

Wenn mehrere Auswahlmöglichkeiten zu berücksichtigen sind, wird eine `if-else`-Verschachtelung unübersichtlich. In diesem Fall ist manchmal eine Mehrfachauswahl besser geeignet.

Die Funktionsweise erkennt man am besten an einem Beispiel: Die Ordnungszahl eines Monats (1...12) soll zur Ausgabe des zugehörigen Monatsnamens führen.

Eingabe: monatszahl					
1	2	3	4	...	monatszahl sonst
Ausgabe: „Januar“	Ausgabe: „Februar“	Ausgabe: „März“	Ausgabe: „April“	und so weiter ...	Ausgabe: „Fehler!“

C++-Code:

```
int monatszahl;

cout << "Bitte eine Monatszahl zwischen 1 und 12 eingeben: ";
cin >> monatszahl;

switch (monatszahl)
{
    case 1:      cout << "Januar";
                 break;
    case 2:      cout << "Februar";
                 break;
    case 3:      cout << "Maerz";
                 break;
    case 4:      cout << "April";
                 break;
    case 5:      cout << "Mai";
                 break;
    case 6:      cout << "Juni";
                 break;
    case 7:      cout << "Juli";
                 break;
    case 8:      cout << "August";
                 break;
    case 9:      cout << "September";
                 break;
    case 10:     cout << "Oktober";
                 break;
    case 11:     cout << "November";
                 break;
    case 12:     cout << "Dezember";
                 break;
    default:     cout << "Fehler!";
}
}
```

In den Klammern nach `switch` kann ein beliebiger Ausdruck stehen, der einen Ganzzahl-Typ ergibt, zum Beispiel `int` oder `char`.

Ein `case`-Block darf mehrere Anweisungen enthalten, ohne dass geschweifte Klammern erforderlich sind. Der Anweisungsteil kann aber auch leer bleiben. Beachten Sie, dass jeder Fall mit `break` abgeschlossen werden muss. Lässt man es weg, dann werden ab dem zutreffenden `case` alle nachfolgenden Anweisungen bis zum nächsten `break` ausgeführt. Man kann diesen Effekt auch absichtlich ausnutzen.