

# SCALA

## A FUNCTIONAL AND OBJECT-ORIENTED PROGRAMMING LANGUAGE

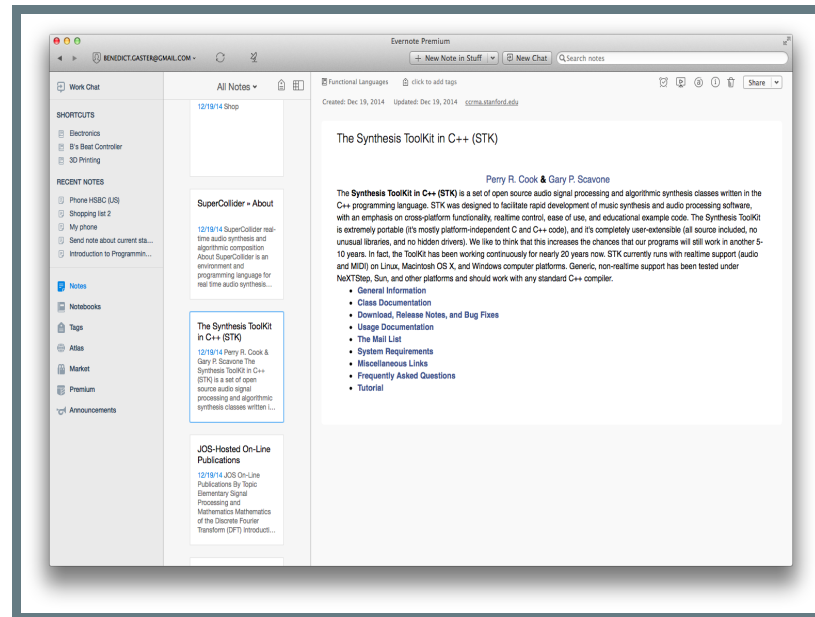
Benedict R. Gaster / @cuberoo\_

# 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

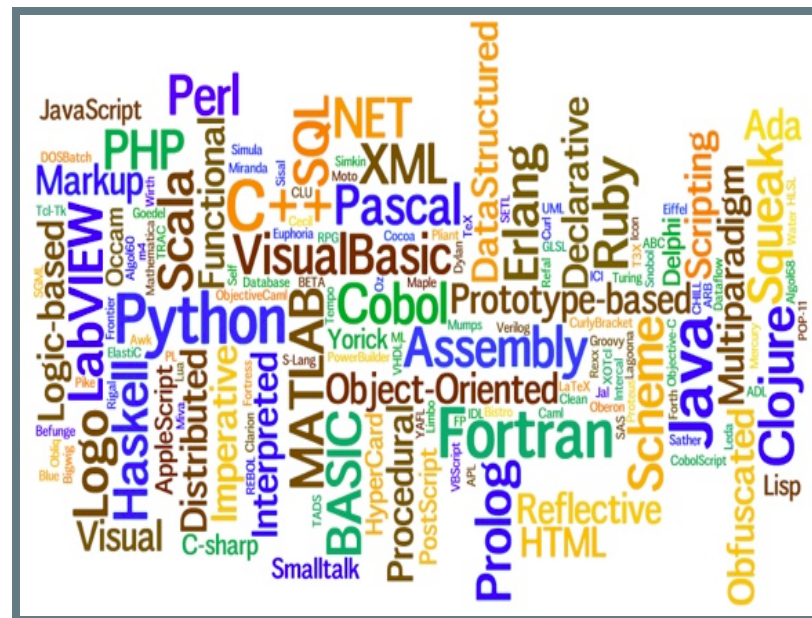
# PROGRAMMING

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

# BEGIN PROGRAMMING

First off, one must learn a programming language

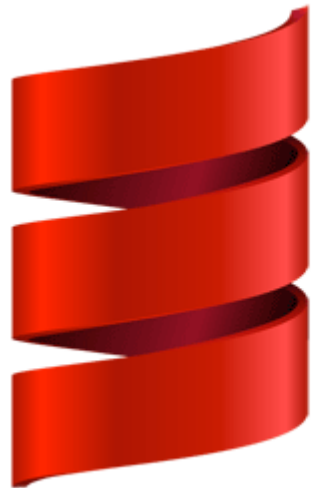
# WHICH PROGRAMMING LANGUAGE?



# 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



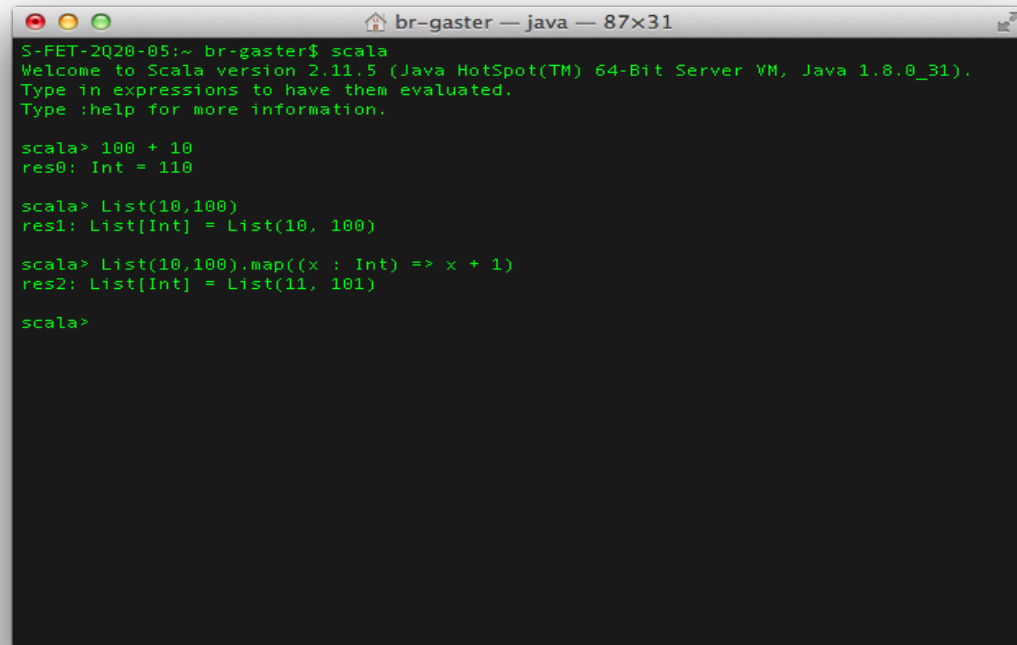
# SETUP

- 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



```
br-gaster — java — 87x31
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)

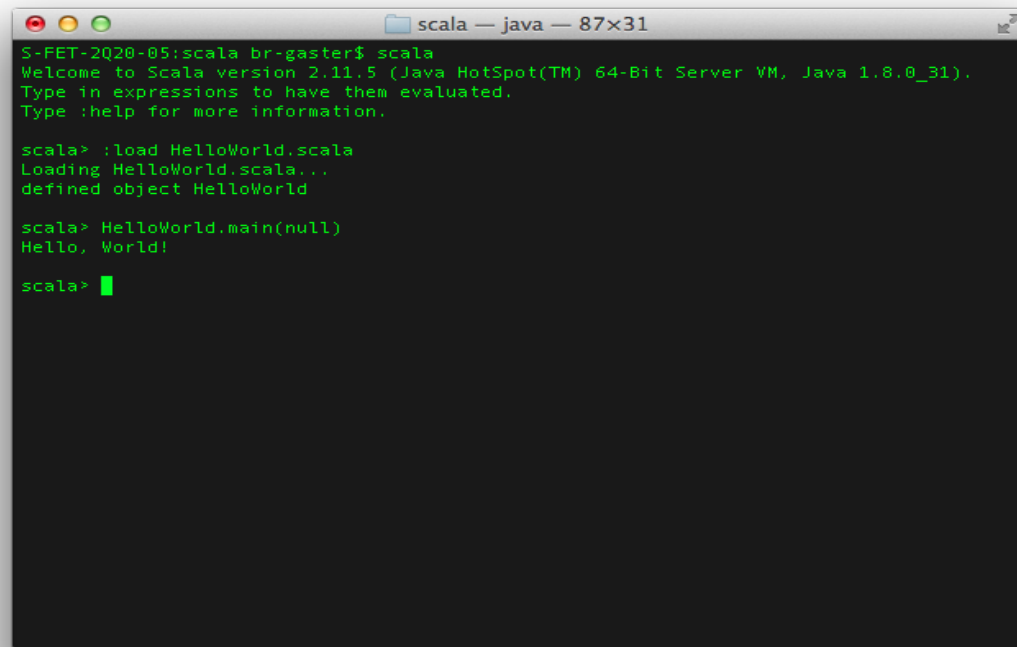
scala>
```

# 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 Scala REPL window titled "scala — java — 87x31". The window shows the following text:

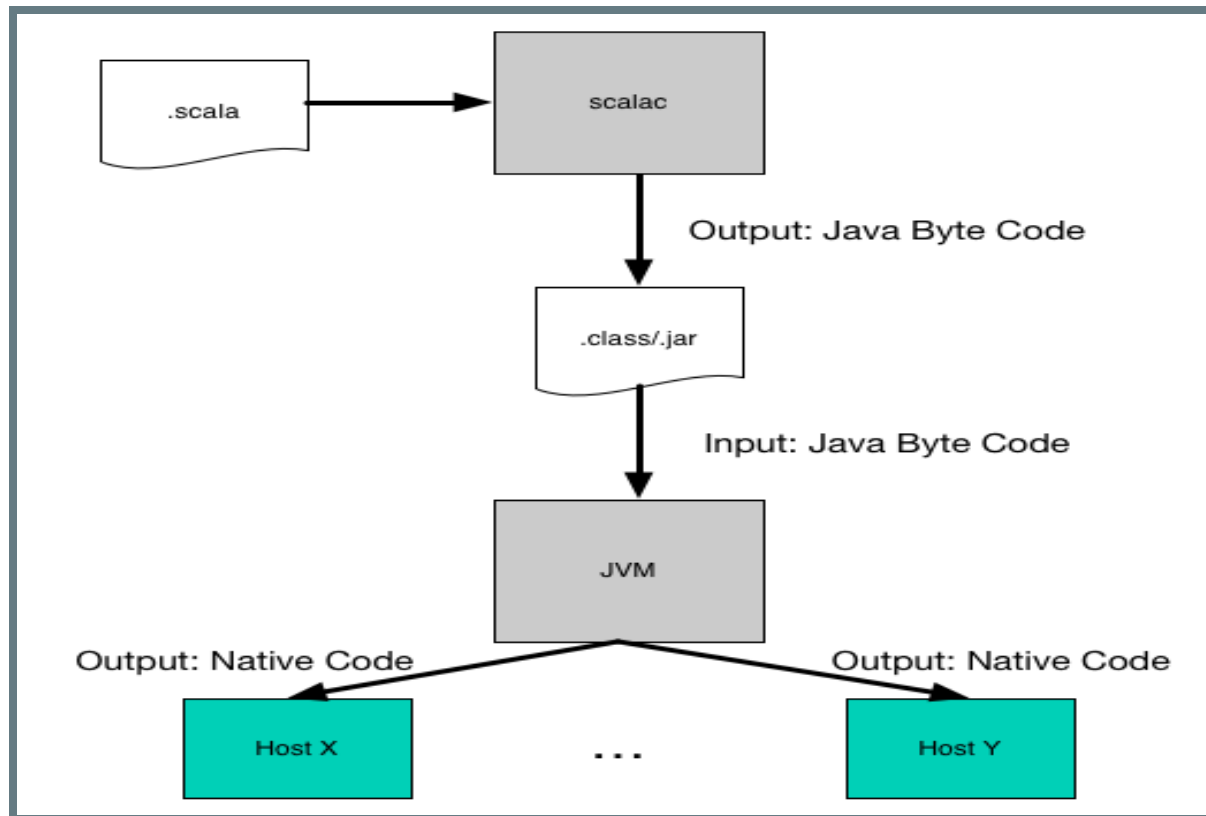
```
S-FET-2020-05:scala br-gaster$ scala
Welcome to Scala version 2.11.5 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_31).
Type in expressions to have them evaluated.
Type :help for more information.

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

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

scala> █
```

# SCALA COMPILATION FLOW



# SBT - THE INTERACTIVE BUILD

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

# SBT - SETUP

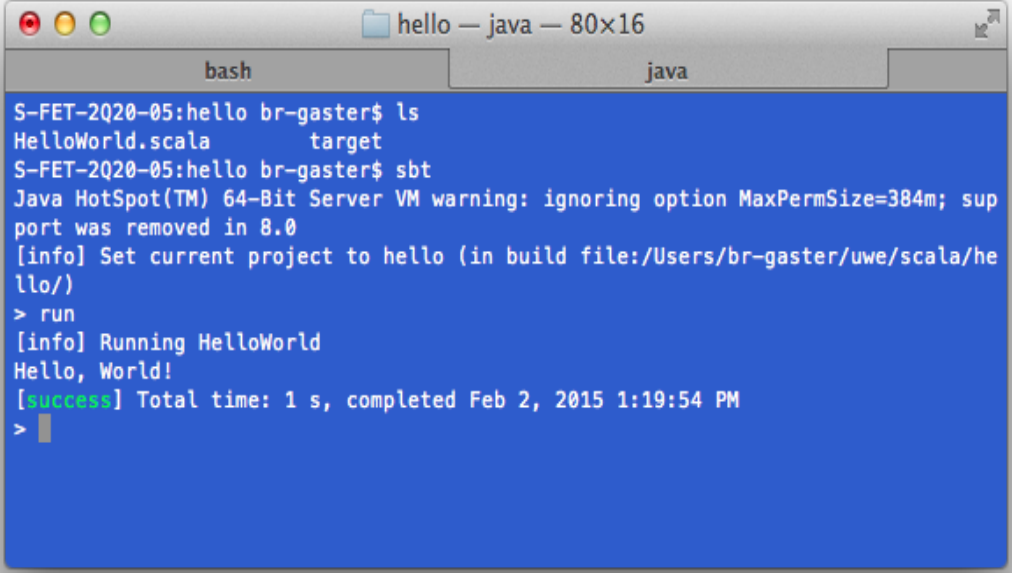
- Main website: <http://www.scala-sbt.org/>
- Download and install SBT binaries for your system
- Again you will need a text editor (e.g. edit, vi, emacs, ...)



# 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



```
hello — java — 80x16
bash      java
S-FET-2020-05:hello br-gaster$ ls
HelloWorld.scala      target
S-FET-2020-05:hello br-gaster$ sbt
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=384m; support was removed in 8.0
[info] Set current project to hello (in build file:/Users/br-gaster/uwe/scala/hello/)
> run
[info] Running HelloWorld
Hello, World!
[success] Total time: 1 s, completed Feb 2, 2015 1:19:54 PM
> 
```

# 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

```
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** - compiles 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