

CODAPPS

Conditional statements, loops and data structures

Clément Levallois

Version 1.0, last modified

Table of Contents

| | |
|--|---|
| 1. Useful additions to your coding toolkit | 1 |
| a. If ... else ... making your app adapt | 1 |
| The end | 3 |



1. Useful additions to your coding toolkit

What we have seen so far is really the essential of coding: variables, methods and classes.

We can code an app with this. But what makes programming really interesting is when we take advantage of three things computers (and mobile phones) are really good at:

1. they can adapt their behavior to changing environments
2. they are really fast
3. they can handle huge volumes of things

In this lesson, we learn how to do these 3 things. Promised, this is the last of the "technical" lessons. After this one you will create a great app.

a. If ... else ... making your app adapt

If your app does always the same thing, it will be quickly boring.

What you need is the ability to code something that says:

"if the user of the app chooses this option, then do this. If not, do that instead"

This need is so universal that all programming languages have a way to express this "if this, then that".

The formal name for it is a **conditional statement**.

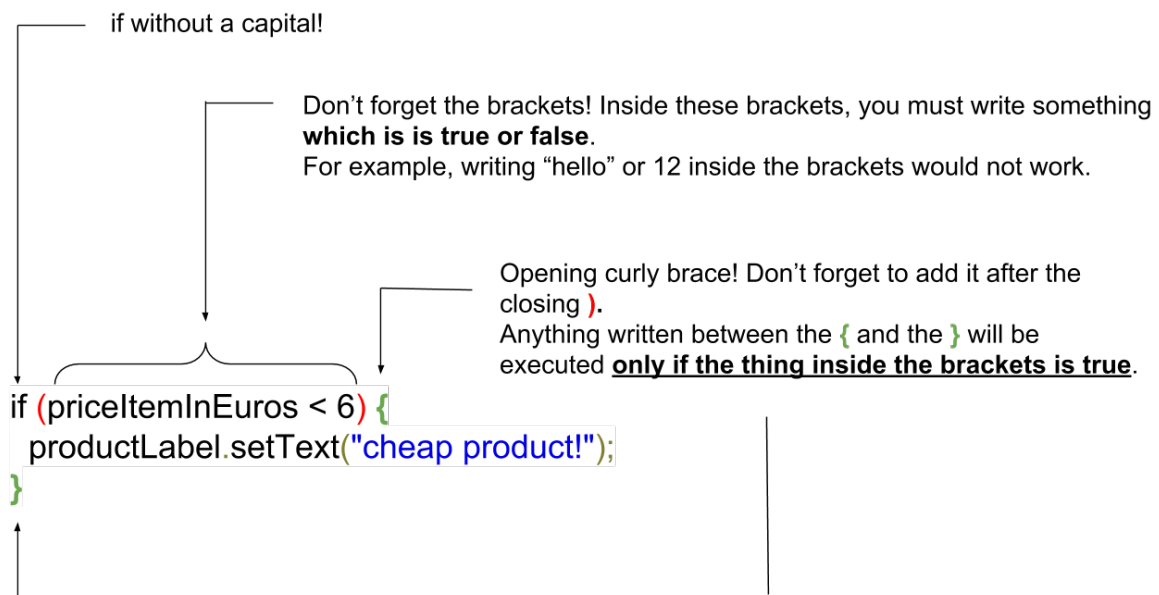
Let's see:

Creating a conditional statement

```
Float priceItemInEuros;  
priceItemInEuros = 5.99f; ①  
Label productLabel = new Label();  
if (priceItemInEuros < 6) {  
    productLabel.setText("cheap product!");  
}
```

This will put the text "cheap product" in the Label, **only if the price of the product is under 6**.

Let's have a closer look at the syntax (the boring part, but where any mistake will make your build fail)



What if `priceItemInEuros` is 8.99 ? Then what's inside the curly braces is just ignored: it does not happen.

Figure 1. Detailing how a conditional statement works

That's about it for conditional statements. We just need to learn to write "if these things are **equal**" or "if these things are **unequal**":

Creating a conditional statement

```
Float priceItemInEuros;  
priceItemInEuros = 5.99f; ①  
Label productLabel = new Label();  
if (priceItemInEuros == 5.99) { ①  
    productLabel.setText("the price is exactly 5.99");  
}  
if (priceItemInEuros != 5.99) { ②  
    productLabel.setText("the price is different from 5.99");  
}
```

① the `==` is surprising. Why twice `=`? Just because `=` is already taken to assign a value to a variable.

② `!=` looks weird, why not just `≠`? Well try to find `≠` on your keyboard

Before we leave conditional statements: we'll see in the app that we use them **a lot**. Some even say that the many products with "artificial intelligence" sold out there actually just have plenty of conditional statements in their code

The end

Questions? Want to open a discussion on this lesson? Visit the forum [here](#) (need a free Github account).

Find references for this lesson, and other lessons, [here](#).

Licence: Creative Commons, [Attribution 4.0 International](#) (CC BY 4.0). You are free to:

- copy and redistribute the material in any medium or format
- Adapt — remix, transform, and build upon the material

⇒ for any purpose, even commercially.



This course is designed by Clement Levallois.

Discover my other courses in data / tech for business: <http://www.clementlevallois.net>

Or get in touch via Twitter: [@seinecle](#)