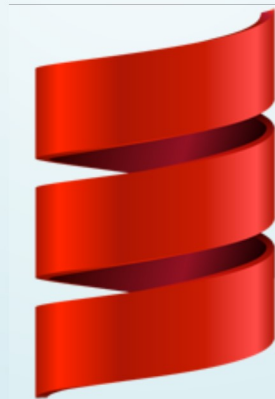


SINGLE PAGE APPLICATION DEVELOPMENT WITH SCALA.JS



WHAT ARE WE GOING TO TALK ABOUT?

- What is Scala
 - For the people that didn't go to Scala 101.
- What is Scala.js
- Why fullstack Type-safe development is desirable
- Tool Chain
- Javascript Library Facades
- Helpful Libraries
- Example app 'Live-Coding'

WHAT IS
SCALA?



```
class Point(xc: Int, yc: Int) {
  var x: Int = xc
  var y: Int = yc
  def move(dx: Int, dy: Int) {
    x = x + dx
    y = y + dy
  }
  override def toString(): String = "(" + x + ", " + y + "
```

PARADIGM

IMPERATIVE

```
def fib( n : Int ) : Int = {  
    var a = 0  
    var b = 1  
    var i = 0  
  
    while( i < n ) {  
        val c = a + b  
        a = b  
        b = c  
        i = i + 1  
    }  
    return a  
}
```

PARADIGM

FUNCTIONAL

```
def fib2( n : Int) : Int = {  
    @scala.annotation.tailrec // Optional  
    def fib_tail( n: Int, a:Int, b:Int): Int = n match {  
        case 0 => a  
        case _ => fib_tail( n-1, b, a+b )  
    }  
    return fib_tail( n, 0, 1)  
}
```

`fib_tail` becomes a `while` loop in the bytecode!

TYPING DISCIPLINE

```
val foo: Int = 10 // Statically Checked!
val bar: String = 10 /* This FAILS */ // Strongly Typed
val bleh = 10 // Types can be inferred!

// Structural Typing!
// (probably shouldn't do this...)
// (Uses JVM reflection so ends up being slow)
def quacker(duck: { def quack: String }) =      println(duck.quack)

quacker(new { def quack = "QUACK!" })

class Goose { def quack = "HONK!" }
quacker(new Goose())
```

PLATFORMS

- Java Virtual Machine (Scala)
 - Windows/Linux/Mac
 - x86(-64), Arm and more
 - Android
 - iOS via BugVM
 - .Net via IKVM
- Javascript (Scala.js)
 - NodeJS
 - Any JS compatible browser
- Bare Metal (Scala-native)
 - Any LLVM Target

WHAT IS SCALA.JS

SCALA.JS

- Scala in the Browser!
- Scala on Node
- Javascript Interopt

WHY FULL STACK SCALA?

TYPICAL WEB STACK

- Client
 - Javascript
 - HTML
 - CSS
- Server
 - PHP
- Database
 - SQL

'COMPILED' WEB STACK

- Client
 - Javascript
 - HTML
 - CSS
- Server
 - C#
- Database
 - ORM -> SQL
 - EntityFramework

SAFER WEB STACK

- Client
 - Typescript
 - HTML
 - CSS
- Server
 - C#
- Database
 - ORM -> SQL
 - EntityFramework

SAFERER WEB STACK

- Client
 - Typescript
 - React
 - CSS
- Server
 - C#
- Database
 - ORM -> SQL
 - EntityFramework

SAFEST WEB STACK

- Client
 - Scala.js
 - Scala-tags/Scala React
 - Scala-CSS
- Server
 - ScalaJVM
- Database
 - ORM -> SQL
 - Slick/Quil/Scala Relational

SO... WHY?

- Push unsafe operations to the periphery of your application.
- Share Code between client and server
 - Models (POJOs)
 - Interfaces
 - Validation Logic
 - Templates
- Keep the tool chain simple

TOOLS/IDES

- IDE's
 - Eclipse
 - Idea IntelliJ <- Recommended
 - Netbeans
- Build scripts
 - Maven
 - Apache Buildr
 - Gradle
 - SBT (Simple/Scala Build Tool) <- Recommended
- Scala-SDK
 - scala/scalac
- scala-js-fiddle.com

SCALA.JS LIMITATIONS

- No Java
- No Java Standard Library
- No Java Libraries
- Must use scala libraries compiled for Scala.js
- Must use pure scala code

PERFORMANCE

- fastOptJS
- fullOptJS

JAVASCRIPT LIBRARY FACADES

HELPFUL LIBRARIES

- HTML
 - Scala-Tags
 - Scala-Reactjs
 - SRI
 - WIDOK
- CSS
 - Scala-css

HELPFUL LIBRARIES

- Database
 - Slick
 - Scalarelational
 - Quil

HELPFUL LIBRARIES

- Server
 - Play
 - Scalatra (Like rubys Sinatra)
 - LIFT
 - Akka-HTTP (formerly Spray.IO)
- RPC
 - Autowire
 - Udash RPC

LIVE CODING

RESOURCES

- coursera.org/specializations/scala
- scala-js-fiddle.com
- docs.scala-lang.org
- scala-exercises.org
- twitter.github.io/scala_school
- twitter.github.io/effectivescala
- Programming in Scala Book:
 - artima.com/pins1ed
- scalatutorials.com/tour
- scala-tour.com
- ktoso.github.io/scala-types-of-types
- Scala Times
 - scalatimes.com