



Tutorial Link <https://course.testpad.chitkara.edu.in/tutorials/Java: conditional statements/5aa3ea8b5ead875d7faaa8ca>

TUTORIAL

Java: conditional statements

Consider the statement – “is it raining?” The answer is either true or false. This is called a two-way decision.

Here is an outline of a two-way decision structure:

```
// statements that are executed before the decision
if ( condition )
// true branch
else
// false branch
// statements that are executed after the branches join together
again
```

Here are some details:

- The condition evaluates to true or false, often by comparing variables and values.
- The else separates the true branch from the false branch.
- The statements after the false branch are always executed.
- A block consists of several statements inside a pair of braces, { and }.
- The true branch can be a block.
- The false branch can be a block.
- There can be as many statements in a block as you need.
- When a block is chosen for execution, the statements in it are executed one by one.

The condition can compare what is held in a variable to other values. You can use the comparisons $<$, $>$, and so on. The first statement after the false branch will be executed no matter which branch is chosen. The if-else is like a fork in the road, but the road always comes together again.

Here is a program that implements the wiper decision in a car:

```
1  import java.util.Scanner;
2  class Main
3  {
4      public static void main (String[] args)
5      {
6          String answer;
7          System.out.print("Is it raining? (Y or N): ");
8          answer = "Y";
9          if ( answer.equals("Y") )           // is
answer exactly "Y" ?
10             System.out.println("Wipers On");           //
true branch
11             else
12                 System.out.println("Wipers Off");       //
false branch
13     }
14 }
15
```

The user is prompted to answer with a single character, Y or N :

```
System.out.print("Is it raining? (Y or N): ");
```

The Scanner reads in whatever the user enters (even if the user enters more than one character):

```
answer = scan.nextLine();
```

The if statement tests if the user entered exactly the character Y (and nothing else):

```
if ( answer.equals("Y") ) // is answer exactly "Y" ?
```

If so, then the statement labeled "true branch" is executed. Otherwise, the statement labeled "false branch" is executed. The "true branch" is separated from the "false branch" by the reserved word else. Here is another program with more statements:

```
1  import java.util.Scanner;
2
3  class Main
4  {
5      public static void main (String[] args)
6      {
7          int num;
8          System.out.println("Enter an integer:");
9          num = 15;
10         if ( num < 0 )
11         {
12             System.out.println("The number " + num + " is
13             negative."); // true-branch
14             System.out.println("Negative numbers are less
15             than zero."); // true-branch
16         }
17         else
18         {
19             System.out.println("The number " + num +
20             // false-branch
21             " is zero or positive.");
22             // false-branch
23             System.out.print ("Positive numbers are greater
24             "); // false-branch
25             System.out.println("than zero. ");
26             // false-branch
27         }
28         System.out.println("Good-bye for now"); //
29         always executed
30     }
31 }
```

To include more than one statement in a branch, enclose the statements with braces, { and }. A group of statements grouped together like this is called a block statement, (or usually, just block). There can be as many statements as you want in a block. A block can go any place a single statement can go. All the statements in the true

block are executed when the answer to the question is true. Of course, all the statements in the false block are executed when the answer to the question is false. The false block consists of the block that follows the else. Notice that the very last statement in the program is not part of the false block.

An if statement makes a two-way decision. Surely you must sometimes pick from more than just two branches? We ran into this problem with a previous example program that divided integers into negative and non-negative. It really should pick one of three choices:

```
Negative: ... -3 -2 -1  
Zero: 0  
Positive: +1 +2 +3 ...
```

Two-way decisions can do this. First divide the integers into two groups (using a two-way decision):

```
Negative: ... -3 -2 -1  
Zero and Positive: 0 +1 +2 +3 ...
```

Then further divide the second group (by using another two-way decision):

```
Negative: ... -3 -2 -1  
Zero and Positive:  
    Zero: 0  
    Positive: +1 +2 +3 ...
```

By repeatedly splitting groups into subgroups, you can split a collection into any number of fine divisions. So the previous program can be rewritten as below:

```
1 import java.util.Scanner;  
2 class Main  
3 {  
4     public static void main (String[] args)  
5     {  
6         int num;  
7         System.out.println("Enter an integer:");
```

Java

```
8      num = 15;
9      if ( num < 0 )
10     {
11         System.out.println("The number " + num + " is
negative."); // true-branch
12         System.out.println("Negative numbers are less
than zero."); // true-branch
13     }
14     else if(num == 0)
15     {
16         System.out.println("The number " + num + " is
zero."); // Second true branch
17     }
18     else
19     {
20         System.out.println("The number " + num + " is
positive."); // Second true branch
21         System.out.print ("Positive numbers are greater
"); // false-branch
22         System.out.println("than zero. ");
// false-branch
23     }
24     System.out.println("Good-bye for now"); //
always executed
25 }
26 }
27
```



CodeQuotient

Tutorial by codequotient.com | All rights

reserved, CodeQuotient 2025