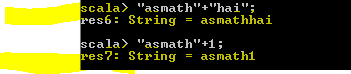




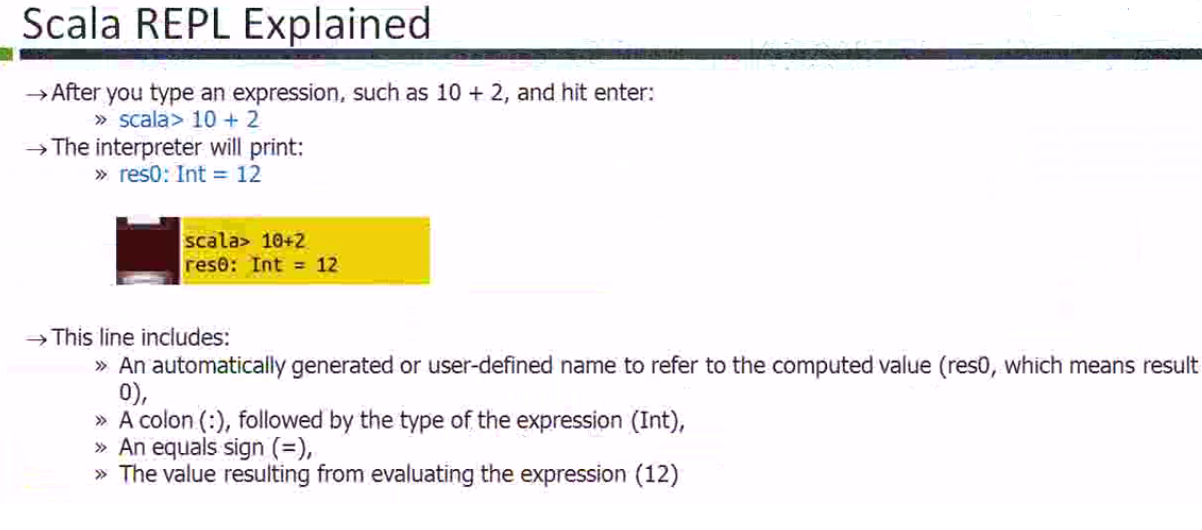
Scala interprets automatically and give the value back. That is whether it is string or int or double etc.



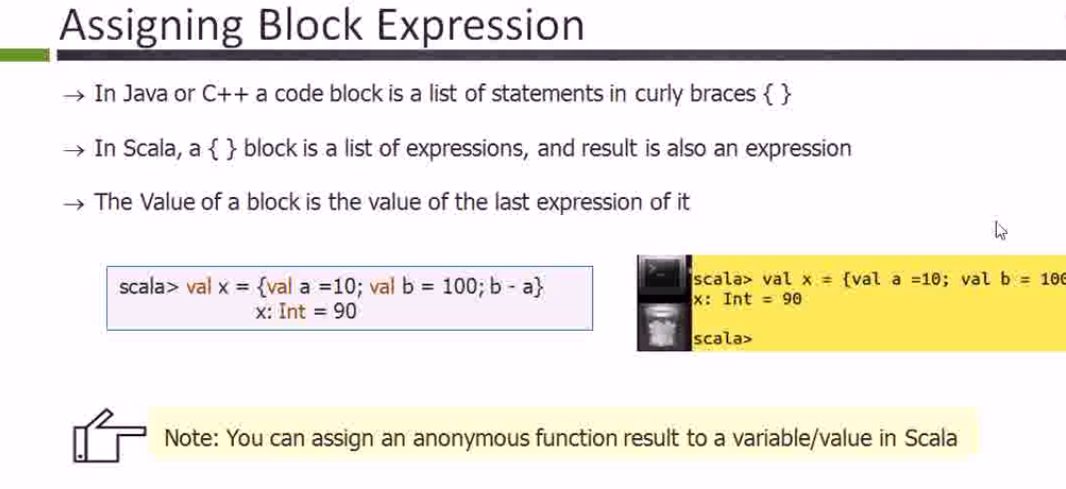
Adding strings in scala.



This is called type inference.



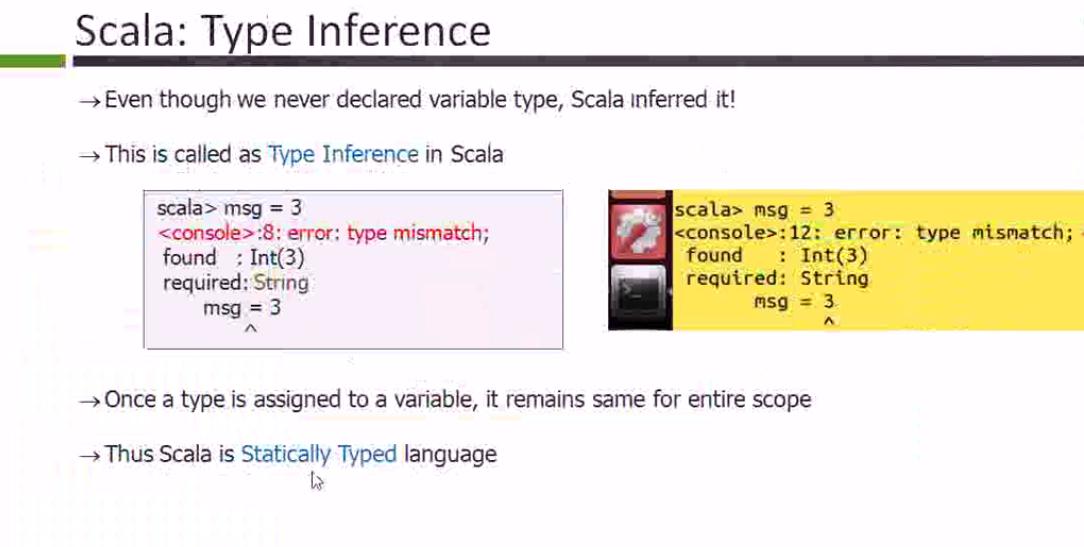
Above, res stands for result. res0 means result 0.

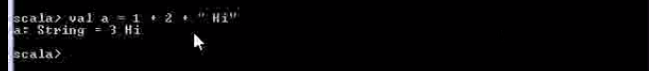


In java they are statements but in scala it is expressions.

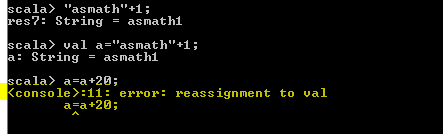
**Varibale Types in Scala:**

Var and val are two different types in scala.



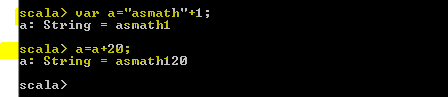


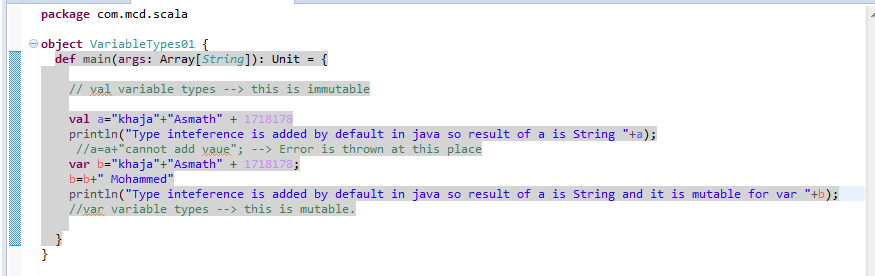
Val is immutable here which means constant.



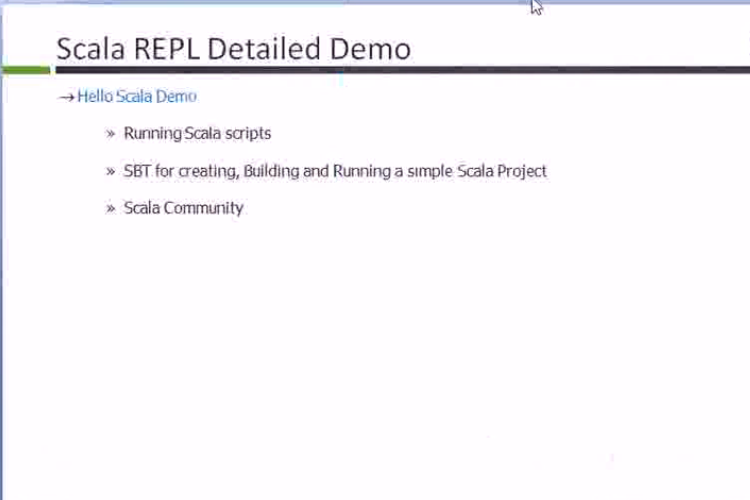
You created variable with some value and reassignment to that value will result in error as it is immutable.

Try the same example with var. It will work.

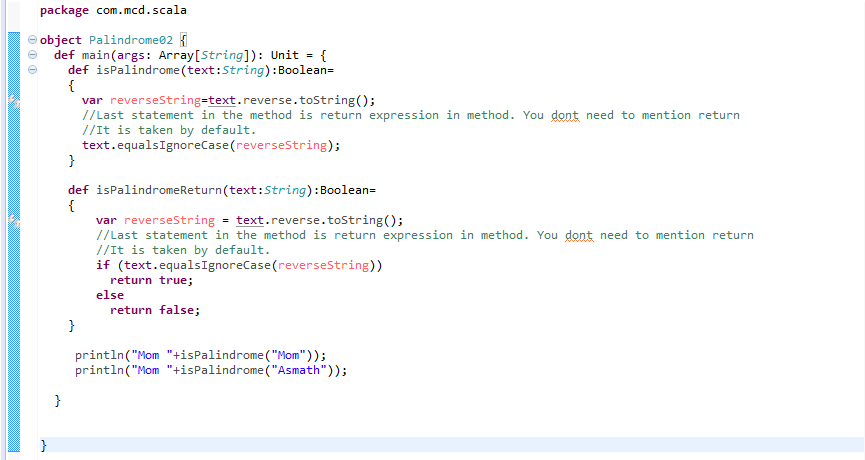




* Scala Demo

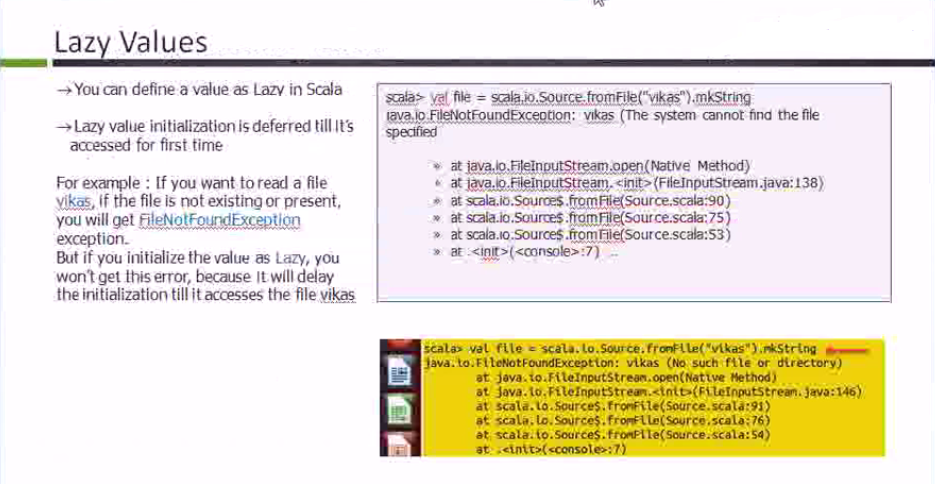


REPL stands for Read, Evaluate, Print and Loop.

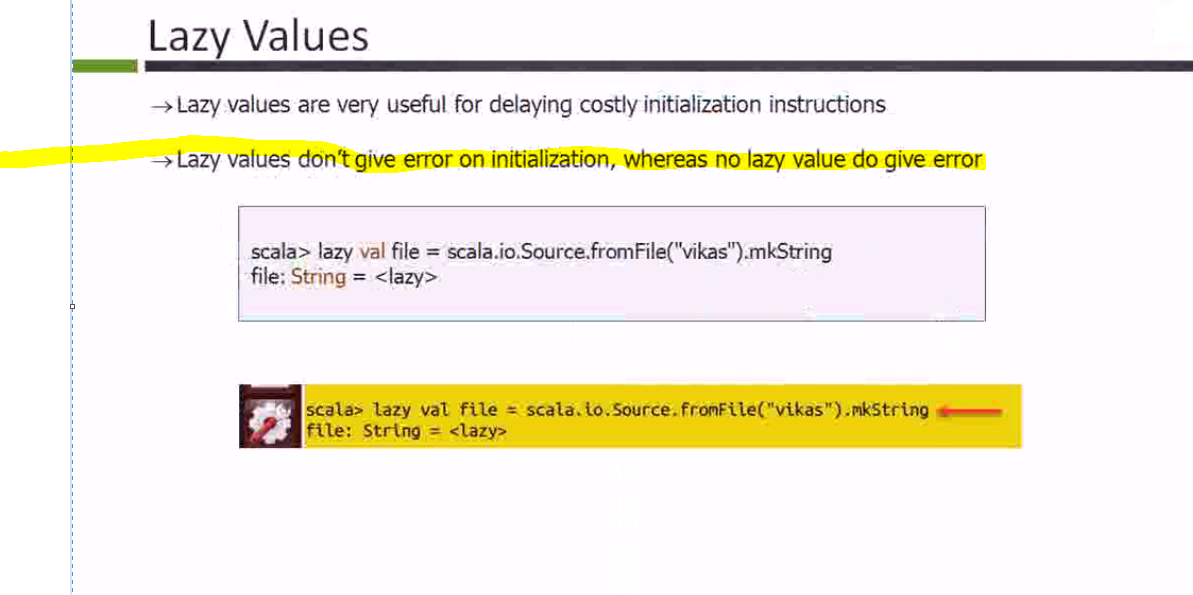


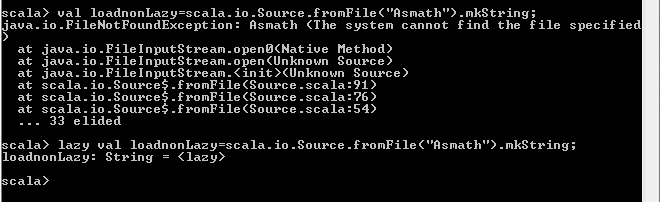
Maven is painful for scala but not sure why it is.

**Lazy Values:**



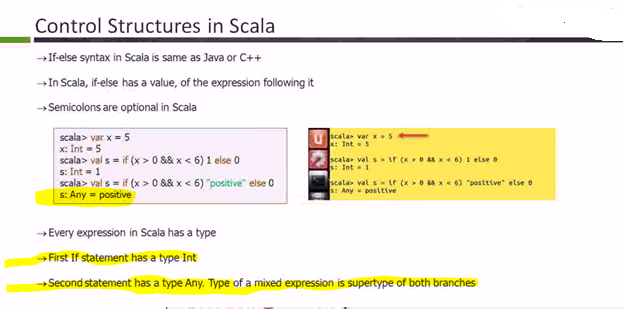
Lazy values don’t give any error while compilation time where as non lazy value give error at compilation time.

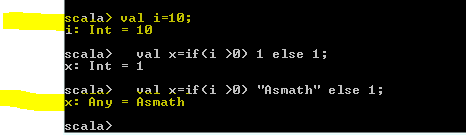




Typically we don’t use much but will be useful in spring applications.

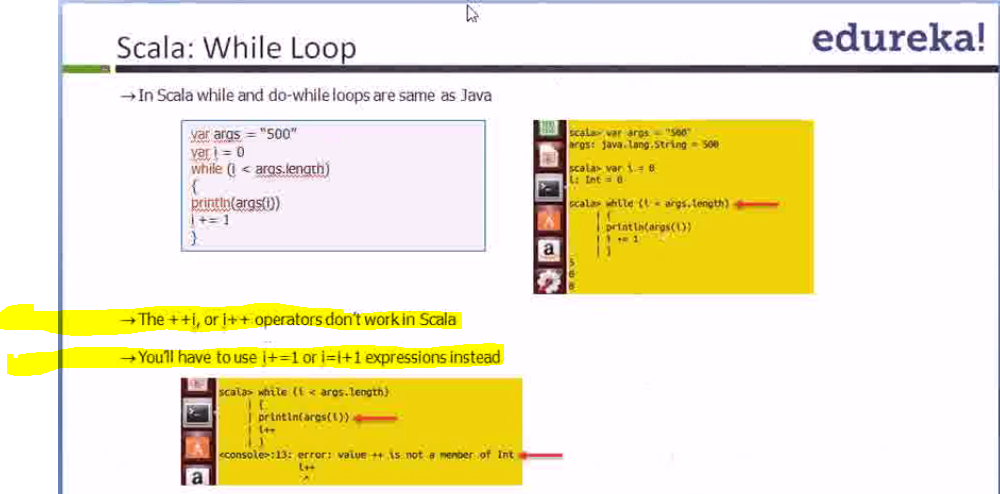
**Control Structures in Scala:**

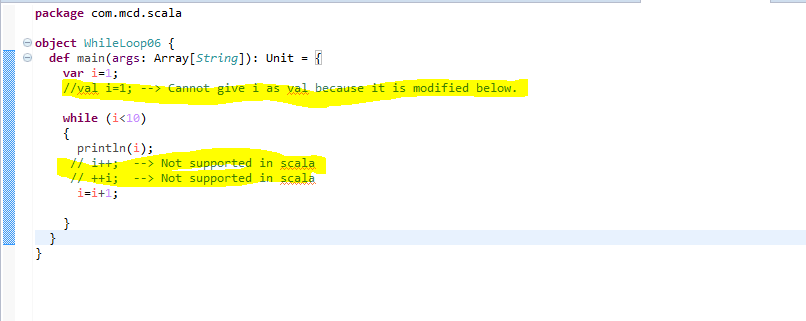


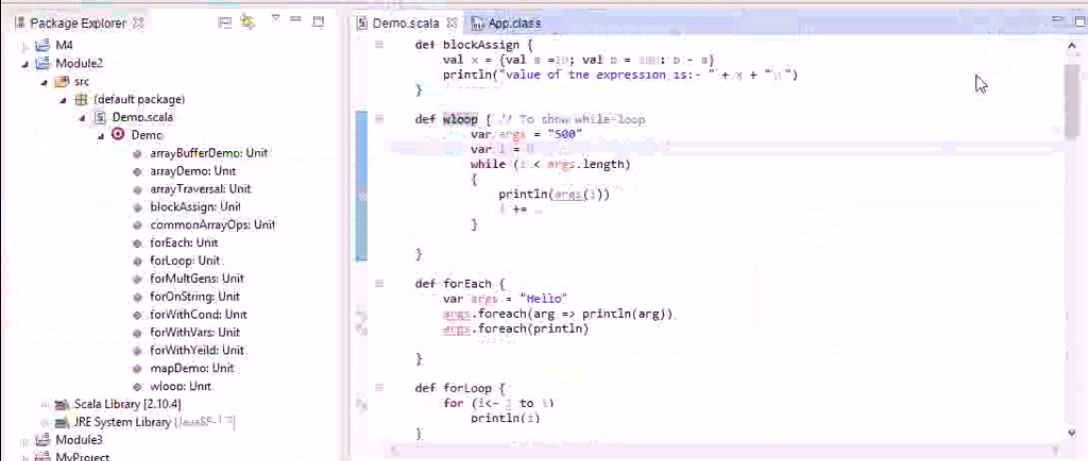


If you want to use Any datatype with other datatype, it should be type casted.

**While loop in Scala:**



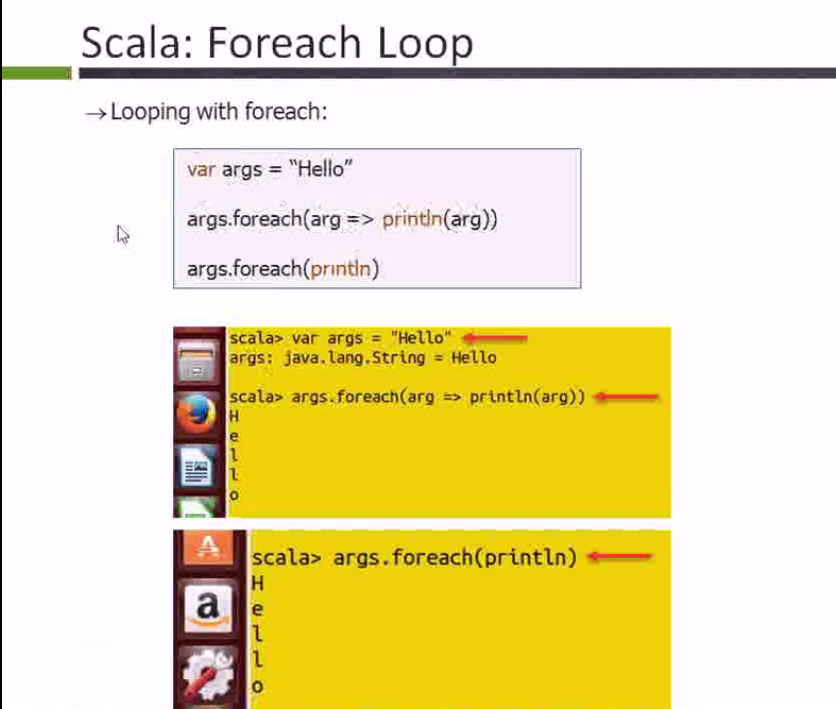




**Asmath 🡪 Try putting all in once class and call them with methods as shown above.**

**Foreach in Scala:**

This loop will take anonymous function. Also read about syntactical sugar.



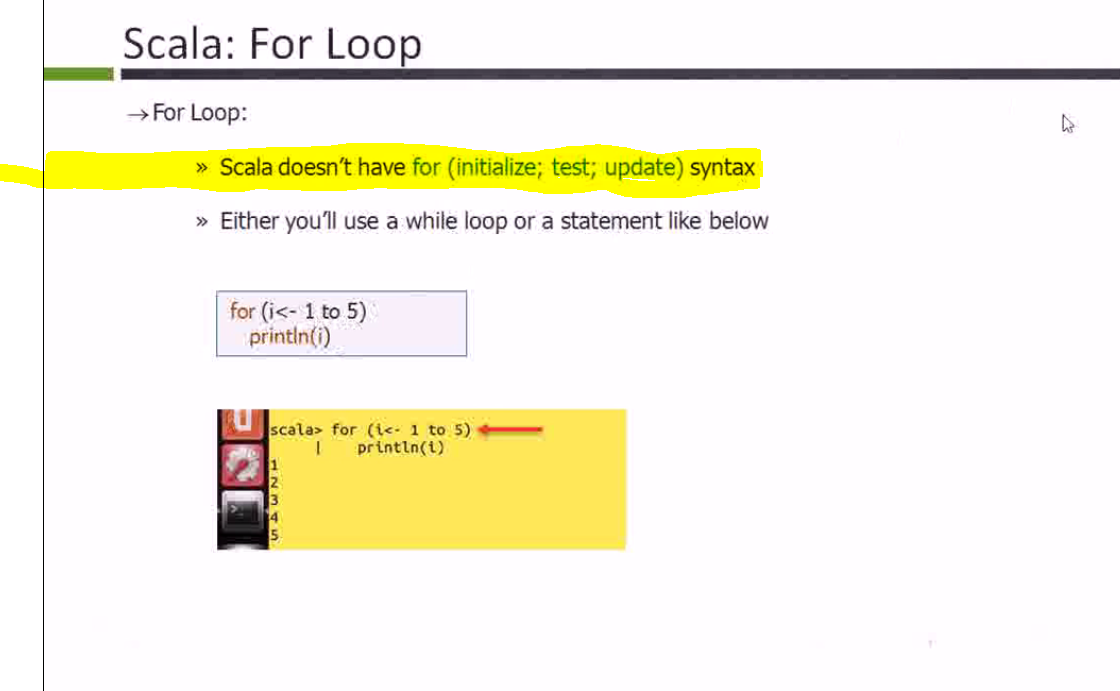
In above example, both return the same output. Lets go deeper.

If args.foreach(i => println(i)), this loop is taking anonymous function with parameter i.

Println(i) is the body of the function and if that body is using the parameter only once after =>, then you can omit both the argument i and (). Which makes both are same now.

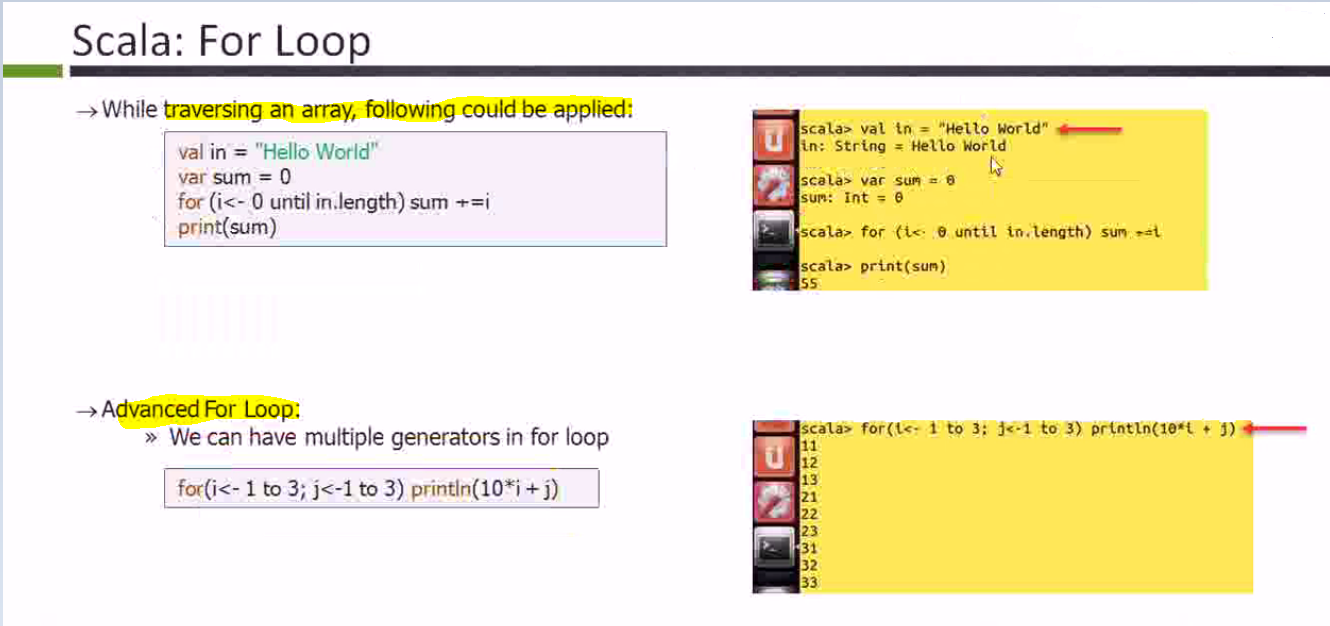
Scala will take argument automatically, you can give any name.

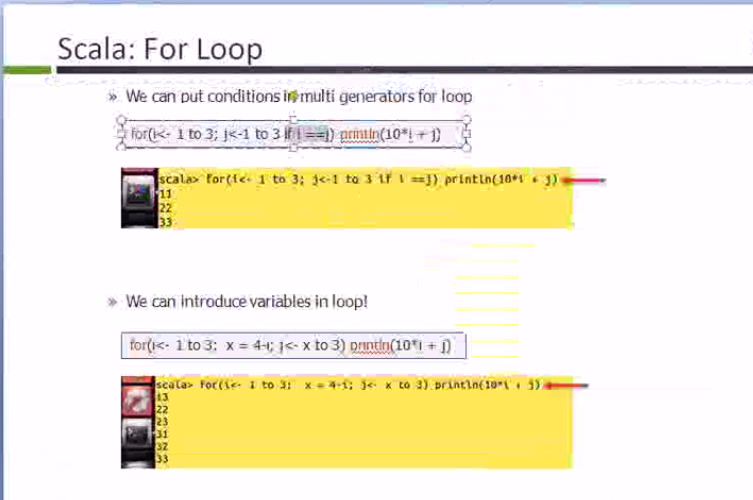
**Scala ForLoop (this is different to foreach loop)**



**No need of initialization in for loop. It defines the range (like java we don’t need i++, i=0)**

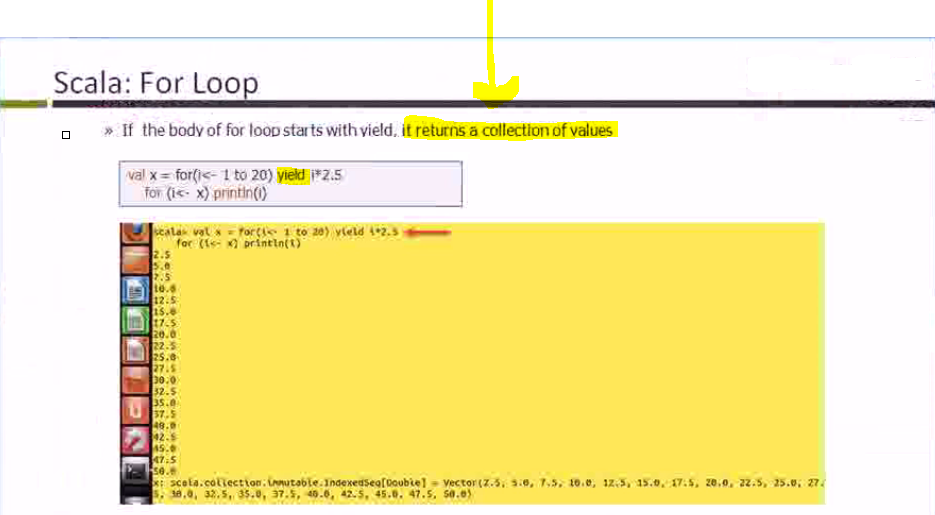
**<- is syntax in scala and => is obtained from the library.**

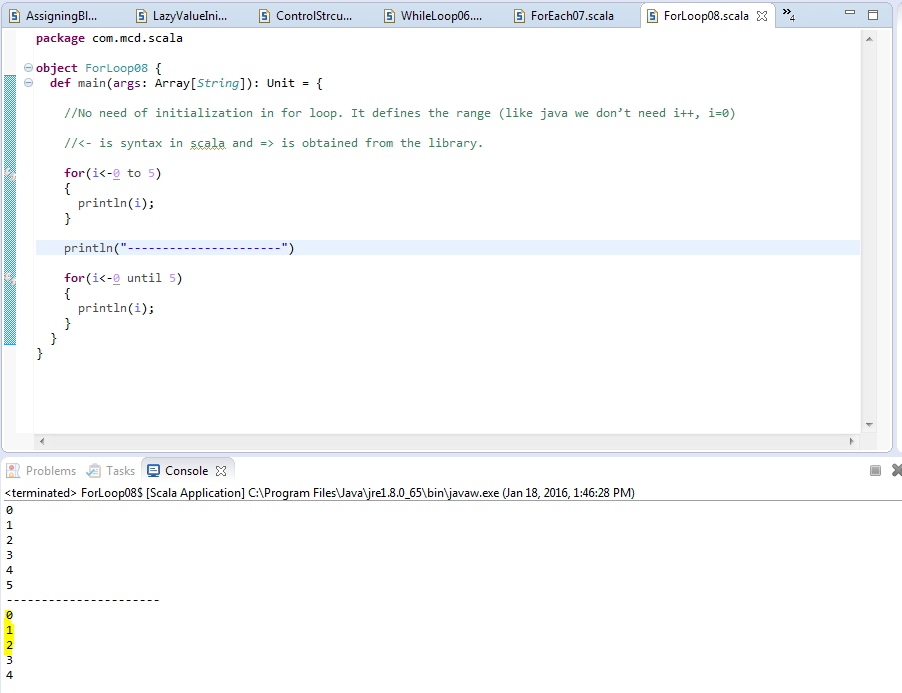




**We can declare variables and do conditional checking too in for loop. See above screenshot and code.**

**Yield will return collection. This is available in ruby and csharp.**





Until will not take last value.

**Yield output**

