

Was ist ein Algorithmus?

Ein Algorithmus ist nicht mehr als eine Folge von Befehlen so abzuarbeiten bis wir zu unserem gewollten Ergebnis kommen.

<i>Resultat</i>	<i>Algorithmus</i>	<i>Einzelschritte</i>
Pullover	Strickmuster	eine links, eine rechts, eine fallen lassen
Kuchen	Rezept	nimm 3 Eier ...
Konzert	Partitur	Noten

Also eben ein Muster...

Nehmen wir das Rezept als Beispiel

1. Nimm 3 Eier
 2. Nimm Mehl
 3. Nimm Milch
 4. Schütte alles in eine Schüssel
 5. Vermische es ordentlich
 6. Packe alles in die Pfanne
- > Wir bekommen einen Pfannkuchen

Unsere Schritte 1-6 sind unser Algorithmus und mit diesem Algorithmus kommen wir immer zu unserem Pfannkuchen.

Welche Operationen können wir überhaupt benutzen?

Wenn wir Matheoperationen durchführen wollen:

<i>symbol</i>	<i>name</i>
*	Multiplikation
/	Division
%	Modulo
+	Addition
-	Subtraktion

Zum Vergleichen:

<i>symbol</i>	<i>name</i>
>	größer
>=	größergleich
<	kleiner
<=	kleinergleich
==	gleich
!=	ungleich

Warum ist unser Vergleich „gleich“ ein „==“ und kein „=“?

Unser normales Gleichzeichen „=“ ist in jedem Fall eine Zuweisung: `x = 5` -> x ist jetzt 5
 Deswegen müssen wir irgendwie ein anderes Schlüsselwort haben : `x == 5` -> ist x = 5 ?

Schleifen

Unser nächstes Thema sind Schleifen. Nehmen wir an wir wollen alle Zahlen von 1 bis 1000 in der Konsole ausgeben, dann wäre es immens umständlich alle Zahlen einzeln aufzuschreiben. Für solche Fälle gibt es Schleifen. Wir können einfach eine Schleife schreiben, die immer das selbe tut.

Was wäre das denn beim ausgeben aller Zahlen von 1 bis 1000?

1. Wir geben 1 aus
2. Wir erhöhen um 1
3. Wir geben 2 aus
4. Wir erhöhen um 1 ...

Kotlin - Kapitel 2

Wir können das zusammenfassen.

1. Erstelle eine Variable 1
2. Schreibe die Variable
3. Erhöhe die Variable

Die Schritte 2 und 3 sollen 1000 wiederholt werden. Wie machen wir das?

In Kotlin gibt es dafür 2 Möglichkeiten...

While-Schleife:

```
// Schritt 1
var i = 1

// Die Schleife
while (i <= 1000){
    // Schritt 2
    println(i)
    // Schritt 3
    i = i + 1
}
```

Die While-Schleife heißt wortwörtlich „Mache ... solange das gilt“

Wir haben also ein Bedingung und solange diese erfüllt wird wird die Schleife ausgeführt.

Was ist die Reihenfolge?

-> Als erstes wird die Bedingung überprüft
-> dann führt das Programm alles in den Klammern aus

-> dann springt das Programm wieder an den Kopf der Schleife (rosa Quadrat) und führt die Schritte aus

Die while Schleife funktioniert wie eine if-Bedingung, die sich immer wiederholt.

For-Schleife:

```
// Schritt 1 und 3 sind im Kopf der Schleife
for (i in 1..1000){
    // Schritt 2
    println(i)
}
```

Die For-Schleife ist eine Zählschleife. Also sie kann nur in bestimmten Zahlintervallen hochzählen. Das bedeutet sie ist perfekt „um etwas genau 10-Mal zu machen“, aber wenn es heißt „Mache ... bis wir ein Ergebnis haben“ bringt sie uns nichts.

-> Merke: Die for Schleife bringt uns nur etwas, wenn wir genau wissen wie häufig wir etwas Ausführen wollen.