CompSci 230
Introduction to Java

Lecture Slides #1: S1 2015

Version 1.1 of 2015-03-02: discussed jitting on slide 14

Introducing myself

- Clark Thomborson:
  - 1973-5 Ass'y language programmer for Nicolet Technology
  - 1975 BS(honors) Chemistry, MS Comp Sci/Eng'g, Stanford
  - 1980 PhD Computer Science, C-MU
  - 1979-86 Asst Prof at UC Berkeley
  - 1983 Married Barbara Borske, shifted surname to Thomborson
    - Not Thomborsonske, Borthomp, or Borson!
  - 1986-94 Prof at U Minnesota-Duluth (incl. 1992-3 Visiting Prof at MIT)
  - 1995 Principal Programmer at LaserMaster (6 mo.)
  - 1995-6 Systems Integrator, contracted to Digital Biometrics (6 mo.)
  - 1996- Prof at U Auckland

Today's Agenda

- Topics:
  - What is Java?
  - Is Java secure?
  - How does Java compare with Python?

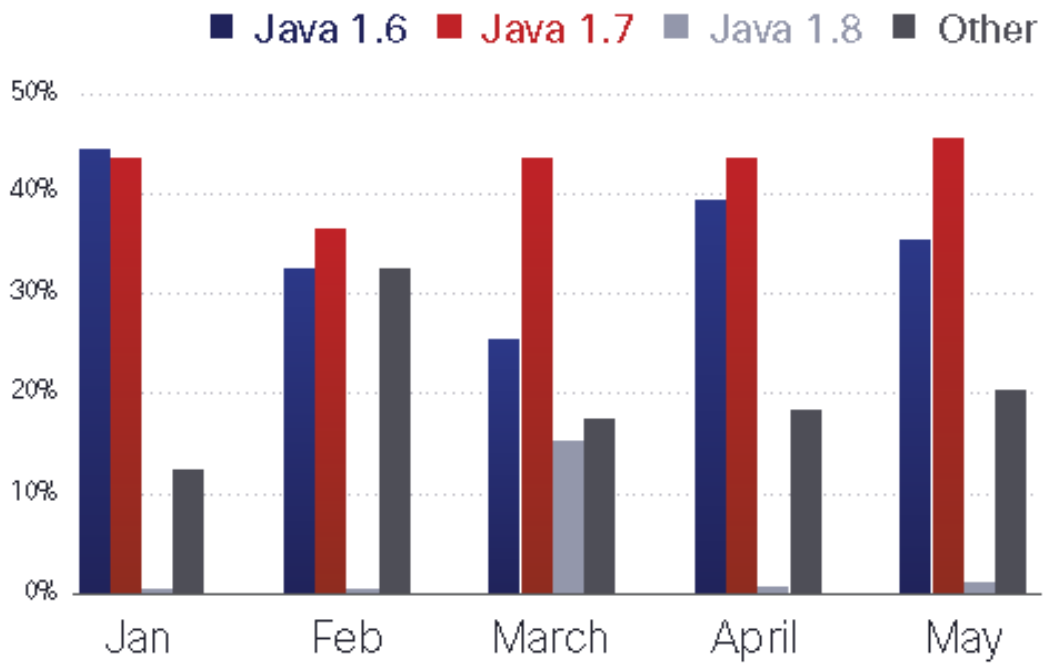
The Java Programming Language is…

Simple	Architecture neutral
Object oriented	Portable
Distributed	High performance
Multithreaded	Robust
Dynamic	Secure

- Source: The Java Tutorials, Oracle, 2015.
- Do you believe everything you read?
  - I believe that the Java Tutorials are an authoritative and reliable source of technical information about Java.
  - I believe that anyone (and any company!) is likely to "oversell" the advantages of their products and inventions.
- What do other authoritative sources say about Java's security?

Cisco's *2014 Mid-year Security Report*

- "Java remains the most exploited piece of software, with 93 percent of all web exploits originating from this service.
  - "Java versions 1.6 and 1.7 remain the most exploited, but exploits tailored for version 1.8 are also on the rise.
  - "With increases in exploit kits that rely first and foremost on non-Java vectors, such as Microsoft Silverlight, we might be seeing a shift away from Java 8 (which has stronger security controls) to other software that is more conducive to attacks."



Cisco's *Annual Security Report 2015*

- "In recent years, Java has played an unwanted starring role in lists of the most prevalent and severe vulnerabilities to exploit.
  - "However, Java appears to be falling out of favour among adversaries searching for the fastest, easiest, and least detectable ways to launch exploits using software vulnerabilities...
- "Of the top 25 vendor- and product-related vulnerability alerts from January 1, 2014, to November 30, 2014, only one was Java-related…
  - "In 2013, Cisco Security Research tracked 54 urgent new Java vulnerabilities;
  - "in 2014, the number of tracked Java vulnerabilities fell to just 19.
- "This should not detract online criminals from the popularity and effectiveness of attacking these older vulnerabilities that persist today."

So… is Java secure? (My opinion)

- Yes, if you're careful:
  - If Java code is well-designed for security, and if you don't give it unnecessary privileges, then an attacker will have a hard time making it "do anything bad" on your fully-patched system (= JVM, browser, OS).
- No, if you're careless (or clueless ;-):
  - If Java code is malicious, and you're running it on an unpatched system.
  - If you have lowered the default security settings on a fully-patched system.
- Yes, if you're comparing it to other sandboxed languages in early 2015:
  - Java 1.8 isn't a very attractive target. All recently-disclosed vulnerabilities have been patched promptly. See http://java-0day.com/.
  - Google's Project Zero recently annoyed Apple and Microsoft but not Oracle, according to an article in ZDnet, by disclosing vulnerabilities in their products after a 90-day notice.
  - Flash: http://threatpost.com/1800-domains-overtaken-by-flash-zero-day/110835

How does Java compare with Python?

- Java:

Simple	Architecture neutral
Object oriented	Portable
Distributed	High performance(?)
Multithreaded	Robust (as defined by Gosling)
Dynamic	Secure

- Python:

Simpler than Java	Architecture neutral
Object oriented	Portable
Distributed	Adequate performance
Multithreaded	Less robust?
More dynamic than Java	More difficult to secure?

- Java programs are "robust" if they are well-tested: reliable behaviour.
  - Python is not a strongly-typed language, so a method can produce strange results if given an unexpected input. More difficult to test, so less "robust"?

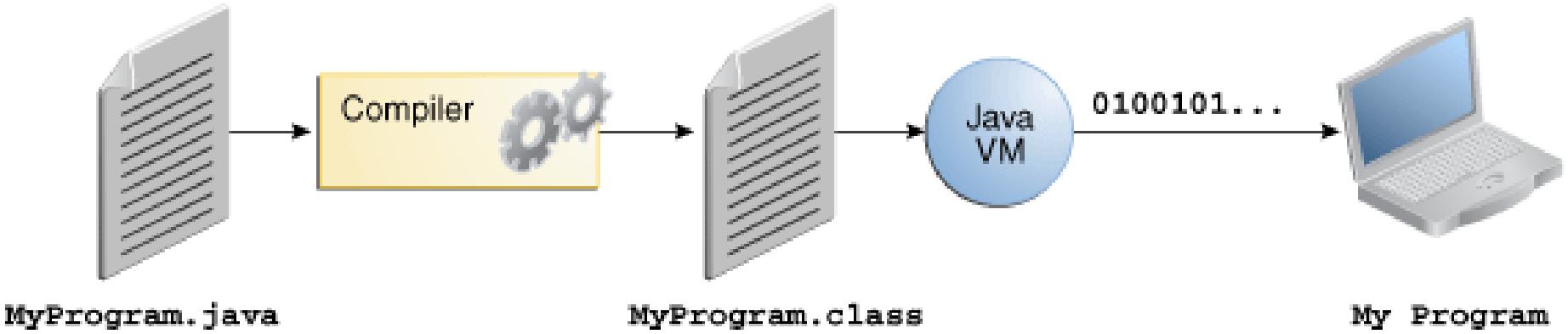
java4Python ("Java for Python Programmers")

- "Python is a nice language for beginning programming for several reasons.
  - "The syntax is sparse and clear.
  - "The underlying model is very simple. Everything is an object.
  - "You can write powerful and interesting programs without a lot of work.
- "Python is representative of a whole class of languages, sometimes referred to as scripting languages.
  - "Other languages in the same category as Python are Ruby and Perl.
- "Java is representative of what I will call industrial strength languages, which include C++, C# and Scala.
  - "Industrial strength languages are good for projects with several people working on the project where being formal and careful about what you do may impact lots of other people."

Static and Dynamic Languages

- java4Python: "Python is representative of one kind of language, called a dynamic language.
  - "Dynamic languages can be interpreted directly, which means that the actual text of the program — the source code — is used while the program is running.
- "In contrast, a static language is executed in two phases:
  - first the program is translated from source code to binary code,
  - and then the binary code is interpreted.
- "Although the terms dynamic and static language are widely used, the distinction is a fuzzy one.
  - "Most execution engines do both translation and interpretation.
- "Static refers to what the translater does.
  - "The translater is called a compiler.
- "Dynamic refers to what the interpreter does."
- Remember: static vs. dynamic is an imprecise way to describe a language, but compiler vs. interpreter is an important technical distinction.

Java: A Compiled and Interpreted Language



- "In the Java programming language, all source code is first written in plain text files ending with the `.java` extension.
  - "Those source files are then compiled into .class files by the javac compiler.
- "A `.class` file does not contain code that is native to your processor;
  - "it instead contains bytecodes — the machine language of the Java Virtual Machine (Java VM).
  - "The java launcher tool then runs your application [by interpreting its bytecode on] an instance of the Java Virtual Machine."
  - Source: http://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html

Is Java a Static or Dynamic Language?

- Dynamic, because Java bytecode (in a `.class` file) is interpreted by a JVM.
- Static, because Java source code (in a `.java` file) is compiled into another language (Java bytecode) before it is executed – it is *not* directly executable.
- So… we might say that Java bytecode is dynamic, and that Java source code is static.

Is Python Static or Dynamic?

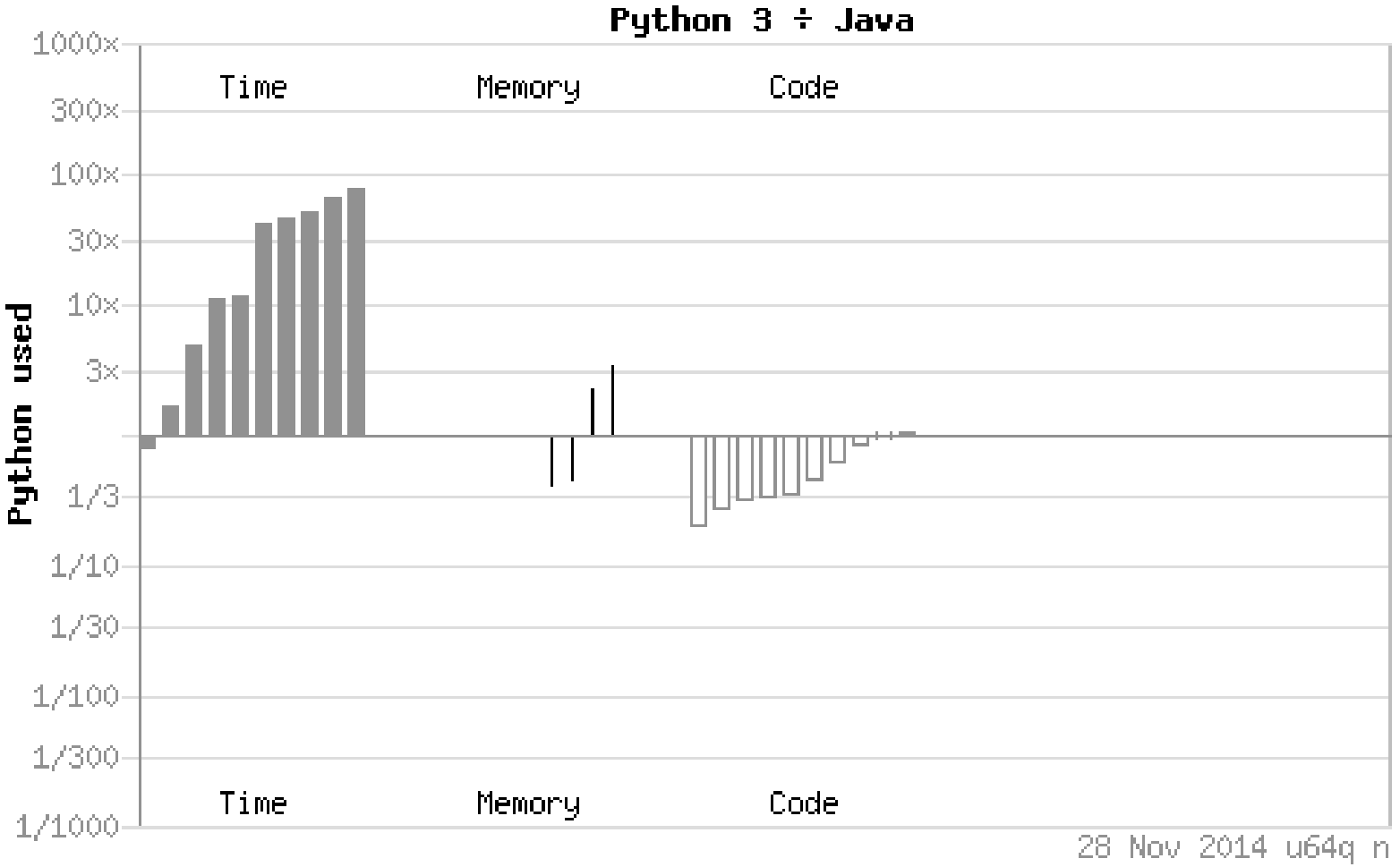
- Python bytecode (in a `.pyc` file) is dynamic, because it is interpreted by the Python runtime system.
  - Note: Python bytecode is *not* portable across versions of Python.
  - The semantics of Java bytecode is very stable.
    - "Old" Java bytecode runs on newer JVMs. (However libraries are versioned, and there are some incompatibilities across major releases of Java; so a recompilation is advisable every year or two.)
- Python source code (in a `.py` file) is static, because it is compiled into bytecode before the bytecode is interpreted.
- However: a Java compilation is more complicated than a Python compilation, and a Python interpretation is more complicated than a Java interpretation.
  - Python source code is often interpreted and executed on a line-by-line basis, in a shell.
  - It is possible to compile Python source into an `.exe`, see Cython v0.22.
- So… Python (but not Cython!) is "more dynamic" than Java.

Performance: Python vs. Java

- Advocates for each language use different ways to measure performance, and (unsurprisingly ;-) get different results.
  - In many applications, performance is unimportant.
  - If performance is very important, you should use a fully-static language such as C or Fortran.
- Python runtime performance is hampered by the limited amount of analysis done by the Python compiler.
  - The Java compiler performs optimisations which are infeasible in Python (because Python variables have no static type – we'll discuss typing in future lectures, but we will not discuss optimisations).
- Java performance is hampered if the source code isn't pre-compiled.
  - Java compilation is much slower than Python compilation.
- Most JVMs (and some PVMs, e.g. PyPy) compile bytecode into machine code, to avoid the overheads of interpretation on tight loops.
  - This is called "just-in-time" compilation, or jitting.
  - A jitting VM may run a bytecoded program 10x faster than a non-jitting VM, because machine-coded loops run *much* faster than interpreted loops.

One Way to Measure Performance

- http://benchmarksgame.alioth.debian.org/u64q/python.html:

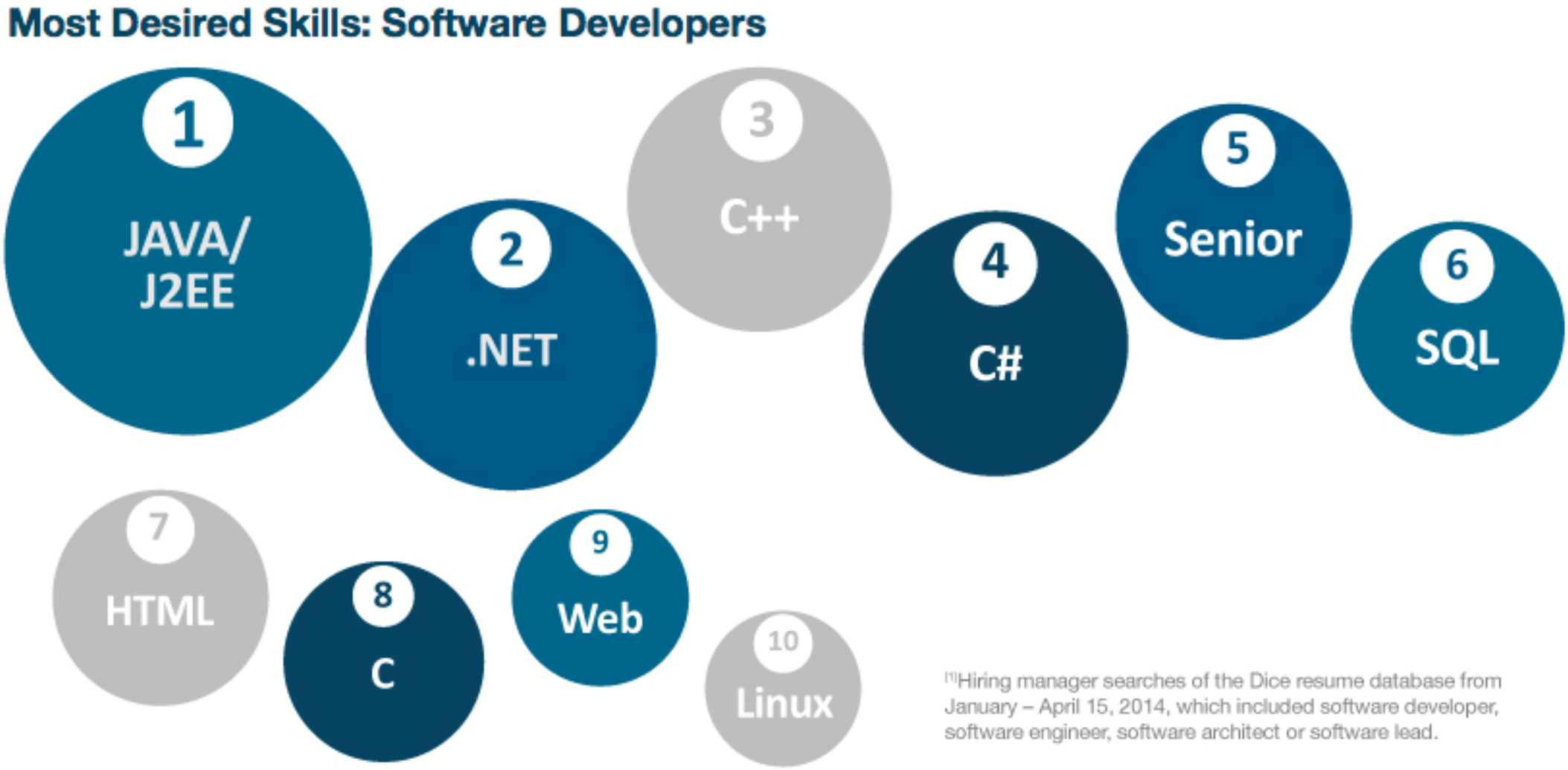


Horses for courses?

- Ted Samson, "Why Netflix is embracing Python over Java", Mar 11, 2013:
  - "Netflix is increasingly turning to Python over Java to power certain aspects of its video-streaming service, such as generating and processing alerts, boosting resilience, securing transactions, producing deployable AMIs (Amazon Machine Images), and for managing and automatic Cassandra clusters.
  - "Python is giving Java a run for its money among developers at Netflix, [due to Python's] 'rich batteries-included standard library, succinct and clean yet expressive syntax, large developer community, and wealth of third-party libraries.'"
- Sean Kelly's Recovery from [a Java] Addiction, 10-minute video, 2006.
  - I recommend you watch this video *after* you learn Java.
  - Sean argues, persuasively, that Python is much better than Java for web development. (He doesn't consider Javascript or compatibility. Nor will we! ;-)
  - Sean doesn't discuss testing and quality assurance. (Maybe watch this video again, after you have completed the software-quality unit in this course?)

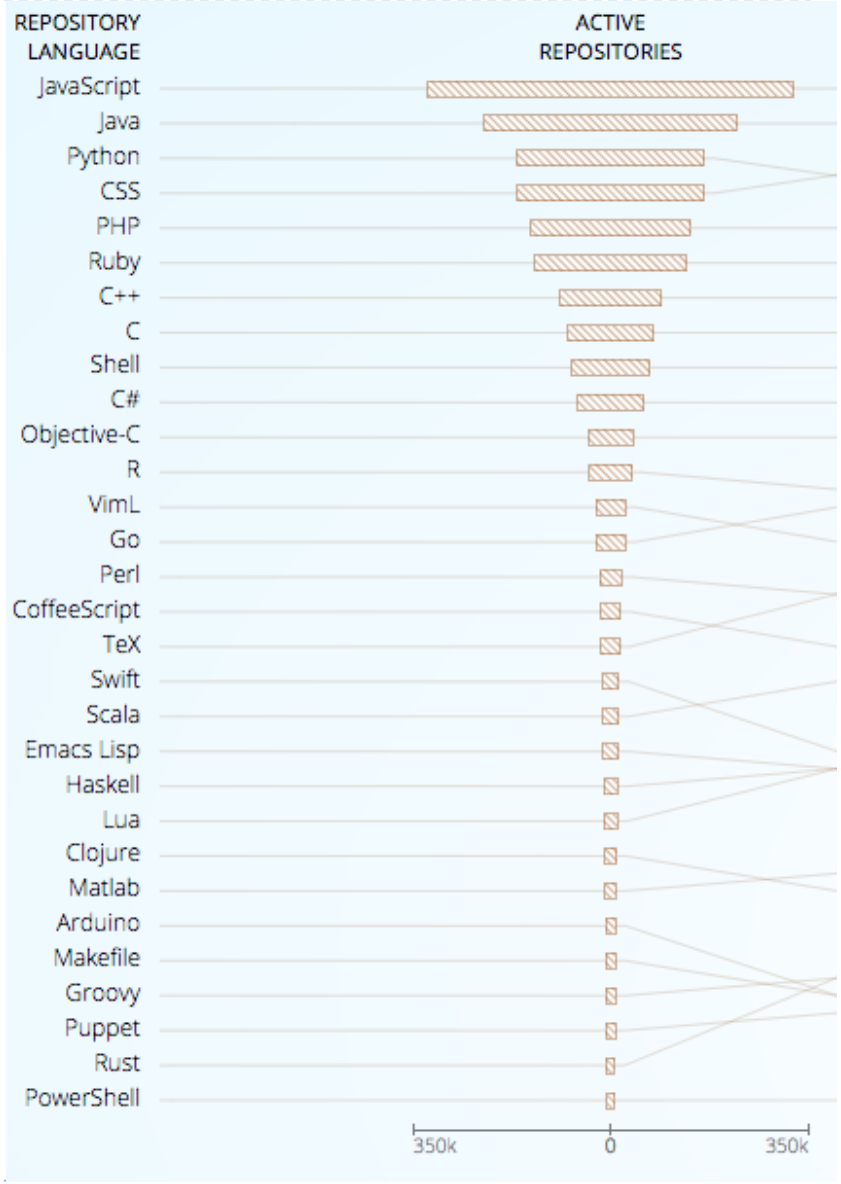
Dice.com (a job-search agency in the US)

http://media.dice.com/wp-content/uploads/2014/05/Screenshot-2014-05-14-14.53.08.png



GitHub (a web-based Git repository hosting service)

- "JavaScript Tops GitHub's Most Popular Languages", by Nick Kolakowski, Dice.com, Feb 12, 2015:
  - "What are the top programming languages on GitHub?
  - "According to GitHut, a website that attempts to estimate and visualize the repository's most popular languages, JavaScript topped the list in the fourth quarter of 2014, followed by Java, Python, CSS, PHP, Ruby, C++, C, and Shell.
  - "With roughly 3.4 million users and 16.7 million repositories, GitHub offers …"
- To learn more about GitHub:
  - https://education.github.com/pack



Review

- Topics:
  - What is Java?
  - Is Java secure?
  - How does Java compare with Python?
- Important technical concepts:
  - Compilers, interpreters, source code, bytecode.
  - If you don't understand these concepts, you'll be lost!
- Important (but fuzzy!) descriptors:
  - Static, dynamic, secure, high performance, simple, robust.
  - If you don't understand these words, you won't be able to communicate.
  - These words have multiple meanings, depending on the context and the motivation of the speaker – be careful!