

SCALA

A FUNCTIONAL AND OBJECT-ORIENTED PROGRAMMING LANGUAGE

Benedict R. Gaster / @cuberoo_



University of the
West of England

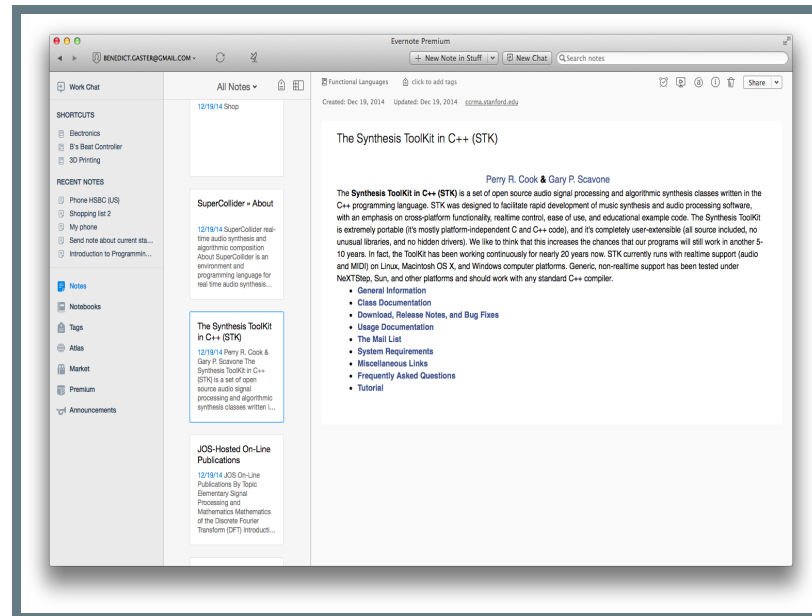
bettertogether

COMPUTER PROGRAMMING

Provide instructions so computer can perform tasks

COMPUTER PROGRAMMING

Prepackged software provides fixed solution, e.g.:



COMPUTER PROGRAMMING

- Allows **you** to write the software
- Allows **you** to tell the computer what to do

MACHINE LANGUAGES

- Low-level
- Verbose
- Prone to errors

NATURAL LANGUAGES

- Examples include English and French
- Ambiguous and hard for computer to process
 - "The peasants are revolting"
(<http://simple.wikipedia.org/wiki/Ambiguity>)

HIGH-LEVEL LANGUAGES

A middle ground

PROGRAMMING

Fun, creative, and satisfying

PROGRAMMING

Enables real world interaction with huge number of

- devices, e.g. smartphones, desktop computers, ...
- sensors
- controllers

PROGRAMMING

- Found in all walks of life
- not only computer scientists write computer programs

BEGIN PROGRAMMING

First off, one must learn a programming language

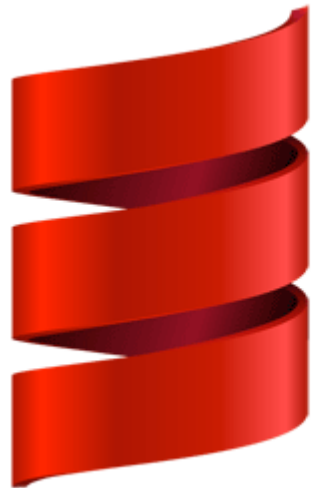
WHICH PROGRAMMING LANGUAGE?



KNOW MANY PROGRAMMING LANGUAGES

Most "real" projects use multiple languages

FOR THIS COURSE



Scala

WHY SCALA?

- Scala is object-oriented---every value is an object
- Scala is an functional---every function is a value
- Scala is statically typed---helps catch errors early
- Scala is based on the Java Virtual Machine (JVM)
Fully compatable with Java!

WHY SCALA?

Widely used, e.g. the web application library Scalatra used by:

- BBC
- Netflix
- Guardian News Paper

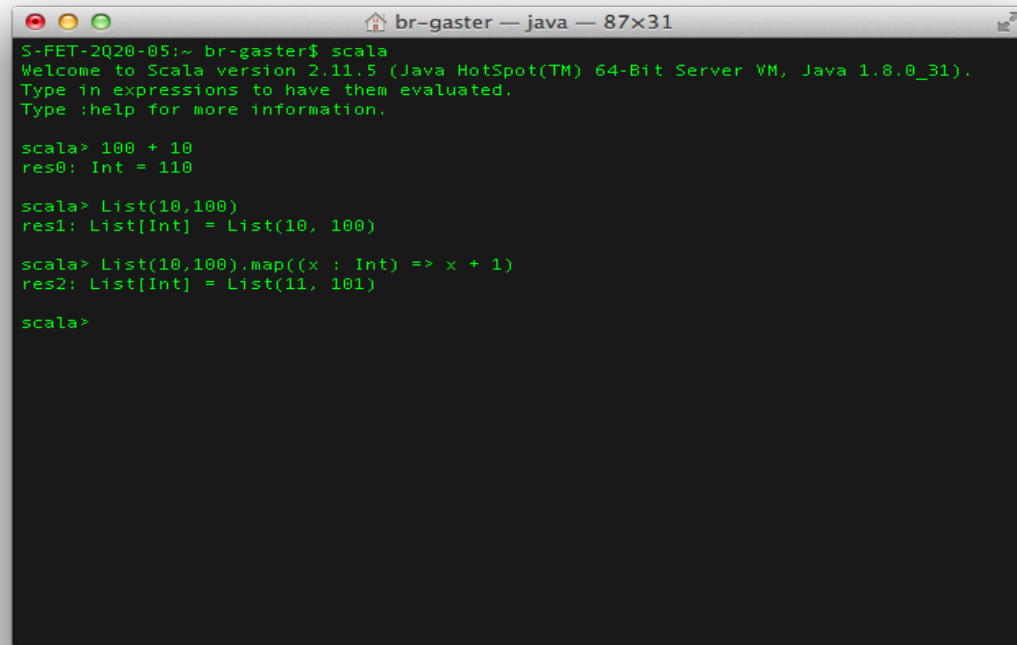
SETUP

- Main website: <http://www.scala-lang.org/>
- Download and install Scala 2.11.5 binaries for your system
- You will need a text editor (e.g. edit, vi, emacs, ...)
 - We will look at Intergrated Development Environments (IDE) later

PLAYING WITH SCALA

- The Scala tools provides a wide selection of utilites for developing Scala progams
- The command tool **scala**, a Read-eval-print loop (REPL) tool, allows interactive development of programs
- Great for learning to program or a new language

PLAYING WITH SCALA



A screenshot of a Scala REPL window titled "br-gaster — java — 87x31". The window shows the following text:

```
S-FET-2020-05:~ br-gaster$ scala
Welcome to Scala version 2.11.5 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_31).
Type in expressions to have them evaluated.
Type :help for more information.

scala> 100 + 10
res0: Int = 110

scala> List(10,100)
res1: List[Int] = List(10, 100)

scala> List(10,100).map((x : Int) => x + 1)
res2: List[Int] = List(11, 101)

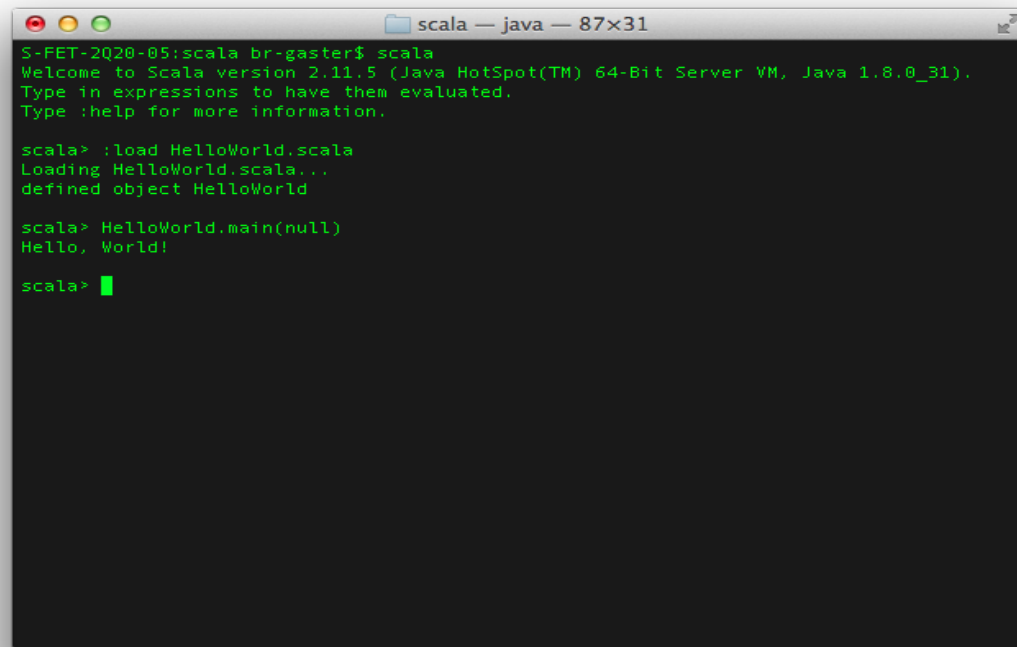
scala>
```

HELLO WORLD

Create a program by typing it into text editor, call it
HelloWorld.scala

```
object HelloWorld {  
  def main(args: Array[String]) {  
    println("Hello, World!")  
  }  
}
```

RUNNING HELLO WORLD



A screenshot of a terminal window titled "scala — java — 87x31". The window shows the Scala REPL interface. The prompt is "S-FET-2020-05:scala br-gaster\$". The user enters "scala", and the REPL responds with "Welcome to Scala version 2.11.5 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_31). Type in expressions to have them evaluated. Type :help for more information." The user then enters "scala> :load HelloWorld.scala", and the REPL responds with "Loading HelloWorld.scala..." and "defined object HelloWorld". The user then enters "scala> HelloWorld.main(null)", and the REPL responds with "Hello, World!". The prompt "scala> " is followed by a green cursor.

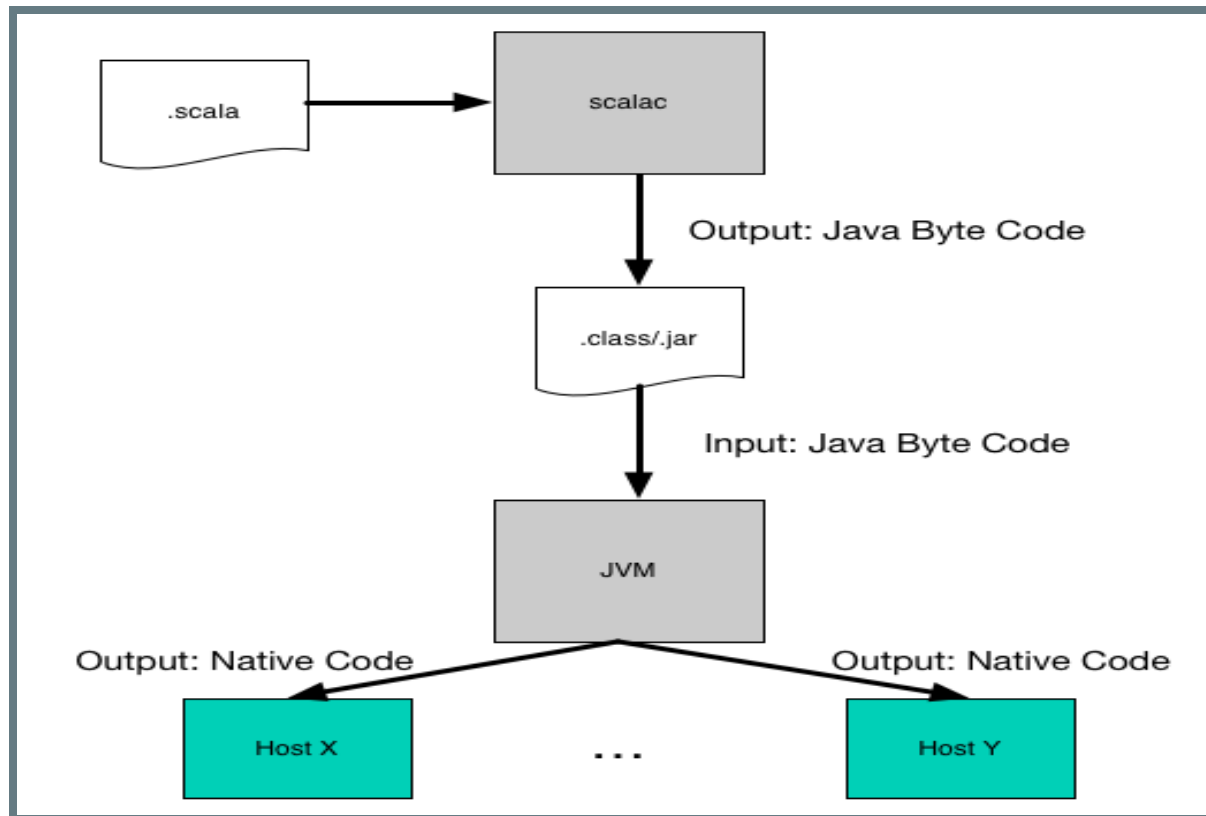
```
S-FET-2020-05:scala br-gaster$ scala
Welcome to Scala version 2.11.5 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_31).
Type in expressions to have them evaluated.
Type :help for more information.

scala> :load HelloWorld.scala
Loading HelloWorld.scala...
defined object HelloWorld

scala> HelloWorld.main(null)
Hello, World!

scala> █
```

SCALA COMPILATION FLOW



SBT - THE INTERACTIVE BUILD