

## Das Java Array

Unter einem Array versteht man einen Container, der eine feste Anzahl von Werten eines einzelnen Typen speichern kann. Du musst bei der Deklaration eines Arrays also sowohl den Datentyp festlegen als auch die Anzahl der Elemente, die du darin speichern möchtest. Bei der Initialisierung speicherst du dann konkrete Werte in deinem Array ab. So viel zur Theorie, aber wie lässt sich das nun in Java umsetzen? Zuerst beginnst du mit der Deklaration. Dabei legst du zuerst den Datentyp fest und gibst deinem Array einen Namen. Den Datentyp kannst du später nicht mehr ändern. Danach bestimmst du mit dem `new` Operator, wie viele Elemente du speichern möchtest. Diese Zahl wird auch `Array Length` genannt. In der Praxis wird diese Prozedur bevorzugt. Du kannst aber auch beide Schritte kombinieren. Dabei ist es egal, ob du die eckigen Klammern direkt hinter den Datentyp oder den Array Namen setzt.

Sehr gut, mit diesem Wissen kannst du jetzt direkt ein Array erstellen und das z.B. 4 Integer Werte abgespeichert werden können. Zuerst liegst den Integer als Datentyp fest, gefolgt von eckigen Klammern. Danach gibst du dem Array einen beliebigen Namen, denn du danach natürlich konstant weiterverwenden musst. Im nächsten Schritt setzt du den Array Namen mit dem `new` Operator gleich der dafür zuständig ist, ein neues Objekt zu erzeugen. Hier musst du dann wiederum den gleichen Datentyp eingeben. Zum Schluss folgen dann eckige Klammern, in denen du nun die Anzahl der Werte eingibst. In deinem Fall also vier. Das war's auch schon. Wenn du beide Schritte kombiniere möchtest, dann sieht deine Deklaration so aus.

Jetzt hast du also dein Array erstellt und kannst es mit Werten füllen. Die Einträge besitzen dabei eine bestimmte Reihenfolge, den sogenannten Index der jeweiligen Position. Bei einem Array wird dabei immer mit der Null begonnen. Wenn du dein Array mit diesen Werten füllen möchtest, kannst du jeder Indexposition direkt einen Wert zuteilen. Der ersten Position, also Index 0, weißt du die 65 zu, der zweiten die 5 und so weiter. Bei diesem Vorgehen hast du übrigens auch die Möglichkeit, einzelne Positionen innerhalb des Arrays leer zu lassen. Du kannst ein Array aber auch direkt bei der Deklaration initialisieren, dabei werden die Werte innerhalb von geschweiften Klammern aufgelistet und den Indexpositionen der Reihenfolge nach zugeteilt. Für das Beispiel sieht das dann so aus.

Hier noch ein kleiner Tipp: wenn du eine bestimmte Zahlenreihe in einem Array abspeichern möchtest, kannst du bei der Initialisierung auch eine For-Schleife nutzen. Als nächstes schauen wir uns an wie du auf die Elemente eines Arrays zugreifen kannst. Dafür verwendest du natürlich auch wieder den Index. Du gibst ganz einfach die genaue Indexposition an. Für die Elemente der zweiten und vierten Position deines Arrays, brauchst du also die Index Positionen 1 und 3. Möchtest du alle Elemente eines Arrays ausgeben, kannst du auch hierfür eine For-Schleife verwenden. Mit dieser Schleife gibst du dann alle Einträge des Arrays der Reihe nach auf dem Bildschirm aus. Wenn du jetzt noch wissen möchtest, wie du in Java auf die Werte eines Arrays mittels einer For-each-Schleife zugreifen kannst, dann schau dir unser Video dazu an. Sehr gut, fassen wir das Gelernte noch einmal in einem vollständigen Java-Code zusammen. Zuerst müssen wir das Array deklarieren und bei der Initialisierung

mit Werten füllen. Anschließend können wir auf einzelne Elemente des Arrays zugreifen. Das war doch gar nicht so schwer, oder? Nun weißt du wie du ein Array in Java problemlos selbst umsetzen kannst.