Advanced Modeling with Java > Java Basics for AnyLogic > Statements

if-then-else

As in any language that supports imperative programming, Java has *control flow statements* that "break up the flow of execution by employing decision making, looping, and branching, enabling your program to conditionally execute particular blocks of code" [The Java Tutorials, Oracle].

The "if-then-else" statement is the most basic control flow statement that executes one section of code if a condition evaluates to true, and another – if it evaluates to false. The statement has two forms, short:

```
if( <condition> )
    <statement if true>
```

and full:

```
if( <condition> )
    <statement if true>
else
    <statement if false>
```

For example, this code assigns a client to a salesman only if the salesmen is currently following up less than 10 clients.

```
if( salesman.clients.size() < 10 )
    salesman.assign( client );
```

And this code tests if there are tasks in a certain queue and, if yes, assigns the first one to a truck, otherwise sends the truck to the parking position:

```
if( tasks.isEmpty() )

    truck.setDestination( truck.parking );

else

    truck.assignTask( tasks.removeFirst() );
```

In case "then" or "else" code section contains more than one statement, they should be enclosed in braces `{ … }` to become a block, which is treated as a single statement, see the code below. We however recommend to always use braces for "then" and "else" sections to avoid ambiguous-looking code. Braces are specifically important when there are nested "if" statements or when lines of code in the "if" neighborhood are added or deleted during editing or debugging.

```
if( friends == null ) {

    friends = new ArrayList< Person >();

    friends.add( john );

} else {

    if( ! friends.contains( john ) )

        friends.add( john );
}
```