Scala:
The Language of Pragmatism

-Vasu
-Brian
Why a new language?

- Goal was to create a language with better support for component software
- Two hypotheses:
  1. Programming language for component software should be scalable
     - The same concepts describe small and large parts
     - Rather than adding lots of primitives, focus is on abstraction, composition, and decomposition
  2. Language that unifies OOP and functional programming can provide scalable support for components
- Adoption is key for testing this hypothesis
- Scala interoperates with Java and .NET
History

- Scala was developed in 2001 by Martin Odersky at École Polytechnique Fédérale de Lausanne.
- Scala was released late 2003/early 2004 on the Java platform, and on the .NET platform in June 2004.
- Scala runs on all Java Machines.
- It also runs on Android Smartphones.
Properties of Scala

- Object-Oriented
- Functional
- Scalable
- Statically Typed
- Executable
- Productive
- Interoperable with Java
- Open Source

```scala
1. import java.util.{Date, Locale}
2. import java.text.SimpleDateFormat
3. import java.text.SimpleDateFormat._

4.
5. object FrenchDate {
6.   def main(args: Array[String]) {
7.     val now = new Date
8.     val df = getDateTime(LONG, Locale.FRANCE)
9.     println(df.format(now))
10.   }
11. }
```
Practical Applications

- Twitter: Backend and Frontend production code
- Linkedin
- Novell
- Foursquare: Almost 100% build in Scala
- UBS
- HSBC
- Web servers
Scala VS Python

Scala: Has the Java libraries
Scala: Very fast only 2 to 3 times slower than C
Scala: Very advanced type system
Scala: Scala is the most adapted functional language
Python: Easiest language to learn
Python: Very Minimal Language
Python: Most people pick it
Examples in Scala

- Hello World
- Writing a bash script
- Stack Implementation
- HTTP Web Server - (commercial Implementation - tiscaf)
- Lift Web framework
Features

- Scala is both functional and object-oriented
  - every value is an object
  - every function is a value—including method
- Scala is statically typed

in Java 1.5:

```java
Pair p = new Pair<Integer, String>(1, "Scala");
```

in Scala:

```scala
val p = new MyPair(1, "scala");
```
More Comparisons

Clojure (both are JVM languages)

- Java frameworks not easy with clojure
- Scala has edge when it comes to featured web frameworks
- Scala uses strong types and is influenced by ML and Haskell
- Lazy initialization in scala
- Clojure has good multi core concurrency that is pretty invisible to the developer
Comparisons contd..

- Ruby
  - Long living processes
  - Ruby's poor garbage collection - uses mark sweep garbage collection algorithm which slowly leaks memory.
  - How about putting ruby on a JVM - JRuby?
Comparison contd..

- **Java**
  - Scala misses control structures like continue and break
  - Scala Keywords not in Java, eg override notation is necessary in scala
  - enhanced for loops in scala
  - Arrays accessed by parenthesis (why did they do this?)
  - Java you need to take care of initializations, they should be in sequence however in scala any code not a part of method or member variable declaration is executed a sprimary constructor
Quiz

- Major difference in ruby and scala?
- What is lazy initializations in Scala?
- Can you write a bash script in scala?
- Can you embed java code into a scala program?
- What platform does scala work/compiles on?
Questions?