

THE ADVENTUROUS **DEVELOPERS** GUIDE TO **JVM LANGUAGES**

SIMON MAPLE
@SJMAPLE



ABOUT ME



SIMON MAPLE
@SJMAPLE



ABOUT **ME**



SIMON MAPLE
@SJMAPLE



VIRTUAL JUG FOUNDER

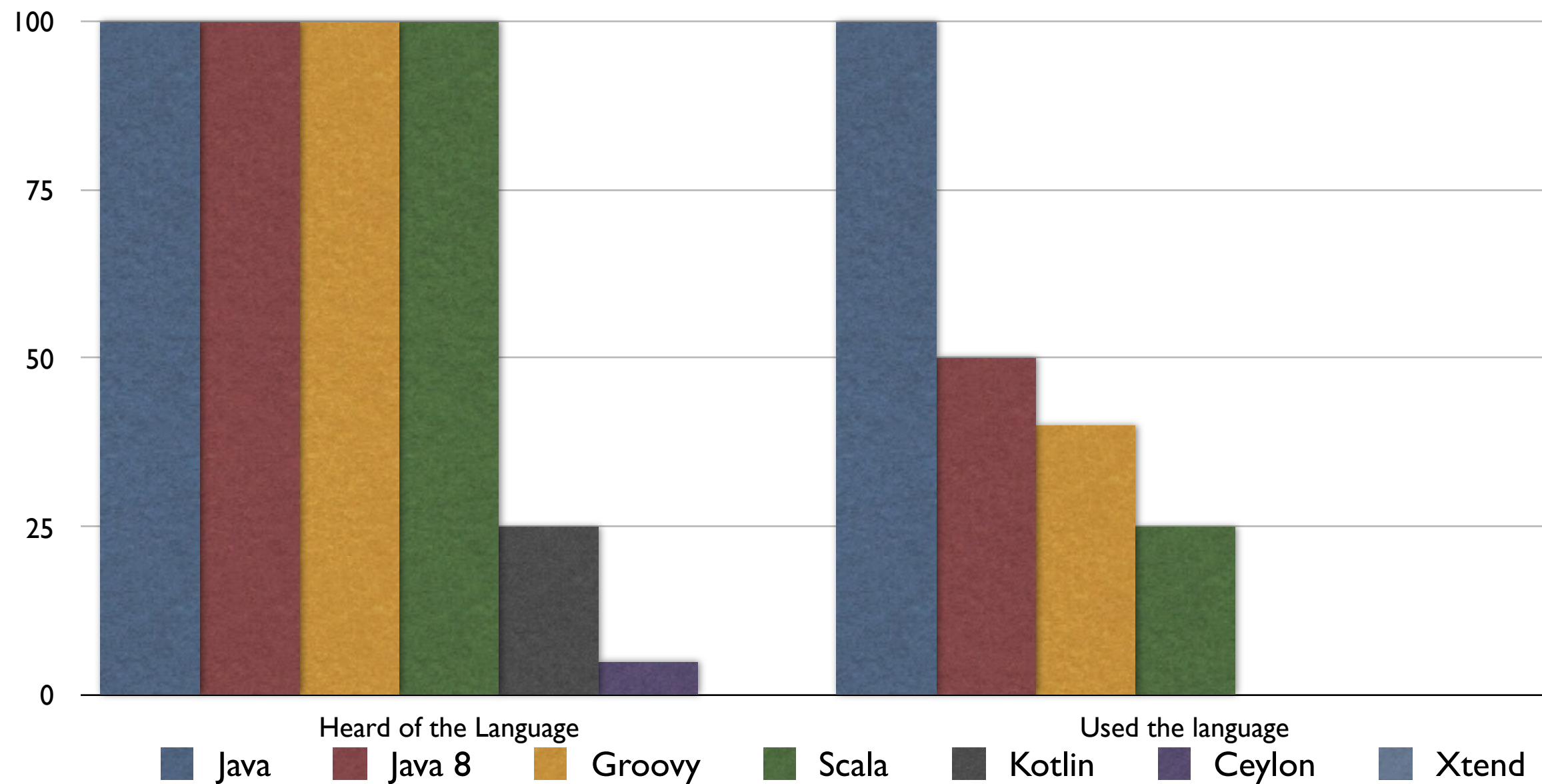
LONDON JUG CO-LEADER

JAVA CHAMPION

JAVAONE ROCKSTAR

REBELLABS AUTHOR

ABOUT YOU



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 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

A word cloud of various programming languages built for the JVM. The word 'Java' is the largest and most prominent, centered in the image. Other languages are arranged around it in various sizes and orientations, including horizontal, vertical, and diagonal. The colors of the text are diverse, ranging from dark browns and greys to bright yellows and greens.

Frege Ioke
Jabaco Pnuts CAL
BeanShell Alef KBML Nice Fortress
Join Libretto PHP.reboot BBj
Judoscript Stab ColdFusion
Joy Noop N.A.M.E. Basic Flow Java Kotlin ObjectScript
> Sleep Ceylon Redline Smalltalk
Ateji PX
NetLogo
Mirah E Jelly Fantom
Gosu Frink Pizza Hecl Xtend
X10 Yeti Jaskell

LANGUAGES PORTED TO THE JVM







JAVA 8

1. Don't break **binary compatibility**
2. Avoid introducing **source incompatibilities**
3. Manage **behavioral compatibility** changes

LET'S **EXPERIMENT**





COMPANION CLASS

NO **STATIC** KEYWORD



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

VARIABLES

THERE IS NO **FINAL** KEYWORD

```
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

```
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



Follow

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.



Reply



Retweet



Favorite



Buffer



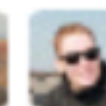
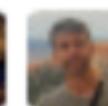
More

59

RETWEETS

22

FAVORITES



```

/*
 * 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))
}

```

```

/** 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)

```



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]
```

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

```
void someMethod() {}
```

```
void test() {
```

```
    someMethod()
```

```
}
```

GROOVY 2.0 - DYNATIC

```
import groovy.transform.TypeChecked
void someMethod() {}

@TypeChecked
void test() {

    sommeeMethod()
}
```

GROOVY 2.0 - DYNATIC

```
import groovy.transform.TypeChecked
void someMethod() {}
@TypeChecked
void test() {
    //compilation error:
    //cannot find matching method sommigeMethod()
    sommigeMethod()
}
```

YAH, WE SAVE LIVES!



Founder/CEO Jevgeni "Hosselhuff" Kabanov gets ready to save more Java developers from redeploy madness with JRebel



LET'S **EXPERIMENT**





PROJECT
Kotlin

LET'S **EXPERIMENT**



»Xtend

LET'S **EXPERIMENT**



SUMMARY

Functions are **first class citizens**
and should be treated as such!

SUMMARY

EvErY-ONe's
5YNT4X SuCkS...

SUMMARY

EvErY-ONe's
5YNT4X SuCkS...

...to someone else!

SUMMARY

The JVM is **AWESOME**

SUMMARY

The JVM is **AWESOME**

BE **ADVENTUROUS!**

YOU, ONE HOUR LATER



RESOURCES

HTTPSERVER EXAMPLES OF EACH LANGUAGE ON GITHUB

<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/>

FINAL QUESTIONS?

1. why are there rings on Saturn?

Because God liked it, so he

put a ring on it. Saturn was

NOT a single

Lady.

FREE STUFF!

JRebel



XRebel



0t.ee/devnexusjr 0t.ee/devnexusxr



HOPE YOU ENJOYED
THE PRESENTATION

IF NOT, AND YOU WOULD LIKE TO
CLAIM YOUR 60 WASTED MINUTES BACK,
PLEASE REDEEM VIA

JRebel