Der Begriff Algorithmus



Du hast nun bereits erste einfache Programme in Snap! geschrieben. Nun wird es Zeit, dass du einen Fachbegriff für das, was hinter einem Programm steht, kennenlernst: *Algorithmus*.

Ein Algorithmus ist eine **Verarbeitungsvorschrift** – also ein Kommando – , die aus einer **endlichen Anzahl** von **eindeutig ausführbaren Anweisungen** besteht. Unter **gleichen Voraussetzungen** liefert ein Algorithmus dabei immer das **gleiche Ergebnis**. Ist der Algorithmus in einer für den Computer verständlichen Form geschrieben, so nennt man ihn *Programm*.

Was wir in Snap! entwickeln sind somit sowohl **Algorithmen**, als auch **Programme**.



Öffne dein Projekt

Starte Snap! und öffne das Projekt der letzten Stunde. Wir werden nun damit weiterarbeiten.

Verwendung des Stifts

Nach der letzten Stunde kannst du bereits das Aussehen deiner Figur bestimmen, sie bewegen und sie sprechen lassen. Nun werden wir eine weitere nützliche Anwendung in Snap kennenlernen: Den *Stift*.

Prinzipiell kannst du in Snap! die Bewegung jeder Figur aufzeichnen lassen. Dazu musst du lediglich vor deren Bewegung den Stift "aktivieren". Dazu findest du unter dem Überpunkt **Pen** den Baustein **pen down**.





Zeichne die Bewegung deiner Figur nach

Füge in deinem Programm vor der Bewegung deiner Figur den Baustein <mark>pen down</mark> ein. Starte nun das Programm und beobachte, was passiert.

Wie du siehst, werden nun alle Bewegungen deiner Figur aufgezeichnet. Dies hinterlässt eine ganze Reihe an Linien im Zeichenfenster. Um sie zu löschen, verwende den Baustein **clear**.

	Lösche deine Zeichnung Lösche nun die in Aufgabe 2 erzeugten Linien im Zeichenfenster wieder.
Aufgabe 3	Tinn • Du musst den Baustein clear dazu nicht unbedinat ins Programm einhinden
	Per Klick lässt er sich auch von Hand ausführen.

Weitere wichtige Bausteine zur Verwendung des Stiftes sind:

- **pen up** : deaktiviert den Stift
- set pen color to ... : ändert die Stiftfarbe
- set pen size to ... : ändert die Stiftgröße
- **stamp** : hinterlässt ein Abbild der Figur an der momentanen Position

Aufgabe "Haus zeichnen"

Im Folgenden soll ein Programm entwickelt werden, welches durch geschicktes Bewegen einer Figur ein Haus zeichnen lässt.

Zur Vereinfachung benutzen wir dazu die dreieckige Standardform, da an ihr die Drehungen am besten sichtbar sind.

pen up

set pen color to

set pen size to 1

stamp

Erinnerung: Deine Figur bewegt sich immer **geradlinig in Blickrichtung**!

200	Zeichne das Haus
	Starte Snap! und öffne ein neues Programm.
Aufgabe 4	Entwickle nun Schritt für Schritt das oben beschriebene Programm zum Zeichnen eines
8	Hauses.
	Teste dabei deinen Programmcode nach jedem Schritt.
	Exportiere anschließend dein Programm unter dem Namen Haus.xml und speichere es auf deinem USB-Stick ab.
	Tipp: Vergiss nicht, deine Figur nach jedem Test wieder in die Ausgangsposition
	zurückzusetzen, sowie deine Zeichnung zu löschen.

Wir haben gerade eine **Folge von Anweisungen** programmiert, die genau **in angegebener Reihenfolge nacheinander** ausgeführt werden sollen. Eine solche Folge von Anweisungen nennt man in der Informatik *Sequenz*.

Im Spezialfall kann sogar eine einzelne Anweisung bereits eine Sequenz sein.

	Gestalte das Haus	\wedge
	Starte Snap und importiere die Datei Haus.xml .	
Hausaufgabe	Verändere das Programm nun derart, dass deinem Haus eine Tür und ein	
8	Fenster hinzugefügt werden.	
	Exportiere dein Programm anschließend unter dem Namen Haus-erweitert.xm	nl und
	speichere es auf deinem USB-Stick ab.	





Tipps und Tricks

- Beim Entwickeln von Programmen ist es oftmals nützlich, einzelne Programmteile sofort zu testen. Damit erleichterst du dir die Fehlersuche, da du dann meist genau weißt, welcher Programmteil falsch umgesetzt wurde.
- Wenn du einzelne Bausteine, oder gar Programmteile mehrfach benötigst, kannst du diese ganz einfach kopieren. Klicke dazu mit rechts auf den entsprechenden Baustein/Programmteil und wähle *duplicate*.

