

In this video, we're going to focus on functional methods and how we call functional methods, because if you remember back earlier in the course I mentioned, functional methods allow us to include the method call in line with our calling programs, standard routines.

And what we're going to do is instead of using the standard method of calling the method like we have here, you're remembering the code we used the exporting and then we use the receiving.

We can swap it around to where our data item to which in our case was the result variable.

We can swap it round like this so we can say the result equals our class name, our method name, and

then specify the methods important parameter, which in this method of calling is the exporting parameter.

Let's switch over to the code and you'll see it in action.

So here we are.

And what I'm going to do, I'm just going to copy and paste this existing piece of code here.

We will take it down to the bottom.

And I will say

a better way of calling a functional method.

So where we have Kawan go slower, export in increment receiving and so on.

Let's change this round so all we need to do is say the result.

Is equal to the Klusener dash, greater than go slower, which is the method name, and then we don't

need all this stuff, let's slow it down just by the number five.

Say how much easier that is, and this is why functional methods are used quite often within the logic

of our code, so we could even say something like, I'm not going to do it, but we could say if this

and then, you know, to do more code depending on the true or false statement.

But we'll stick with this.

Would you say the result equals and then we're going to call it this way.

So where we had the receiving and where we were passing the returning methods parameter back into the

result this way when assigning it directly.

Just by doing this.

Let's do a syntax check.

And then we'll write you down.

Let's put a little comment here.

The results of go slower,

let's just say functional method in brackets is then the result.

Execute him.

Scroll to the bottom and here you go.

So it's executed again, the speed was 10 miles per hour or 10.

The speed is now five because we reduced it by five.

And then he writes out the results of go slower.

The functional method is five.

So there you have two ways of calling your functional methods.

And really, this last way is a lot better than this one.