X-Platform Development in Scala.js

Li Haoyi

9 August 2014

Scala by the Bay

What is Scala.js?

- Scala to JavaScript, run in browser
- Share code client/server!
- Get typechecking in your web apps!
- 1-4x slower than JS, 10x slower than Scala-JVM, 2-6x faster than Python

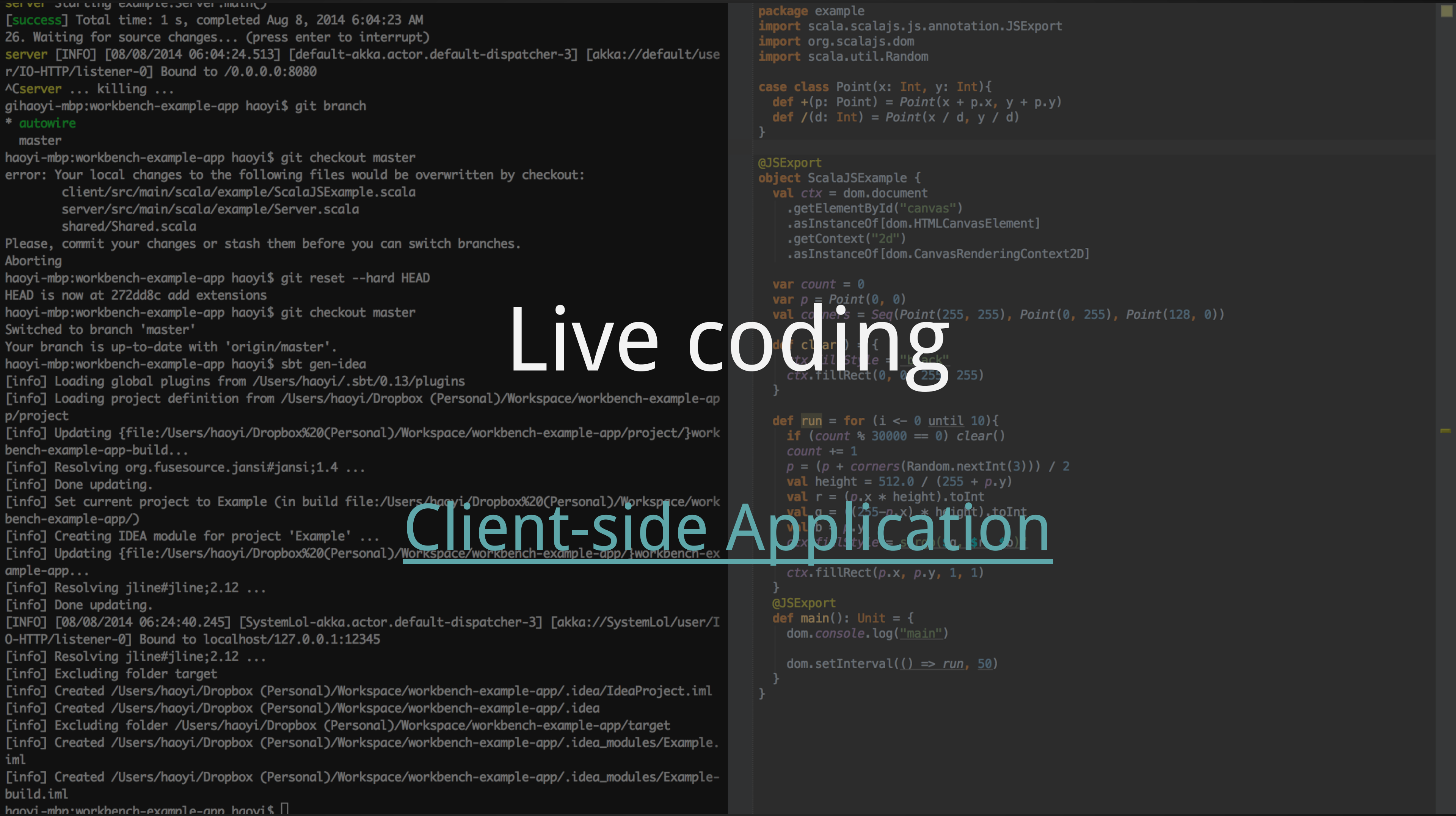
What’s wrong with (my) JS?

- “Who is doing this?”
- “Where did this variable come from?”
- “Why is it undefined?”
- “Why is renaming this method so hard =(”
- “I want to refactor this but I’m scared!”

- “WTF is going on -.-”

Live coding

Client-side Application



```
server Starting example.Server.main()
[success] Total time: 1 s, completed Aug 8, 2014 6:04:23 AM
26. Waiting for source changes... (press enter to interrupt)
server [INFO] [08/08/2014 06:04:24.513] [default-akka.actor.default-dispatcher-3] [akka://default/use
r/IO-HTTP/listener-0] Bound to /0.0.0.0:8080
^Cserver ... killing ...
gihaoyi-mbp:workbench-example-app haoyi$ git branch
* autowire
  master
haoyi-mbp:workbench-example-app haoyi$ git checkout master
error: Your local changes to the following files would be overwritten by checkout:
        client/src/main/scala/example/ScalaJSExample.scala
        server/src/main/scala/example/Server.scala
        shared/Shared.scala
Please, commit your changes or stash them before you can switch branches.
Aborting
haoyi-mbp:workbench-example-app haoyi$ git reset --hard HEAD
HEAD is now at 272dd8c add extensions
haoyi-mbp:workbench-example-app haoyi$ git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.
haoyi-mbp:workbench-example-app haoyi$ sbt gen-idea
[info] Loading global plugins from /Users/haoyi/.sbt/0.13/plugins
[info] Loading project definition from /Users/haoyi/Dropbox (Personal)/Workspace/workbench-example-ap
p/project
[info] Updating {file:/Users/haoyi/Dropbox%20(Personal)/Workspace/workbench-example-app/project/}work
bench-example-app-build...
[info] Resolving org.fusesource.jansi#jansi;1.4 ...
[info] Done updating.
[info] Set current project to Example (in build file:/Users/haoyi/Dropbox%20(Personal)/Workspace/work
bench-example-app/)
[info] Creating IDEA module for project 'Example' ...
[info] Updating {file:/Users/haoyi/Dropbox%20(Personal)/Workspace/workbench-example-app/}workbench-ex
ample-app...
[info] Resolving jline#jline;2.12 ...
[info] Done updating.
[INFO] [08/08/2014 06:24:40.245] [SystemLol-akka.actor.default-dispatcher-3] [akka://SystemLol/user/I
O-HTTP/listener-0] Bound to localhost/127.0.0.1:12345
[info] Resolving jline#jline;2.12 ...
[info] Excluding folder target
[info] Created /Users/haoyi/Dropbox (Personal)/Workspace/workbench-example-app/.idea/IdeaProject.iml
[info] Created /Users/haoyi/Dropbox (Personal)/Workspace/workbench-example-app/.idea
[info] Excluding folder /Users/haoyi/Dropbox (Personal)/Workspace/workbench-example-app/target
[info] Created /Users/haoyi/Dropbox (Personal)/Workspace/workbench-example-app/.idea_modules/Example.
iml
[info] Created /Users/haoyi/Dropbox (Personal)/Workspace/workbench-example-app/.idea_modules/Example-
build.iml
haoyi-mbp:workbench-example-app haoyi$
```

```scala
package example
import scala.scalajs.js.annotation.JSExport
import org.scalajs.dom
import scala.util.Random

case class Point(x: Int, y: Int){
  def +(p: Point) = Point(x + p.x, y + p.y)
  def /(d: Int) = Point(x / d, y / d)
}

@JSExport
object ScalaJSExample {
  val ctx = dom.document
    .getElementById("canvas")
    .asInstanceOf[dom.HTMLCanvasElement]
    .getContext("2d")
    .asInstanceOf[dom.CanvasRenderingContext2D]

  var count = 0
  var p = Point(0, 0)
  val corners = Seq(Point(255, 255), Point(0, 255), Point(128, 0))

  def clear() = {
    ctx.fillStyle = "black"
    ctx.fillRect(0, 0, 255, 255)
  }

  def run = for (i <- 0 until 10){
    if (count % 30000 == 0) clear()
    count += 1
    p = (p + corners(Random.nextInt(3))) / 2
    val height = 512.0 / (255 + p.y)
    val r = (p.x * height).toInt
    val g = ((255-p.x) * height).toInt
    val b = p.y
    ctx.fillStyle = s"rgb($g, $r, $b)"

    ctx.fillRect(p.x, p.y, 1, 1)
  }
  @JSExport
  def main(): Unit = {
    dom.console.log("main")

    dom.setInterval(() => run, 50)
  }
}
```

Can/Can't Use

Can use

- Most of java.lang.*
- Almost all of scala.*
- Some of java.util.*
- Scala Macros: upickle, scala-async, scalaxy, etc
- Pure-Scala ecosystem: shapeless, scalaz, scalatags, utest

Can't use

- j.l.Thread, j.l.Runtime, ...
- s.c.parallel, s.tools.nsc
- org.omg.CORBA, sun.misc.*
- Reflection: scala-pickling, scala-reflect
- Java-dependent: Scalatest, Scalate

Can/Can't Use

Can use

- JS stuff: XmlHttpRequest, Websockets. Localstorage
- HTML DOM, Canvas, WebGL
- JavaScript libraries: chipmunk.js, hand.js, react.js, jquery
- IntelliJ, Eclipse, SBT
- Chrome console, firebug

Can't use

- JVM stuff: Netty, akka, spray, file IO, JNI
- AWT, Swing, SWT, OpenGL
- Java ecosystem: guice, junit, apache-commons, log4j
- Yourkit, VisualVM, JProfiler

Show & Tell

TodoMVC, Roll, Scala-Js-Fiddle, Ray-Tracer

Tutorial

On the right are some of the common types of materials and joints you will encounter in the game.

Roll your ball left and right using the arrow keys, and draw paths using the mouse or touchscreen for your ball to roll on. Esc restarts the current level, and pg up and pg down allow you to zoom in or out.

When ready, roll your ball to the bottom right corner of the level and hit the light blue "Goal" block to proceed to the next level.

Gusts

These pick up your ball, and, any other objects that end up within them, and push them around

You control a ball which you can roll left or right using the arrow keys

Touch the cyan-colored goal block to turn it yellow and progress to the next level

Joints and Shapes

The properties of each joint (left) or shape (right) is indicated by its color. The brighter the color, the stronger that property.

Mixed colors take on all the properties of their constituents e.g. purple shapes being both bouncy and rough.

Lasers

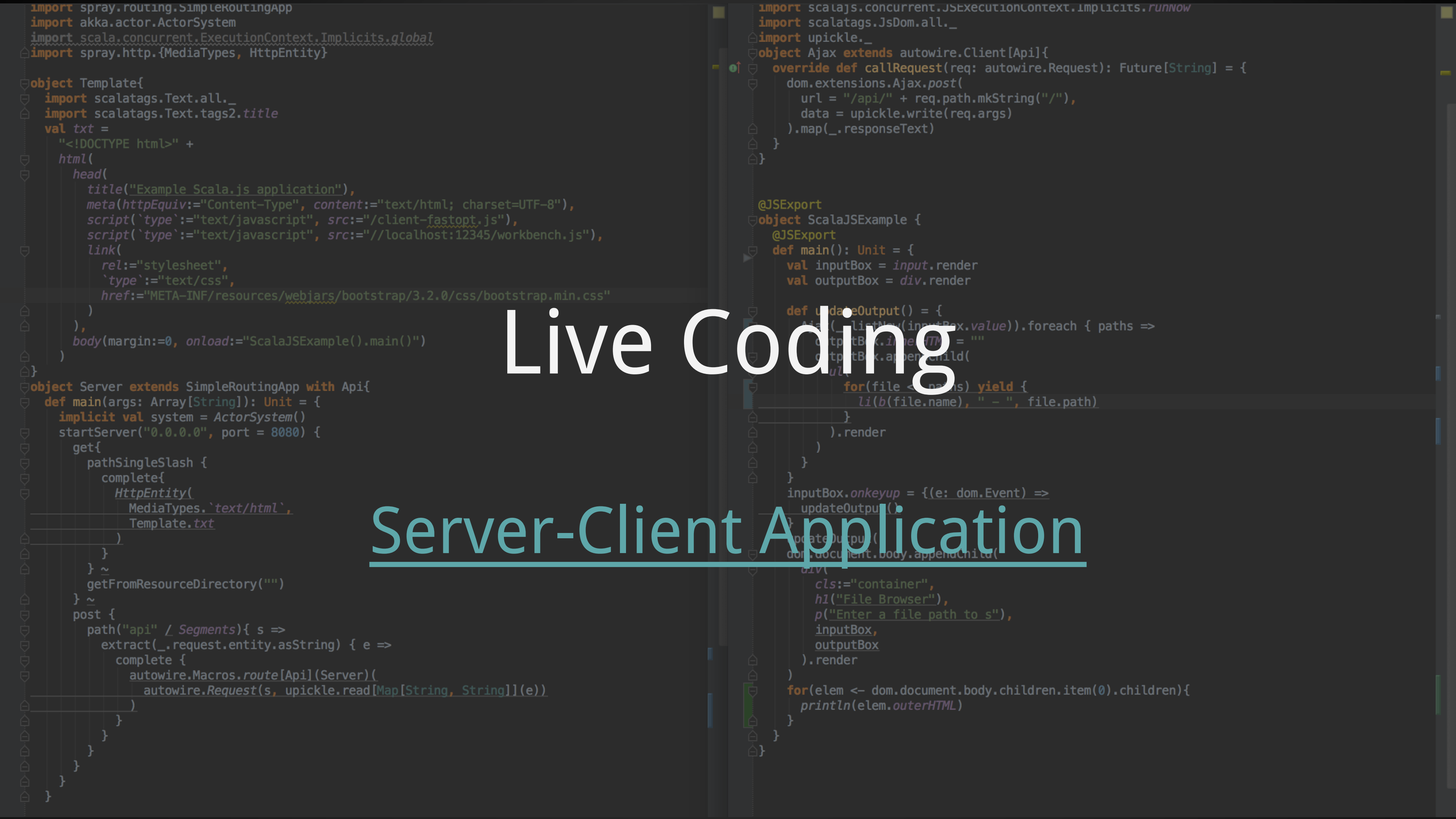
These destroy your ball if you touch them, forcing you to restart from your last check point. They can be blocked by drawn paths or other shapes

Why Scala.js

- Scala's great and JavaScript not so much
- Huge ecosystem of libraries and tools available for free (because Scala, and JS!)
- Web apps > Swing apps for deployment
- Front-end development in Scala is fun!

Live Coding

Server-Client Application



```
import spray.routing.SimpleRoutingApp
import akka.actor.ActorSystem
import scala.concurrent.ExecutionContext.Implicits.global
import spray.http.{MediaTypes, HttpEntity}

object Template{
  import scalatags.Text.all._
  import scalatags.Text.tags2.title
  val txt =
    "<!DOCTYPE html>" +
    html(
      head(
        title("Example Scala.js application"),
        meta(httpEquiv:="Content-Type", content:="text/html; charset=UTF-8"),
        script(`type`:="text/javascript", src:="/client-fastopt.js"),
        script(`type`:="text/javascript", src:="//localhost:12345/workbench.js"),
        link(
          rel:="stylesheet",
          `type`:="text/css",
          href:="META-INF/resources/webjars/bootstrap/3.2.0/css/bootstrap.min.css"
        )
      ),
      body(margin:=0, onload:="ScalaJSExample().main()")
    )
}
object Server extends SimpleRoutingApp with Api{
  def main(args: Array[String]): Unit = {
    implicit val system = ActorSystem()
    startServer("0.0.0.0", port = 8080) {
      get{
        pathSingleSlash {
          complete{
            HttpEntity(
              MediaTypes.`text/html`,
              Template.txt
            )
          }
        } ~
        getFromResourceDirectory("")
      } ~
      post {
        path("api" / Segments){ s =>
          extract(_.request.entity.asString) { e =>
            complete {
              autowire.Macros.route[Api](Server)(
                autowire.Request(s, upickle.read[Map[String, String]](e))
              )
            }
          }
        }
      }
    }
  }
}
```

```
import scalajs.concurrent.JSExecutionContext.Implicits.runNow
import scalatags.JsDom.all._
import upickle._
object Ajax extends autowire.Client[Api]{
  override def callRequest(req: autowire.Request): Future[String] = {
    dom.extensions.Ajax.post(
      url = "/api/" + req.path.mkString("/"),
      data = upickle.write(req.args)
    ).map(_.responseText)
  }
}


@JSExport
object ScalaJSExample {
  @JSExport
  def main(): Unit = {
    val inputBox = input.render
    val outputBox = div.render

    def updateOutput() = {
      Ajax[[illegible].listNow(inputBox.value)).foreach { paths =>
        outputBox.innerHTML = ""
        outputBox.appendChild(
          ul(
            for(file <- paths) yield {
              li(b(file.name), " - ", file.path)
            }
          ).render
        )
      }
    }
    inputBox.onkeyup = {(e: dom.Event) =>
      updateOutput()
    }
    updateOutput()
    dom.document.body.appendChild(
      div(
        cls:="container",
        h1("File Browser"),
        p("Enter a file path to s"),
        inputBox,
        outputBox
      ).render
    )
    for(elem <- dom.document.body.children.item(0).children){
      println(elem.outerHTML)
    }
  }
}
```

Scala.js

- Able to use strengths of each platform
- Sharing code/libraries/data-structures between client as server is awesome
- Static typing keeps things straight and keeps you sane

Cool Demos

- Shared libraries between client & server
- Auto-rename routes and Ajax calls!
- Find-usages for Ajax endpoints!
- Tons of Safety
- Tons of Toolability (and Tools!)

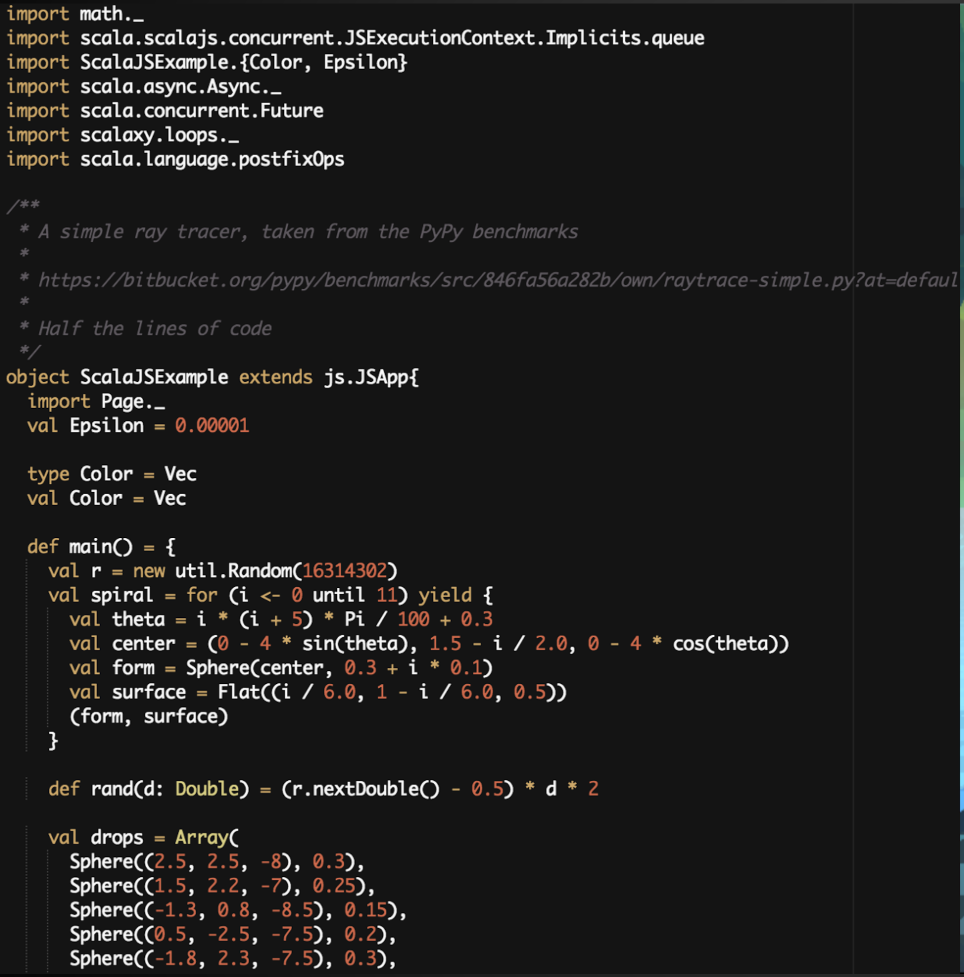
Conclusion

- X-Platform dev in Scala.js is awesome

- www.scala-js.org
  - Fork it, make cool stuff
  - Come hang out in the google group

- https://github.com/lihaoyi/workbench-example-app
  - master -> Client example
  - todomvc
  - raytracer
  - autowire -> Server-Client example

Questions?



```
import math._
import scala.scalajs.concurrent.JSExecutionContext.Implicits.queue
import ScalaJSExample.{Color, Epsilon}
import scala.async.Async._
import scala.concurrent.Future
import scalaxy.loops._
import scala.language.postfixOps

/**
 * A simple ray tracer, taken from the PyPy benchmarks
 *
 * https://bitbucket.org/pypy/benchmarks/src/846fa56a282b/own/raytrace-simple.py?at=defaul
 *
 * Half the lines of code
 */
object ScalaJSExample extends js.JSApp{
  import Page._
  val Epsilon = 0.00001

  type Color = Vec
  val Color = Vec

  def main() = {
    val r = new util.Random(16314302)
    val spiral = for (i <- 0 until 11) yield {
      val theta = i * (i + 5) * Pi / 100 + 0.3
      val center = (0 - 4 * sin(theta), 1.5 - i / 2.0, 0 - 4 * cos(theta))
      val form = Sphere(center, 0.3 + i * 0.1)
      val surface = Flat((i / 6.0, 1 - i / 6.0, 0.5))
      (form, surface)
    }

    def rand(d: Double) = (r.nextDouble() - 0.5) * d * 2

    val drops = Array(
      Sphere((2.5, 2.5, -8), 0.3),
      Sphere((1.5, 2.2, -7), 0.25),
      Sphere((-1.3, 0.8, -8.5), 0.15),
      Sphere((0.5, -2.5, -7.5), 0.2),
      Sphere((-1.8, 2.3, -7.5), 0.3),
```