

# **Arrays in Java**

## Verzinstes Guthaben in einem Sparplan

Jahr	Guthaben in €
0	100,00
1	105,00
2	110,25
3	115,76



**Gesucht:** Variable(n) für das Guthaben

## Verzinstes Guthaben in einem Sparplan

Jahr	Guthaben in €
0	100,00
1	105,00
2	110,25
3	115,76

### Möglichkeit 1: Mehrere Variablen

- Guthaben0 = 100.0
- Guthaben1 = 105.0
- Guthaben2 = 110.25
- Guthaben3 = 115.76
- ...

**Bewertung?**

→ **Ist das praktisch?**

## Verzinstes Guthaben in einem Sparplan

Jahr	Guthaben in €
0	100,00
1	105,00
2	110,25
3	115,76

Wenn man mehrere Objekte **vom gleichen Typ** speichern und verwalten möchte, kann man **Arrays** oder **Feldvariablen** verwenden.

Deklaration:

```
double Guthaben[] = new double[10]; // 10 „Speicherplätze“
```

Verwendung:

```
Guthaben[0] = 100.0;  
Guthaben[1] = 105.0;  
Guthaben[2] = 110.25;
```

Guthaben[ ... ]

Index	0	1	2	3
Wert	100	105	110,25	115,76

The diagram illustrates an array structure. It consists of two rows of boxes. The top row is labeled 'Index' and contains four boxes with values 0, 1, 2, and 3. The bottom row is labeled 'Wert' and contains four boxes with values 100, 105, 110,25, and 115,76. Arrows point from each index box to its corresponding value box below it.

## Datentyp " Feld" oder " Array"

- Das erste Element besitzt Index 0
- Größe wird bei der Deklaration festgelegt. Nachträglich nicht veränderbar.
- Direkte Initialisierung geht auch: `int[] zahlenarray = {11,32,42,2,4};`

## Array mit Schleifen durchlaufen for-Schleife

```
// Beispiel 1
String[] name = {"Heinrich", "Margarete", "Joe"};
for(int i = 0; i < name.length; i++)
{
System.out.println("Name " + i + ": " + name[i]);
}
```



Zahl der Elemente