# Arrays / Listen

Werte vom gleichen Typ als Liste speichern



# Liste / eine Dimension

#### char Beispiel

```
// Deklaration
char[] signs = new char[10];

// Zuweisung
signs[0] = 'a';
signs[signs.length - 1] = 'j';

// Deklaration + Direktzuweisung
char[] signs = {
    'a', 'b', 'c', 'd', 'e',
    'f', 'g', 'h', 'i', 'j'
};

// Zugriff
char firstValue = signs[0];
char lastValue = signs[signs.length - 1];
```

#### **Eine Reihe**



Notenliste, Messwerte, usw.

mit [] erkennt man Arrays



#### Länge startet bei 1

```
int size = 100;
int[] values = new int[size];
```

#### **Index** startet bei **0**

```
int firstValue = values[0]
// index: 100 - 1 = 99
int lastValue = values[size - 1];
```

# **Durch Array iterieren** (schrittweise)

## Mit for -Schleife

```
int values = new values[5];
for (int i = 0; i < values.length; i++) {</pre>
    // Zuweisung
    values[i] = Math.rand();
    // Zugriff
    System.out.println(values[i]);
```

Zugriff und Zuweisung via Index i

## Mit foreach -Schleife

```
int values = new values[5];
for (int value : values) {
    // nur Zugriff
    System.out.println(value);
```

🥑 Nur Zugriff dafür übersichtlicher



# Arrays sind Magier!

Wieso denkt Ihr?



#### Konventionell

```
int value1;
int value2;
int value3;
// immer weiter so
int value100;
```

Es müssen 100 Zeilen geschrieben werden für 100 Variablen vom gleichen Typ

### **Mit Array**

```
int[] values = new int[100];
```

Eine Zeile reicht aus!



# **Zuweisungszauber**

#### Konventionell

```
value1 = Math.rand();
value2 = Math.rand();
value3 = Math.rand();
// immer weiter so
value100 = Math.rand();
```

Es müssen 100 Zeilen geschrieben werden um 100 Variablen einen neuen Wert zuzuweisen



```
for (int i = 0; i < values.length; i++) {</pre>
   values[i] = Math.rand();
```

Drei Zeilen reichen aus! Und zwar auch für 1 Mio Werte



#### Konventionell

```
System.out.println(value1);
System.out.println(value2);
System.out.println(value3);
// immer weiter so
System.out.println(value100);
```

Es müssen 100 Zeilen geschrieben werden um 100 Variablen auszugeben

#### ★ Mit Array und foreach

```
for (int value : values) {
    System.out.println(value);
}
```

- Drei Zeilen reichen aus!
  Und zwar auch für 1 Mio Werte
- Da wir nur auf Daten zugreifen können wir mit foreach uns den index sparen



Wenn eine **manuelle Nummerierung** in Variablennamen oder Methodennamen vorkommt, sollte man an **Arrays** denken.



# Ab hier nur für Interessierte

# 👱 Feld / zwei Dimensionen - 🚨 Nicht Pflicht!

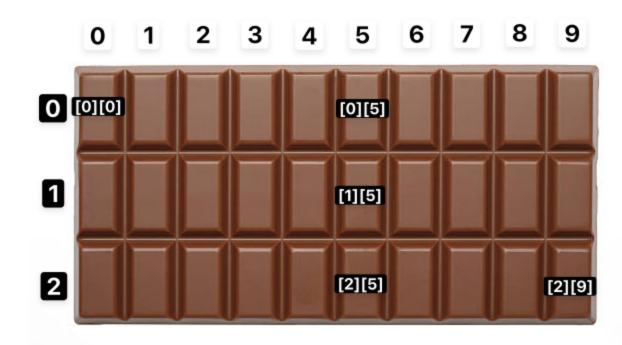
#### int Beispiel

```
// Deklaration
int[][] numbers = new int[3][10];

// Zuweisung (expliziter Index)
numbers[0][0] = 1000;
numbers[2][9] = 30000;
// oder (impliziter Index)
int index1 = numbers.length - 1;
int index2 = numbers[0].length - 1;
numbers[index1][index2] = 30000;

// Zugriff
int firstValue = numbers[0][0];
int lastValue = numbers[index1][index2]
```

#### Ein Feld mit mehreren Reihen



Schiffchen versenken, Schachbrett,Koordinatensystem