

## Einfache Alternative (Programm-Verzweigung)

Die einfache Programmverzweigung kennt zwei Varianten:

- ohne Alternativzweig
- mit Alternativzweig

Wenn kein Alternativzweig angegeben ist, dann fährt der Ablauf nach dem Ende der Verzweigung fort. Wenn ein Alternativzweig ( `else` ) angegeben ist, dann wird dieser abgearbeitet. Die Entscheidung darüber, welcher Ausführungspfad konkret ausgeführt wird, ergibt sich zur Laufzeit durch Abarbeitung des Bedingungsausdrucks.

```
if( <Bedingungsabfrage> ) {
    <Befehle>
}
else {
    <Befehle>
}
```

Die Verzweigungsbedingung innerhalb der runden Klammern wird als Wahrheitswert interpretiert. Ergibt sich der Wert `TRUE`, dann wird der hinter `if(..)` stehende Anweisungsblock ausgeführt, wenn `FALSE`, dann der hinter `else` stehende. Die `else`-Anweisung kann weggelassen werden, wenn keine alternativen Anweisungen existieren.

Die Bedingung kann als einzelner Wert (Variable oder Konstante) oder als Ausdruck angegeben werden, der erst „berechnet“ werden muss:

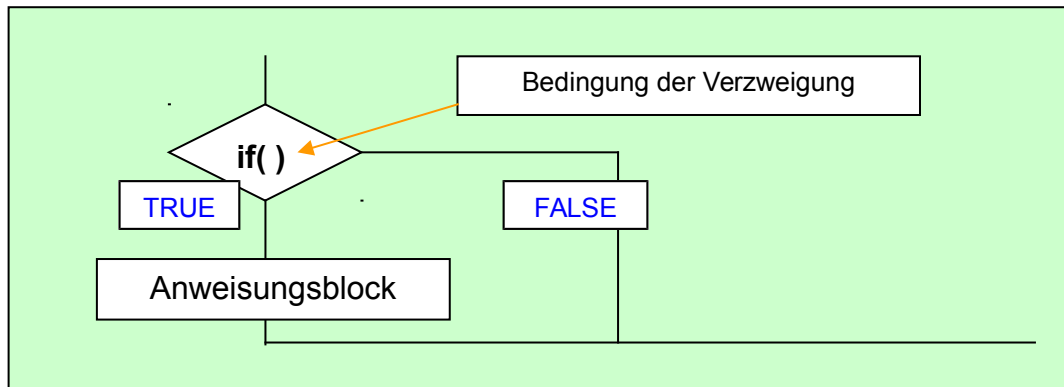
```
int a=4, b=11;

if (a < 5 && b >= 10) {
    System.out.println(""+a+" < 5 UND "+b+" <= 10");
}
```

### Darstellung im Programmablaufplan

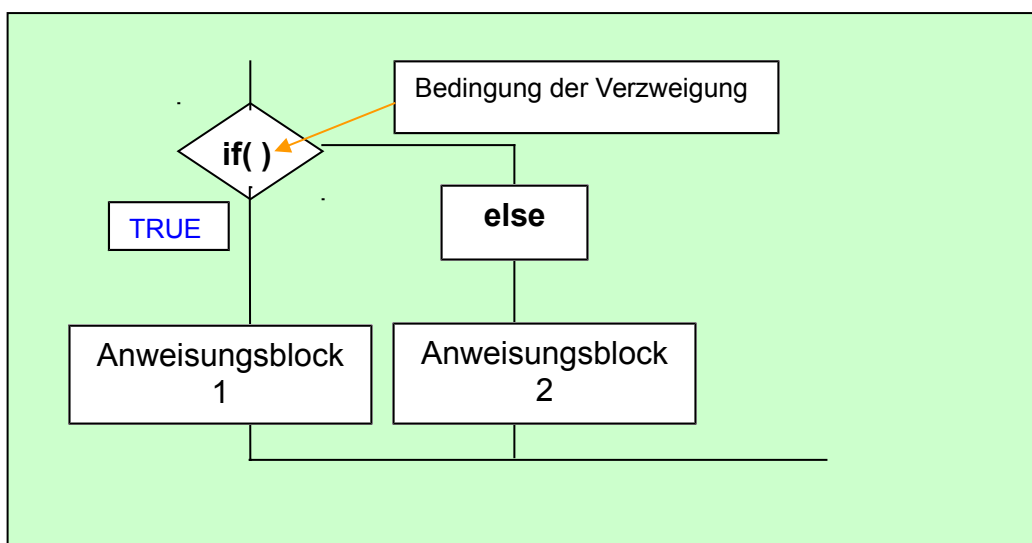
```

if( <Bedingung> ) {
    <Anweisungen>
}
    
```



```

if ( <Bedingung> ) {
    <Anweisungen 1>
}
else {
    <Anweisungen 2>
}
    
```



```
int a = 5;

if ( a < 4 ) {
    System.out.println(„a kleiner 4“);
}
else {
    System.out.println(„a groesser gleich 4“);
}
```

### Mehrfache Alternative (switch - case)

Bei der mehrfachen Alternative sind nur **integrale** Werte (ganze Zahlen, Buchstaben oder als solche interpretierbare Ausdrücke) erlaubt. Wichtig ist die Anweisung **break** am Ende jedes case-Blocks, die das unbeabsichtigte “Durchfallen“ verhindert.

```
char c = 'x';

switch( c ) {
    case 'a': ... ; break;
    case 'x':
        puts(„c ist x \n“);
        printf(„Wollen Sie weitermachen“);
        break;
    default:
        puts(„Falscheingabe“);
}
```

