

Julia, Rust, Python, R, Ruby, Javascript of toch beter Java?

Een gids voor programmeurs

Seppe vanden Broucke

23 februari 2017



seppe vanden broucke

Seppe vanden Broucke, KU Leuven

Faculty of Economics and Business

Management Informatics Department

Research:

- Big Data & Analytics
- Process Mining
- Business Process Management
- Programming: Java, Python, R, Lua

Contact:

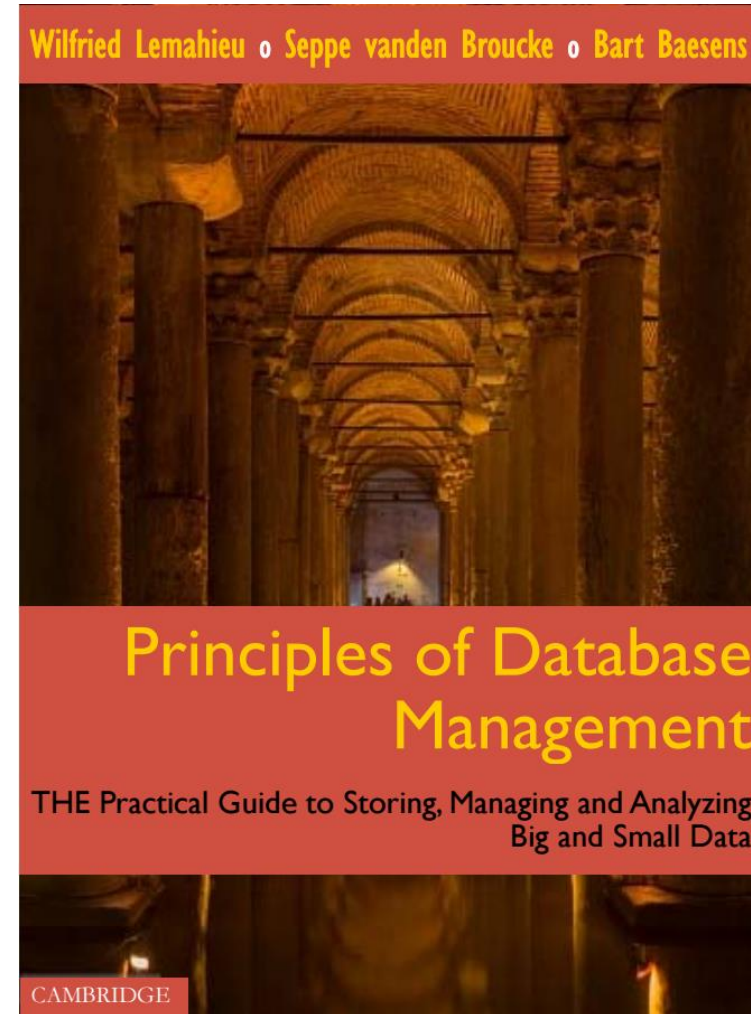
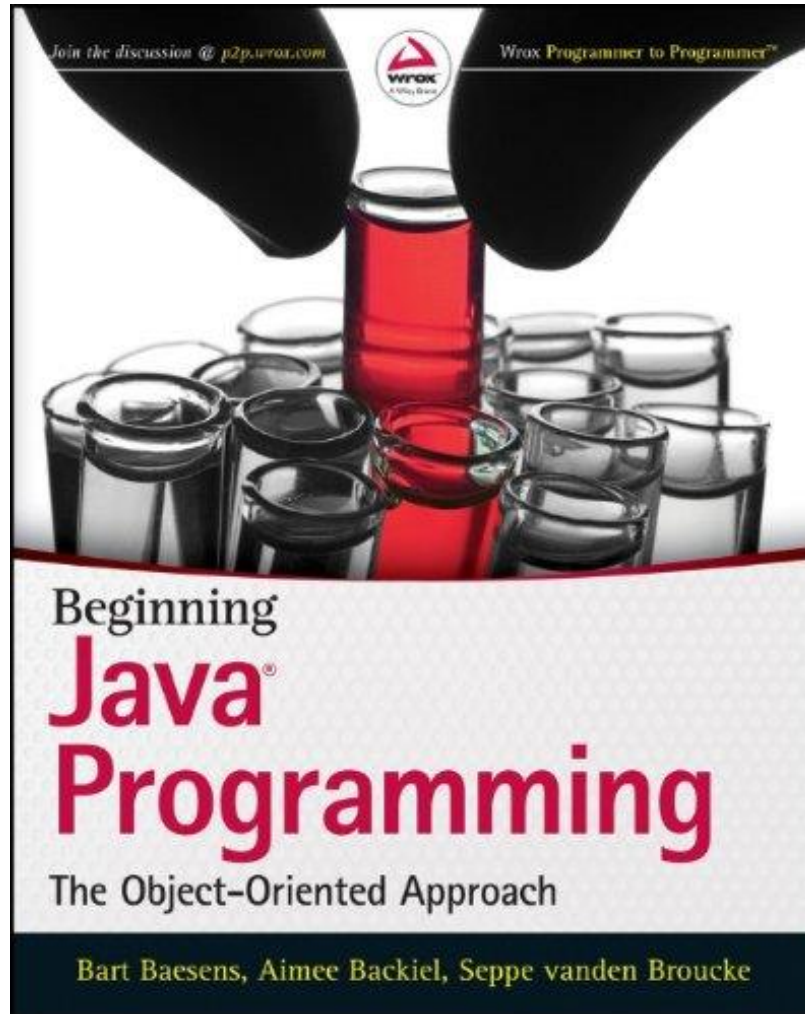
seppe.vandenbroucke@kuleuven.be

www.processmining.be

www.dataminingapps.com

www.seppe.net





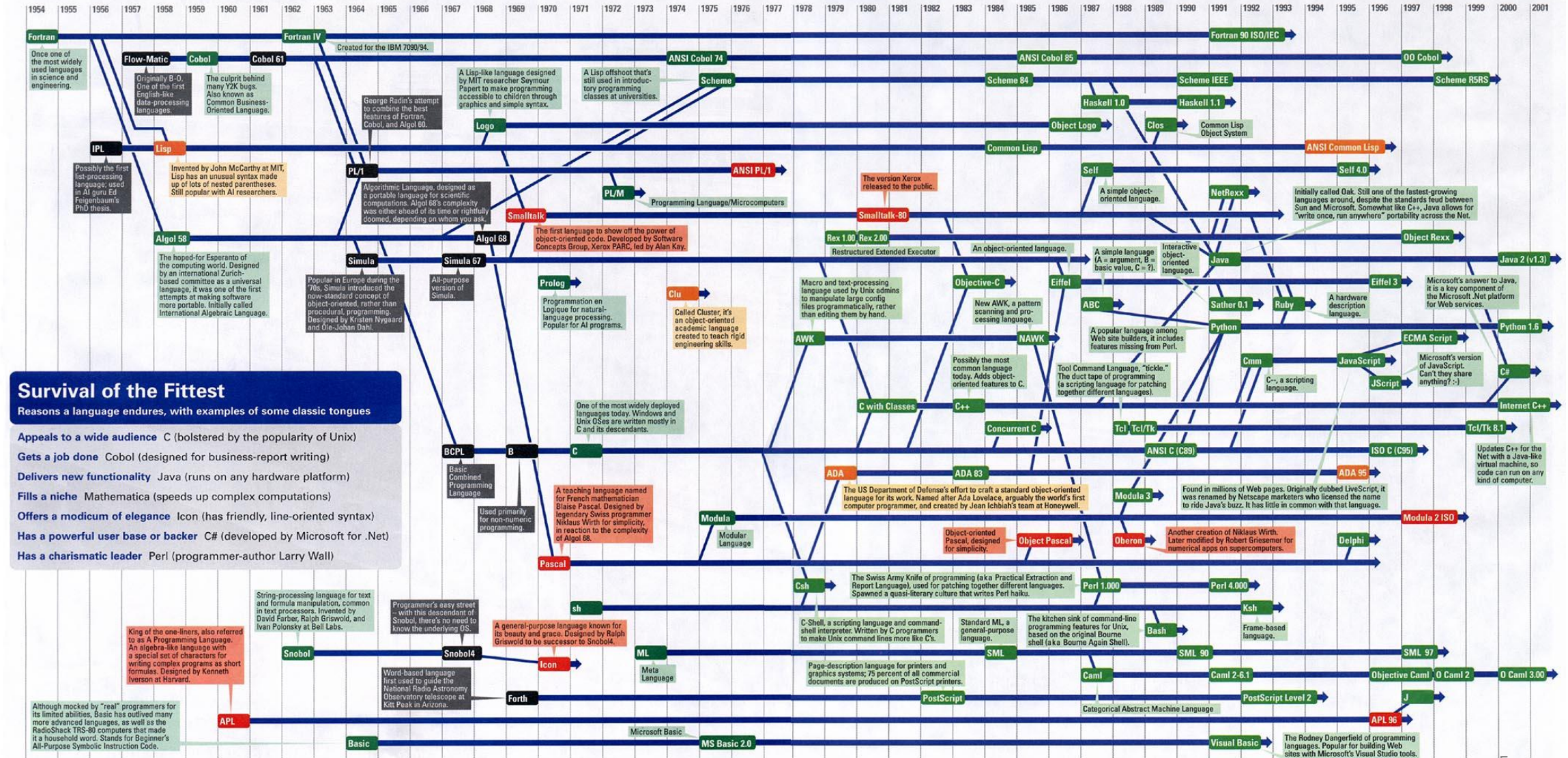
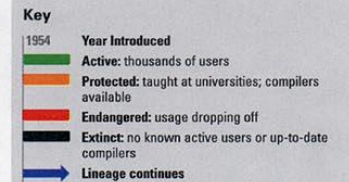
Mother Tongues

Tracing the roots of computer languages through the ages

Just like half of the world's spoken tongues, most of the 2,300-plus computer programming languages are either endangered or extinct. As powerhouses C/C++, Visual Basic, Cobol, Java, and other modern source codes dominate our systems, hundreds of older languages are running out of life.

An ad hoc collection of engineers – electronic lexicographers, if you will – aim to save, or at least document, the lingo of classic software. They're combing the globe's 9 million developers in search of coders still fluent in these nearly forgotten lingua francas. Among the most endangered are Ada, APL, B (the predecessor of C), Lisp, Oberon, Smalltalk, and Simula.

Code-raker Grady Booch, Rational Software's chief scientist, is working with the Computer History Museum in Silicon Valley to record and, in some cases, maintain languages by writing new compilers so our ever-changing hardware can grok the code. Why bother? "They tell us about the state of software practice, the minds of their inventors, and the technical, social, and economic forces that shaped history at the time," Booch explains. "They'll provide the raw material for software archaeologists, historians, and developers to learn what worked, what was brilliant, and what was an utter failure." Here's a peek at the strongest branches of programming's family tree. For a nearly exhaustive rundown, check out the Language List at www.informatik.uni-freiburg.de/Java/misc/lang_list.html. – Michael Menduno



Trends

2016

1. JavaScript
2. SQL
3. Java
4. C#
5. PHP
6. Python
7. C++
8. AngularJS
9. Node.js
10. C



1. Rust
2. Swift
3. F#
4. Scala
5. Go
6. Clojure
7. React
8. Haskell
9. Python
10. C#

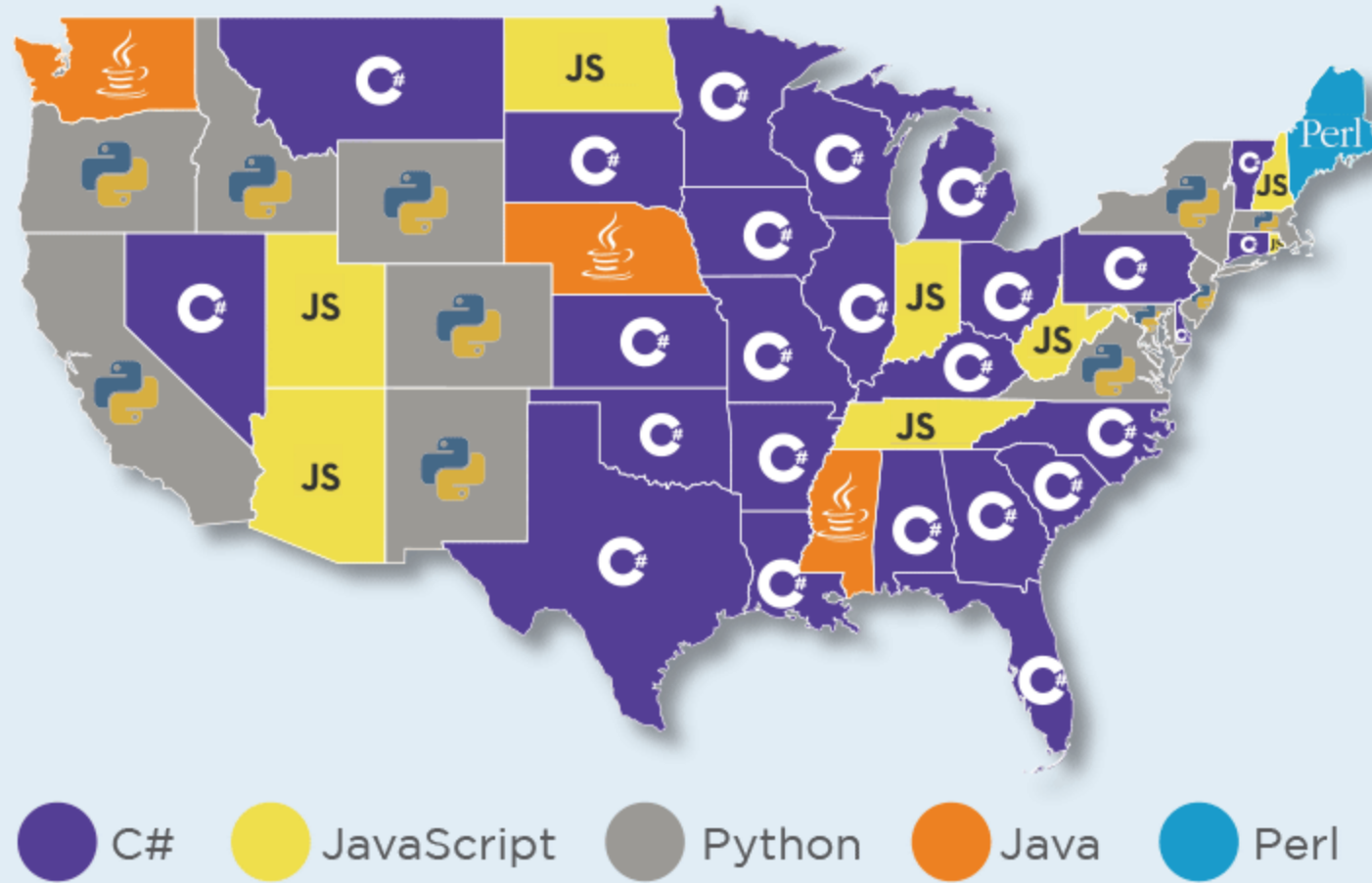


1. Visual Basic
2. WordPress (PHP)
3. Matlab
4. Sharepoint (ASP)
5. CoffeeScript
6. PHP
7. Cordova
8. Salesforce
9. Perl
10. SQL

TIOBE Index (februari 2017)

Feb 2017	Feb 2016	Change	Programming Language	Ratings	Change
1	1		Java	16.676%	-4.47%
2	2		C	8.445%	-7.15%
3	3		C++	5.429%	-1.48%
4	4		C#	4.902%	+0.50%
5	5		Python	4.043%	-0.14%
6	6		PHP	3.072%	+0.30%
7	9	+	JavaScript	2.872%	+0.67%
8	7	-	Visual Basic .NET	2.824%	+0.37%
9	10	+	Delphi/Object Pascal	2.479%	+0.32%
10	8	-	Perl	2.171%	-0.08%
11	11		Ruby	2.153%	+0.10%
12	16	++	Swift	2.125%	+0.75%
13	13		Assembly language	2.107%	+0.28%
14	38	++	Go	2.105%	+1.81%
15	17	+	R	1.922%	+0.73%
16	12	--	Visual Basic	1.875%	+0.02%
17	18	+	MATLAB	1.723%	+0.63%
18	19	+	PL/SQL	1.549%	+0.49%
19	14	--	Objective-C	1.536%	+0.13%
20	23	+	Scratch	1.500%	+0.71%

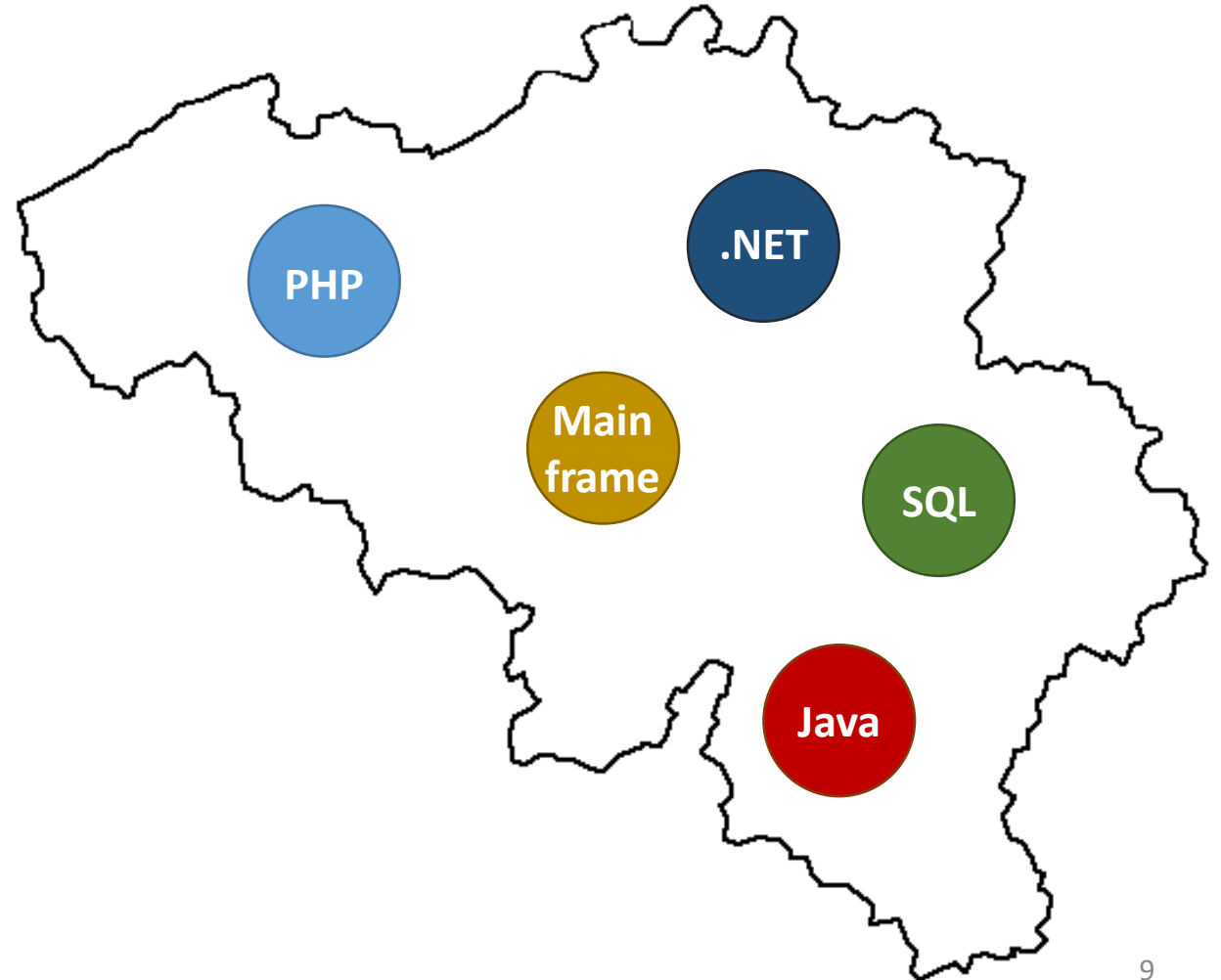
Programming Language By State With The Most Jobs and Least Competition



(Brightwing)

“Programmeur” in België

- .NET (VB.Net en C#)
- Java
- SQL
- PHP
- “Mainframe” (Fortran, Cobol)



Nieuwe evoluties

- Object georiënteerd versus functioneel?
- Server-side JavaScript?
- Analytics?
- Memory safe languages?
- Concurrency?

Onze gids



Frontend

Web
Mobile
UI

Systems
Databases
Processing



Backend



Niche

Analytics
Concurrent systems
High-scalability

Frontend



Frontend

- Web
- Mobile
- UI



Frontend: Web

- 10 jaar geleden: PHP, ASP, JSP
- Ruby on Rails als “disruptor”
- Vandaag: JavaScript, PHP, Python, ASP.NET



Frontend: Web



- **Vermindering in gebruik** van Ruby on Rails
 - Gelimiteerd in toepassingsbereik
 - Moeilijk schaalbaar
 - Dure aanwervingskosten
 - Vernieuwingen inzake ORM e.d. geadopteerd door andere ecosystemen
 - Migratie terug naar PHP en diens web-frameworks (Symfony, Zend)
 - Overschakeling naar Python (Django)



Frontend: Web

- **PHP blijft een van de meest gebruikte talen op het web**
 - Makkelijk te leren
 - Snelle prototypen
 - Groot ecosysteem en community
 - “Dynamisch” aspect nuttig voor snelle time-to-market applicaties
 - Maar: PHP 4 versus 5, 6 en 7 wordt een probleem
 - PHP 7 voegt interessante functies toe zoals type hints en een algemene “opschoning” van de standard library
 - Belangrijk om programmeurs te selecteren en op te leiden volgens laatste ontwikkelingen!

Frontend: Web



- **“HHVM” als performanter alternatief**
 - “HHVM is an open-source virtual machine designed for executing programs written in Hack and PHP. HHVM uses a just-in-time (JIT) compilation approach to achieve superior performance while maintaining the development flexibility that PHP provides. HHVM supports Hack, PHP 5 and the major features of PHP 7”
 - Facebook open source project
 - Sneller, performanter, correcter
 - Doch: “minor incompatibilities”

Frontend: Web



- **ASP.NET blijft ook populair**
 - Zeker binnen Microsoft / .NET omgeving
 - Grotendeels met C# als sturende taal (minder VB.NET)
 - Open source implementaties beschikbaar (Mono), doch minder in gebruik

Frontend: Web



- **In Java is er een beweging naar “lean”**
 - (Langzaam) weg van “zware EE stacks” (JSF + Struts + Spring MVC + Hibernate + Tapestry) naar lichtere omgevingen
 - Java 8
 - HikariCP en jOOQ als moderne DB libraries
 - Spark Framework, Play Framework en Ratpack als moderne web frameworks

Frontend: Web



- **JavaScript verandert het landschap**

- Node.JS, EcmaScript 6, TypeScript, Elm, Dart, CoffeeScript
- Server-side JavaScript
- “Single-page apps”, reactive programming
- AngularJS, React, Vue.JS: “JavaScript to write user interfaces”
- Snelle adoptie door startups
- Moeilijk om gerichte keuze te maken (e.g. Angular 2 versus React)

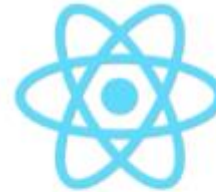
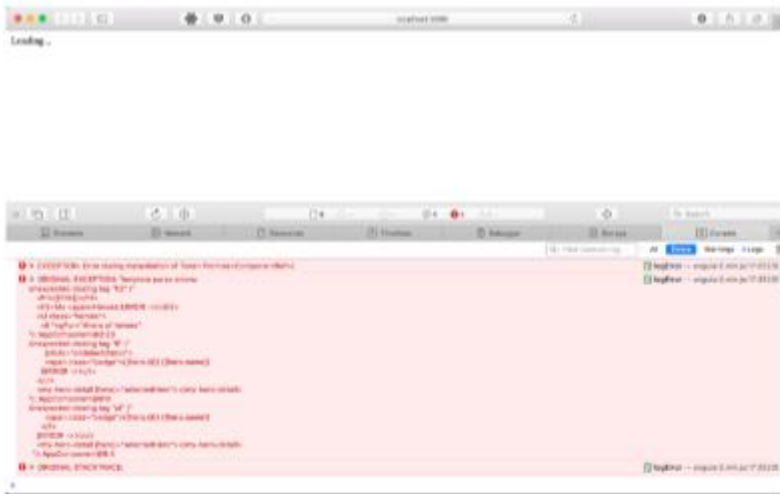
Frontend: Web



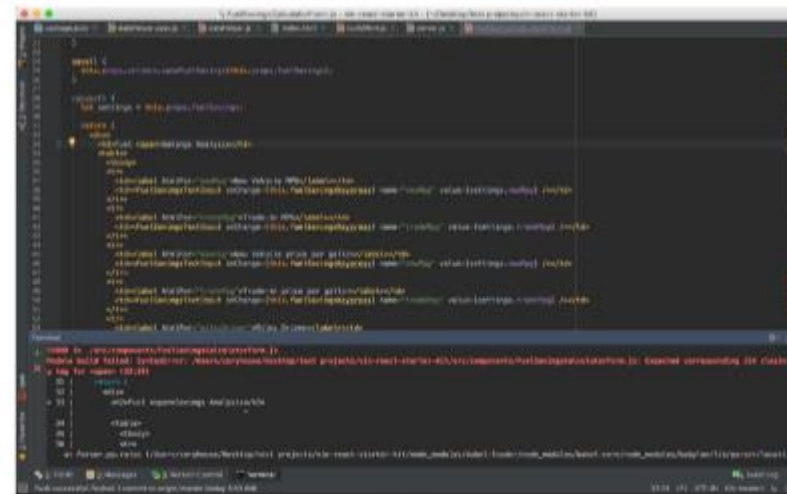
Frontend: Web



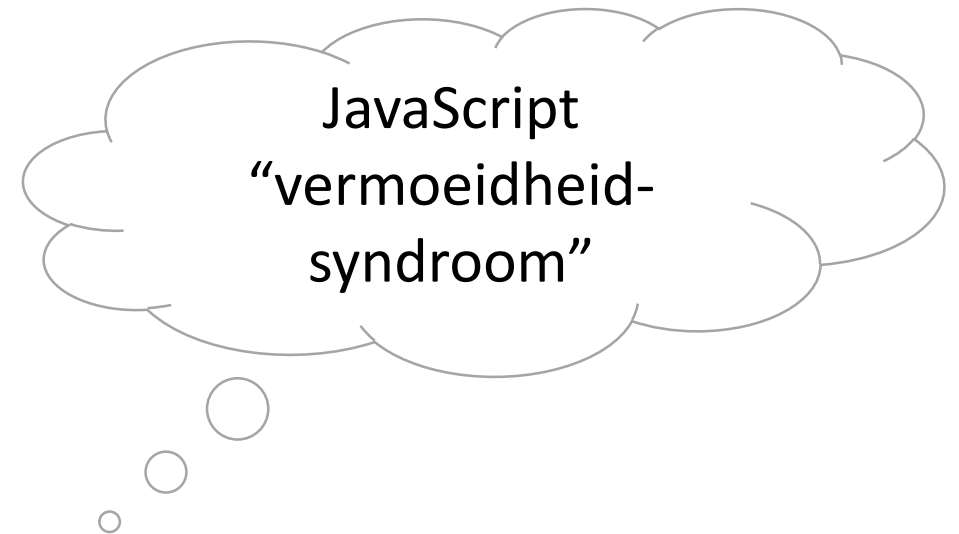
- Fails at runtime.
- No line number provided.
- No unclosed tag mentioned.



Fails at compile-time.
Line number provided.
Unclosed tag mentioned.



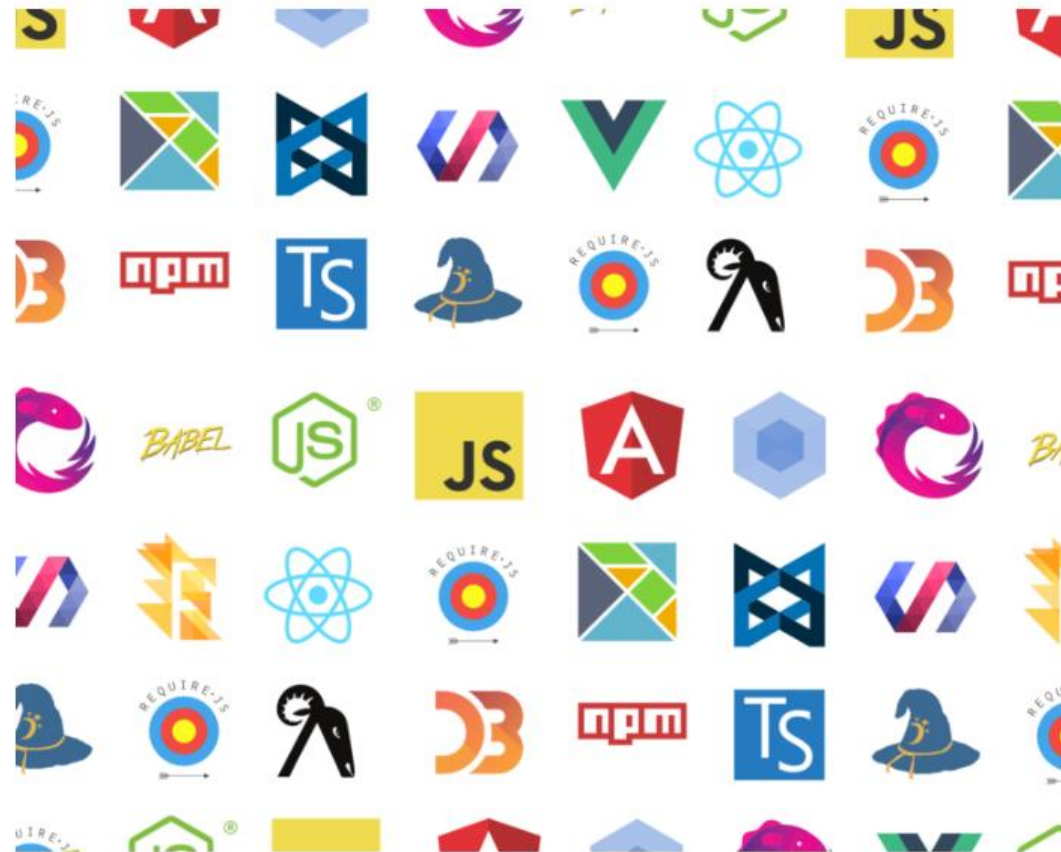
Frontend: Web



Ultimately, the problem is that **by choosing React (and inherently JSX), you’ve unwittingly opted into a confusing nest of build tools, boilerplate, linters, & time-sinks to deal with before you ever get to *create* anything.**

Frontend: Web

How it feels to learn JavaScript in 2016



Frontend: Web

- “I need to create a page that displays the latest activity from the users, so I just need to get the data from the REST endpoint and display it in some sort table”
 - “Oh my god no, no one uses jQuery anymore. You should try learning React”
 - “But first you need to add React and React DOM as a library in your webpage”
 - “JSX is just a JavaScript syntax extension that looks pretty much like XML. It’s kind of another way to describe the DOM, think of it as a better HTML”
 - “It’s 2016. No one codes HTML directly anymore”
 - “Babel is a transpiler that allows you to target specific versions of JavaScript, while you code in any version of JavaScript”
- “ES5? ES2016+? I’m getting lost over here. What’s ES5 and ES2016+?”
 - “Well, you COULD use ES6, but to use cool features like async and await, you need to use ES2016+. Otherwise you are stuck with ES6 generators with coroutines to block asynchronous calls for proper control flow”
- “What is Browserify? npm registry? Bower?”
 - “Angular is so 2015. But yes. Angular would be there, alongside VueJS or RxJS and other cool libraries”
- “That seems overly complicated to just grab a bunch of dependencies and tie them together”
 - “It is, that’s why you use a task manager like Grunt or Gulp or Broccoli to automate running Browserify. Heck, you can even use Mimosa”
- “SystemJS is a dynamic module loader that allows you to tie multiple modules in multiple files instead of bundling them in one big file”
- “I would transpile it from Typescript using a Webpack + SystemJS + Babel combo”
- “Typescript? I thought we were coding in JavaScript!”
 - “Typescript IS JavaScript, or better put, a superset of JavaScript, more specifically JavaScript on version ES6. You know, that sixth version we talked about before?”
- “I thought ES2016+ was already a superset of ES6! WHY we need now this thing called Typescript?”
 - “Oh, because it allows us to use JavaScript as a typed language, and reduce runtime errors. It’s 2016, you should be adding some types to your JavaScript code”
- Sigh... and Flow is?
 - “It’s a static type checker made by some guys at Facebook. They coded it in OCaml, because functional programming is awesome”
- “jTemplates? jQote? PURE? Transparency? JSRender? MarkupJS? KnockoutJS? That one had two-way binding. PlatesJS? jQuery-tmpl? Handlebars? Some people still use it. Mustache, underscore? I think now even lodash has one to be honest, but those are kind of 2014. Jade? DustJS? DotJS? EJS? Nunjucks? ECT?”

Frontend: Web

The screenshot displays a web browser window with the URL "Ons online contacteren" and a search bar for "Vind een kantoor of maak". The main content area shows "Kantoren Belfius Bank - Belfius" with a map of Brussels. The map includes markers for "Brussel Rogier Tower" and "Brussel Arenberg". A search bar for "Postcode of gemeente" is visible. A legend on the right side of the map lists various locations: "Open", "Op afspraak", "Geldautomaat", and "Gesloten".

Overlaid on the right side of the browser window is a network developer tool. The "Network" tab is active, showing a list of requests. The table below represents the data shown in the network tool:

Name	Status	Type	Initiator	Size	Time	Waterfall
index.aspx	304	document	Other	727 B	27 ms	
kernel.css?param=20170112141615	304	stylesheet	index.aspx	739 B	28 ms	
iws_framework.css?param=20170112141615	304	stylesheet	index.aspx	739 B	31 ms	
iws_style.css?param=20170112141615	304	stylesheet	index.aspx	739 B	28 ms	
iws_info_components.css?param=20170112141615	304	stylesheet	index.aspx	739 B	145 ms	
iws_framework_desktop.css?param=20170112141615	304	stylesheet	index.aspx	739 B	30 ms	
external_libs.js?param=20170112141615	304	script	index.aspx	739 B	143 ms	
external_query_libs.js?param=20170112141615	304	script	index.aspx	739 B	49 ms	
iws_kernel.js?param=20170112141615	304	script	index.aspx	739 B	54 ms	
iws_framework.js?param=20170112141615	304	script	https://www.belfius.be/common/nl/fw/lib/js/iws_kernel.js?param=20170112141615	739 B	378 ms	
external-angular.js?param=20170112141615	304	script	index.aspx	738 B	154 ms	
internal-angular.js?param=20170112141615	304	script	index.aspx	738 B	154 ms	
js?libraries=places&language=nl®ion=BE	200	script	index.aspx	(from ...)	1 ms	
satelliteLib-1fae8e32fe8c598c4506b077e890...	200	script	index.aspx	(from ...)	7 ms	
Contact%201300x350_tcm_86-110659.png	304	png	index.aspx	739 B	47 ms	
Card_Stop_content_tcm_86-8463.jpg	304	jpeg	index.aspx	738 B	29 ms	
default-user.png	304	png	index.aspx	322 B	24 ms	
iws_framework_tablet_landscape.css?param=...	304	stylesheet	index.aspx	738 B	27 ms	
iws_framework_tablet_portrait.css?param=20...	304	stylesheet	index.aspx	738 B	29 ms	
iws_framework_phone.css?param=20170112141615	304	stylesheet	index.aspx	739 B	28 ms	
iws_framework_print.css?param=20170112141615	304	stylesheet	index.aspx	738 B	29 ms	
satellite-5731b2a164746d2e02001bd5.js	200	script	satelliteLib-1...	(from ...)	3 ms	
satellite-5757bb7764746d64e200f469.js	200	script	satelliteLib-1...	(from ...)	5 ms	
satellite-56c7203c64746d4bf0000aa9.js	200	script	satelliteLib-1...	(from ...)	2 ms	
satellite-57d9466864746d192600e1.js	200	script	satelliteLib-1...	(from ...)	2 ms	
s-code-contents-81be4fc9258474b721bcc80...	200	script	satelliteLib-1...	(from ...)	2 ms	
csi?v=2&cs=mapsapi3&v3v=28.1&action=api...	(failed)	Other	0B	334 ms		
belfiusalternative_bold.woff?v=14c77daf	304	font	onloadwoffjs...	739 B	26 ms	

Summary: 129 requests | 262 KB transferred | Finish: 13.60 s | DOMContentLoaded: 1.04 s | Load: 1.07 s

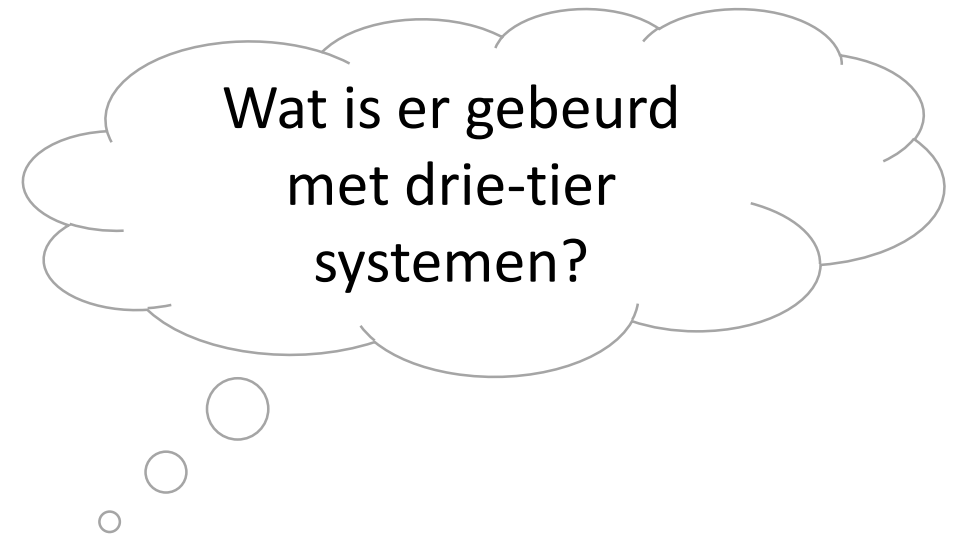
Frontend: Web

The image displays a web browser window showing the Belfius Bank website. The page title is "Kantoren Belfius Bank - Belfius". The main content area features a map of Brussels with markers for various Belfius branches. A search bar is visible at the top left, and a sidebar on the right lists different services like "Open TAXIS", "Op afspraak", "Geldautomaat", and "Gesloten".

Below the map, there is a section titled "Brussel Rogier Tower" with a Google Maps link. At the bottom of the page, there are links for "Algemene voorwaarden en informatie", "Tarieven", "Gedragscodes", "Cookiebeleid", and "Sitemap".

On the right side of the image, a network inspector (likely Chrome DevTools) is open, showing the "Network" tab. It displays a list of network requests, with the selected request being a JSON response from the Belfius API. The response contains a list of branch locations, each with details like "SSBOpenHours", "Longitude", "Latitude", "City", "ContactUrl", "CrossDeviceFlow", "Fax", "HasHomeAgent", "Id", "IsHtc", "IsNotHtc", "IsSelfServiceOpen", "Latitude", "Longitude", "Modified", "Name", "NeedsAttention", "Objective", "OfficeType", "OpenType", "Phone", "PostalCode", "SSBOpenHours", "SelfServiceTodayHours", "ShowContactButton", "ShowSelectButton", "Street", "Street2", "Street3", "TabbedPane", "Wifi", and "Wifi Indoor".

Frontend: Web



More than 10,000 exposed MongoDB databases deleted by ransomware groups ([PCWorld](#))

Five groups of attackers are competing to delete as many publicly accessible MongoDB databases as possible

Frontend: Web

- Java
- PHP
- ASP.NET
- Python
- JavaScript

Frontend: Mobile

- In de iOS wereld blijft Swift stevig doorgroeien
- Binnen Android begint Kotlin te groeien
- Aandacht voor cross-platform oplossingen



Frontend: Mobile

- **Swift blijft groeien**
 - Objective-C staat nog steeds hoog in de rangschikking
 - Doch Swift groeit gestaag verder
 - Makkelijker te leren syntax, betere structuur



Frontend: Mobile



- **Kotlin als “Java 8 op Java 6”?**
 - Kotlin werkt bovenop de Java 6 JVM
 - Lijkt sterk op Java maar
 - Veiliger, null-pointer bescherming
 - Korter, betere syntax
 - 1.1 RC uit
 - Sterke kandidaat binnen Android ecosysteem!

Frontend: Mobile

- **Cross-platform blijft belangrijk, doch niet “perfect”**
 - Grotendeels HTML 5 en web-gebaseerde toepassingen
 - Sencha, Corona, Qt, Xamarin
 - Unity3D, Cocos2D, Haxe
 - Niet dezelfde vlotheid en “UX” als een “native” app
 - Doch: “mobile web” technologie kan zeker een optie zijn

Frontend: UI

- **Java**

- Zeer trage adoptie van JavaFX

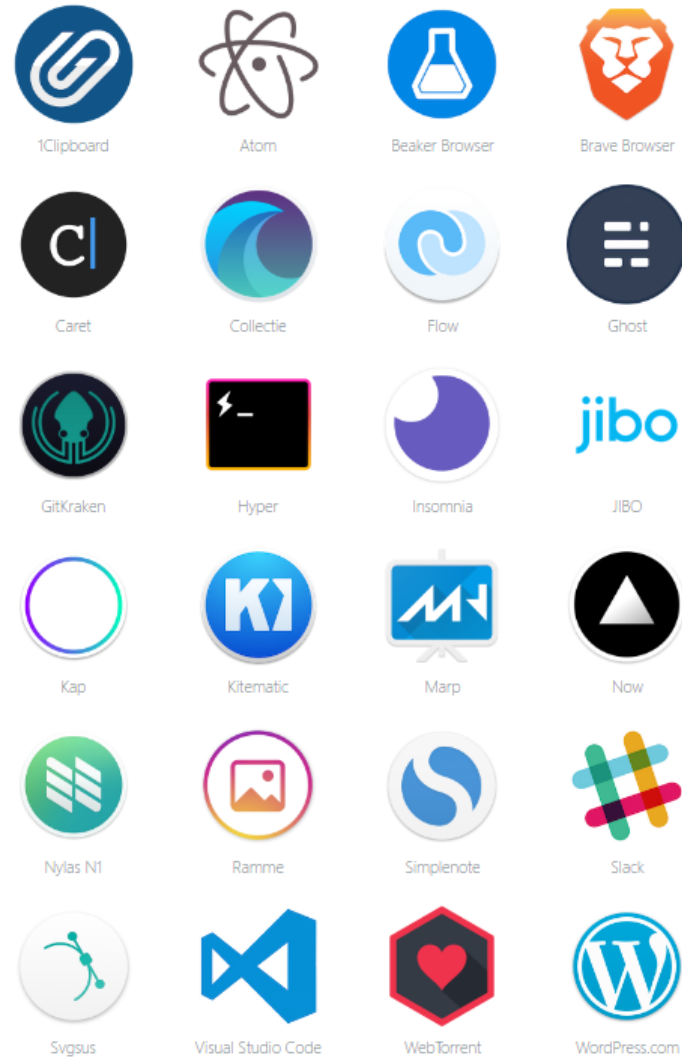
- **.NET**

- Windows Forms

- **Electron**

- “Build cross platform desktop apps with JavaScript, HTML, and CSS”
- Chromium + Node.JS
- Maar: verlies van “native” gevoel

Frontend: UI



Backend



Backend

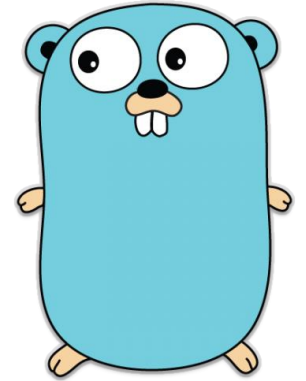
- **Traditionele talen blijven sterk**
 - Java
 - C++
 - C
 - C#
- Doch met evoluties
(lambda expressies in Java 8, coroutines in C++ 17)

Backend

- **SQL als database lingua franca**
 - Moeilijker binnen huidige generatie programmeurs
 - NoSQL, ORM, “fluent API’s”

Backend

- **“Veilige” alternatieven komen op als alternatief**
 - Go als even snelle, prettigere, veiligere C
 - Maar geen generics en andere minpunten
 - Rust als C/C++ vervanger
 - Memory safety, geen null-pointers
 - Maar C/C++ compilers evolueren ook (LLVM, Clang)
 - Nim als C/C++ vervanger
 - Crystal als snellere Ruby
 - Maar nog vrij jong / immatuur

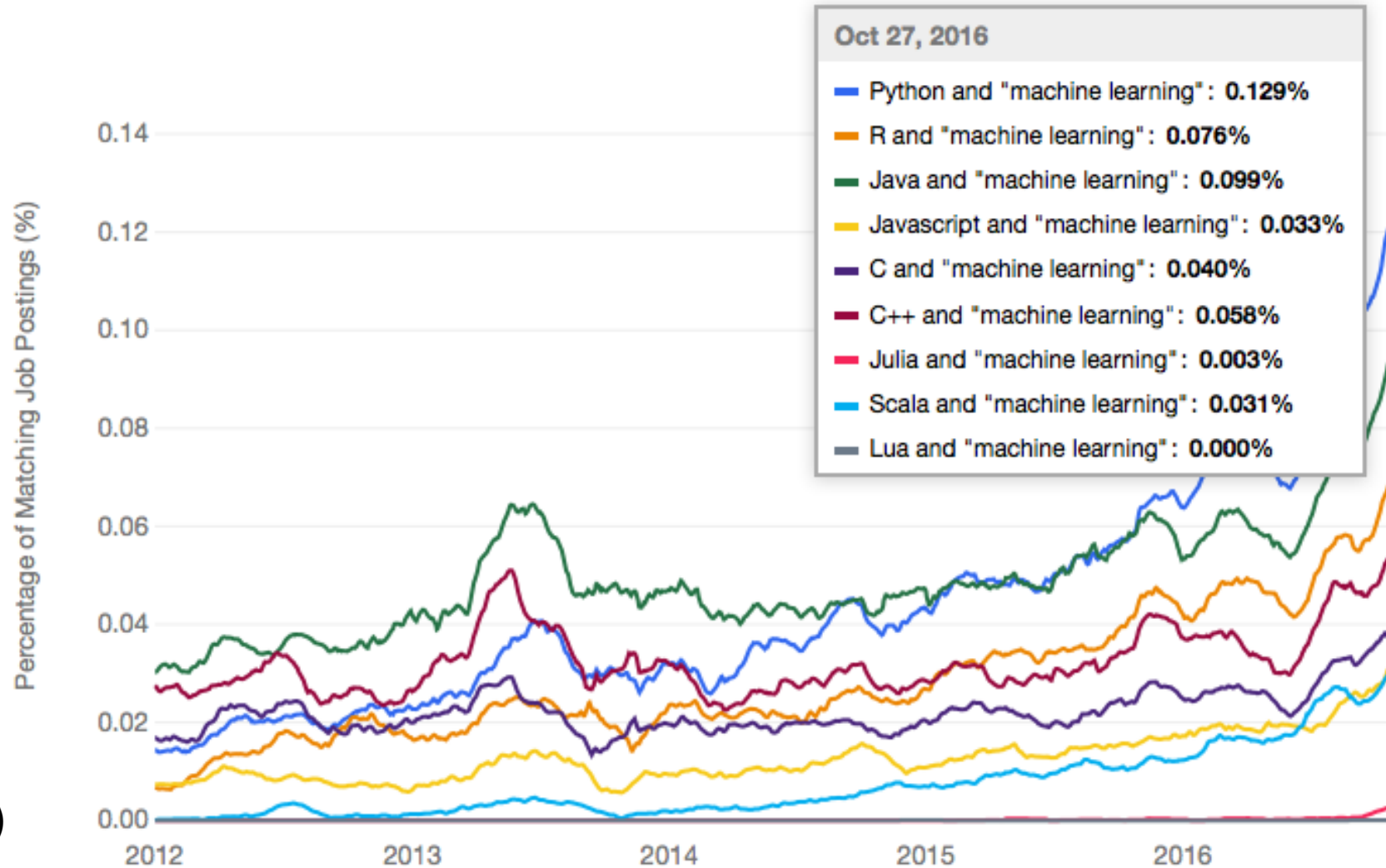


Niches

**When i wrote this code, only God
and i understood what it does**

Now, God only knows

Analytics, Big Data



(KDNuggets)

Analytics, Big Data

- **R en Python als meest-gebruikte open source talen**
 - Voor in-memory analytics
 - R in SQL Server
- **Naast commerciële producten**
 - SAS, SPSS, MatLAB, Mathematica



Analytics, Big Data



- **Julia als snellere R**

- Julia is a high-level, high-performance dynamic language for technical computing
- Zeer snel maar ecosysteem nog niet zo compleet
- Trage adoptie

- **Lua in deep-learning**

- Libraries zoals Torch

Analytics, Big Data



- **In de Big Data wereld blijven Python, Java en Scala belangrijk**
 - Scala als object-georiënteerd + functioneel
 - Java en Spark voornamelijk als “natieve taal” in Spark / Hadoop omgevingen
 - Voordeel JVM in termen van deployment
 - .NET kent hier minder ondersteuning
- **SQL als missing gap**
 - Nood aan grondige opleiding

Concurrency, schaalbaarheid



- **Talen zoals Erlang en Go als nieuwe “systems programming”**
 - “Build massively scalable soft real-time systems”
 - Telefonie, messaging, e-commerce
 - “Let it crash” principe
 - “The view is that you don't need to program defensively. If there are any errors, the process is automatically terminated, and this is reported to any processes that were monitoring the crashed process. In fact, defensive programming in Erlang is frowned upon”
 - “Chaos monkey”
 - “Always be able to ‘just restart’”

Functionele talen

- **Functionele talen en LISP dialecten zullen langzaam blijven doorgroeien**
 - F#, Racket, Ocaml, Haskell, Scheme
 - “What you want” en niet “How you want it”
 - Recursie, lambda functies, currying
 - Sterke wiskundige ondersteuning
 - Correctheid en concurrency
 - Doch lage adoptie in industrie, buiten specifieke toepassingen
 - Sommige functionaliteit vind haar weg naar OOP talen: Java, Swift, Scala, PHP
 - functies als “first class citizens”

Mainframe programming

- **Talen zoals COBOL en Fortran blijven bestaan**
 - OpenCobol, GNUCobol als moderne IDEs
 - COBOL brug naar Node.JS
 - COBOL in Docker, in MicroServices

Wrap-up

Conclusies

- Traditionele talen zoals Java, C, C++ en C# blijven sterk in gebruik
 - Goed gekend en getest
 - Langzame vernieuwing
- JavaScript blijft doorgroeien op het web, servers, en desktopapplicaties
 - Enorme stijging in populariteit
 - Voordelen afwegen tegen nadelen
- Dynamische talen zoals Python en PHP blijven populair
 - Lage barrière
 - Type hint uitbreidingen

Conclusies

- Talen zoals Go en Rust beginnen te groeien als systeemtalen
 - Concurrency, null-pointers, thread en memory safety
- Swift en Kotlin worden in mobiele wereld belangrijker
 - Samen met web-gebaseerde ontwikkelomgevingen
- Binnen analytics blijven R, Python, Java belangrijk
 - Scala en Julia als interessante ontwikkelingen
- Functionele talen blijven grotendeel “hobby”-talen maar bieden antwoord voor unieke behoeften
 - Fault tolerance, self-healing

Q & A

