ARRAYS: EINE REIHE VON VARIABLEN



Wir lernen jetzt, wie man auf einen Schlag sehr viele Variablen speichern und auswerten kann. Das braucht man in der Informatik ziemlich häufig.

Wir schauen uns das am Beispiel eines Computer-Spiels an. In der Abbildung sehen Sie unsere Spielfigur. Diese hat natürlich ein *Inventar*, dessen Inhalt im Spiel gespeichert werden muss. Wie macht man das?

Eine erste Lösungsidee

Mit unserem bisherigen Wissen können wir es so probieren: Wenn wir vier Gegenstände speichern wollen, erstellen wir dafür vier Variablen. Also so:

(Verwenden Sie hierfür das vorbereitete Projekt 7000_Array-DatenOhneArraySpeichern-AUFGABE)



Warum ist das keine gute Idee?



Hoffentlich ist Ihnen beim Erstellen der Variablen aufgefallen, dass das alle keine gute Idee ist. Mit vier Variablen geht das ja noch. Aber was passiert, wenn unsere Spielfigur ein paar Level aufsteigt und plötzlich 100 Gegenstände tragen kann? Sollen wir dann 100 Variablen anlegen? Das muss doch auch irgendwie praktischer gehen!

Die Lösung: Arrays

Natürlich gibt es in der Programmierung ein Konzept, das genau dieses Problem löst. Das sind die **Arrays** (deutsch: Feldvariablen). Ein Array ist dabei eine Reihe von Variablen, die man "auf einen Rutsch" erstellt. Diese Variablen heißen alle fast gleich, man unterscheidet sie nur über eine Zahl in Klammern (den sogenannten *Index*). Das kann man sich so vorstellen:





Der Computer fängt übrigens bei 0 an zu zählen, nicht bei 1. Deshalb würde das erste Element des Arrays in den meisten Programmier-Sprachen *inventar*[**0**] heißen! In Scratch ist das anders, deshalb fangen wir bei 1 an zu zählen.

Array in Scratch erstellen

Weil wir die Gegenstände des Inventars also nicht mit Variablen verwalten werden, löschen wir die vier Variablen wieder (*Rechte Maustaste / Lösche die Variable…*).

Dann erstellen wir unser Array mit Variablen/Neue Liste:



Im Struktogramm stellt man die Erzeugung des Arrays so dar:



Scratch zeigt uns das Array sofort auf dem Spielfeld:



Das Array füllen

Wir können das Array ganz einfach füllen, indem wir den entsprechenden Befehl aus der Variablen-Kategorie verwenden:



Das Füllen des Arrays stellt man im Struktogramm so dar:

Array füllen		
	Zuweisung: inventar[1] = "Bogen"	

Wenn Sie das Programm starten, sehen Sie, wie das Array gefüllt wird. Wenn Sie es aber ein zweites Mal starten, werden die Gegenstände an das bereits gefüllte Inventar angehängt. Das Array wird also immer größer. Das Problem lösen wir, indem wir den Inhalt des Arrays löschen, wenn das Programm gestartet wird.



Aufgaben:

1.	Füllen Sie das Inventar mit drei weiteren Gegenständen! Verwenden Sie dazu das Projekt, mit dem Sie bisher gearbeitet haben oder nutzen Sie als Vorlage 7102_Array-AUFGABE
2.	Erstellen Sie ein weiteres Array, in dem die Fähigkeiten (= skills) unserer Spielfigur gespeichert werden können! Orientieren Sie sich hierbei am folgenden Struktogramm: skills-Array erzeugen und füllen Deklaration und Initialisierung: skills als Array = [] Zuweisung: skills[0] = "Klettern" Zuweisung: skills[1] = "Bogenschießen" Zuweisung: skills[2] = "Schwimmen" Zuweisung: skills[3] = "Nachtsicht" Zuweisung: skills[4] = "Schnelligkeit"

Einzelne Elemente im Array ändern

Bisher fügen wir immer Elemente ein, die direkt ans Ende des Arrays angehängt werden (dieser Befehl heißt in den meisten Programmiersprachen *append*). Wir können über den Index (also die Position im Array) auch direkt auf Elemente des Arrays zugreifen.

Anwendungsbeispiel:

Unsere Spielfigur findet eine bessere Waffe. Der "Bogen" im Array soll durch einen "Langbogen" ersetzt werden!



Struktogramm:



Aufgaben:

