wir, dass die Kinder gerne die Schleifenstruktur „übersehen“. Hier wäre es wichtig, im Unterricht noch einmal Darstellung und Lesart der Schleifenblöcke im Programm-Ablauf-Plan zu besprechen.

Darüber hinaus kann besprochen werden, dass solche Programm-Ablauf-Pläne unterschiedlich detailliert sein können. Was genau „Rate eine neue kleinere mittige Zahl“ ist, wird nicht erläutert. Hier könnte man genauer beschreiben, wie dieser Teilschritt funktioniert – oder eine Art „Unterprogramm“ definieren, das eben beschreibt, wie „mittige“ Zahlen gefunden werden können. An dieser Stelle ist dies aber noch nicht notwendig, es reicht aus, dass hier die Zone der nächsten Entwicklung angeregt wird, über die Programmideen nachzudenken. Ggf. könnten die Kinder später (nach MZ8/ MZ9) versuchen, selbst den Algorithmus der Suchraumhalbierung in der Rolle des „Ratekindes mit KI“ in einem Programm-Ablauf-Plan aufzuschreiben. Vorlagen dazu können auf promaprim.de unter *Materialien* heruntergeladen werden.

MZ7 In dieser Aufgabe sollen die Kinder nun selbst ein Programm schreiben, das Dee in der Rolle des Spielleiters eine *gedachte Zahl* festlegen und auf die Rateversuche der Kinder passende Antworten geben lässt. Dazu finden die Kinder in Scratch eine halb-vorgefertigte Lernumgebung vor, die nur noch im algorithmischen Kern, der „wiederhole bis ...“-Schleife, fertiggestellt werden muss. Alle notwendigen Elemente liegen bereit. Dazu sollen sie Arbeitsblatt 6 zur Rate ziehen, und die Strukturelemente vergleichen. Dann wird das Programm getestet und mindesten fünfmal gegen Dee gespielt. Dies dient zum einen dazu, die korrekte Funktionsweise des Programms zu überprüfen, zum anderen aber auch dazu, während des Spiels schon strategische Vorgehensweisen zu überlegen/ anzuwenden.

MZ8 Wenn die Kinder MZ8 in Scratch öffnen, spielt Dee nun in der Rolle des Ratekindes gegen sie. Die ikonische Darstellung der Rateversuche ist dabei analog zu den Darstellungen „die Mitte finden“ in MZ2, so dass hier bereits durch die Darstellung die Verbindung von Ratestrategie und Suchraumhalbierung angeregt wird. Zunächst sollen die Kinder fünfmal gegen Dee spielen, um ein Gespür für das Programm zu bekommen. Aus der Erprobung wissen wir, dass die Kinder Dees Behauptung, er werde maximal 6 Versuche brauchen, zunächst ignorieren. Fokussieren Sie darauf, und lassen Sie die Kinder dies überprüfen. Ein Vergleich mit den eigenen Rateversuchsanzahlen aus MZ5 wird vermutlich aufzeigen, dass Dee deutlich schneller die Zahl rät, denn in der Regel entwickeln die Kinder die Strategie der Suchraumhalbierung nicht von selbst.

In Teilaufgabe 2 sollen die Kinder einen weiteren Durchgang spielen, aber dabei ganz langsam Schritt für Schritt die Änderung der Zahlen in der Zahldarstellung verbunden mit der Darstellung am 60er-Feld beobachten und dabei versuchen, Dees Trick zu verstehen. Dieser Vorgang kann auch gut Gegenstand einer gemeinsamen Reflexion im Unterricht werden. So wäre es sinnvoll, noch vor Teilaufgabe 2 das

Berechnen der Mitte zwischen zwei Zahlen zu wiederholen (siehe MZ3). In einem gemeinsamen langsamen Durchgang können Sie die Kinder immer zuerst die mittlere Zahl ausrechnen lassen, bevor die Antwort gegeben wird – wird Dee wieder die mittlere Zahl als neue *Ratezahl* wählen? Aus der Erprobung wissen wir ebenfalls, dass Kinder hier auch im 4. Schuljahr noch Schwierigkeiten haben, die Rechenschritte im Kopf auszuführen. Regen Sie hier ggf. geeignete Stützmaßnahmen (halbschriftliches Rechnen, Taschenrechner-App) an.

Wie bereits bei Aufgabe MZ5 angeregt, könnte ein eigenes Erstellen eines Programm-Ablauf-Plans (Vorlagen auf promaprim.de unter *Materialien*) zu diesem Algorithmus als differenzierende Zusatzaufgabe möglich sein.

- MZ9** Das in MZ8 noch angewendete Programm ist hier nun in MZ9 Gegenstand einer Programmieraktivität. Wieder finden die Kinder in Scratch eine halb-vorgefertigte Lernumgebung vor, die ebenfalls nur noch im algorithmischen Kern, der „wiederhole bis ...“-Schleife, fertiggestellt werden muss. Allerdings ist die Schleife nun komplexer. Neben den richtigen Rechenschritten müssen auch die beiden Bedingungen richtig gesetzt werden, wenn passend zu den Antworten der Suchraum oben oder unten abgeschnitten wird, und die genutzte *Ratezahl* dann die neue untere oder obere Zahl wird. Beim Fertigstellen des Programms werden sich die Kinder daher intensiv mit den einzelnen Schritten des Algorithmus der Suchraumhalbierung beschäftigen. Aus der Erprobung wissen wir, dass manche Kinder nur oberflächlich lesen, was in den bereitliegenden Blöcken steht, und beispielsweise die Operatoren (wie + und -) übersehen. Regen Sie ein genaues/ langsames Lesen an, oder besprechen Sie die bereitliegenden Blöcke mit den Kindern gemeinsam. Eine weitere Hürde stelle die Blockgruppe „abrunden von (untere Z. + halbe Diff) – erinnern Sie hier daran, dass die Mitte zwischen der oberen und der unteren Zahl auch zwischen zwei Zahlen liegen kann, und beim Rechnen dann eine Zahl mit „Komma 5“ herauskommt. Besprechen Sie, was hier passiert – wählt Dee dann die untere oder die obere der beiden möglichen Zahlen?
- MZ10** Die Aufgabe MZ10 (freiwillige Aufgabe) dient zur Differenzierung als Zusatzangebot – hier wird das nahezu gleiche Programm aus MZ9 mit dem additiven Rechenweg aus MZ4 fertiggestellt. Diese Aufgabe ist also nur dann sinnvoll, wenn sich die Kinder zuvor auch mit der freiwilligen Aufgabe MZ4 beschäftigt haben.
- MZ11** Mit den 4 Teilaufgaben in MZ11 beginnt die reflektive Betrachtung der Auseinandersetzung mit dem *Zahlen raten* Algorithmus der Suchraumhalbierung. Als Transfer wird nun eine Zahl zwischen 1 und eine Million gewählt. Die Kinder sollen zunächst selbst schätzen, wie viele Rateschritte man nun brauchen wird. Sie werden deutlich höhere Zahlen angeben als jene 20 Schritte, die Dee maximal brauchen wird. Dass selbst bis eine Million nur 20 Schritte notwendig sind, ist für Kinder erstaunlich. Es reicht aus, dieses Phänomen wahrzunehmen – zur differenzierenden Auseinandersetzung damit können sich die Kinder in MZ12 noch unterschiedlich tief mit der mathematischen Struktur dahinter auseinandersetzen. Die Teilaufgabe 4 (Diskussion über KI, sinnvoll auch im Plenum) stellt eine Art Meta-Reflexionsfrage dar. Ist Dee (bzw. das Programm dahinter) wirklich „intelligent“ oder handelt es sich hier nur um ein Problem, dass sich mit einem Algorithmus gut und sicher lösen lässt? Tatsächlich stecken auch hinter „intelligenten“ Suchen bei Suchmaschinen oder „intelligenten“ Chat-Bots wie ChatGPT solche (allerdings dann wesentlich kompliziertere) Algorithmen.
- MZ12** Freiwillig können die Kinder in MZ12 noch das Muster der mathematischen Struktur hinter dem Algorithmus der Suchraumhalbierung vertiefen. Hier wird dann auch ersichtlich, warum nur 6 Schritte bis 60, 10 Schritte bis 1000, und so weiter, bis 20 Schritte beim Raten bis zu einer Million notwendig sind. Alle Kinder sollten versuchen, die Tabelle bis 4096 (12 Versuche) auszufüllen, das geht noch gut mit Kopfrechnen. Die höheren Zahlen könnte ein Team aus Freiwilligen mit einer Taschenrechner-App ausrechnen. In der Reflexion dieser Aufgabe sind 2 Dinge wichtig: Zum einen sind es nun Suchräume

zwischen 0 und X – damit sind 1 und X auch mögliche *gedachte Zahlen*, während vorher „zwischen 1 und 60“ ja nur Zahlen von 2 bis 59 zuließ. Diese Variante ist etwas abstrakter, ermöglicht aber erst die Zuordnung zu den Zweierpotenzen. Zum anderen müssen Sie besprechen, dass bspw. „Ratezahl zwischen 0 und 64“ bedeutet, dass auch 60 (analog 1000/ 4000) im gleichen Bereich liegt.

Ratezahl zwischen 0 und ...	Maximal benötigte Rateversuche:
2	1
4	2
8	3
16	4
32	5
64	6
128	7
256	8
512	9
1.024	10
2.048	11
4.096	12
8.192	13
16.384	14
32.768	15
65.536	16
131.072	17
262.144	18
524.288	19
1.048.576	20

Damit ist die Arbeit am Moduls MZ – Zahlen raten – abgeschlossen. Es bietet sich an, noch einen gemeinsame Abschlussreflexion am Ende durchzuführen.

HINWEISE ZUM COPYRIGHT

Einige Materialien enthalten Screenshots von Scratch/CC-BY-SA-4.0 Creative Commons. Scratch ist ein Projekt der Scratch Foundation und der Lifelong Kindergarten Group am MIT Media Lab. Es steht kostenlos unter <https://scratch.mit.edu> zur Verfügung.

Urheber dieses Kurses ist das Projekt ProMaPrim, vertreten durch Dr. Ulrich Schwätzer, Universität Duisburg-Essen, Didaktik der Mathematik, Thea-Leymann-Straße 9, D-45127 Essen. Die Urheberrechte an diesem Programmierkurs stehen unter der Creative Commons Lizenz CC BY-NC-SA 4.0 (Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 international). Sie dürfen:

- Teilen — das Material in jedwedem Format oder Medium vervielfältigen und weiterverbreiten
- Bearbeiten — das Material remixen, verändern und darauf aufbauen

Unter folgenden Bedingungen:

- Namensnennung — Sie müssen angemessene Urheber- und Rechteangaben machen, einen Link zur Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden. Diese Angaben dürfen in jeder angemessenen Art und Weise gemacht werden, allerdings nicht so, dass der Eindruck entsteht, der Lizenzgeber unterstütze gerade Sie oder Ihre Nutzung besonders.
- Nicht kommerziell — Sie dürfen das Material nicht für kommerzielle Zwecke nutzen.
- Weitergabe unter gleichen Bedingungen — Wenn Sie das Material remixen, verändern oder anderweitig direkt darauf aufbauen, dürfen Sie Ihre Beiträge nur unter derselben Lizenz wie das Original verbreiten.
- Keine weiteren Einschränkungen — Sie dürfen keine zusätzlichen Klauseln oder technische Verfahren einsetzen, die anderen rechtlich irgendetwas untersagen, was die Lizenz erlaubt.

Das ProMaPrim Icon wurde erstellt unter Verwendung des Icons pc.png von Freepik auf www.flaticon.com (freie Lizenz bei Namensnennung).