

Infoblatt

Diese Einheit ist auf ca. drei Zeitstunden ausgerichtet, kann aber je nach Fortschritt der Schülerinnen und Schüler (SuS) auch kürzer unterrichtet werden.

Je nach Ausstattung der Schule haben Sie in Klassenstufe 6 die Wahl zwischen den graphischen Programmiersprachen *Scratch* oder *Pocket Code*.

Das Mkid-Programmiermodul in Klassenstufe 7 kann jedoch nur mit Scratch umgesetzt werden. Dies ist auch deshalb sinnvoll, weil in Klasse 7 im Rahmen des Informatikunterrichts ebenfalls Scratch verwendet wird.

Scratch läuft direkt online im Browser oder kann auf Computern und Tablets offline genutzt werden. Letzteres empfiehlt sich für den Einsatz in der Schule, damit man nicht von Verfügbarkeit und Bandbreite des Internetzugangs der Schule abhängt. Pocket Code gibt es zurzeit nur für Android-Geräte. Es soll aber auch für Apple entwickelt werden. Voraussetzung zum Einsatz von Pocket Code ist also, dass zumindest zwei SuS gemeinsam ein Android-Smartphone oder Android-Tablet zur Verfügung haben. Man findet das Programm im Google Play Store.

Bevor Sie mit den SuS die Arbeitsblätter (AB) bearbeiten, sollten Sie die Lösungen zunächst selbst einmal programmiert haben. Diese Selbsterfahrung der Lehrkraft ist wichtig.

Scratch ist weitgehend intuitiv bedienbar. Wir verzichten hier auf eine Einführung, diese ist ausführlich auf den Arbeitsblättern zu finden.

Bei Pocket Code muss man den SuS unbedingt eine kleine Einführung zur Bedienung geben. Denn man kann sich ziemlich schnell in den vielen Bildschirmen verirren und die Menüpunkte sind nicht immer einleuchtend. Dafür hat die Verwendung des eigenen Smartphones und die Nutzbarkeit unterwegs einen gewissen Anreiz.

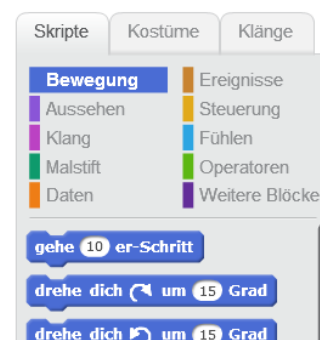
Hinweise zu Scratch:

Scratch kann online genutzt werden https://scratch.mit.edu/projects/editor/?tip_bar=home oder kostenlos auf Geräten mit den Betriebssystemen Windows, macOS, Android und iOS installieren werden. Den dazu benötigten Offline-Editor finden Sie auf der Scratch Homepage <https://scratch.mit.edu> unter Support beziehungsweise im App Store bzw. Play Store.

Auf der Scratch-Homepage stehen neben der aktuellen Version auch ältere Versionen von Scratch zum Download zur Verfügung. Falls an Ihrer Schule noch eine ältere Version in Gebrauch ist, empfiehlt es sich, die Version der Schule zu installieren. Der Online-Editor nutzt ausschließlich die aktuelle Scratch-Version.

Um mit Scratch zu programmieren, werden einfach die gewünschten Programmblöcke in den Skriptbereich gezogen. Die Programmblöcke sind in verschiedene Bereiche wie Bewegung, Aussehen, Steuerung, Daten, Fühlen etc. gruppiert. Die Zuordnung ist nicht in allen Fällen intuitiv. Machen Sie sich daher mit den verschiedenen Bereichen vertraut.

Um einen Programmblock zu entfernen, schiebt man ihn einfach wieder zurück in den Block-Vorrat.



Eine kurze Einführung in die Benutzung von Scratch findet sich auf dem Arbeitsblatt 0 in zwei verschiedenen Versionen, einmal für Scratch 2.x und einmal für Scratch 3.x.

Verschaffen Sie sich auf jeden Fall einen ausführlichen Überblick über Scratch, bevor Sie es zum ersten Mal einsetzen. Klicken Sie dabei durch alle Bereiche und beachten Sie, dass die Zuordnung einzelner Blöcke zu den Bereichen ist nicht immer intuitiv einleuchtet. **Man findet z.B. "Sage" unter "Aussehen" und "Frage" bei "Fühlen".** Insbesondere sollten Sie sich **mit "Daten" vertraut machen**. Mit der Erzeugung einer Variablen entstehen auch neue Programmierblöcke wie "Setze ... auf" und "Ändere ... um".

In Scratch können allen Figuren und Bühnen (so heißen die Hintergründe) Programm-Skripte zugeordnet werden. Damit man die Skripte, die z.B. zur Figur Katze gehören, sehen und verändern kann, muss man vorher im Bereich Figuren auf die Katze klicken. Sind die gesuchten Skripte nicht zu sehen, ist wahrscheinlich gerade eine andere Figur oder eine Bühne aktiviert.

Hinweise zu Pocket Code:

Hier ist es nicht intuitiv, ein neues Programm zu beginnen. Man klickt zuerst auf „Neu“, vergibt einen Namen für das Programm und wählt die Ausrichtung. Dann klickt man auf „Hintergrund“ und wählt „Skripte“ und danach auf „+“, nun erst bekommt man die Bereiche aus denen Anweisungen gewählt werden können. Hat man dann ein Programm zusammengestellt und lässt es laufen (Pfeil am unteren Rand), dann bekommt man am Ende unter Umständen einen leeren Bildschirm und ist etwas hilflos. Hier hilft dann nur die Zurücktaste des Smartphones. Am unteren Rand erscheint dann eine Bedienleiste, bei der man auf „Zurück“ klicken kann.

Zum Löschen einer Anweisung klickt man auf die Anweisung und wählt dann „Baustein löschen“ (hier findet man auch „Baustein verschieben“).

In der Hilfe findet man eine YouTube-Einführung in die vielfältigen Möglichkeiten, die wir nur zum kleinen Teil nutzen.


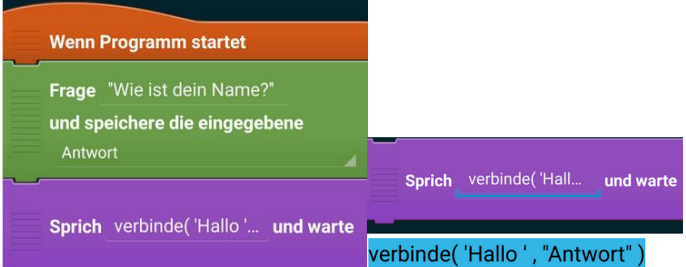

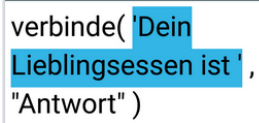
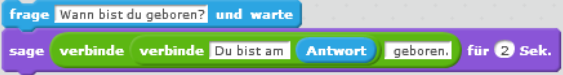
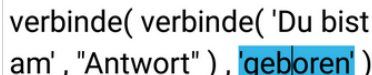

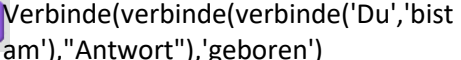
Tipps für den Unterricht:

Besprechen und üben Sie bei erster Gelegenheit das Speichern eigener Scratch-Projekte an einer Stelle, an der die SuS das Programm auch Wochen später wiederfinden.


Sie können die Lösungen zu den einzelnen Aufgaben an der Tafel aushängen, sodass die SuS bei kleinen Problemen oder Fragen selbständig nachschauen können. Das kann Sie entlasten, weil so weniger Fragen auf einmal von Ihnen beantwortet werden müssen. Sinnvollerweise geben Sie vor, dass die Lösungen von den SuS nicht mit an den eigenen Platz genommen werden dürfen.



Wie schnell die SuS vorankommen, hängt auch von den Vorkenntnissen ab und wird bei den SuS variieren. Daher sind am Ende der Einheit zusätzliche Aufgaben für die schnelleren SuS vorgesehen.

Lösungen der Aufgaben:

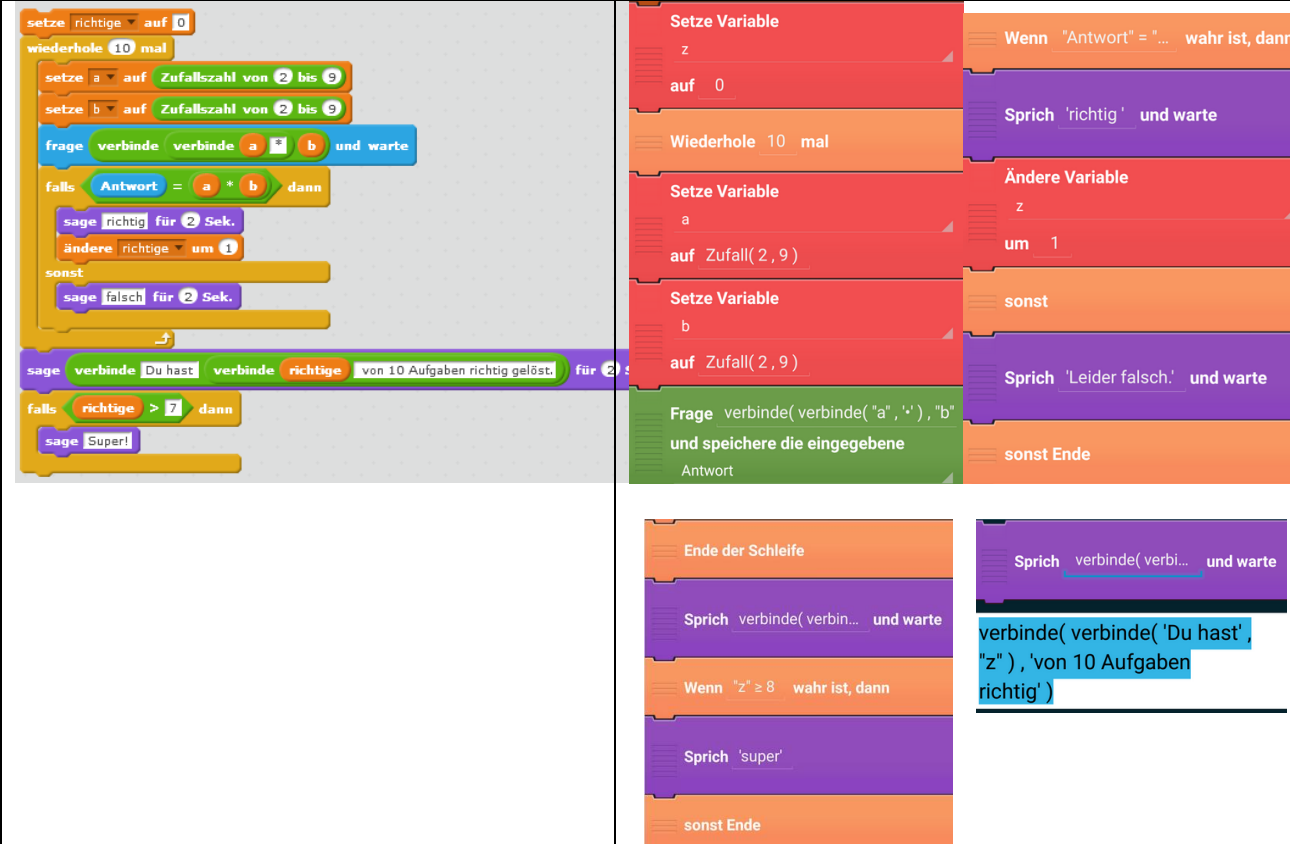
Scratch	Pocket Code
AB1 Aufg.1	
	
Die Antwort kommt jetzt nicht in zwei Schritten.	
AB1 Aufg. 2	
	
Auf das Leerzeichen nach „ist“ achten.	
AB1 Aufg. 3	
	
Auf Leerzeichen nach „am“ und vor „geboren“ achten.	
AB1 Aufg. 4	
Noch gibt es kein Beispiel, wo dies notwendig wäre. Hier können die SuS schon erkennen, dass <i>Antwort</i> immer nur das Ergebnis der letzten Eingabe enthält. Es kann also das Bedürfnis entstehen, einen Wert zu speichern (siehe Pocket Code 2 Arbeitsblatt – Aufgabe 3). Man kann aber trotzdem die Aufgabe einfach lösen:	
	

AB2 Aufg. 1 In der linken Version wird immer „Schade“ ausgegeben, in der rechten Version nur wenn man nicht „Ja“ sagt.	
AB2 Aufg. 2 	 <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="flex: 1;"> <p>Antwort = Ja</p> <p>Junge</p> <p>Mädchen</p> </div> </div>
AB2 Aufg. 3 	 <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="flex: 1;"> <p>= Ja</p> <p>'Guten Tag Herr', "Antwort1"</p> </div> </div>

AB2 Aufg. 4 	
Diese Aufgabe dient als Vorbereitung auf AB3 und kann natürlich schon in diesem Sinne variiert werden	
AB3 Aufg. 1 	
AB3 Aufg. 2 	

AB3 Aufg. 3 	 <p>"Antwort" = a x b</p>
AB3 Aufg. 4 Es zählt im Sekundentakt von 0 bis 4 hoch. Man könnte einen Countdown anregen.	3-Sekundentakt

AB3 Aufg. 5



Ein Fehler in Pocket Code: Das "Warte" funktioniert in der Sprich-Anweisung nicht so richtig.

Deshalb ist es günstiger, die Anweisung in zwei Teile aufzuspalten:



Didaktischer Kommentar

Ziele:

Die Stunden haben offensichtliche und verdeckte Ziele.

Einerseits sollen die Schülerinnen und Schüler (SuS) Freude am Programmieren entwickeln und im besten Fall anschließend eigene Programme entwickeln. Dabei lernen sie die wesentlichen Elemente eines Programms (Eingabe, Ausgabe, Variable, Verzweigung und Schleife) zu verwenden.

Andererseits müssen sie beim Testen ihres Programms „im Kopf“ nachrechnen, ob dieses richtig rechnet. Dabei üben sie das kleine Einmaleins. Ein wesentliches, übergeordnetes Ziel von Mkid ist eine Stärkung der Grundfertigkeiten, zu denen das kleine (oder auch das große) Einmaleins gehört.

Sachanalyse:

Eine **Eingabe** besteht in der Regel aus einer Aufforderung, z.B. über die Tastatur etwas einzugeben und aus einem weiteren Element (z.B. einem Button) mit dem signalisiert werden kann, dass die Eingabe beendet worden ist. D.h. das Programm „wartet“ während der Eingabe. Wenn die Eingabe sinnvoll verwendet werden soll, dann muss das Eingegebene in einer Variablen gespeichert werden. Bei Scratch gibt es die vorgegebene Variable *Antwort*, bei Pocket Code muss man eine Variable dafür benennen.

Variablen sind beim Programmieren Speicher, in denen etwas aufbewahrt und wieder abgerufen werden kann. Sie unterscheiden sich also inhaltlich vom Variablenbegriff in der Mathematik.

Eine **Ausgabe** kann auf den Bildschirm (Scratch) oder auch akustisch (Pocket Code) erfolgen. Gibt es zwei Ausgaben hintereinander, dann muss sichergestellt sein, dass die zweite Ausgabe die erste nicht überschreibt. Dazu kann z.B. das Warten für eine bestimmte Zeitdauer dienen.

Die **Verzweigung** besteht aus einer Bedingung und ein bis zwei Blöcken. Der erste Block wird ausgeführt, wenn die Bedingung wahr ist, der zweite, wenn sie falsch ist. Danach geht das Programm weiter. Die Bedingung spielt dabei keine weitere Rolle mehr.

Die **Schleife** um einen Anweisungsblock sorgt dafür, dass der Block wiederholt wird. Dabei kann er fortlaufend bis zum Eintritt einer formulierten Abbruchbedingung oder n-mal ausgeführt werden.

Didaktische Reduktion:

Für SuS ist der Begriff einer Programmvariablen schwierig.

Sie dürfen sich bei den Programmvariablen eine Sammlung von Papieren unterschiedlicher Farbe vorstellen, auf denen ein „Variablen-Verwalter“ nach Anweisung etwas (z.B. eine Zahl) notiert.

Die „Variablen-Verwaltungskraft“ kann bei Nennung der Farbe auch Auskunft darüber geben, was auf diesem Papier im Moment steht.

Als Kombination dieser beiden Aktionen kann sie auch einen Variablenwert *aktualisieren*, das bedeutet, den bisherigen Wert nach Anweisung zu verändern.

Dazu muss sie den bisherigen Wert zunächst lesen, die Veränderung durchführen, den bisherigen Wert streichen und schließlich den neuen Wert aufschreiben.

Man kann das mit einer Schülerin oder einem Schüler als Verwaltungskraft und einigen verschiedenfarbigen Papieren durchspielen:

„Notiere die Zahl 5 auf dem gelben Papier.“ | „Lies vor, welche Zahl auf dem blauen Papier steht.“ |

„Erhöhe die Zahl, die auf dem grünen Papier steht, um 3.“ usw.

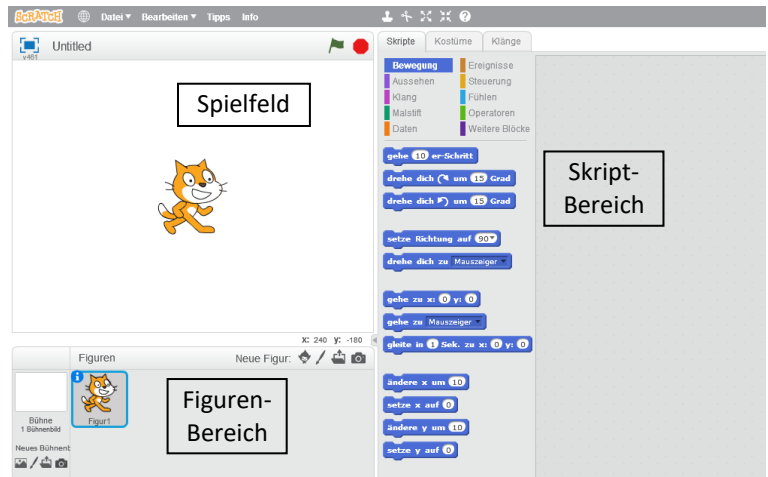
Informationen zu Scratch 2.0

Mit Scratch 2.0 programmieren

Der Bildschirm ist bei Scratch in drei Teile aufgeteilt. Rechts befindet sich der **Skript-Bereich**, in dem die Programme (auch Skripte genannt) erstellt werden.


Links ist oben das **Spiefeld**, in dem später die Programme ablaufen. Du kannst die Programme mit der grünen Fahne starten und mit dem roten Stopp-Zeichen anhalten.

Links unten ist der **Figuren-Bereich**. Dort kannst du beim Programmieren verschiedenen Figuren und Bühnen auswählen.



Um ein Programm zu schreiben, zieht man die farbigen Blöcke aus dem Vorrat nach rechts in den Skript-Bereich.

Probiere das kurz aus: Benutze aus dem Bereich **Ereignisse** den „Wenn grüne Fahne angeklickt“-Block und aus dem Bereich **Bewegung** den „gehe 10-er Schritt“-Block, um ein Programm zu schreiben, bei dem die Katze ein Stück nach rechts geht. Ergänze anschließend aus dem Bereich **Aussehen** den „sage Hello! Für 2 Sek.“-Block.

Achte darauf, dass du alle Blöcke aneinandergehängt hast – ähnlich wie bei Puzzleteilen. Um das fertige Programm zu starten, klicke auf die grüne Fahne  oberhalb des Spielfelds.

Nicht mehr benötigte Blöcke löschst du einfach, indem du sie wieder zurück in den Vorrat schiebst.



Informationen zu Scratch 3.0

Mit Scratch 3.0 programmieren

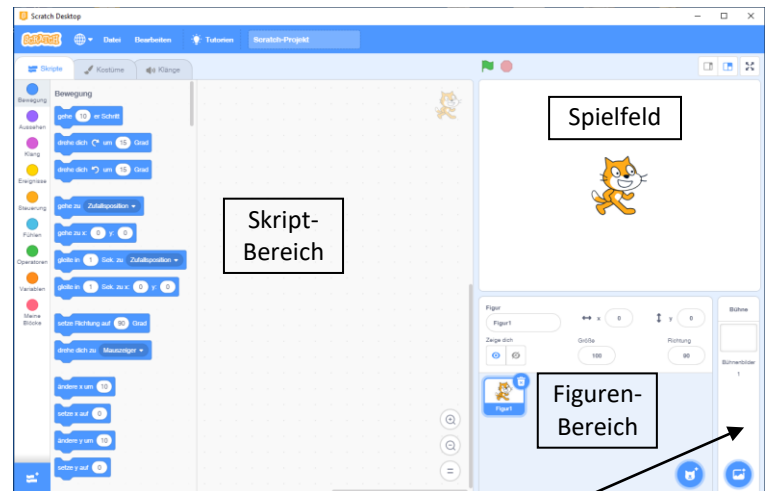
Der Bildschirm ist bei Scratch 3.0 in vier Teile aufgeteilt. Links befindet sich der **Skript-Bereich**, in dem die Programme (auch Skripte genannt) erstellt werden.

Rechts ist oben das **Spielfeld**, in dem später die Programme ablaufen. Du kannst die Programme mit der grünen Fahne starten und mit dem roten Stopp-Symbol anhalten.



Rechts unten ist der **Figuren-Bereich**.

Dort kannst du beim Programmieren verschiedenen Figuren für das

Programmieren auswählen. Ganz unten rechts ist neben dem Figuren-Bereich der **Bühnen-Bereich**. Hier können verschiedene Hintergründe eingestellt werden.




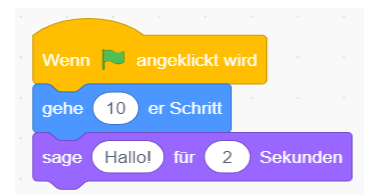
Um ein Programm zu schreiben, zieht man die farbigen Blöcke aus dem Vorrat nach rechts in den Skript-Bereich.


Probiere das kurz aus: Benutze aus dem Bereich  den „Wenn grüne Fahne angeklickt“-Block und aus dem Bereich  den „gehe 10-er Schritt“-Block, um ein Programm zu schreiben, bei dem die

Katze ein Stück nach rechts geht. Du musst dabei die Blöcke

aneinanderhängen – wie bei Puzzleteilen. Ergänze anschließend aus dem

Bereich  den „sage Hello! Für 2 Sek.“-Block.




Um das Programm zu starten, klicke auf die grüne Fahne  oberhalb des Spielfelds.

Um nicht mehr benötigte Blöcke zu löschen, kannst du sie einfach wieder in den Vorrat schieben.

Programmieren mit Scratch (1)

Ziehe als erstes aus dem Bereich „Ereignisse“ den Block  in den Skript-Bereich.

Wähle nun in der Mitte den Bereich "Fühlen" aus und ziehe  unter den vorherigen Block.


Schreibe statt "What's your name?" "Wie heißt du?".

Klicke dann mit der Maus darauf.


Merke dir: Eine Anweisung führt man aus, indem man auf sie klickt.

Gib dann deinen Namen ein und klicke auf den Haken.



Ziehe nun aus dem Bereich "Aussehen"  unter die Frage und ersetze "Hello" durch "Hallo".




Merke dir: Elemente kann man wie Puzzle-Teile aneinander setzen. Klickt man auf ein Element, so werden die Puzzleteile nacheinander ausgeführt. Man nennt das ein Programm.

Wozu gibt man eigentlich seinen Namen ein? Im Bereich "Fühlen" gibt es eine Element , das man statt eines Textes einfügen kann. Das würde dann z.B. so aussehen:



Merke dir: Runde Elemente kann man statt eines Textes einsetzen.

Aufgabe 1

Ziehe  vom Programm weg. Im Bereich Operatoren findest du das Element . Setze dieses für "Hallo" in deinem Programm ein. Ziehe nun  in das "world"-Feld. Wie hat sich dein Programm verändert?

Aufgabe 2

Nun sollst du ein Programm schreiben, das nach deinem Lieblingsessen fragt und dann sagt: "Dein Lieblingsessen ist ...", wobei ... durch die Antwort ersetzt werden soll.

Aufgabe 3

Jetzt wird es schwierig: Das Programm soll dich nach deinem Geburtstag fragen und dann ausgeben "Du bist am ... geboren." Leider gibt es hinter "world" kein weiteres Feld. Tipp:



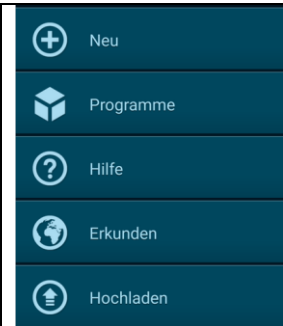
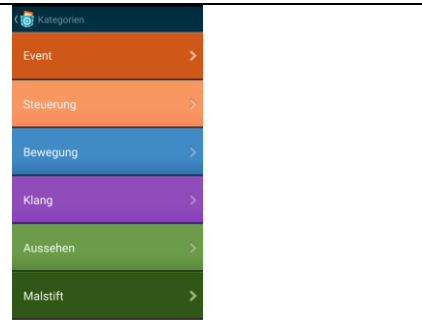
Aufgabe 4

Denke dir ein Programm mit drei ineinander geschachtelten "verbinde" aus.

Programmieren mit Pocket Code (1)

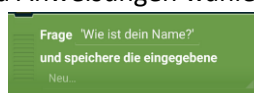
Installiere auf deinem Android Smartphone oder Tablet die App Pocket Code.

Du kannst mit dieser Seite eigene Programme schreiben. Der Bildschirm sieht zum Beginn so aus.

	<p>Klicke auf "Neu" und gib dem Programm einen Namen, z.B. <i>Test</i>. Wähle dann die Ausrichtung für die Anzeige. Klicke dann auf Hintergrund und schließlich auf Skripte. Nun musst du auf das „Plus“ klicken. Der Bildschirm sieht dann wie rechts aus.</p>	
---	---	---

Hier kannst du Bereiche auswählen, aus denen du Anweisungen wählen kannst.

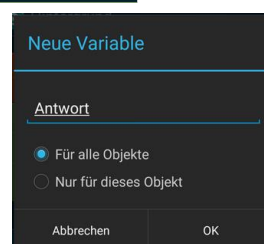
Wähle nun den Bereich "Aussehen" aus und



und dort .

Schreibe als Frage "Wie heißt du?" und statt neu Antwort.

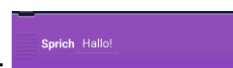
Dein Programm sieht nun so aus:



Klicke dann auf 

Merke dir: Ein Programm führt man aus, indem man auf  klickt. Zurück kommst du über den Zurück-Knopf auf deinem Handy.

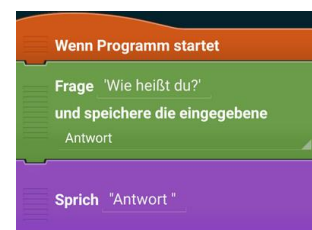
Addiere nun mit „+“ eine weitere Anweisung aus dem Bereich Klang:



und ziehe sie unter die Frage.

Merke dir: Elemente kann man wie Puzzle-Teile aneinander setzen. Die Puzzleteile werden nacheinander ausgeführt. Man nennt das ein Programm.

Wozu gibt man eigentlich seinen Namen ein? Er wurde in der Variablen Antwort gespeichert. Du kannst die Variable in das "Sagen" einbauen. Dazu musst du auf „Hallo“ klicken und dann im Eingabefeld „Daten“ „Antwort“ auswählen.



Merke dir: Variablen kann man über die Eingabe "Daten" einfügen

Aufgabe 1

Klicke auf "Antwort". Im Bereich Funktionen findest du `verbinde('Hallo','Welt')`. Setze dieses für "Antwort" in deinem Programm ein. Setze nun "Antwort" in das "Welt"-Feld. Wie hat sich dein Programm verändert?

Aufgabe 2

Nun sollst du ein Programm schreiben, das nach deinem Lieblingsessen fragt und dann sagt: "Dein Lieblingsessen ist ...", wobei ... durch die Antwort ersetzt werden soll.

Aufgabe 3

Jetzt wird es schwierig: Das Programm soll dich nach deinem Geburtstag fragen und dann ausgeben "Du bist am ... geboren." Leider gibt es hinter "Welt" kein weiteres Feld. Tipp: Man kann "verbinde" ineinander schachteln.

Aufgabe 4

Denke dir ein Programm mit drei ineinander geschachtelten "verbinde" aus.

Programmieren mit Scratch (2)

Öffne im Browser den Link https://scratch.mit.edu/projects/editor/?tip_bar=home

Du sollst nun ein Programm schreiben, das abhängig von der Antwort auf eine Frage unterschiedlich reagiert.

Dazu benötigt man die Verzweigung (Bereich Steuerung):



Damit kann man abhängig von einer Bedingung unterschiedliche Reaktionen auslösen.


Aufgabe 1: Überlege zuerst, was die folgenden Programme machen, klicke sie dann zusammen und überprüfe ob deine Vorhersage stimmt. Was unterscheidet die beiden Varianten?



Aufgabe 2: Schreibe ein Programm, das fragt, ob man männliches Geschlecht hat. Anschließend soll ausgeben: "Ah, du bist ein Junge!" bzw. "Ah, du bist ein Mädchen".

Aufgabe 3: Nun sollst du das Programm so ändern, dass zuerst nach dem Nachnamen gefragt wird und dann nach dem Geschlecht. Abhängig vom Geschlecht soll dann ausgegeben werden: "Guten Tag Herr (Nachname)" bzw. "Guten Tag Frau (Nachname)".

Hier tritt ein Problem auf, das man lösen kann, indem man sich die Antwort merkt:



Scratch interface showing variable management for 'nachname'. It includes blocks for 'setze nachname auf 0', 'ändere nachname um 1', 'zeige Variable nachname', and 'verstecke Variable nachname'.

Klicke im Bereich Daten auf Neue Variable und gib dann nachname ein. Es erscheinen nun neue Elemente

Das runde Element nachname kannst du wie Antwort verwenden.

Vorher muss du allerdings festlegen, was nachname enthalten soll. Du kannst z.B. sagen

setze nachname auf Antwort

Nun enthält nachname die Antwort.

So kann man beliebig viele Variablen erzeugen, die irgendwelche Informationen enthalten.

Aufgabe 4: Schreibe ein Programm, das eine Rechenaufgabe stellt. Wenn sie richtig gelöst ist, dann soll es "Super" ausgeben, sonst soll es "Leider falsch" ausgeben.

Programmieren mit Pocket Code (2)

Du sollst nun ein Programm schreiben, das abhängig von der Antwort auf eine Frage unterschiedlich reagiert.

Dazu benötigt man die Verzweigung (Bereich Steuerung):



Damit kann man abhängig von einer Bedingung unterschiedliche Reaktionen auslösen.

Aufgabe 1: Überlege zuerst, was die folgenden Programme machen, klicke sie dann zusammen und überprüfe ob deine Vorhersage stimmt. Was unterscheidet die beiden Varianten?



Aufgabe 2: Schreibe ein Programm, das fragt, ob man männliches Geschlecht hat. Anschließend soll ausgeben: "Ah, du bist ein Junge!" bzw. "Ah, du bist ein Mädchen".

Aufgabe 3: Nun sollst du das Programm so ändern, dass zuerst nach dem Nachnamen gefragt wird und dann nach dem Geschlecht. Abhängig vom Geschlecht soll dann ausgegeben werden: "Guten Tag Herr (Nachname)" bzw. "Guten Tag Frau (Nachname)".

Hier benötigt man nun zwei unterschiedliche Variablen *Antwort1* und *Antwort2*:




Klicke auf „Antwort“ und wähle dann „Neu...“. Gib der Variablen den Namen *Antwort1*. Wähle nun *Antwort1* statt *Antwort*.


So kann man beliebig viele Variablen erzeugen, die irgendwelche Informationen enthalten.

Aufgabe 4: Schreibe ein Programm, das eine Rechenaufgabe stellt. Wenn sie richtig gelöst ist, dann soll es "Super" ausgegeben, sonst soll es "Leider falsch" ausgegeben.

Programmieren mit Scratch (3)

Öffne im Browser den Link https://scratch.mit.edu/projects/editor/?tip_bar=home


Computer heißen so, weil sie besonders gut rechnen können. Bei der Multiplikationsaufgabe haben wir diese Fähigkeit gar nicht verwendet. Mit dem Operator  kann man den Computer ein Produkt ausrechnen lassen.

Überprüfe, was die Anweisung  macht.

Aufgabe 1

Schreibe ein Programm, das fragt "Was ist 4*7?" und dann das Ergebnis ausrechnet und abhängig davon, ob die Eingabe richtig oder falsch ist eine Rückmeldung gibt.

Aufgabe 2

Schön wäre es, wenn der Computer eine beliebige Multiplikationsaufgabe anzeigen könnte. Auch dafür findet man einen Operator . Diese Aufgabe ist super schwierig und es kann sein, dass du eine Hilfe benötigst. Diese Elemente benötigst du zur Lösung:



Aufgabe 3

Nun kannst du ein Programm schreiben, das eine beliebige Multiplikationsaufgabe und auf die Eingabe reagiert.

Dazu musst du die Zufallszahl speichern (Daten, Neue Variable, a bzw. b)



Mit der **Schleife** (Bereich Steuerung) kann man einen Programmteil wiederholen.



Aufgabe 4

Was macht wohl dieses Programm? (Daten, Neue Variable, a)



Aufgabe 5

Schreibe nun ein Programm, das 10 Multiplikationsaufgaben der Reihe nach stellt und zählt, wie viele Aufgaben richtig gelöst wurden. Ab 8 richtigen Aufgaben gibt es "Super" aus.

Programmieren mit Pocket Code (3)

Computer heißen so, weil sie besonders gut rechnen können. Bei der Multiplikationsaufgabe haben wir diese Fähigkeit gar nicht verwendet. Mit dem Formel-Editor kann man den Computer rechnen lassen.

7	8	9	÷
4	5	6	×
1	2	3	-
,	0	=	+
Abc	OK		

Sprich 4×7 und warte

Überprüfe, was die Anweisung macht.

Aufgabe 1

Schreibe ein Programm, das fragt "Was ist 4×7 ?" und dann das Ergebnis ausrechnet und abhängig davon, ob die Eingabe richtig oder falsch ist eine Rückmeldung gibt.

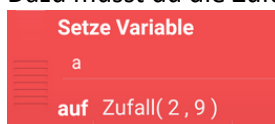
Aufgabe 2

Schön wäre es, wenn der Computer eine beliebige Multiplikationsaufgabe anzeigen könnte. Auch dafür findet man eine Funktion `Zufall(0,1)`. Diese Aufgabe ist super schwierig und es kann sein, dass du eine Hilfe benötigst. Diese Elemente benötigst du zur Lösung:
Die Anweisung Frage..., die Funktionen Verbinde (2mal) und Zufall (2,9) (2mal)

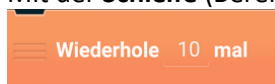
Aufgabe 3

Nun kannst du ein Programm schreiben, das eine beliebige Multiplikationsaufgabe stellt und auf die Eingabe reagiert.

Dazu musst du die Zufallszahl speichern (Daten, Setze Variable, a bzw. b)

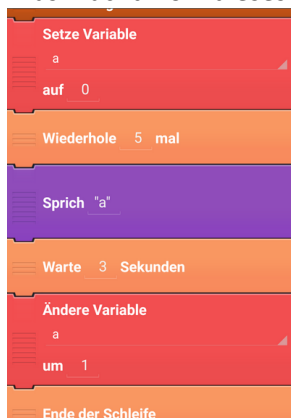


Mit der **Schleife** (Bereich Steuerung) kann man einen Programmteil wiederholen.



Aufgabe 4

Was macht wohl dieses Programm? (Daten, Neue Variable, a)



Aufgabe 5

Schreibe nun ein Programm, das 10 Multiplikationsaufgaben der Reihe nach stellt und zählt, wie viele Aufgaben richtig gelöst wurden. Ab 8 richtigen Aufgaben gibt es "Super" aus.

Verlaufsplan – alle drei Stunden

SuS ... Schülerinnen und Schüler L ... Lehrerin bzw. Lehrer

EA ... Einzelarbeit PA ... Partnerarbeit FEU ... fragendentwickelnder Unterricht

Die Zeitangaben dienen nur zur groben Orientierung! Insbesondere hängt es sehr stark vom individuellen Unterrichtsverlauf ab, ob 2 oder 3 Nachmittage benötigt werden.

Phase / Zeit	L / SuS	Medien
Erste Stunde		
1. Einstieg Demonstration 10 Min.	L erklärt, was man unter "Programmieren" versteht, und stellt die verwendete Programmierumgebung an einem Beispiel vor.	Beamer ggf. Arbeitsblatt 0
2. Erarbeitung PA 30 Min.	SuS bearbeiten das Arbeitsblatt 1. Evtl. wird die erste Aufgabe im Plenum erledigt. L gibt individuelle Hilfestellungen	Arbeitsblatt 1 Computer bzw. Smartphone (evtl. mit Kopfhörer)
3. Plenum FEU 15 Min.	Die Lösungen der Aufgaben werden vorgestellt.	Beamer
Zweite Stunde		
1. Einstieg FEU 10 Min.	Wiederholung der wesentlichen Ergebnisse der letzten Stunde.	Beamer
2. Problemstellung FEU 10 Min.	Intensive Vorbereitung der Partnerarbeit, z.B. indem man die Verzweigung an Alltagssituationen bespricht, die zugehörigen Anweisungsblöcke verdeutlicht, usw.	
3. Erarbeitung PA 25 Min.	SuS bearbeiten das Arbeitsblatt 2.	Arbeitsblatt 2 Computer bzw. ...
Dritte Stunde		
1. Einstieg FEU 10 Min.	Wiederholung der wesentlichen Ergebnisse der letzten Stunde.	Beamer
2. Erarbeitung PA 15 Min.	Arbeitsblatt 3 – Aufgabe 1 bis 3	Arbeitsblatt 3 Computer bzw. ...
3. Plenum FEU 10 Min.	Besprechung der Aufgaben	Beamer
4. Lehrervortrag 5 Min.	Die Schleife als Steuerungselement	
5. Erarbeitung PA 20 Min.	Arbeitsblatt 3 – Aufgabe 4 und 5	Computer bzw. ...
6. Plenum FEU 5 Min.	Vorstellung der fertigen Programme	Beamer
7. Ergänzungen	Mögliche Erweiterungen der Aufgabe mit Lob auf verschiedenen Stufen	Computer bzw. ...