# THE ADVENTUROUS DEVELOPERS GUIDE TO JVM LANGUAGES

SIMON MAPLE
@SJMAPLE

ZEROTURNAROUND

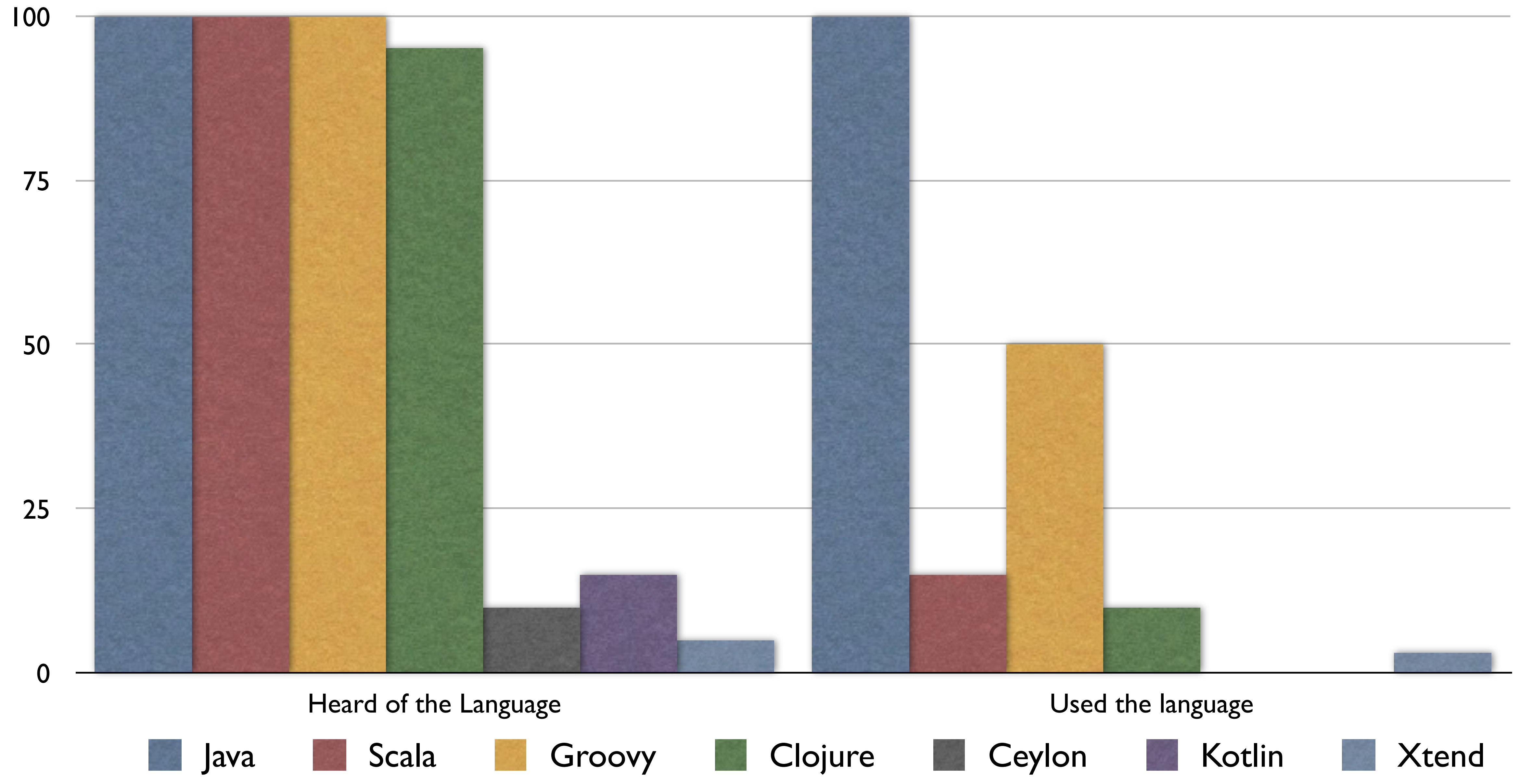
## YOUR SPEAKER

SIMON MAPLE
@SJMAPLE

ZEROTURNAROUND

IBM
SIMON MAPLE
I'M THE TECHNICAL EVANGELIST
PAST JOB: SOMMELIER
Cheers
+44 7974 761027
SIMON.MAPLE@ZEROTURNAROUND.COM
LINKEDIN.COM/IN/SIMONMAPLE
@SJMAPLE
ZEROTURNAROUND

# MY AUDIENCE


100
75
50
25
0
Heard of the Language
Used the language
Java
Scala
Groovy
Clojure
Ceylon
Kotlin
Xtend

# JAVA

*“Most people talk about Java the language, and this may sound odd coming from me, but I could hardly care less. At the core of the Java ecosystem is the JVM.”*

**James Gosling,**

creator of the Java programming language (2011, TheServerSide)

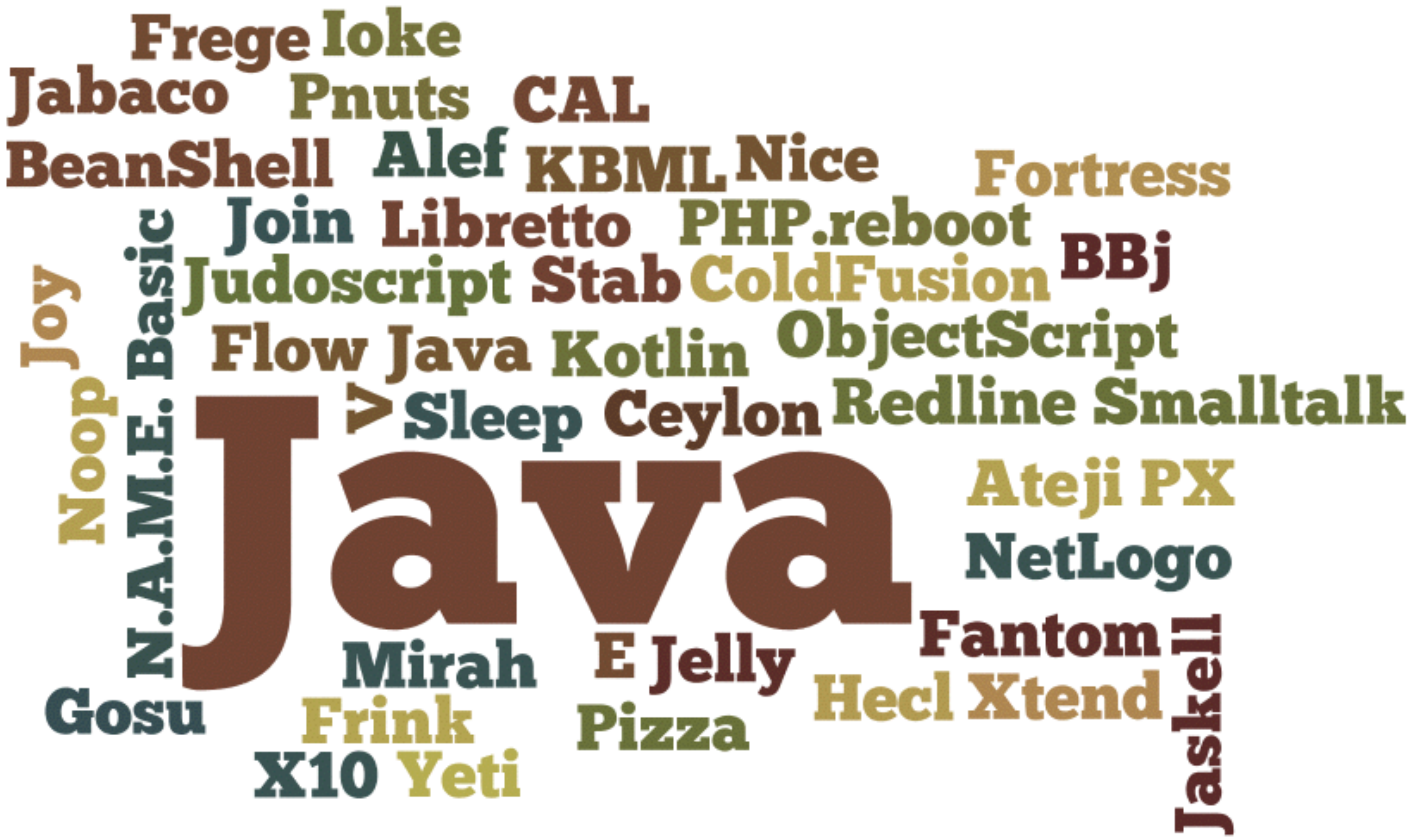
## ~~JAVA~~ THE JVM

*"Most people talk about Java the language, and this may sound odd coming from me, but I could hardly care less. At the core of the Java ecosystem is the JVM."*

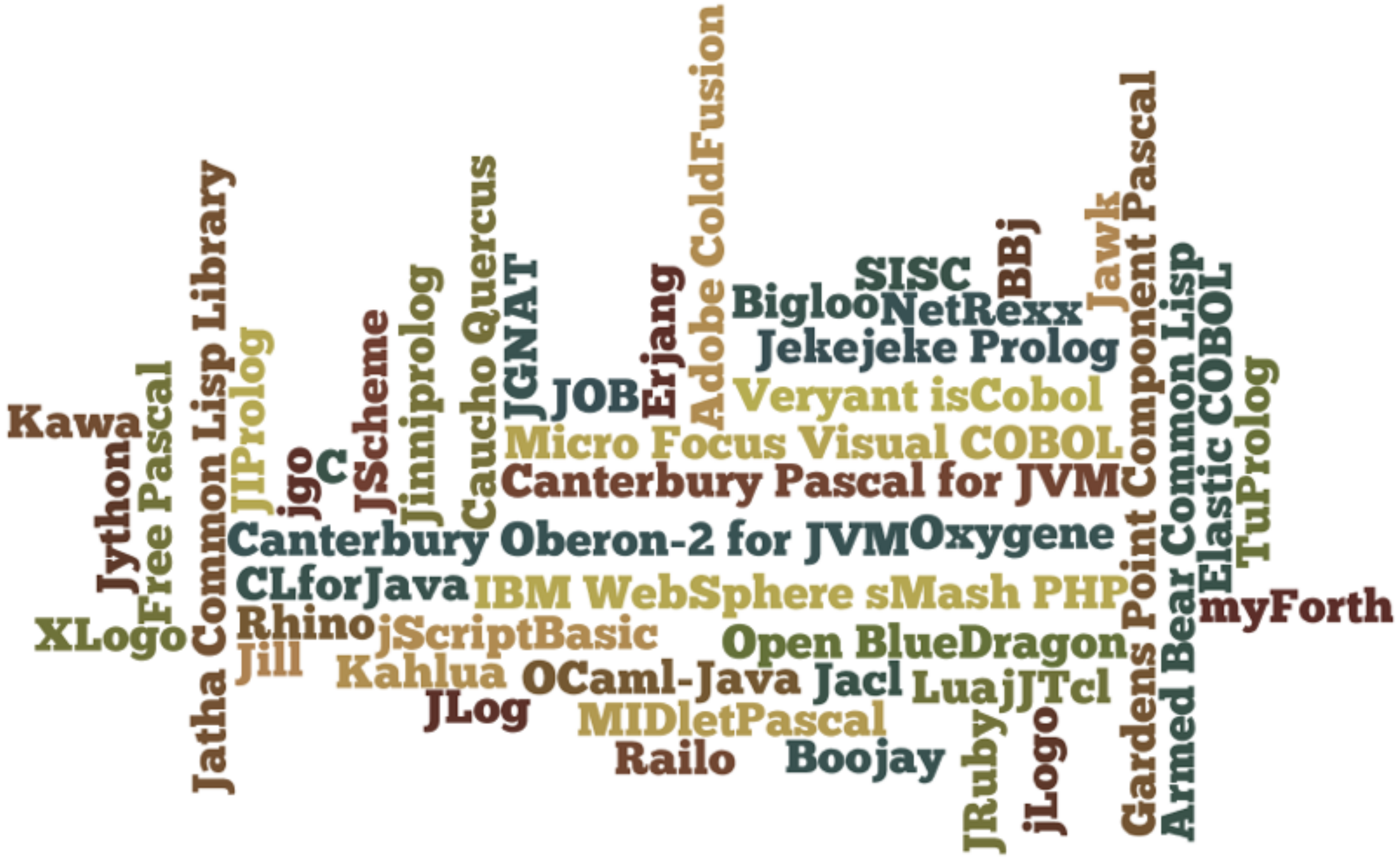
**James Gosling,**

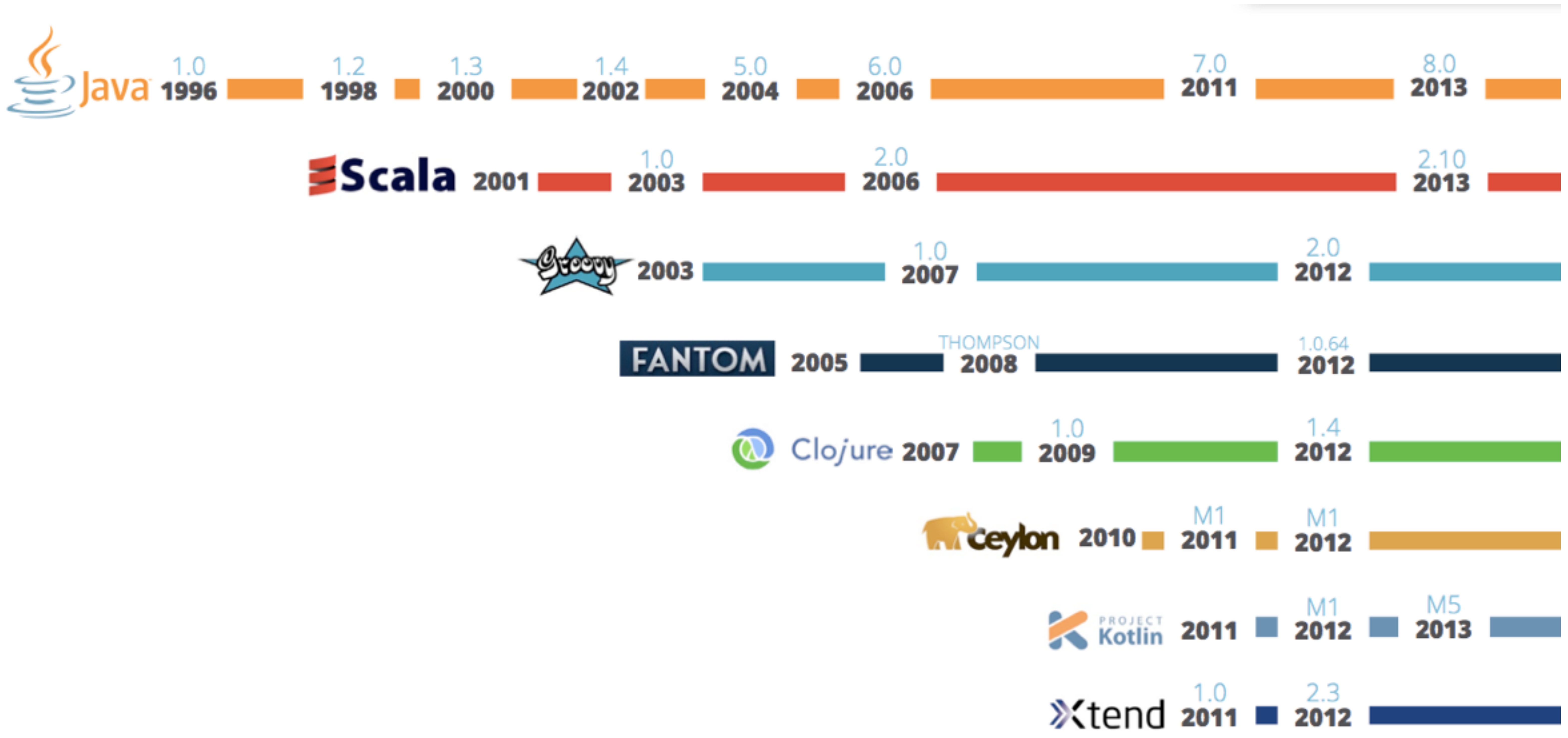
creator of the Java programming language (2011, TheServerSide)

## LANGUAGES BUILT FOR THE JVM



# LANGUAGES PORTED TO THE JVM





Java
1.0 1996
1.2 1998
1.3 2000
1.4 2002
5.0 2004
6.0 2006
7.0 2011
8.0 2013
Scala
2001
1.0 2003
2.0 2006
2.10 2013
Groovy
2003
1.0 2007
2.0 2012
FANTOM
2005
THOMPSON 2008
1.0.64 2012
Clojure
2007
1.0 2009
1.4 2012
ceylon
2010
M1 2011
M1 2012
PROJECT Kotlin
2011
M1 2012
M5 2013
Xtend
1.0 2011
2.3 2012

REBELLABS

HTTP://REBELLABS.ORG

Java™ 8

# JAVA 8

1. DON’T BREAK **BINARY** COMPATIBILITY
2. AVOID INTRODUCING **SOURCE** INCOMPATIBILITIES
3. MANAGE **BEHAVIORAL** COMPATIBILITY CHANGES

## LET'S EXPERIMENT

Scala

# COMPANION CLASS

THERE IS NO STATIC

```
import HttpServer._
// import statics from companion object
```

## VARIABLES

THERE IS NO FINAL

```
val name: Type = initializer // immutable value

var name: Type = initializer // mutable variable
```

## CASE CLASS

```
case class Status(code: Int, text: String)
```

```
case method @ ("GET" | "HEAD") =>
...
case method =>
respondWithHtml(
Status(501,
       "Not Implemented"),
       title = "501 Not Implemented",
)      body = <H2>501 Not Implemented: { method } method</H2>
...
```

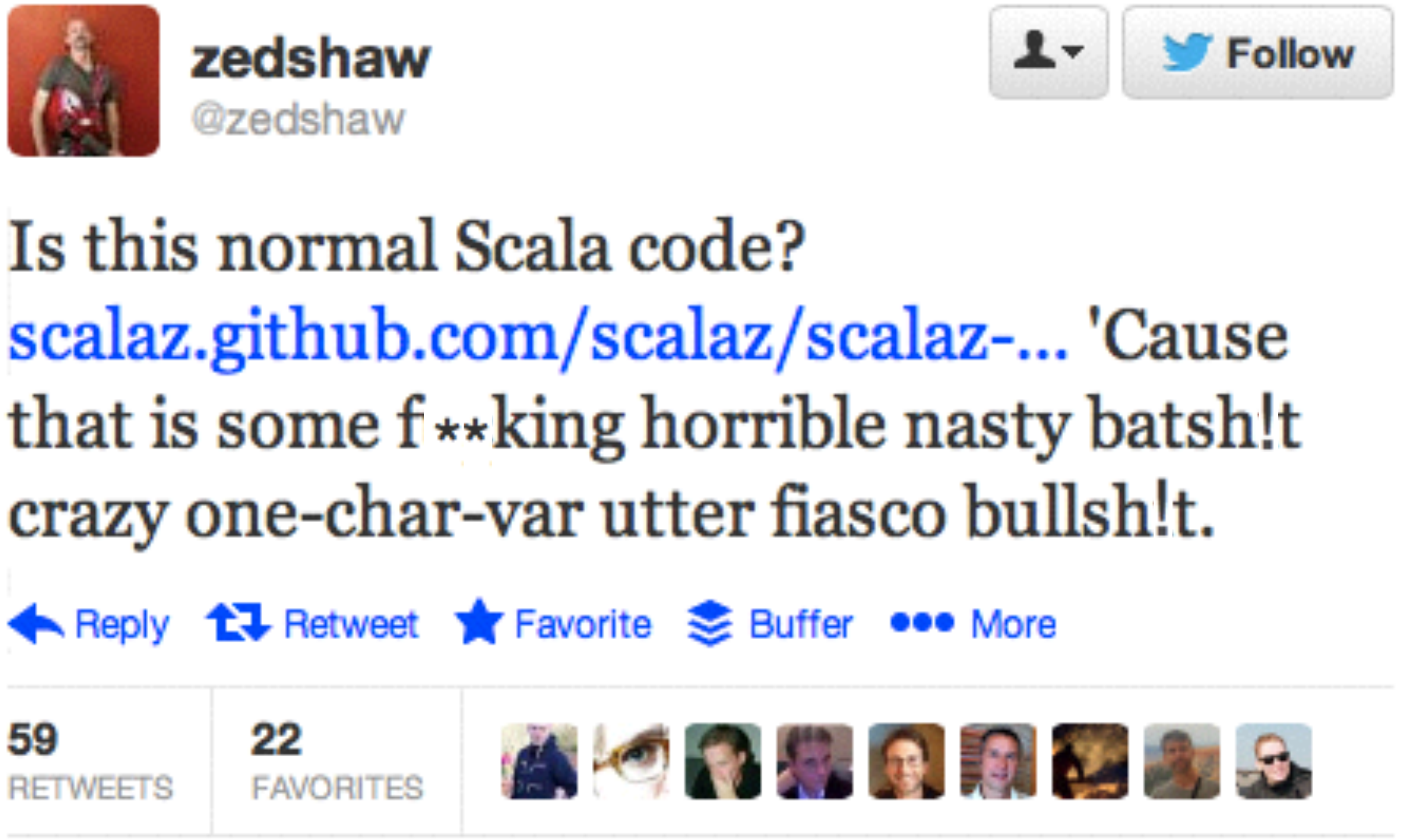
## STRINGS

```scala
val header = s"""
        |HTTP/1.1 ${status.code} ${status.text}
        |Server: Scala HTTP Server 1.0
        |Date: ${new Date()}
        |Content-type: ${contentType}
        |Content-length: ${content.length}
        """.trim.stripMargin + LineSep + LineSep
```

## NULL

```
def toFile(file: File, isRetry: Boolean = false): Option[File] =
 if (file.isDirectory && !isRetry)
  toFile(new File(file, DefaultFile), true)
  else if (file.isFile)
  Some(file)
  else
    None
```

## COMPLEXITY



zedshaw
@zedshaw

Is this normal Scala code? scalaz.github.com/scalaz/scalaz-... 'Cause that is some f**king horrible nasty batsh!t crazy one-char-var utter fiasco bullsh!t.

```
/*
 * Product Categories
 */

/** Index for a product category */
sealed trait P[+IX, +IY] { type _1 = IX; type _2 = IY }

case class ProductCategory[UX <: Hom, UY <: Hom](
  _1: GeneralizedCategory {type U = UX}, _2: GeneralizedCategory {type U = UY}
) extends GeneralizedCategory with Hom {
  type _1 = _1.type
  type _2 = _2.type
  type L = P[UX#L, UY#L]
  type H = P[UX#H, UY#H]
  case class C[A >: L <: H, B >: L <: H](
    _1: UX#C[A#_1, B#_1], _2: UY#C[A#_2, B#_2]
  ) extends P[UX#C[A#_1, B#_1], UY#C[A#_2, B#_2]]
  type U = ProductCategory[UX, UY]

  def id[A>:U#L<:U#H] = C(_1.id[A#_1],_2.id[A#_2])
  def compose[A >: U#L <: U#H, B >: U#L <: U#H, C >: U#L <: U#H](
    f: B =>: C, g: A =>: B
  ) = C(_1.compose(f._1, g._1), _2.compose(f._2, g._2))
}
```

```scala
/** Isomorphism for arrows of kind * -> * -> * */
case class Iso[Arr[_,_], A, B](to: Arr[A, B], from: Arr[B, A])

/** Isomorphism for arrows of kind (* -> *) -> (* -> *) -> * */
case class Iso2[Arr[_[_], _[_]], F[_], G[_]](to: Arr[F,G], from: Arr[G,F])

/** Isomorphism for arrows of kind (* -> * -> *) -> (* -> * -> *) -> * */
case class Iso3[Arr[_[_,_], _[_,_]], F[_,_], G[_,_]](to: Arr[F,G], from: Arr[G,F])

/** Set isomorphism */
type <=>[A, B] = Iso[Function1, A, B]

/** Natural isomorphism between functors */
type <~>[F[_], G[_]] = Iso2[~>, F, G]

/** Isomorphism natural in both sides of a bifunctor */
type <~~>[F[_,_], G[_,_]] = Iso3[~~>, F, G]

/** Set isomorphism is commutative */
implicit def flipIso[A, B](implicit i: A <=> B): B <=> A =
  new Iso[Function1, B, A](i.from, i.to)

/** Natural isomorphism is commutative */
implicit def flipFunctorIso[F[_], G[_]](implicit i: F <~> G): G <~> F =
  new Iso2[~>, G, F](i.from, i.to)
```

Groovy

## JAVA SUPERCHARGED!

## NULL

```
def streetName = user?.address?.street
```

## ELVIS LIVES

```
def displayName = user.name ?: "Anonymous"
```

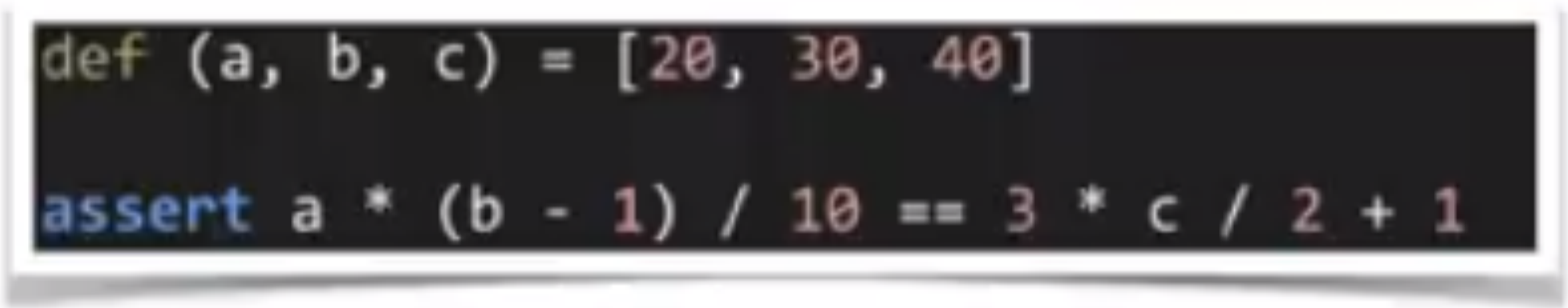
## CLOSURES

```
square = { it * it }

[ 1, 2, 3, 4 ].collect(square) // [1, 4, 9, 16]
```

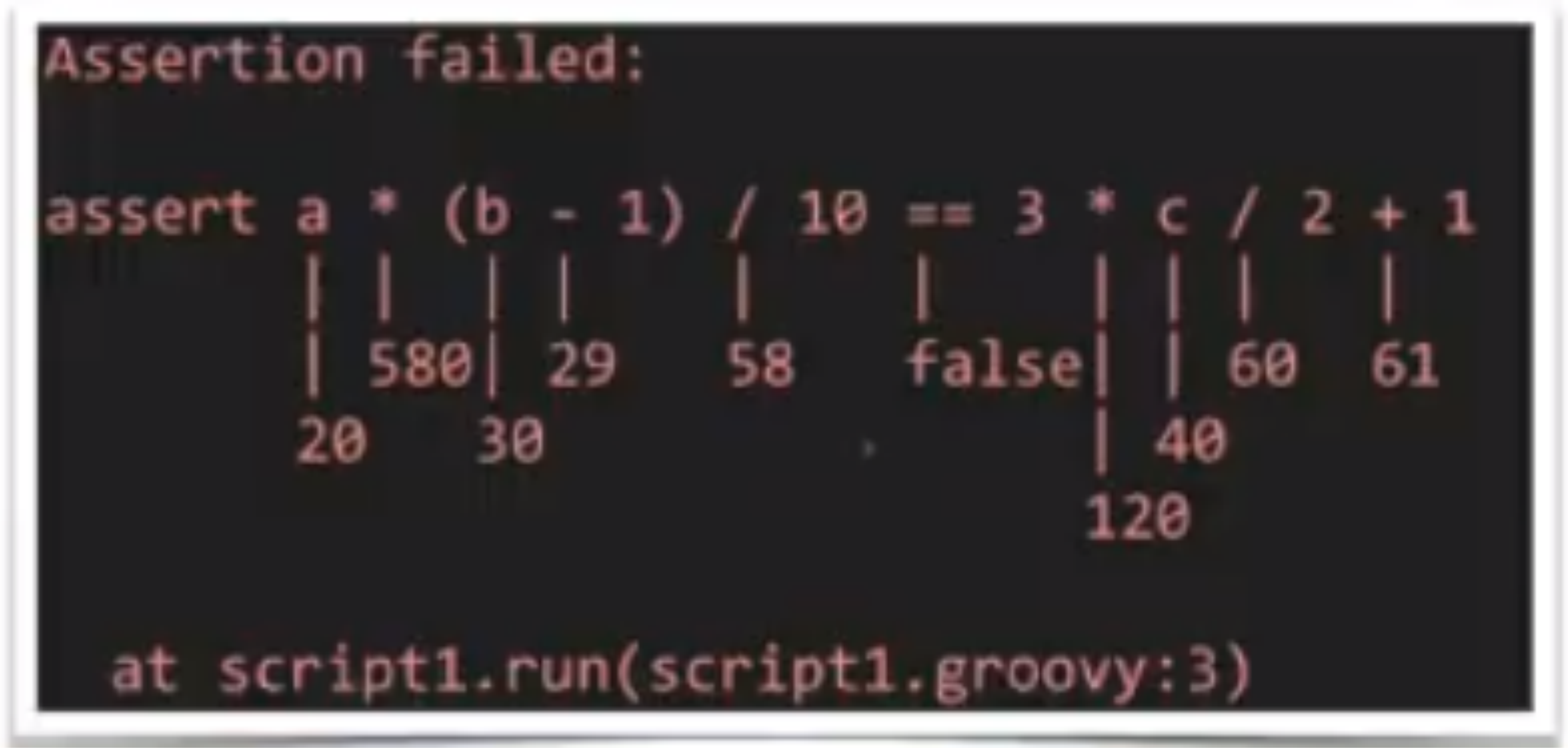
## POLYMORPHIC CLOSURES

```
def adder = { a, b -> a + b }

assert adder(1, 2) == 3
assert adder(‘A’, ‘B’) == ‘AB’
```

# POWER ASSERTS



```groovy
def (a, b, c) = [20, 30, 40]

assert a * (b - 1) / 10 == 3 * c / 2 + 1
```



```
Assertion failed:

assert a * (b - 1) / 10 == 3 * c / 2 + 1
       | |  | |    |    |    | | |   |
       | 580| 29   58   false| | 60  61
       20   30               | 40
                             120

  at script1.run(script1.groovy:3)
```

## COLLECTIONS

```
def names = ["Ted", "Fred", "Jed", "Ned"]

println names //[Ted, Fred, Jed, Ned]

def shortNames = names.findAll { it.size() <= 3 }

shortNames.each { println it } // Ted
                                // Jed
                                // Ned
```

# GROOVY 2.0 - DYNATIC

```
import groovy.transform.TypeChecked

void someMethod() {}

@TypeChecked
void test() {
  // compilation error:
  // cannot find matching method sommeeMethod()
  sommeeMethod()
}
```

Ceylon

## LET’S EXPERIMENT

PROJECT
Kotlin

## LET'S EXPERIMENT

Xtend

# LET’S EXPERIMENT

## SUMMARY

EVERYONE’S SYNTAX SUCKS…

## SUMMARY

EVERYONE’S SYNTAX SUCKS…

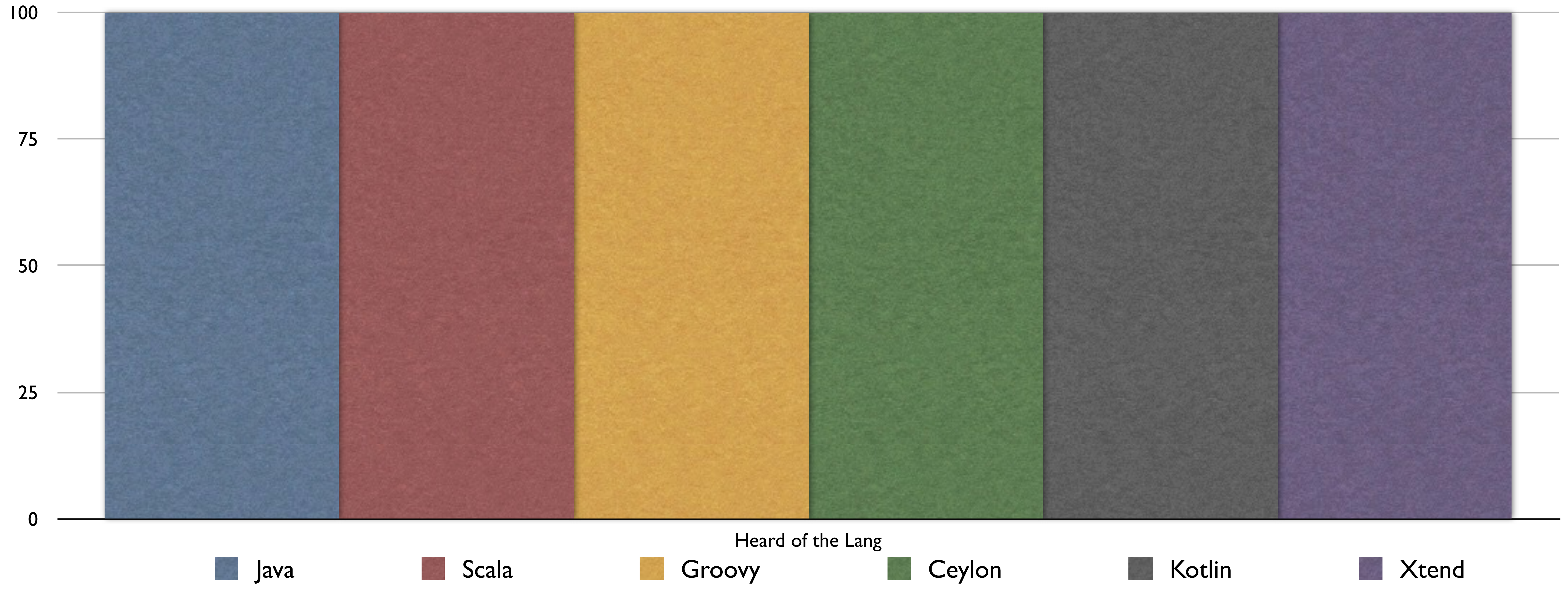
TO SOMEONE ELSE.

## SUMMARY

THE JVM IS AWESOME

BE ADVENTUROUS!

# YOU, ONE HOUR LATER



## REBEL LABS == AWESOME

FREE TECHNICAL REPORTS ON:

JAVA 8,
CONTINUOUS DELIVERY,
APP SERVER DEBATE,
JVM WEB FRAMEWORKS,
PRODUCTIVITY REPORTS...

## REBEL LABS == AWESOME

AND...

THE ADVENTUROUS DEVELOPERS GUIDE TO JVM LANGUAGES

# RESOURCES

HTTPSERVER EXAMPLES OF EACH LANGUAGE ON GITHUB

  [https://github.com/zeroturnaround/jvm-languages-report](https://github.com/zeroturnaround/jvm-languages-report)

THE ADVENTUROUS DEVELOPERS GUIDE TO JVM LANGUAGES

  [http://zeroturnaround.com/rebellabs/devs/the-adventurous-developers-guide-to-jvm-languages/](http://zeroturnaround.com/rebellabs/devs/the-adventurous-developers-guide-to-jvm-languages/)

# RESOURCES

SIMON MAPLE
@SJMAPLE

# FREE STUFF - BOOTH 8

## FREE STUFF - ONLINE

bit.ly/tshirt-sjm