

Object Meets Function

Scala

Steven Lolang

`steven.lolang(at)uni-tuebingen.de`

**Programming Language Research Group
Tuebingen University**

Oct. 30, 2023



Summary

- 
- 1 REPL and Compiler
 - 2 REPL

- 3 Compiler
- 4 Entry Point



REPL and Compiler

`scala` open scala terminal for REPL, or run Scala's bit code.

`scalac` compile Scala's source code to Scala's bit code.



Read - Evaluate - Print - Loop – REPL

```
1 1 + 1
2 5 + (2 * 2) - 9 * 1
3
4 // value
5 val a = 5
6
7 // variable
8 var b = 7
9 a + b
10
11 // Method
12 def aPlusB(a: Int, b: Int): Int = a + b
13 aPlusB(4, 2)
14
15 val max: (Int, Int) => Int = (a: Int, b: Int) => {
16     if (a > b) a
17     else b
18 }
19 max(7, 4)
```

Listing: REPL



REPL

scala> :help
:help print this summary
:load *<path>* interpret lines in a file
:quit exit the interpreter
:type *<expression>* evaluate the type of the given expression
:doc *<expression>* print the documentation for the given expression
:imports show import history
:reset [options] reset the repl to its initial state, forgetting all session entries
:settings *<options>* update compiler options, if possible



Compile and Run

Scala compiler will compile Scala's source code into virtual-machine byte code (VMBCode), not a native-machine code (NBCode).

Compiling a file

```
scalac sourceCode.sc
```

Execute virtual-machine byte code

```
scala fileName
```

fileName only without file extension (.class)

ex.: VMBCode **foo.class** and Scala source code **bar.scala**

```
>scala foo
```

```
>scala bar.scala
```

Note: The program file (.class or .sc) to be run requires one method as a starting point.



Entry Point

- Method as an entry point means this method will automatically call.
- It needs an annotation **@main** in front of the function definition to make a function become an entry point.

```
1 @main def hello() =  
2   println("Hello world!")
```

Listing: @main annotation

