

Arrays

Deklaration

Ein Array ist eine Variable, die aus einer Anzahl von Datenelementen gleichen Typs besteht.

```
int Wert1;  
int Wert2;  
:  
:  
int Wert10;
```




```
// Deklaration  
int Werte[10];
```

Deklaration

Ein Array ist eine Variable, die aus einer Anzahl von Datenelementen gleichen Typs besteht.

```
int Wert1 = 0;  
int Wert2 = 10;  
:  
:  
int Wert = 90;
```



```
// Definition  
int Werte[10] = {0, 10, ..., 90};
```

Zugriff auf die Daten

```
int Werte[10];    // Deklaration des Arrays
Werte[0] = 10;    // Initialisierung
Werte[1] = 20;
:
Werte[9] = 100;
// alternativ
for(int i=0; i<10; i++) Werte[ i ] = ( i+1 ) * 10;
```

Der Zugriff erfolgt über
den **Indexoperator []**

Zugriff auf die Daten

```
int Werte[10];  
Werte[0] = 10;  
Werte[1] = 20;  
:  
Werte[9] = 100;  
// alternativ
```

```
for(int i=0; i<10; i++) Werte[ i ] = ( i+1 ) * 10;
```

Wichtig!

Es wird zwar ein Array mit 10 Elementen erstellt, der Index läuft aber von 0 bis 9.

Anzahl der Elemente

Bei der Deklaration wird die Größe des Arrays festgelegt:

- durch einen festen Wert
- durch eine konstante Variable

```
int iAnzahl;  
cout << "Bitte Feldgröße eingeben:" << endl;  
cin >> iAnzahl;  
int Werte[iAnzahl];
```

Das ist so nicht möglich!

Zufallszahlen

```
#include <ctime>
```

```
srand ((unsigned) time(NULL) );
```

```
int Werte [10];
```

```
for (int i = 0; i < 10; i++)
```

```
{
```

```
    Werte[ i ] = rand();
```

```
    cout << Werte[ i ] << endl;
```

```
}
```

```
41
18467
6334
26500
19169
15724
11478
29358
26962
24464

41
18467
6334
26500
19169
15724
11478
29358
26962
24464

41
18467
6334
26500
19169
15724
11478
29358
26962
24464
```

Zufallszahlen

```
#include <ctime>

srand ((unsigned) time(NULL) );

int Werte [10];
for (int i = 0; i < 10; i++)
{
    Werte[ i ] = rand();
    cout << Werte[ i ] << endl;
}
```

```
5454
22590
22663
31695
26425
8230
21266
29609
11909
3582
5421
13410
7862
20438
21583
1731
22909
26113
10633
32295
5157
27525
2656
4622
5303
11345
177
1079
16682
6000
```


Aufgabe:

Erstellen Sie ein Array von 10 Integer-Werten und füllen Sie dieses mit Zufallszahlen. Anschließend ermitteln Sie den maximalen und den minimalen Wert und geben ihn und seine Position im Array auf dem Bildschirm aus.

```
int[] numbers = new int[10];  
for (int i = 0; i < numbers.length; i++)  
    numbers[i] = (int) (Math.random() * 1000);  
int max = numbers[0];  
int min = numbers[0];  
int maxIndex = 0;  
int minIndex = 0;  
for (int i = 1; i < numbers.length; i++)  
    if (numbers[i] > max) {  
        max = numbers[i];  
        maxIndex = i;  
    }  
    if (numbers[i] < min) {  
        min = numbers[i];  
        minIndex = i;  
    }  
System.out.println("Der groesste Wert der Liste ist " + max + " an der Stelle " + maxIndex + ".");  
System.out.println("Der kleinste Wert der Liste ist " + min + " an der Stelle " + minIndex + ".");
```

195	735	294	856	114	887	812	906	743	381
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Der groesste Wert der Liste ist 906 an der Stelle 8.

Der kleinste Wert der Liste ist 114 an der Stelle 5.

Übergabe an Funk

```
void Ausgabe(int [], int anz);
```

Es wird die Speicher-
adresse des ersten
Elementes übergeben

```
int werte[anz] = { 0 };
```

// Definition

```
Ausgabe(werte, anz);
```

// Funktionsaufruf

```
void Ausgabe(int ausg[], int anz)
```

// Funktionskopf

```
{  
    for (int i = 0; i < anz; i++)  
    {  
        cout << ausg[i] << "\\t";  
    }  
}
```

Kennzeichnet die
Übergabe einer
Speicheradresse

Übergabe an Zeigervariablen

```
void Ausgabe(int *, int anz);           // Prototyp der Funktion
```

```
int werte[anz] = { 0 };                // Definition
```

```
Ausgabe(werte, anz);                  // Funktionsaufruf
```

```
void Ausgabe(int *ausg, int anz)       // Funktionskopf
```

```
{  
    for (int i = 0; i < anz; i++)  
    {  
        cout << ausg[i] << "\t";      // Ausgabe der Werte  
    }  
}
```

Zweidimensionales Array

```
1  #include <iostream>
2  #include <ctime>
3  using namespace std;
4
5  void SetzeWerte(int[][4], int);
6  void Ausgabe(int[][4], int);
7
8  int main()
9  {
10     int werte[3][4] = { 0 };
11
12     SetzeWerte(werte, 3);
13     Ausgabe(werte, 3);
14 }
15
16 void SetzeWerte(int zahlen[][4], int anZeilen) { ... }
17
18
19 void Ausgabe(int zahlen[][4], int anZeilen)
20 {
21     for (int j = 0; j < anZeilen; j++)
22     {
23         for (int i = 0; i < 4; i++)
24         {
25             cout << zahlen[j][i] << "\t";
26         }
27         cout << endl;
28     }
29 }
```

Zweidimensionales dynamisches Array

```
5 void SetzeWerte(int**, int, int);
6 void Ausgabe(int**, int, int);
7
8 int main()
9 {
10     int **werte;
11     werte = new int*[3];
12
13     for (int j = 0; j < 3; j++)
14     {
15         werte[j] = new int[4];
16
17         for (int i = 0; i < 4; i++)
18         {
19             werte[j][i] = 0;
20         }
21     }
22
23     SetzeWerte(werte, 3, 4);
24     Ausgabe(werte, 3, 4);
25 }
26
27 void SetzeWerte(int **zahlen, int anZeilen, int anSpalten) { ... }
39
40 void Ausgabe(int **zahlen, int anZeilen, int anSpalten)
41 {
42     for (int j = 0; j < anZeilen; j++)
43     {
44         for (int i = 0; i < anSpalten; i++)
45         {
46             cout << zahlen[j][i] << "\t";
47         }
48         cout << endl;
49     }
50 }
```