CS320

Introduction to Scala

Jaemin Hong

August 31, 2020

What is Scala?

Scala stands for "**Sca**lable **La**nguage."

- Statically typed
- Functional
- Object-oriented
- Java-interoperable

To study in detail, refer to the book "Programming in Scala."

- Available online (`https://www.artima.com/pins1ed/`)

Why Should I Learn Scala?

Is Scala popular? Maybe not...

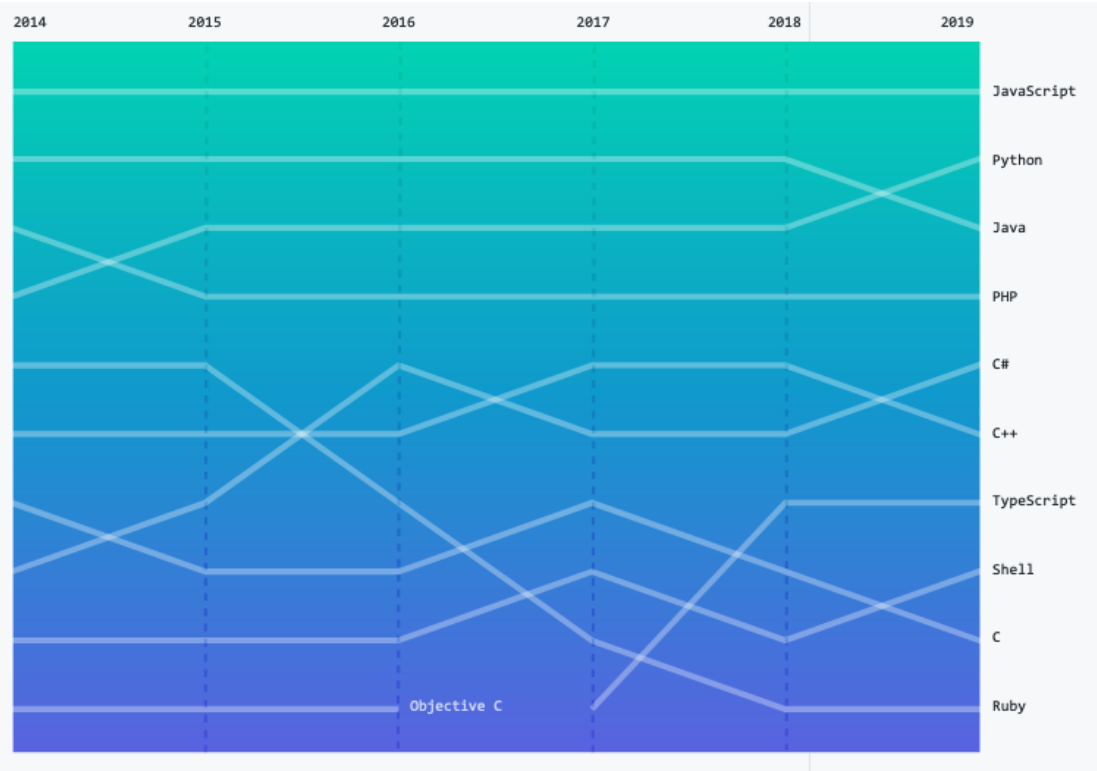
- 28th (TIOBE index, March 2020)
  https://www.tiobe.com/tiobe-index/

Position	Programming Language	Ratings
21	SAS	0.70%
22	Scratch	0.69%
23	D	0.68%
24	Dart	0.60%
25	Transact-SQL	0.53%
26	COBOL	0.48%
27	ABAP	0.45%
28	Scala	0.42%

Why Should I Learn Scala?

Is Scala popular? Maybe not...

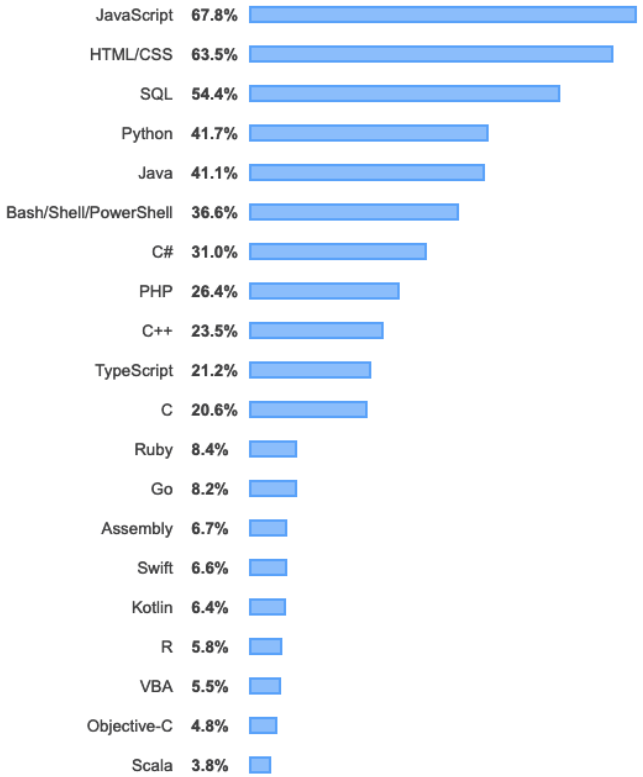
- Outside 10th (GitHub Octoverse, 2019)
  https://octoverse.github.com/#top-languages



Why Should I Learn Scala?

Is Scala popular? Maybe not...

- 20th (Stackoverflow survey, 2019) `https://insights.stackoverflow.com/survey/2019`



Why Should I Learn Scala?

Is Scala popular? Maybe yes!

- In the industry:
  Twitter, Gilt, Foursquare, Coursera, Apple, Guardian, New York Times, The Huffington Post, UBS, LinkedIn, Meetup, Milk, Verizon, Airbnb, Zalando, SoundCloud, Databricks, Morgan Stanley, Google, Walmart, Duolingo, HMRC, $\cdots$ are using Scala.
  `https://en.wikipedia.org/wiki/Scala_(programming_language)#Companies`

Why Should I Learn Scala?

Is Scala popular? Maybe yes!

- In academia:
  ACM SIGPLAN Programming Languages Software Award (2019) `https://www.sigplan.org/Awards/Software/`

Recipients

Scala 2019

Citation:

> Scala is one of the few programming languages from academia that has had a significant impact on the world as well as on programming languages research. It enjoys significant industrial adoption, with early adopters like Twitter and LinkedIn serving as catalysts; it forms the basis of the widely used Apache Spark data analytics platform. Scala's impact on PL research includes the idiom of implicits, which have found their way into other languages; attempts to formalize Scala's type system have pushed the boundaries of type systems research, culminating in the Dependent Object Types (DOT) calculus. In addition, Scala has been a fertile research ground for m etaprogramming, macros, staging, and embedded domain-specific languages, including DSLs for machine learning and GPU execution (Delite and OptiML).

Why Should I Learn Scala?

Is Scala popular?

Yes, Scala is popular for specific domains:

- processing highly structued data, such as programs
- parallel computing, such as large-scale data processing and web servers

because of its functional features:

- immutability
- first-class functions
- ADTs and pattern matching

Why Should I Learn Scala?

Is Scala popular?

Yes, Scala is popular for specific domains:

- **processing highly structued data, such as programs**
- parallel computing, such as large-scale data processing and web servers

because of its functional features:

- immutability
- first-class functions
- ADTs and pattern matching

We are going to write interpreters in Scala.

Scala vs Python

	Scala	Python
how to run	compiler	interpreter
type system	static	dynamic
functional?	○	△
immutable?	mostly	rarely
first-class functions	○	○
ADTs and pattern matching	○	×
object-oriented?	○	○

You need a different apporach to programming!

Installation

1. Install JDK 8.
   - `https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html`
   - For JDK compatability, refer to `https://docs.scala-lang.org/overviews/jdk-compatibility/overview.html`
   - To check: `$ java -version`
2. Install Scala 2.13.
   - `https://www.scala-lang.org/download/`
   - To check: `$ scala -version`
3. Install SBT 1.3.
   - `https://www.scala-sbt.org/download.html`

REPL

REPL stands for "**R**ead-**E**val-**P**rint-**L**oop."

1. Read: read input.
2. Eval: evaluate the input, and get the result.
3. Print: print the result.
4. Loop: go to the first step.

Type `scala` in your terminal to start the REPL.

```
scala
Welcome to Scala 2.13.1 (Java HotSpot(TM) 64-Bit Server VM, Java 11.0.6).
Type in expressions for evaluation. Or try :help.

scala>
```

REPL

Evaluate `1 + 1` in the REPL.

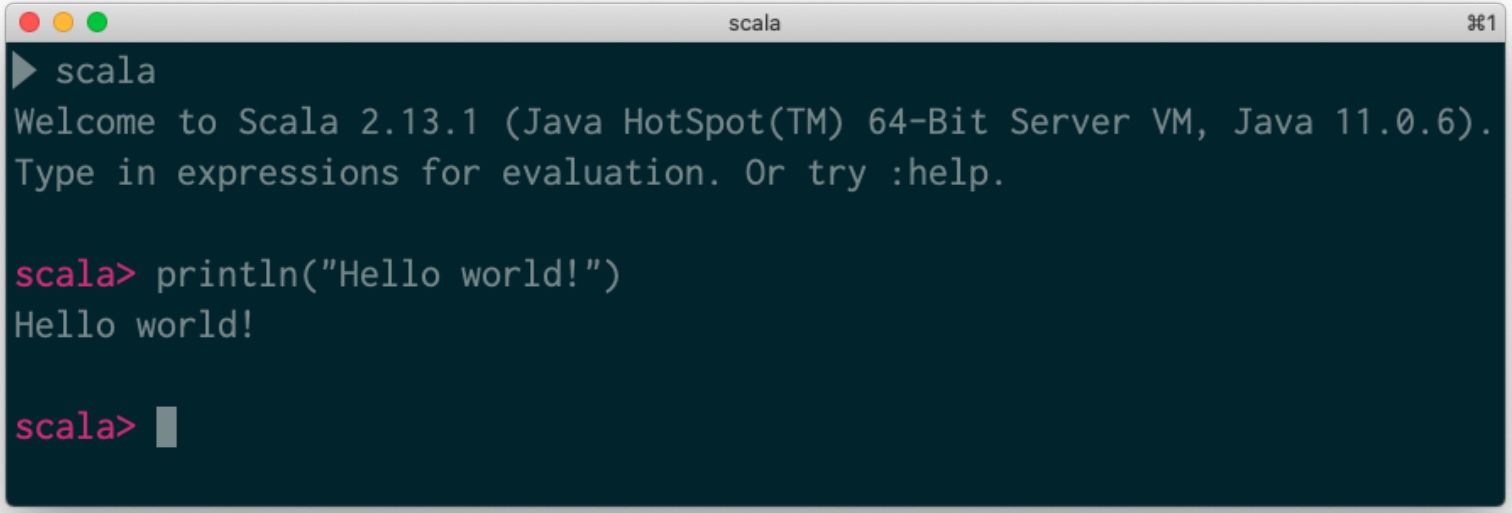
1. Read: read `1 + 1`.
2. Eval: evaluate `1 + 1`, and get 2.
3. Print: print 2.
4. Loop: go to the first step.

```
scala
Welcome to Scala 2.13.1 (Java HotSpot(TM) 64-Bit Server VM, Java 11.0.6).
Type in expressions for evaluation. Or try :help.

scala> 1 + 1
res0: Int = 2

scala>
```

REPL

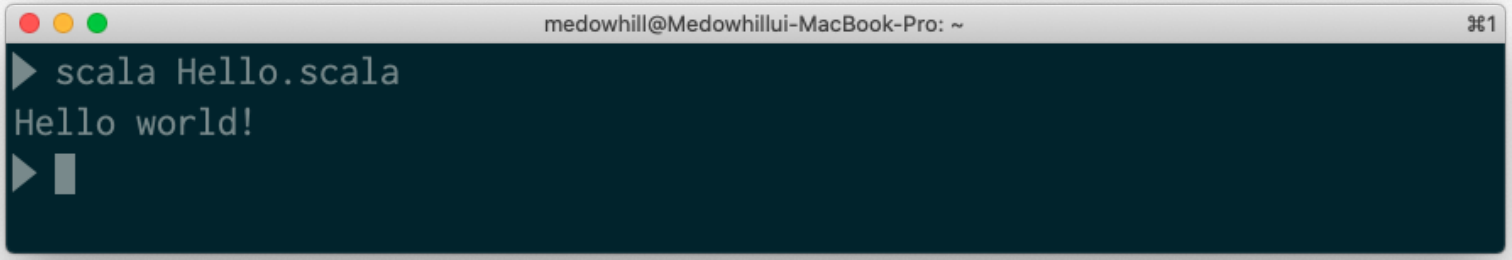
Evaluate `println("Hello world!")` in the REPL.

1. Read: read `println("Hello world!")`.
2. Eval: evaluate `println("Hello world!")`, print `"Hello world!"` with a line break, and get `()`.
3. Print: print nothing (`()` is not printed).
4. Loop: go to the first step.



Writing a Scala Program

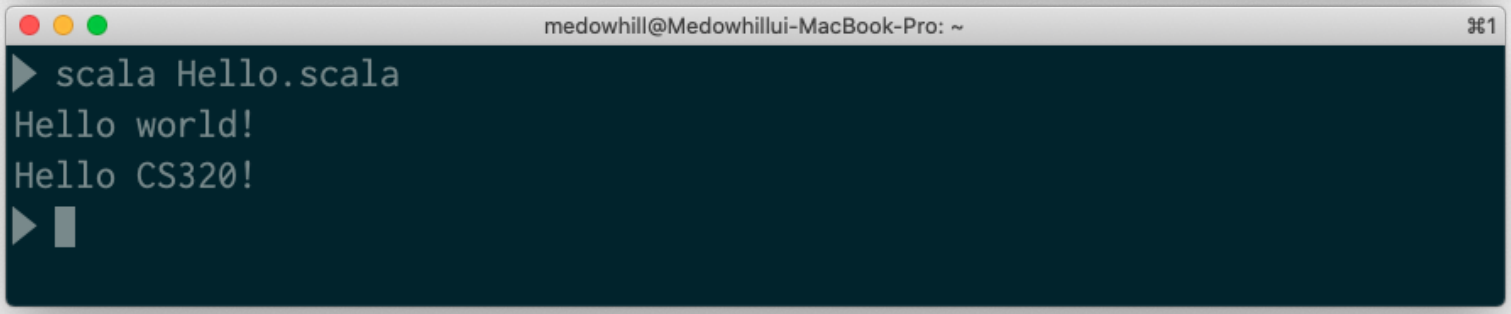
1. Open your favorite text editor.
2. Write `println("Hello world!")` in a new file.
3. Save the file as `Hello.scala`.
4. Type `scala Hello.scala` in your terminal.



Writing a Scala Program

You can write multiple lines.

1. Open `Hello.scala`.
2. Add `println("Hello CS320!")` after the existing line.
3. Save the file.
4. Type `scala Hello.scala` in your terminal.



Writing a Scala Program

Why it takes so long?

1. You type `scala Hello.scala` in your terminal.
2. The Scala compiler compiles the code and generates class files.
3. The JVM load the class files and runs the program.
4. You see `Hello world!\nHello CS320!`.
5. The class files are removed.

The second step is needed only once even if you execute the program multiple times.

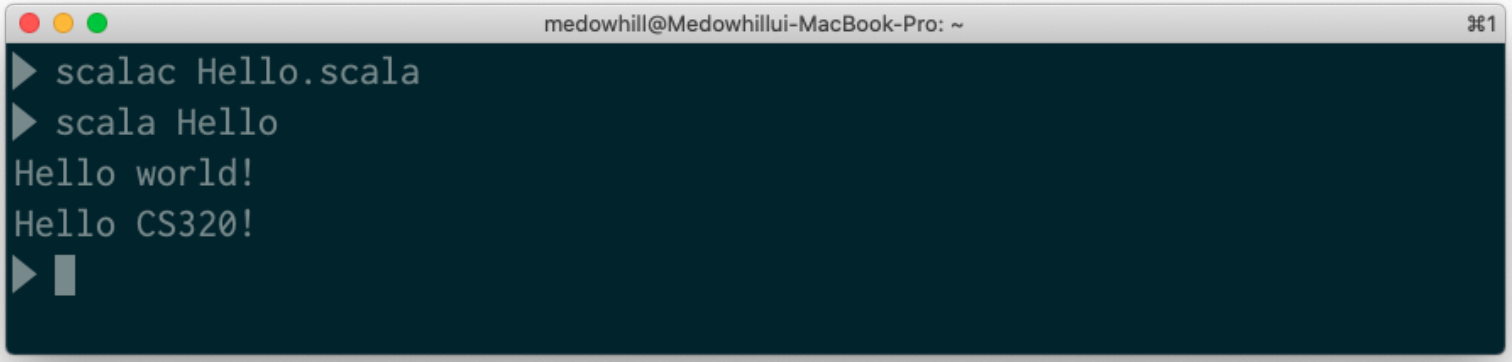
Compiling a Scala Program

```
object Hello {
  def main(args: Array[String]): Unit = {
    println("Hello world!")
    println("Hello CS320!")
  }
}
```

1. Open `Hello.scala`.
2. Change the content of the file into the above code.
3. Save the file.
4. Type `scalac Hello.scala` in your terminal.
5. Type `scala Hello` in your terminal.

Compiling a Scala Program

- The command `scalac Hello.scala` compiles the program.
- `Hello.class` and `Hello$.class` are the generated class files.
- The command `scala Hello` runs the program.
- The execution is much faster than before.
- For future executions, you need only `scala Hello`.



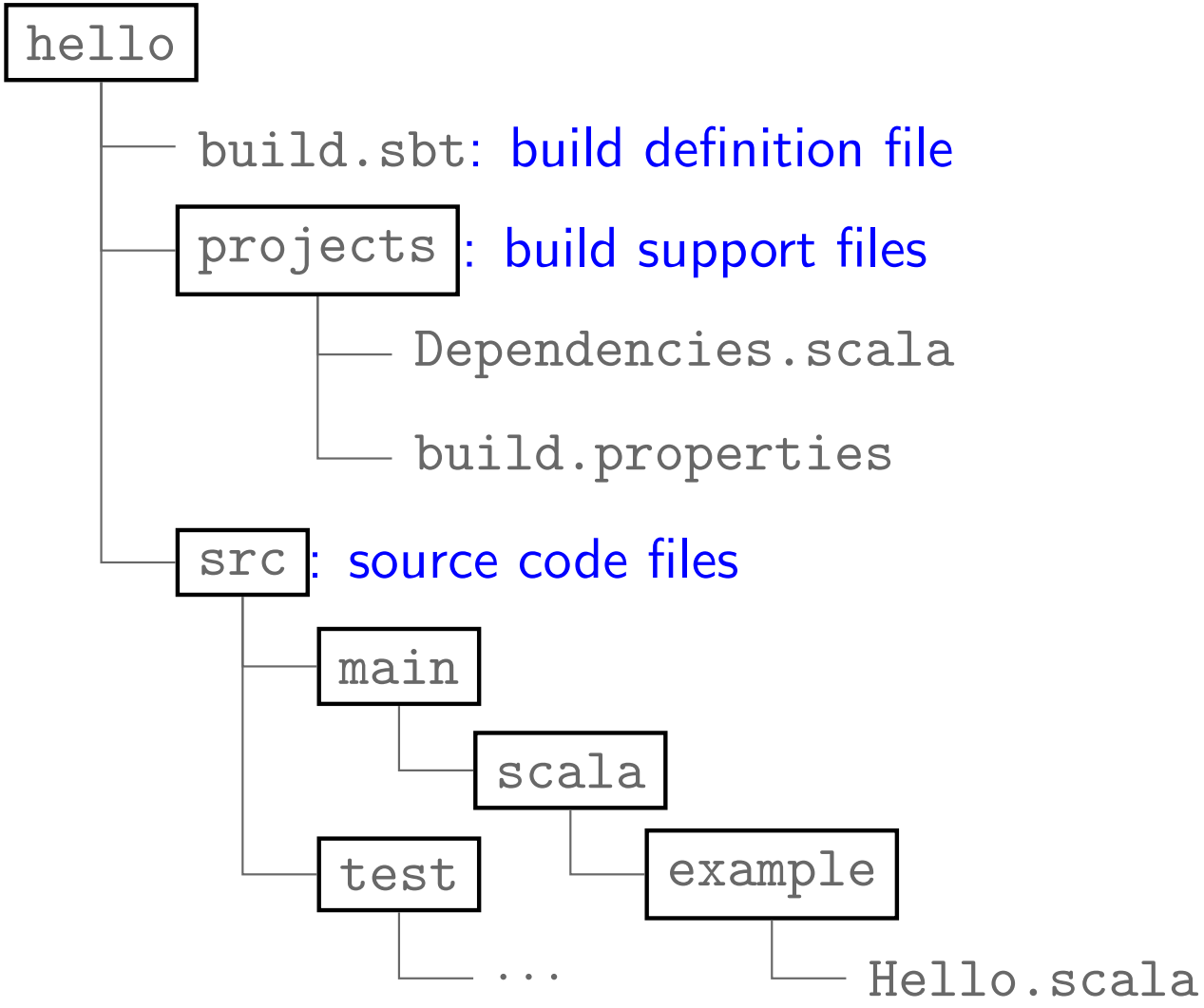
Creating a Scala Project

Use SBT (**S**imple **B**uild **T**ool).

1. Type `sbt new scala/scala-seed.g8` in your terminal.
2. Choose `Hello` as the name of your project.
3. Change the current directory to `hello`.

```
medowhill@Medowhillui-MacBook-Pro: ~/hello                    ⌘1
▶ sbt new scala/scala-seed.g8
[info] Loading global plugins from /Users/medowhill/.sbt/1.0/plugins
[info] Set current project to medowhill (in build file:/Users/medowhill/)
[info] Set current project to medowhill (in build file:/Users/medowhill/)

A minimal Scala project.

name [Scala Seed Project]: Hello

Template applied in /Users/medowhill/./hello

▶ cd hello
▶ 
```

Creating a Scala Project



Creating a Scala Project

```
package example

object Hello {
  def main(args: Array[String]): Unit = {
    println("Hello world!")
    println("Hello CS320!")
  }
}
```

1. Open `src/main/scala/example/Hello.scala`.
2. Change the content of the file into the above code.
3. Save the file.
4. Type `sbt` in your terminal to start an SBT server.

Creating a Scala Project

Useful SBT commands:

- `compile`: compile the program.
- `run`: compile (if necessary) and run the program.
- `test`: compile (if necessary) and run test cases.
- `console`: start the REPL.
- `exit`: terminate the server.
- `~[cmd]`: continuously execute a given command. For example,
  - `~compile`
  - `~run`
  - `~test`

Creating a Scala Project

```
sbt
[info] Loading global plugins from /Users/medowhill/.sbt/1.0/plugins
[info] Loading project definition from /Users/medowhill/hello/project
[info] Loading settings for project root from build.sbt ...
[info] Set current project to Hello (in build file:/Users/medowhill/hello
/)
[info] sbt server started at local:///Users/medowhill/.sbt/1.0/server/917
fd08bbaff396a8dcd/sock
sbt:Hello> compile
[info] Compiling 1 Scala source to /Users/medowhill/hello/target/scala-2.
13/classes ...
[success] Total time: 3 s, completed 2020. 3. 15. 오후 9:15:32
sbt:Hello> run
[info] running Hello
Hello world!
Hello CS320!
[success] Total time: 0 s, completed 2020. 3. 15. 오후 9:15:33
sbt:Hello>
```

Jaemin Hong
jaemin.hong@kaist.ac.kr
https://hjaem.info