SCALA

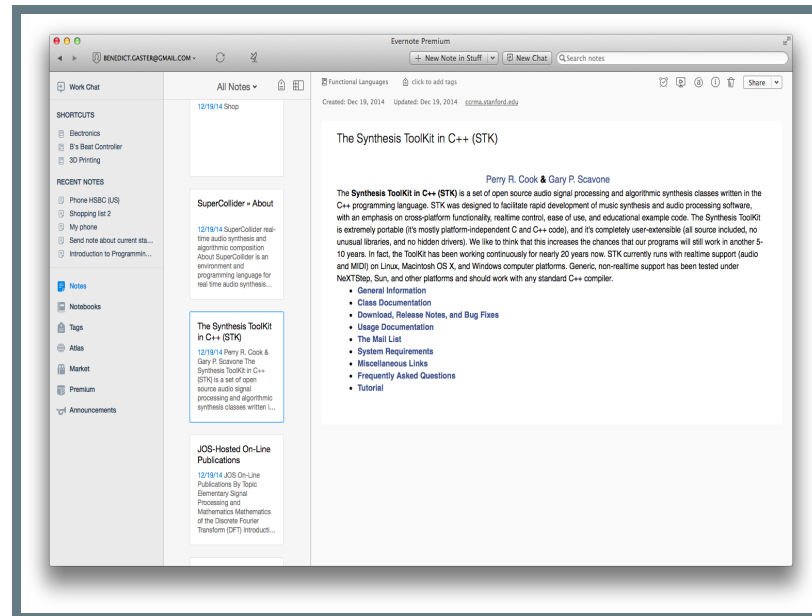## A FUNCTIONAL AND OBJECT-ORIENTED PROGRAMMING LANGUAGE

Benedict R. Gaster / @cuberoo_

COMPUTER PROGRAMMING

Provide instructions so computer can peform 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
- Ambigious 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

PROGRAMMABLE DEVICES

PROGRAMMING

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

BEGIN PROGRAMMING

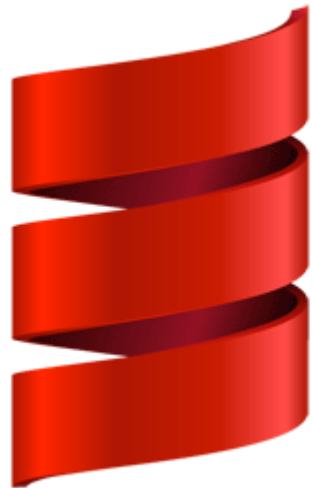First off, one must learn a programming language

WHICH PROGRAMMING LANGUAGE?

KNOW MANY PROGRAMMING LANGUAGES

Most "real" projects use multiple languages

FOR THIS COURSE

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:
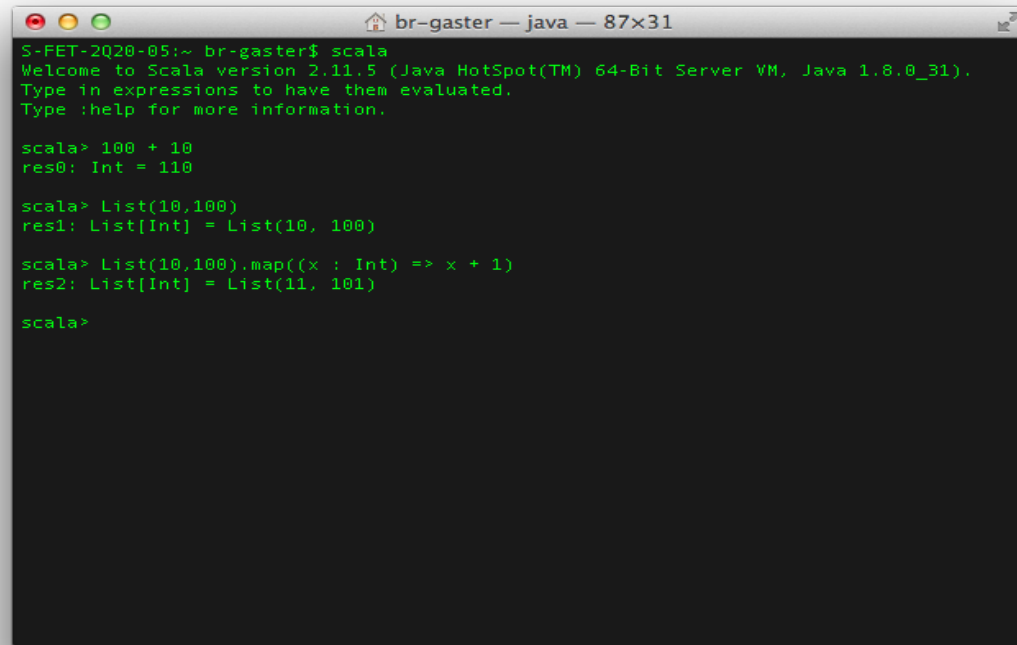
- BBC
- Netflix
- Guardian News Paper

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
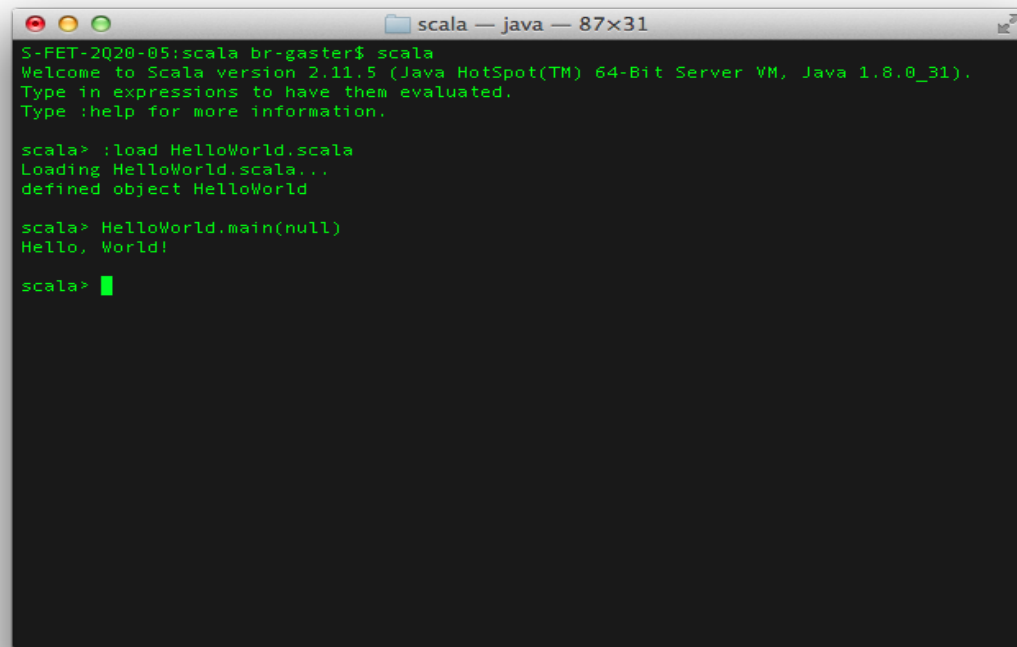- Great for learning to program or a new language

PLAYING WITH SCALA

HELLO WORLD

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

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

RUNNING HELLO WORLD

SCALA COMPILIATION FLOW
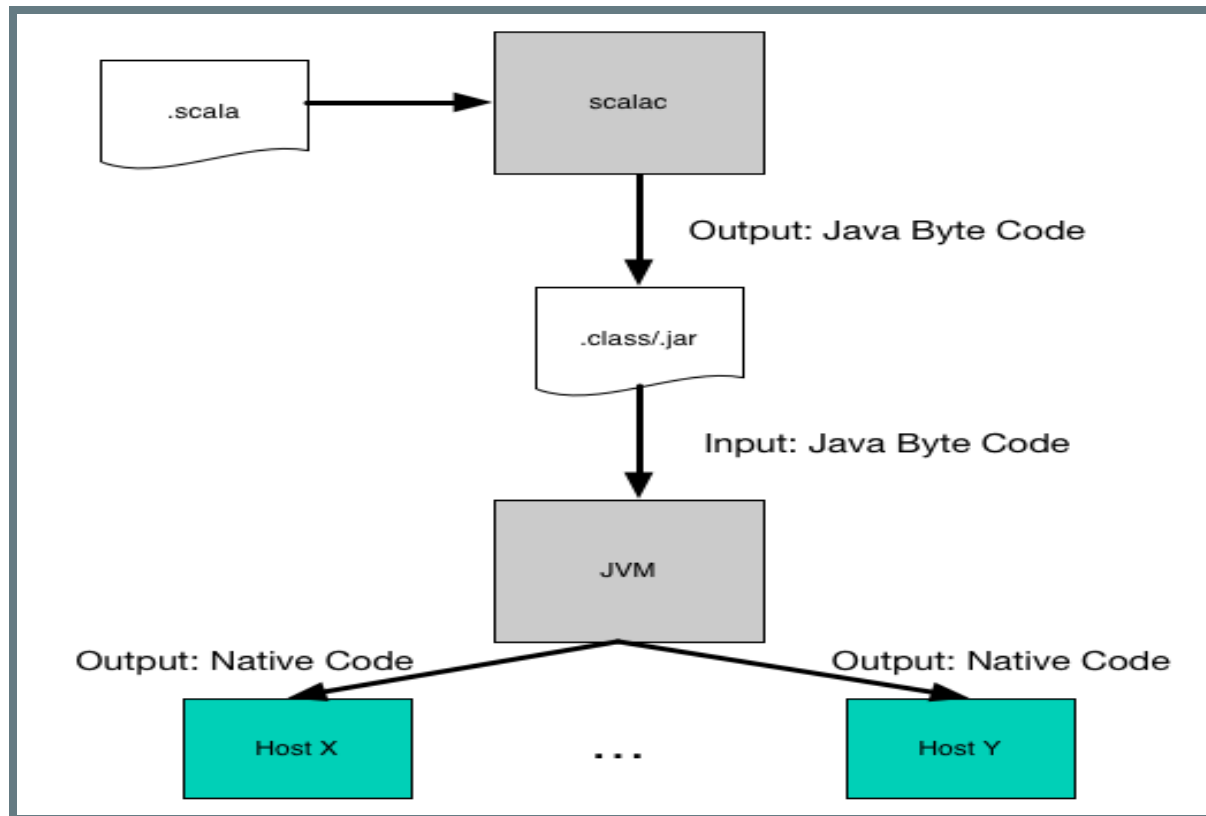
SBT - THE INTERACTIVE BUILD

- Tool to build Scala applications
- Interative
- Extendable
- Simple

SBT - SETUP

- Main website: http://www.scala-sbt.org/
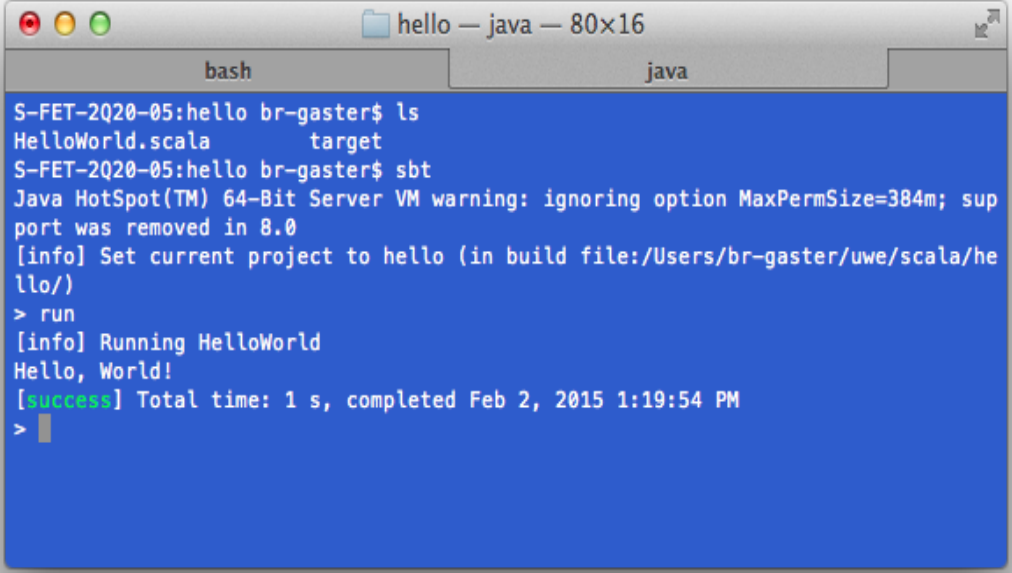- Download and install SBT binaries for your system
- Again you will need a text editor (e.g. edit, vi, emacs, ...)

BASIC SBT - HELLO WORLD AGAIN

- Change to directory with HelloWorld.scala
- Run the command sbt and then the command run

BASIC SBT - HELLO WORLD AGAIN

BASIC SBT

- Standard convention
    - Sources in the base directory (not really a good idea)
    - Sources in src/main/scala or src/main/java

BUILD DEFINITION FILE

- Most projects require some manual setup
- Basic build settings go in build.sbt in project root

EXMAPLE BUILD DEFINITION FILE

```scala
lazy val root = (project in file(".")).
  settings(
    name            := "helloworld",
    version         := "1.0",
    scalaVersion    := "2.11.5",
    organization    := "com.bendict.gaster",
  )
```

PROJECT DIRECTORY STRUCTURE

```
src/
  main/
    resources/
     files to include in main jar here
  scala/
     main Scala sources
  java/
     main Java sources
  test/
     resources
        files to include in test jar here
     scala/
        test Scala sources
     java/
        test Java sources
```

BASIC COMMANDS (INTERACTIVE)

- **compile** - complies application
- **run** - run application
- **Ctrl-D** (Unix/Max) - exit SBT
- **Ctrl-Z** (Windows) - exit SBT

FLATTEN CLASS FILES

- Standard Scala Class File cannot run with java
- Use SBT plugin assembler to address this

ASSEMBLER PLUGIN

- Create directory project in root
- Create file project/project/assembly.sbt
- Open and add the following:

```
addSbtPlugin("com.eed3si9n" % "sbt-assembly" % "0.12.0")
```

- Now you have the new SBT command assembly
- Try It!

ASSEMBLER PLUGIN

- SBT builds into direcotry ./target/scala-2.11
- To run resulting JAR file:

```
java -cp $SCALA_HOME/lib/scala-library.jar -jar helloworld-assembly-1.0.jar
```

SCALA INTRODUCTION REPOSITORY

- A small but growing set of examples can be found here:
  - https://github.com/bgaster/scala-intro
  - Try and have these all build and running by next week

FURTHER READING TUTORIALS

Try to complete at least these two before next week

- http://scalatutorials.com/tour/
- http://www.scala-sbt.org/0.13/tutorial/sbt-tutorial.pdf
- Many more online