

# Feld (Datentyp)

Wechseln zu: [Navigation](#), [Suche](#)

Dieser Artikel erfüllt die [GlossarWiki-Qualitätsanforderungen](#) **nur teilweise**:

**Korrektheit:** 5  
(vollständig  
überprüft)

**Umfang:** 3  
(einige wichtige  
Fakten fehlen)

**Quellenangaben:**  
2  
(wichtige Quellen  
fehlen)

**Quellenarten:** 4  
(sehr gut)

**Konformität:** 4  
(sehr gut)

## Inhaltsverzeichnis

- 1 Definition
- 2 Bemerkungen
- 3 Beispiele
  - 3.1 Eindimensionales Array
  - 3.2 Zweidimensionales Array
- 4 Quellen
- 5 Siehe auch

## 1 Definition

Ein Feld oder Array ist ein spezieller [Container](#), der endlich viele [Elemente](#) zu einer durchnummerierten Folge zusammenfasst. Dabei wird dem ersten Element der Folge im Allgemeinen die Position ([null-basierte Zählweise](#)) oder die Position 1 ([eins-basierte Zählweise](#)) zugeordnet.

Ein Feld hat i. Allg. dauerhaft eine feste Länge. Das heißt, die Zahl der Elemente, die in einem Feld gespeichert werden können, wird bei der Erzeugung des Feldes festgelegt und kann nachträglich i. Allg. nicht mehr geändert werden.

## 2 Bemerkungen

Der Begriff [Array](#) wird nicht einheitlich gebraucht. In einigen Programmiersprachen, wie z.B. [ActionScript](#) oder [Lingo](#), verhalten sich Arrays eher wie [Listen](#), als wie [Felder](#).

Auf ein Element eines Feldes kann direkt zugegriffen werden, wenn seine Position im Feld bekannt ist. Üblicherweise geschieht der Zugriff mit Hilfe des so genannten [Indexoperators](#) `[]`.

```
let a = ['r', 's', 't']; // Definition eines neuen Feldes in Pseudo-Code
print(a[1]);           // Ergebnis: "s", wenn die Positionsnummer des
// ersten Elements 0 ist
                        // Ergebnis: "r", wenn die Positionsnummer des
// ersten Elements 1 ist
```

Mit Hilfe einer **For-Schleife** kann auf alle Elemente eines Feldes nacheinander zugegriffen werden:

```
let a = ['r', 's', 't']; // Definition eines neuen Feldes in Pseudocode
for i := 0 to a.length -1
{ print a[i]; }
```

Es kann darüber hinaus spezielle Operatoren wie z.B. **For-each-Schleifen** geben, mit denen nacheinander auf alle Elemente eines Array zugegriffen werden kann.

In vielen Programmiersprachen ist es möglich, den **Datentyp** der Elemente eines Feldes zu beschränken:

```
String[5], Array<3,Integer> ... // längenbeschränkte und typisierte
Felder in Pseudocode,
// zwei alternative Schreibweisen
```

Ein Feld, das andere Felder als Elemente hat, wird **mehrdimensionales Feld** genannt:

```
String[5][6][3] // drei-dimensionales Feld mit 5*6*3
Elementen in Pseudocode
String[5,6,3] // Kurzschreibweise
Array<5, Array<6, Array<3, String>>>
// alternative Schreibweise
```

## 3 Beispiele

---

### 3.1 Eindimensionales Array

---

Ein eindimensionales Array kann z.B. einen **Vektor** darstellen.

```
vektor: Integer[3] := [0,1,-4];
```

### 3.2 Zweidimensionales Array

---

Ein zweidimensionales kann z.B. eine **Matrix** darstellen.

```
matrix: Integer[5][5]
:= [[0,0,0,0, 0],
    [0,0,7,0, 0],
    [1,1,1,1, 0],
    [0,2,0,0, -4],
    [0,0,0,0, -3]
]
```

## 4 Quellen

---

1. **Schneider, Werner (2001)**: Taschenbuch der Informatik; Hrsg.: [Uwe Schneider](#) und [Dieter Werner](#); Auflage: 4; Verlag: [Fachbuchverlag Leipzig im Carl Hanser Verlag](#); ISBN: 3446217533; [2001](#); Quellengüte: 5 (Buch)
2. **Kowarschick (MMProg)**: [Wolfgang Kowarschick](#); Vorlesung „Multimedia-Programmierung“; Hochschule: [Hochschule Augsburg](#); Adresse: [Augsburg](#); [Web-Link](#); [2018](#); Quellengüte: 3 (Vorlesung)

## 5 Siehe auch

---

[Assoziatives Array](#)

[Wikipedia:Feld \(Datentyp\)](#)

Kategorien:

[Datentyp](#)

[Glossar](#)

Diese Seite wurde zuletzt am 31. Juli 2019 um 17:49 Uhr bearbeitet.

Inhalt verfügbar unter [CC BY-SA 4.0](#).

