



Van Ada tot Zig

Welke programmeertalen voor wat?

Prof. dr. Seppe vanden Broucke

26 mei 2021



seppe vanden broucke



Seppe vanden Broucke

Professor @ Faculty of Economics and Business Administration, Ugent
Department of Business Informatics

Guest Lecturer @ Faculty of Economics and Business, KU Leuven
Management Informatics Department

Research:

- Business Analytics (fraud, marketing and geospatial analytics)
- Process Mining and BPM
- Big Data
- Met Python, Java, R, Julia, ...

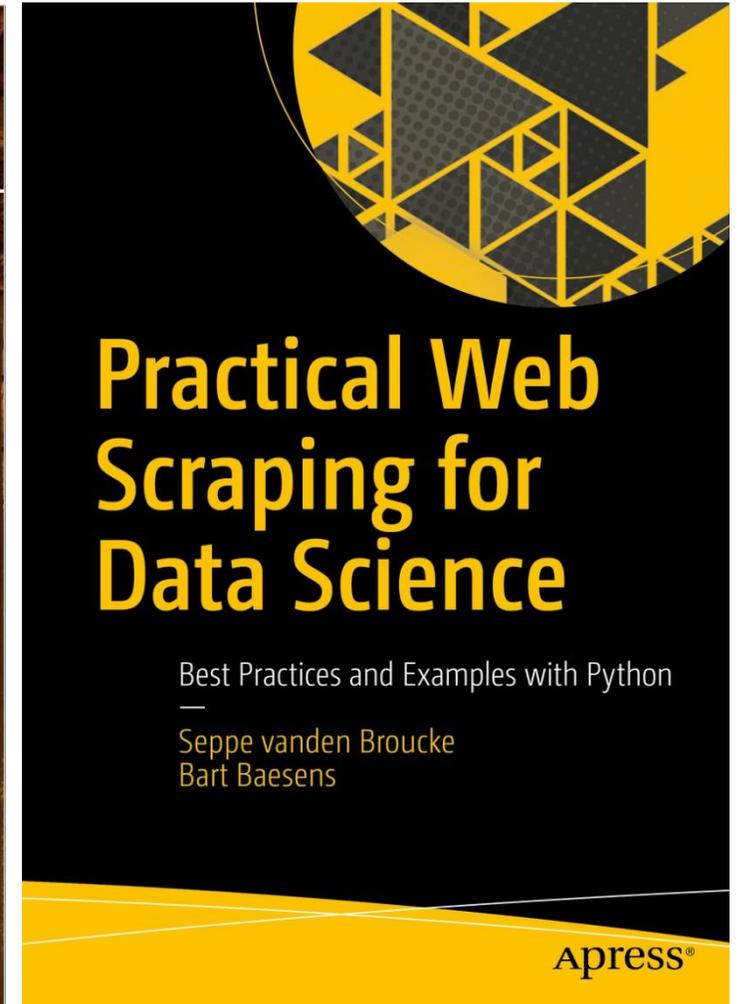
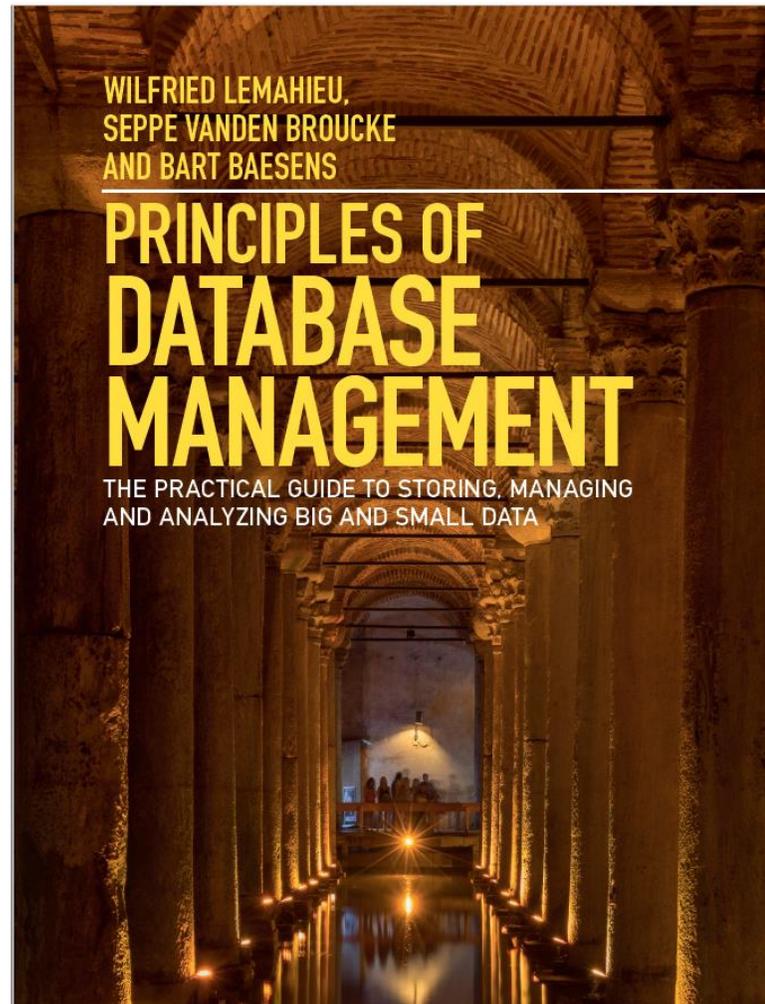
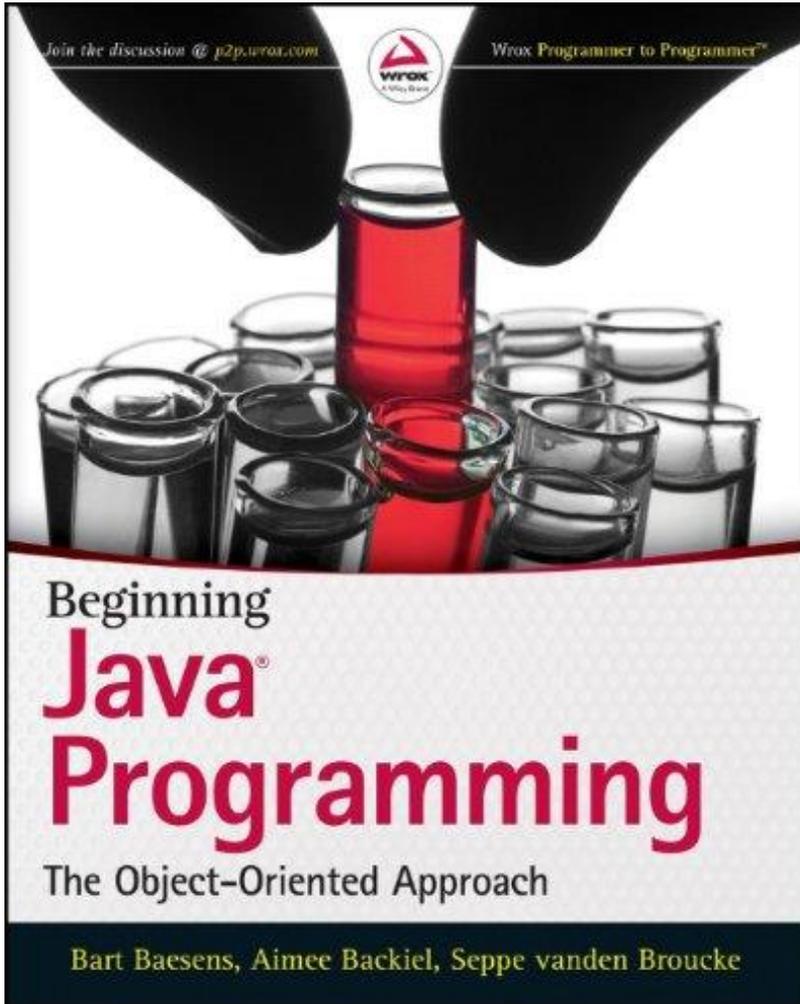
Contact:

seppe.vandenbroucke@ugent.be

www.processmining.be

www.dataminingapps.com

www.seppe.net



Een paar jaren geleden

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%

Een paar jaren geleden



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



Met welke taal / talen werkt u / uw organisatie momenteel?



Met welke taal zou u graag leren
werken / mee experimenteren /
uitproberen?



Met welke taal wilt u liever niet langer werken / wilt u uitfaseren?

Een paar jaren geleden

- Traditionele talen zoals Java, C, C++ en C# blijven sterk in gebruik
- JavaScript blijft doorgroeien op het web, servers, en desktopapplicaties
- Dynamische talen zoals Python en PHP blijven populair
- Talen zoals Go en Rust beginnen te groeien als systeemtalen
- Swift en Kotlin worden in mobiele wereld belangrijker
- Binnen analytics blijven R, Python, Java belangrijk
- Functionele talen blijven grotendeel “hobby”-talen maar bieden antwoord voor unieke behoeften

Vandaag

- IEEE en Tiobe index, Stack overflow developers survey, PYPL, RedMonk Ranking en GitHub State of the Octoverse

PYTHON VOERT LIJST PROGRAMMEERTALEN AAN

Python, gevolgd door Java en vervolgens JavaScript. Dat is de top drie in de lijst van populairste programmeertalen. Typescript is de grote stijger.

Ranking	Ranking in 2020	Programmeertaal	Totaal	Vershil met 2020
1	(1)	Python	113	1%
2	(2)	Java	109	-2%
3	(3)	JavaScript	108	1%
4	(4)	C++	91	-3%
5	(6)	C	88	-1%
6	(7)	PHP	80	-2%
7	(5)	C#	79	-12%
8	(9)	Ruby	55	10%
9	(15)	Typescript	52	68%
10	(10)	R	50	2%
11	(11)	GO	44	-6%
12	(8)	Swift	44	-23%
13	(14)	Shell	33	3%
14	(16)	Matlab	26	-7%
15	(13)	SQL	26	-28%
16	(12)	Objective-C	25	-36%
17	(19)	Kotlin	22	10%
18	(21)	Visual Basic	15	15%
19	(18)	Scala	15	-25%
20	(-)	Arduino	14	-
21	(17)	Assembly	12	-43%
22	(22)	Dart	10	67%
23	(23)	Rust	9	50%
24	(24)	Perl	5	0%
25	-	Julia	2	-

Nieuwe evoluties

- Toe of afname van het object georiënteerd paradigma?
- Wat met de toekomst van JavaScript?
- Draait Windows binnenkort volledig op Rust?
- Is Swift aan de terugtocht begonnen?
- Wordt Python niet hoe langer hoe meer een rommeltje?
- Is Oracle Java volledig aan het doodknijpen?

Onze gids



Frontend

Web
Mobile
Desktop

Systems
Databases
Processing



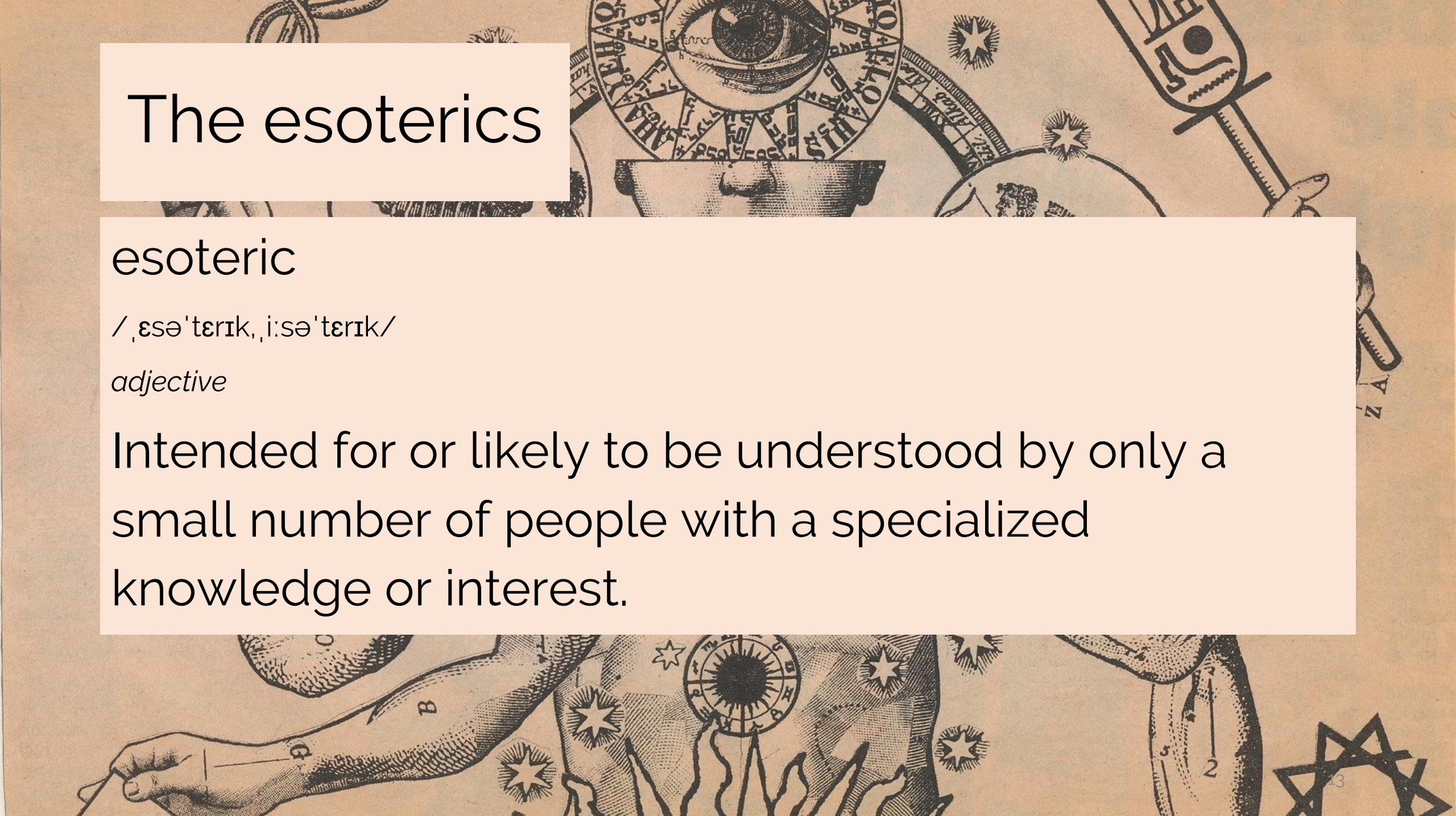
Backend



Niche

Analytics & Big Data
Concurrent systems
High-scalability
Others

The esoterics



esoteric

/,ɛsə'tɛrɪk, i:sə'tɛrɪk/

adjective

Intended for or likely to be understood by only a small number of people with a specialized knowledge or interest.

Brainfuck, Urban Müller, 1993

```
+++++[>++++[>+>+++>+++>+<<<<
-]>+>+>->>+ [<]<-]>>.>---.+++++
..+++.>>.<-.<..+++.------.------
-.>>+.>+.
```

Een taal waarvoor de compiler zo klein als mogelijk kon zijn (voor Amiga OS). Compiler werd uiteindelijk 240 bytes, taal zelf 8 symbolen en Turing-compleet. Geïnspireerd door FALSE, (1024-byte compiler).

INTERCAL, Don Woods & James Lyon, 1972

```
DO ,1 <- #13
PLEASE DO ,1 SUB #1 <- #238
DO ,1 SUB #2 <- #108
DO ,1 SUB #3 <- #112
DO ,1 SUB #4 <- #0
DO ,1 SUB #5 <- #64
DO ,1 SUB #6 <- #194
DO ,1 SUB #7 <- #48
PLEASE DO ,1 SUB #8 <- #22
DO ,1 SUB #9 <- #248
DO ,1 SUB #10 <- #168
DO ,1 SUB #11 <- #24
DO ,1 SUB #12 <- #16
DO ,1 SUB #13 <- #162
PLEASE READ OUT ,1
PLEASE GIVE UP
```

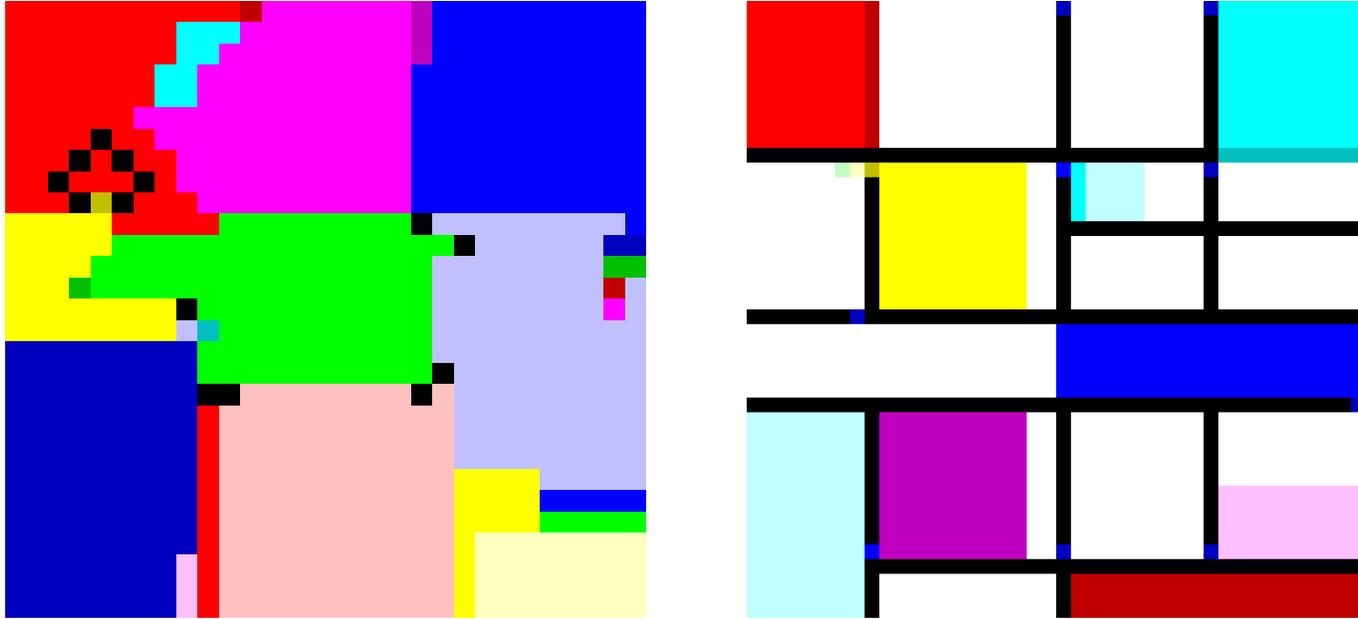
Compiler Language With No Pronounceable Acronym (INTERCAL) werd gemaakt als parodie (voornamelijk van COBOL). Satire van gangbare talen rond 60-70s.

Befunge, Chris Pressey, 1993

```
&>:1-:v v *_$.@  
^      _$>\:^
```

FALSE en Brainfuck wilden een zo klein als mogelijke compiler, terwijl Befunge het maken van een compiler zo moeilijk als mogelijk wou maken. De belangrijkste manier is door de taal "2d" te maken.

Piet, David Morgan-Mar, 2003



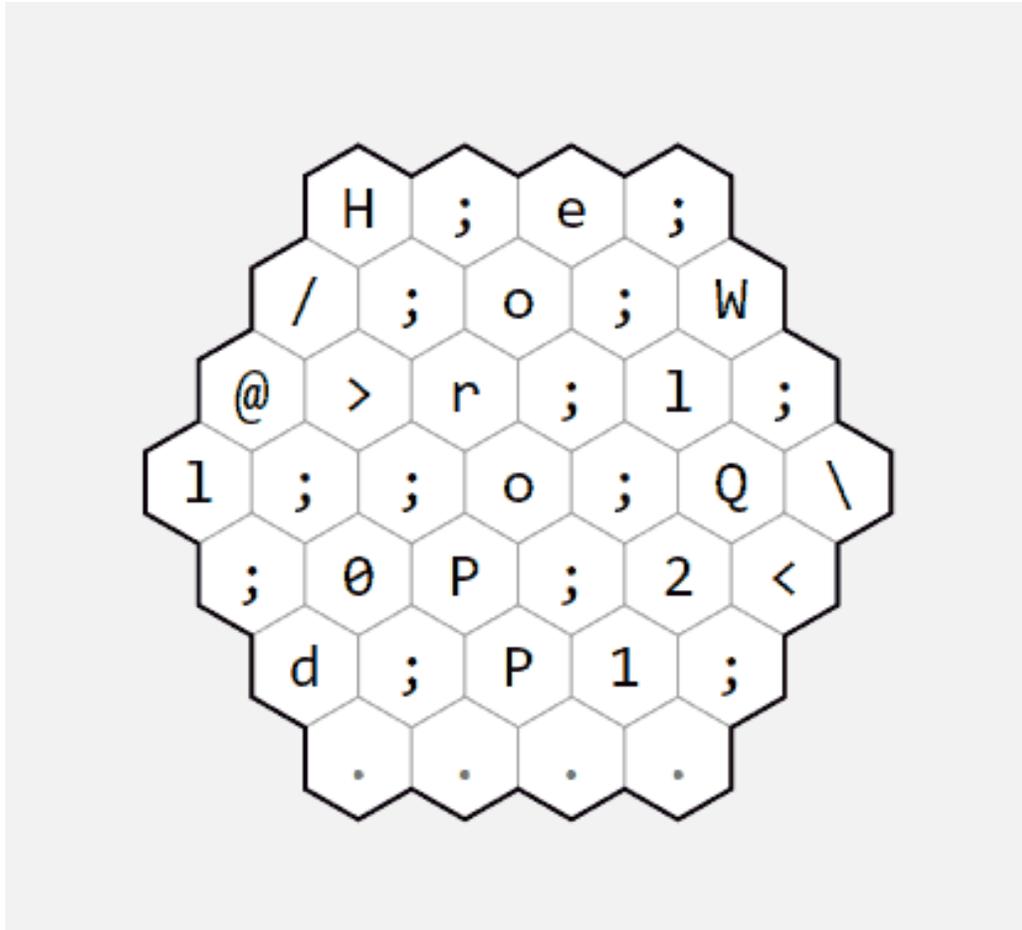
Piet is hier een evolutie van: visueel en 2d.

ArnoldC, Lauri Hartikka, 2013

```
40 ENOUGH TALK
41 BULLSHIT
42 GET YOUR ASS TO MARS mod5
43 DO IT NOW modulo n 5
44 GET TO THE CHOPPER divisibleBy5
45 HERE IS MY INVITATION mod5
46 YOU ARE NOT YOU YOU ARE ME 0
47 ENOUGH TALK
48 BECAUSE I'M GOING TO SAY PLEASE divisibleBy5
49 GET TO THE CHOPPER sum
50 HERE IS MY INVITATION sum
51 GET UP n
52 ENOUGH TALK
53 YOU HAVE NO RESPECT FOR LOGIC
54 YOU HAVE NO RESPECT FOR LOGIC
55 CHILL
56 TALK TO THE HAND "sum:"
57 TALK TO THE HAND sum
58 YOU HAVE BEEN TERMINATED
59
60 LISTEN TO ME VERY CAREFULLY modulo
61 I NEED YOUR CLOTHES YOUR BOOTS AND YOUR MOTORCYCLE dividend
62 I NEED YOUR CLOTHES YOUR BOOTS AND YOUR MOTORCYCLE divisor
63 GIVE THESE PEOPLE AIR
64 HEY CHRISTMAS TREE quotient
65 YOU SET US UP 0
66 HEY CHRISTMAS TREE remainder
67 YOU SET US UP 0
68 HEY CHRISTMAS TREE product
69 YOU SET US UP 0
70 GET TO THE CHOPPER quotient
71 HERE IS MY INVITATION dividend
72 HE HAD TO SPLIT divisor
73 ENOUGH TALK
74 GET TO THE CHOPPER product
75 HERE IS MY INVITATION divisor
76 YOU'RE FIRED quotient
77 ENOUGH TALK
78 GET TO THE CHOPPER remainder
79 HERE IS MY INVITATION dividend
80 GET DOWN product
81 ENOUGH TALK
82 I'LL BE BACK remainder
83 HASTA LA VISTA, BABY
84
```

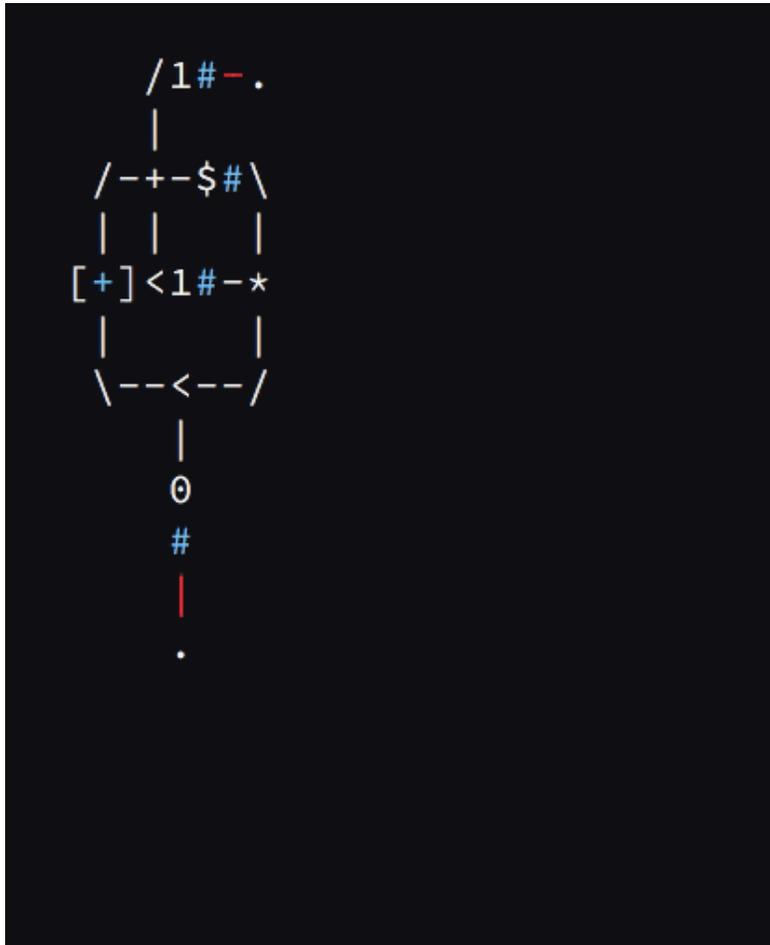
Gebaseerd op de one-liners van Arnold Schwarzenegger.

Hexagony, Martin Ender, 2015



Hexagony (hexagon + agony) is een 2d taal die op een hexagonal-grid wordt geplaatst en 6 i.p.v. 1 instruction pointers gebruikt.

AsciiDots, Aaron Janse, 2017



AsciiDots is gebaseerd op ascii art en laat dots bewegen over een schakeling van instructies.

ide.wy-lang.org
🏠 🔍 ↻

文言 Wenyan +

- 嵌套调用示例 nested_fun
- 对象示例 obj
- 賈憲三角 pascal_triangle
- 萊布尼茲圓周率估算 pi_lei
- 劉徽割圓術求圓周率 pi_liu
- 快速排序 quicksort
- 快速排序二 quicksort_inple
- 自產生程式 quine
- 自產生程式二 quine2
- 選擇排序 selectionsort
- 渾沌經 serialization
- 埃氏篩 sieve
- 牛頓求根法 sqrt_newton
- 畫樹 tree
- 畫樹二 tree2
- 異常處理示例 try
- 圖靈機 turing
- 增乘開平方 zh_sqrt
- ▶ 文淵閣 Packages
- 交互秘術 by GLanguage
- 刻漏 by akira-cn
- 子曰 by antfu
- 柯裡化法 by akira-cn
- 物之類 by XingZiLong
- 符經 by GLanguage
- 簡體秘術 by lymolive
- 腳本秘術 by akira-cn
- 解析整數 by alainsaas
- 質問 by alainsaas
- 造類秘術 by GLanguage
- 連加連乘 by XingZiLong

畫樹二 *by examples* ▶ Compile

```

1 吾嘗觀「畫譜」之書。方悟「備紙」「擇筆」「蘸色」「落筆」「運筆」「提筆」「裱
2 吾嘗觀「算經」之書。方悟「正弦」「餘弦」「半圓周率」「四分圓周率」之義。
3 吾嘗觀「易經」之書。方悟「運」「占」之義。
4
5 施「運」於十六。
6
7 吾有一術。名之曰「畫樹法」。欲行是術。必先得一物。曰「紙」。
8 六數。曰「東」「南」「長」「粗」「向」「向變」。是術曰。
9
10 施「餘弦」於「向」。乘其以「長」。加其以「東」。名之曰「末東」
11 施「正弦」於「向」。乘其以「長」。減其於「南」。名之曰「末南」
12
13 減「末南」以二。名之曰「雪上」。
14 減「南」以二。名之曰「雪下」。
15
16 施「擇筆」於「紙」於「粗」。
17 施「蘸色」於「紙」於「鈿白」。
18 施「落筆」於「紙」於「東」於「雪下」。
19 施「運筆」於「紙」於「末東」於「雪上」。
20 施「提筆」於「紙」。
21
22 施「擇筆」於「紙」於「粗」。
23 施「蘸色」於「紙」於「黑」。
24 施「落筆」於「紙」於「東」於「南」。
25 施「運筆」於「紙」於「末東」於「末南」。
26 施「提筆」於「紙」。
27
28 加「向」以零又三分。乘其以零又五分。加其以「四分圓周率」。名之曰「左向」。
29 減「向」以零又三分。乘其以零又五分。加其以「四分圓周率」。名之曰「右向」。
30 施「占」。減其以零又五分。乘其以「向變」。名之曰「左歪」。
31 施「占」。減其以零又五分。乘其以「向變」。名之曰「右倒」。
32

```

Compiled ▶

```

1 /* module 易經 is hidden */
2 /* module 算經 is hidden */
3 /* module 畫譜 is hidden */
4 var 備紙 = 畫譜.備紙;
5 var 擇筆 = 畫譜.擇筆;
6 var 蘸色 = 畫譜.蘸色;
7 var 落筆 = 畫譜.落筆;
8 var 運筆 = 畫譜.運筆;
9 var 提筆 = 畫譜.提筆;
10 var 裱畫 = 畫譜.裱畫;
11 var 正弦 = 算經.正弦;
12 var 餘弦 = 算經.餘弦;
13 var 半圓周率 = 算經.半圓周率;
14 var 四分圓周率 = 算經.四分圓周率;
15 var 運 = 易經.運;
16 var 占 = 易經.占;
17 const _ans1 = 運(16);
18 var 畫樹法 = _ => {};
19 畫樹法 = 紙 => 東 => 南 => 長 => 粗 => 向 => 向變 => {
20   const _ans2 = 餘弦(向);
21   const _ans3 = _ans2 * 長;
22   const _ans4 = _ans3 + 東;
23   var 末東 = _ans4;
24   const _ans5 = 正弦(向);
25   const _ans6 = _ans5 * 長;
26   const _ans7 = 南 - _ans6;
27   var 末南 = _ans7;
28   const _ans8 = 末南 - 2;
29   var 雪上 = _ans8;
30   const _ans9 = 南 - 2;
31   var 雪下 = _ans9;
32   const ans10 = 擇筆(紙)(粗);

```

Output Render Book



- https://esolangs.org/wiki/Language_list
- <https://esoteric.codes/>
- <https://www.hillelwayne.com/talks/esolangs/>

Frontend



Frontend



Web



Mobile



Desktop

Frontend: Web

- **In het verleden eerder een backend/server-side verhaal**
 - Frontend was simpelweg HTML+CSS
 - Met jQuery voor wat client-side opsmukwerk
 - PHP, ASP, JSP
 - Ruby on Rails als “disruptor”
- **Vandaag: single page apps als afnemer van server-side API**
 - JavaScript frameworks
 - “Backend” nog wat gevarieerder

Ranking	Ranking in 2020	Programmeertaal	Totaal	Vershil met 2020
1	(1)	Python	113	1%
2	(2)	Java	109	-2%
3	(3)	JavaScript	108	1%
4	(4)	C++	91	-3%
5	(6)	C	88	-1%
6	(7)	PHP	80	-2%
7	(5)	C#	79	-12%
8	(9)	Ruby	55	10%
9	(15)	Typescript	52	68%
10	(10)	R	50	2%
11	(11)	GO	44	-6%
12	(8)	Swift	44	-23%
13	(14)	Shell	33	3%
14	(16)	Matlab	26	-7%
15	(13)	SQL	26	-28%
16	(12)	Objective-C	25	-36%
17	(19)	Kotlin	22	10%
18	(21)	Visual Basic	15	15%
19	(18)	Scala	15	-25%
20	(-)	Arduino	14	-
21	(17)	Assembly	12	-43%
22	(22)	Dart	10	67%
23	(23)	Rust	9	50%
24	(24)	Perl	5	0%
25	-	Julia	2	-

Frontend: Web



- **PHP blijft een veelgebruikte taal op het web**
 - Relatief makkelijk te leren, groot ecosysteem en community
 - PHP 7 voegt interessante functies toe zoals type hints en een algemene “opschoning” van de standard library
 - PHP 8 voegt een nieuwe JIT compiler, union types, en null safety checks toe
 - PHP 4 versus 5, 6, 7 en 8 is een probleem
 - En: hoeveel wordt gedreven door legacy code (en WordPress...)?
 - Paradigma-gewijs is de taal een complete melting pot

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

Ending PHP Support, and The Future Of Hack

Posted September 12, 2018

Frontend: Web



- **Gebruik van Ruby on Rails lijkt ook stabiel**

- Vernieuwingen inzake ORM e.d. geadopteerd door andere ecosystemen
- Migratie terug naar PHP en diens web-frameworks (Symfony, Zend)
- Overschakeling naar Python (Django etc.)
- Echter wel een samenhangende interessante visie op het bouwen van moderne web-apps (Stimulus -> Turbolinks -> Hotwire/Turbo)
 - "Hotwire is an alternative approach to building modern web applications without using much JavaScript by sending HTML instead of JSON over the wire. This makes for fast first-load pages, keeps template rendering on the server, and allows for a simpler, more productive development experience in any programming language, without sacrificing any of the speed or responsiveness associated with a traditional single-page application."
 - Zie bijvoorbeeld ook <https://htmx.org/>

Frontend: Web



- **De beweging naar “lean” in het Java ecosystemen heeft impact wat gemist**
 - (Langzaam) weg van “zware EE stacks” (JSF + Struts + Spring MVC + Hibernate + Tapestry) naar lichtere omgevingen
 - HikariCP en jOOQ als moderne DB libraries
 - Spark Framework, Play Framework en Ratpack als moderne web frameworks
 - Maar enigszins verwaterd door grote verschillen en wijzigingen in Java (8, 11, 15) en vage licentiestructuur

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~~
 - Nu een sterke combo bovenop Linux
 - Knappe ORM componenten zoals Entity Framework en Dapper bovenop ASP.NET MVC

Frontend: Web



- **Ook Python blijft populair**

- Maar dan vooral als backend REST API provider
- E.g. FastAPI, Flask en Django
- Ook hier mede dankzij sterke ORM componenten en type checking functionaliteit

Frontend: Web



- **JavaScript zet de groei verder door**

- Node.JS, EcmaScript 2021, TypeScript, Elm, Dart, CoffeeScript...
- "Single-page apps", reactive programming
- Als API-afnemer voor de SPA maar ook meer om meer server-side JavaScript
- Angular, React, Vue.JS, Next.JS, Nuxt.JS
- Leren leven met het JavaScript "vermoeidheidssyndroom"

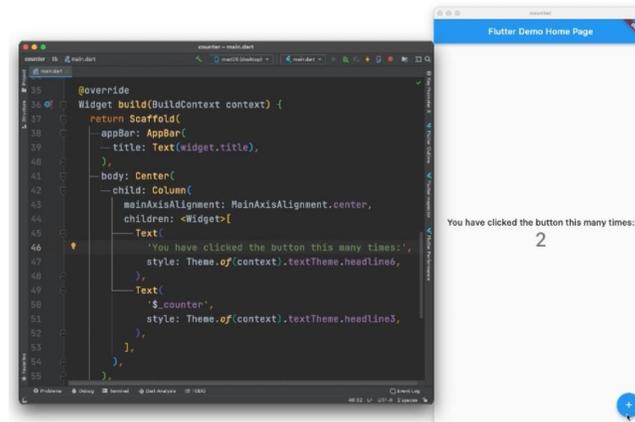


Frontend: Web

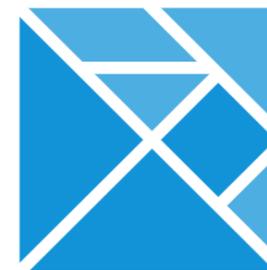


- **TypeScript als stichtend voorbeeld**

- Classen, types, ... weg aangeduid voor EcmaScript 2020 en 2021
- CoffeeScript vrijwel compleet vervangen
- Dart + Flutter (Google) ook interessant (web, desktop en mobiel)

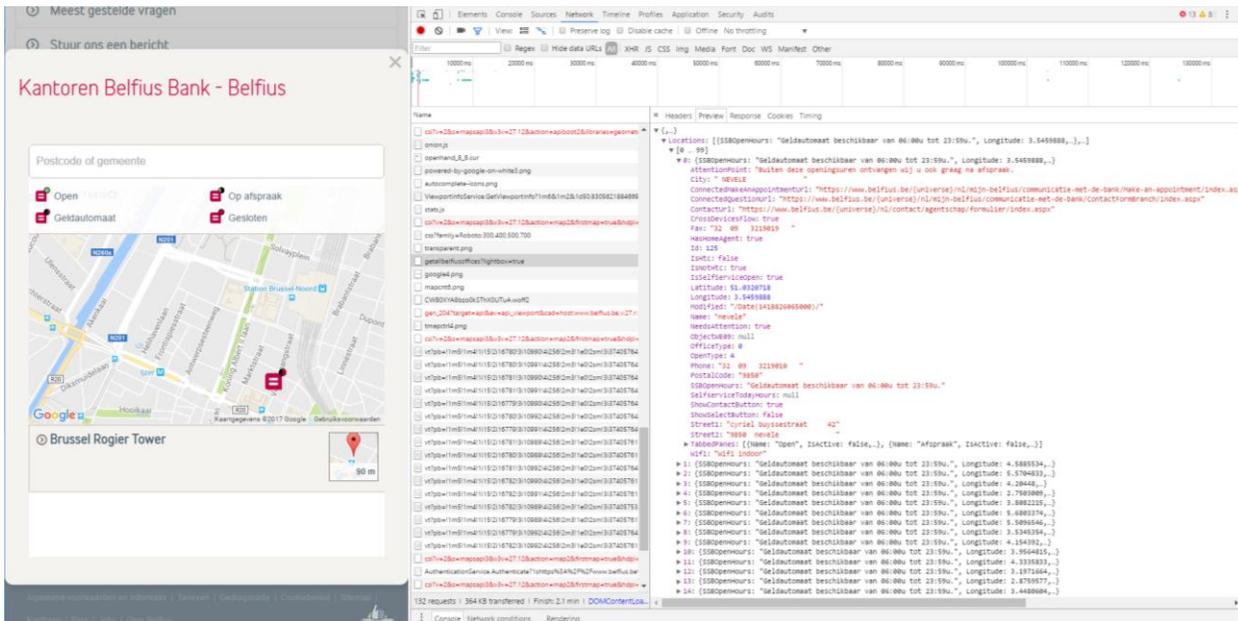


Frontend: Web



- **Elm als interessant buitenbeentje**

- “Elm is a functional language that compiles to JavaScript. It helps you make websites and web apps. It has a strong emphasis on simplicity and quality tooling.”
- “Elm is totally new and different. It couldn't be this good any other way. The hardest thing you'll have to do is unlearn all the OO patterns you use to handle problems that are (together!) nonexistent in Elm. Seriously.”
- “If you are already using React or Angular in a functional reactive way, you'll find that Elm is where everybody got their ideas from and how they look like in their purest form.”



[*]Parsing Search Engines

Interesting URL found:
[+]https://hacker-news.firebaseio.com/v0/topstories.json

site not reachable
Interesting URL found:
[+]https://hacker-news.firebaseio.com/v0/item/9884165.json

site not reachable
Interesting URL found:
[+]https://hacker-news.firebaseio.com/v0/jobstories.json

site not reachable
Interesting URL found:
[+]https://hacker-news.firebaseio.com/v0/beststories.json

site not reachable
Interesting URL found:
[+]https://hacker-news.firebaseio.com/v0/askstories.json

site not reachable
Interesting URL found:
[+]https://hacker-news.firebaseio.com/v0/item/9321149.json

site not reachable
Interesting URL found:
[+]https://hacker-news.firebaseio.com/v0/item/15723926.json

site not reachable
Interesting URL found:
[+]https://hacker-news.firebaseio.com/v0/item/14751340.json

site not reachable
Interesting URL found:
[+]https://hacker-news.firebaseio.com/v0/item/12154137.json

site not reachable
Interesting URL found:
[+]https://hacker-news.firebaseio.com/v0/item/8863.json

Dependency Confusion: How I Hacked Into Apple, Microsoft and Dozens of Other Companies

The Story of a Novel Supply Chain Attack



Alex Birsan Feb 9 · 11 min read



Frontend: Web

- **Andere client side talen?**
WebAssembly?

 .Net
 AssemblyScript
 Astro Unmaintained
 Brainfuck
 C
 C#
 C++
 Clean
 Co
 COBOL
 D
 Eel
 Elixir
 F#
 Faust

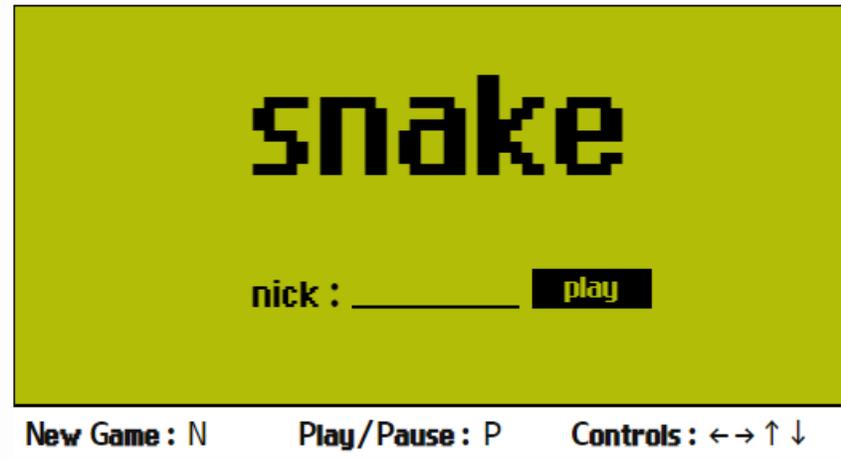
 Forest
 Forth
 Go
 Grain
 Haskell
 Java
 JavaScript
 Julia
 Idris Unmaintained
 Kotlin/Native
 Kou
 Lisp
 Lobster
 Lua
 Lys
 Never
 Nim
 Ocaml

 Perl
 PHP
 Plorth
 Poetry
 Python
 Prolog
 Ruby
 Rust
 Scheme
 Scopes
 Swift
 TypeScript
 Zig

<https://github.com/appcypher/awesome-wasm-langs>

Frontend: Web

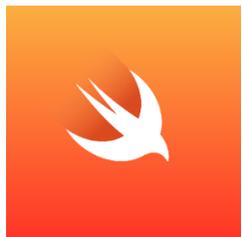
- Andere client side talen? WebAssembly?



<https://nim-lang.org/features.html>

Frontend: Mobile

- **Java, Objective-C en Swift verliezen terrein**
 - Aandacht voor cross-platform oplossingen voornamelijk via "native" JS raamwerken
- Kotlin heeft wel een stevige plaats binnen Android veroverd



Ranking	Ranking in 2020	Programmeertaal	Totaal	Vershil met 2020
1	(1)	Python	113	1%
2	(2)	Java	109	-2%
3	(3)	JavaScript	108	1%
4	(4)	C++	91	-3%
5	(6)	C	88	-1%
6	(7)	PHP	80	-2%
7	(5)	C#	79	-12%
8	(9)	Ruby	55	10%
9	(15)	Typescript	52	68%
10	(10)	R	50	2%
11	(11)	GO	44	-6%
12	(8)	Swift	44	-23%
13	(14)	Shell	33	3%
14	(16)	Matlab	26	-7%
15	(13)	SQL	26	-28%
16	(12)	Objective-C	25	-36%
17	(19)	Kotlin	22	10%
18	(21)	Visual Basic	15	15%
19	(18)	Scala	15	-25%
20	(-)	Arduino	14	-
21	(17)	Assembly	12	-43%
22	(22)	Dart	10	67%
23	(23)	Rust	9	50%
24	(24)	Perl	5	0%
25	-	Julia	2	-

Frontend: Desktop

- **Java**
 - Amper adoptie van JavaFX
- **.NET**
 - Windows Forms, maar zelfs door Microsoft wat minder gebruikt
- **Electron**
 - Chromium + Node.JS
- **React Native**
- **Flutter**



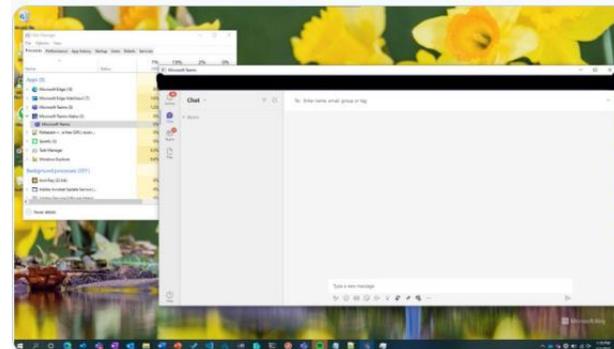
React Native at Microsoft
@ReactNativeMSFT

We ship both! The updated inbox Windows Mail/Calendar apps were written from the ground-up using RN. Within Office, there are several mission critical components that have migrated to RN - one example is the Comments experience in Word.



Jesse Mikael Järvi
@jessejarvi

Performance is better with [#EdgeWebView2](#) compared to the Electron pile of garbage. Native code would be a lot better tho. [#MSTeams](#) [#MSTeamsV2](#)



Backend



Backend

- **Traditionele talen blijven sterk**

- Java
- C++
- C
- C#
- Python

- Doch met evoluties

Ranking	Ranking in 2020	Programmeertaal	Totaal	Vershil met 2020
1	(1)	Python	113	1%
2	(2)	Java	109	-2%
3	(3)	JavaScript	108	1%
4	(4)	C++	91	-3%
5	(6)	C	88	-1%
6	(7)	PHP	80	-2%
7	(5)	C#	79	-12%
8	(9)	Ruby	55	10%
9	(15)	Typescript	52	68%
10	(10)	R	50	2%
11	(11)	GO	44	-6%
12	(8)	Swift	44	-23%
13	(14)	Shell	33	3%
14	(16)	Matlab	26	-7%
15	(13)	SQL	26	-28%
16	(12)	Objective-C	25	-36%
17	(19)	Kotlin	22	10%
18	(21)	Visual Basic	15	15%
19	(18)	Scala	15	-25%
20	(-)	Arduino	14	-
21	(17)	Assembly	12	-43%
22	(22)	Dart	10	67%
23	(23)	Rust	9	50%
24	(24)	Perl	5	0%
25	-	Julia	2	-

Backend

- **SQL als database lingua franca**

- Met de aankomst van NewSQL en cloud databases zelfs een revival
- Time series data bases, geospatial data bases, graph data bases met hun bespoke talen

- En nieuwe evoluties:

```
@OrderBy(Mentions, "mentions desc");  
@Limit(Mentions, 10);  
Mentions(person:, mentions? += 1) distinct :-  
  gdelt-bq.gdeltv2.gkg(persons:, date:),  
  Substr(ToString(date), 0, 4) == "2020",  
  the_persons == Split(persons, ";"),  
  person in the_persons;
```

Logica is an open source declarative logic programming language for data manipulation. Logica is a successor to [Yedalog](#), a language created at Google earlier.

Logica is for engineers, data scientists and other specialists who want to use logic programming syntax when writing queries and pipelines to run on [BigQuery](#).

Logica compiles to SQL and gives you access to the power of Google BigQuery engine with the convenience of logic programming syntax. This is useful because BigQuery is magnitudes more powerful than state of the art native logic programming engines.

Backend

- **“Veilige” alternatieven in volle opmars**
- **Rust als C/C++ vervanger**
 - Memory safety, geen null-pointers
 - Voted StackOverflow's Most Loved Programming Language
 - The Linux project is currently considering a proposal to add Rust as the second language to the kernel to enable writing safer driver and kernel-space code
 - Google has announced that it now supports building low level components of the Android OS in Rust
 - Microsoft: Here's why we love programming language Rust and kicked off Project Verona
 - Maar C/C++ compilers evolueren ook (LLVM, Clang)
- **Go als even snelle, prettigere, veiligere C**
 - Maar ondertussen hier en daar al afgeschreven



Niche

```
[0] BKPROP1 I;INPUT;OUTPUT;O;OUT;E;d;NT;NO;ER;N;E1
[1] E←0.02
[2] NT←1++/N←LAYERS
[3] L:'Input value: ',⊘IN[I;]
[4] INPUT←1,IN[I;],(N[2]+N[3])ρ0
[5] 'Output value: ',⊘OU[I;]
[6] OUTPUT←OU[I;]
[7] L1:O←(-N[3])†OUT←CONEC COMPUTE INPUT
[8] ER←0.5×+/(E1←O-OUTPUT)*2
[9] →(ER<1E-10)/O
[10] d←E×OUT◦.×E1
[11] NO←1+N[1]+N[2]+ιN[3]
[12] d←d×CONEC[;NO]≠0
[13] CONEC[;NO]←CONEC[;NO]-d
[14] E1←(-NT)†E1
[15] NO←(v/d≠0)/ι†ρd
[16] d←E×OUT◦.×(CONEC+.×E1)[NO]
[17] d←d×CONEC[;NO]≠0
[18] CONEC[;NO]←CONEC[;NO]-d
[19] →L1
```

ML, AI, Big Data

- **Python als meest-gebruikte taal**
 - ML en deep learning raamwerken
- **R wat op de terugweg**
- **Commerciële producten omarmen eveneens vaak Python**



ML, AI, Big Data



- **Julia als snellere optie**
 - Ecosysteem is aanzienlijk gegroeid maar adoptie vrij traag
 - “Time to first plot” probleem

LOOKS LIKE MATH —

The unreasonable effectiveness of the Julia programming language

Fortran has ruled scientific computing, but Julia emerged for large-scale numerical work.

LEE PHILLIPS - 10/9/2020, 1:15 PM

ML, AI, Big Data



- **In de Big Data wereld blijven ook Java en Scala belangrijk**
 - Voornamelijk als “natieve taal” in Spark / Hadoop omgevingen
 - Voordeel JVM in termen van deployment
 - .NET kent hier minder ondersteuning
 - Maar: algemene vermindering van “big data on prem” aanpak

Systemetalen om in de gaten te houden

- **Naast Rust en Go...**

- **Erlang + Elixir**

- “Elixir is a dynamic, functional language for building scalable and maintainable applications. Elixir leverages the Erlang VM, known for running low-latency, distributed, and fault-tolerant systems. Elixir is successfully used in web development, embedded software, data ingestion, and multimedia processing, across a wide range of industries.”
- “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” (Netflix)
- Forget about Kafka



**How Discord Scaled Elixir to
5,000,000 Concurrent Users**

Systemetalen om in de gaten te houden

- **Naast Rust en Go...**
- **Crystal: Ruby maar beter**
 - Elegantie van Ruby en
 - Statically type checked
 - No Nulls
 - Go-like concurrency
 - Easy C ffi
 - High performance



Systemetalen om in de gaten te houden

- Naast Rust en Go...



- **Nim: Python maar beter**

- Nim is a statically typed compiled systems programming language
- It combines successful concepts from mature languages like Python, Ada and Modula

```
import strformat

type
  Person = object
    name: string
    age: Natural # Ensures the age is positive

let people = [
  Person(name: "John", age: 45),
  Person(name: "Kate", age: 30)
]

for person in people:
  # Type-safe string interpolation,
  # evaluated at compile time.
  echo(fmt"{person.name} is {person.age} years old")
```

Notable mentions...

- Catalana: “A programming language for the law”
- The V Programming Language: “Simple, fast, safe, compiled. For developing maintainable software”
- Gravity: “An embeddable programming language”
- F*: “A general-purpose functional programming language with effects aimed at program verification”
- Wuffs: “An imperative C-like memory-safe language”
- Zig: “A general-purpose programming language and toolchain for maintaining robust, optimal, and reusable software”
- Lua: Wren, Moonscript, Fennel: “MoonScript is a dynamic scripting language that compiles into Lua”

TerrariaClone

Discussed on: [Hacker News](#), [/r/programming](#), [/r/programminghorror](#), [/r/shittyprogramming](#).

Back when I was first learning to program in Java, I decided to try to make a clone of the excellent PC game [Terraria](#). Of course, I was convinced that my version would have many more features than the official one.

But before I realized how silly that idea was, I produced 11,000 lines of, to date, the **most atrocious code I have ever seen in my life**. I make it available here mostly as a cautionary tale of what can happen if you don't pay attention to the quality of your code. (Lesson learned, in my case!) Here are some of the highlights:

```
public ArrayList<Short> drops() {
    ArrayList<Short> dropList = new ArrayList<Short>();
    Random random = TerrariaClone.getRandom();
    if (name == "blue_bubble") {
        for (i=0; i<random.nextInt(3); i++) {
            dropList.add(new Short((short)97));
        }
    }
    if (name == "green_bubble") {
        for (i=0; i<random.nextInt(3); i++) {
            dropList.add(new Short((short)98));
        }
    }
    if (name == "red_bubble") {
        for (i=0; i<random.nextInt(3); i++) {
            dropList.add(new Short((short)99));
        }
    }
    if (name == "yellow_bubble") {
        for (i=0; i<random.nextInt(3); i++) {
            dropList.add(new Short((short)100));
        }
    }
    if (name == "black_bubble") {
        for (i=0; i<random.nextInt(3); i++) {
            dropList.add(new Short((short)101));
        }
    }
    if (name == "white_bubble") {
        for (i=0; i<random.nextInt(3); i++) {
            dropList.add(new Short((short)102));
        }
    }
    if (name == "shooting_star") {
        for (i=0; i<random.nextInt(2); i++) {
            dropList.add(new Short((short)103));
        }
    }
    if (name == "zombie") {
        for (i=0; i<random.nextInt(3); i++) {
            dropList.add(new Short((short)104));
        }
    }
}
```

```
Param P;
Person* Hosts;
std::vector<PersonQuarantine> HostsQuarantine;
Household* Households;
PopVar State, StateT[MAX_NUM_THREADS];
Cell* Cells; // Cells[i] is the i'th cell
Cell ** Celllookup; // Celllookup[i] is a pointer to the i'th populated cell
Microcell* Mcells, ** Mcelllookup;
std::vector<uint16_t> mcell_country;
Place** Places;
AdminUnit AdUnits[MAX_ADUNITS];
//// Time Series defs:
//// TimeSeries is an array of type results, used to store (unsurprisingly) a time series of every quantity in results. Mostly u
//// TSMeanNE and TSVarNE are the mean and variance of non-extinct time series. TSMeanE and TSVarE are the mean and variance of
Results* TimeSeries, * TSMean, * TSVar, * TSMeanNE, * TSVarNE, * TSMeanE, * TSVarE; //// TimeSeries used in RecordSample, Record
Airport* Airports;
BitmapHeader* bmh;
//added declaration of pointer to events log: ggilani - 10/10/2014
Events* InfEventLog;
```

▲ abainbridge on May 7, 2020 [-]

This code is great fun. Start here to explore the horror: <https://github.com/mrc-ide/covid-sim/blob/master/src/CovidSi...>

I like the way InitModel() crashes (I think) if a global pointer called bmh (short for bitmap header) isn't first initialized by calling InitBMHead() from Bitmap.cpp. I guess it's obvious to academics with giant brains that InitModel() depends on a bitmap existing.

But it gets worse - the pointer is declared in CovidSim.cpp but Bitmap.cpp reaches across the module boundary to initialize it. Sweet.

Edit: I feel bad now. It's great that this code was made public. I should be encouraging that. I think academics writing code for things like this should probably take an undergrad course on software engineering. I think the one that taught me about Abstract Data Types would be most beneficial the author(s) of this code.

▲ spatulon on May 7, 2020 [-]

I was part of the GitHub team that helped get the code ready for public release. We fixed a few bugs, reduced memory consumption, made it portable across operating systems, etc., but the code you see is largely what was written by Neil Ferguson and his team.

Given your concerns, I would like to mention two things:

- 1) the code was originally a single source file, so I'm not surprised the module boundaries are imperfect - the code is many years old and the module boundaries have existed for only a few weeks.
- 2) this was not written by professional software engineers, but epidemiologists, and they were working with limited budget, time, and programming experience.

```
infcountry[MAX_COUNTRIES], infcountry_av[MAX_COUNTRIES], infcountry
][MAX_GEN_REC];
+ 1], denom_household[MAX_HOUSEHOLD_SIZE + 1];
ZE + 1], AgeDist[NUM_AGE_GROUPS], AgeDist2[NUM_AGE_GROUPS];
+ 1], case_household_av[MAX_HOUSEHOLD_SIZE + 1][MAX_HOUSEHOLD_S
PES];
YPES], AirTravelDist[MAX_DIST];
```

<https://github.com/mrc-ide/covid-sim/blob/master/src/CovidSim.cpp>

Trends

- Talen geven de richting aan:
 - Portability
 - Concurrency
 - Scalability
 - Null safety
 - Static types
 - Memory safe
- Niet enkel de taal alleen! Ecosysteem, community, documentatie, adoptie

En ook

A great aspect about Ethereum is that smart contracts can be programmed using relatively developer-friendly languages. If you're experienced with Python or any [curly-bracket language](#) [↗], you can find a language with familiar syntax.

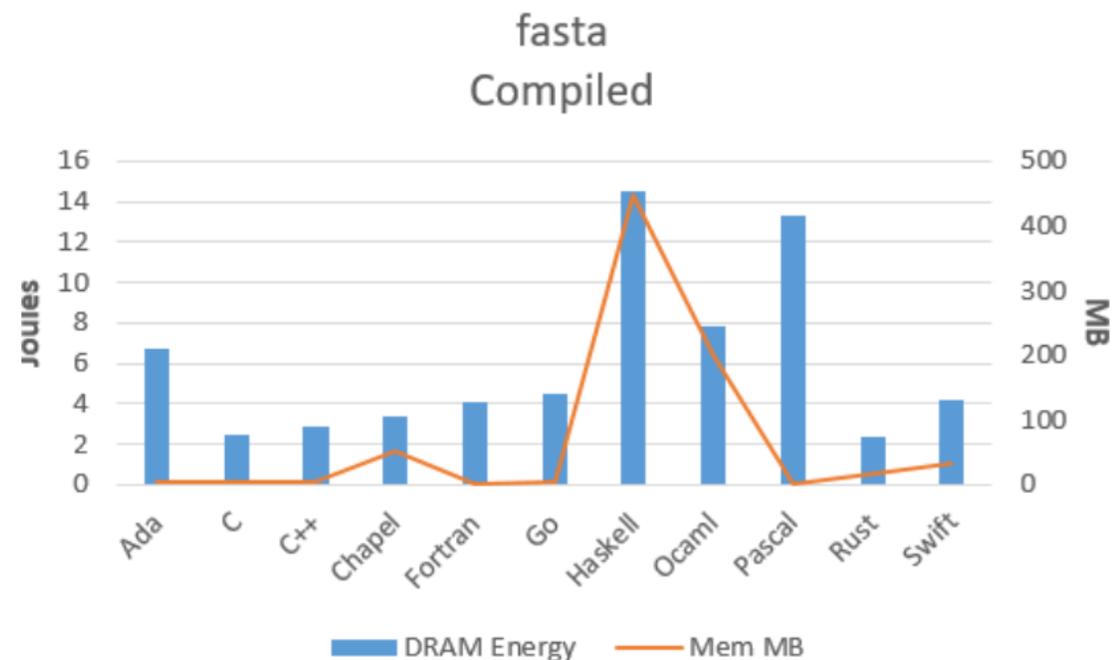
The two most active and maintained languages are:

- Solidity
- Vyper

More experienced developers also might want to use Yul, an intermediate language for the [Ethereum Virtual Machine](#), or Yul+, an extension to Yul.

<https://ethereum.org/en/developers/docs/smart-contracts/languages/>

En ook



Energy Efficiency across Programming Languages

How Do Energy, Time, and Memory Relate?

Rui Pereira
HASLab/INESC TEC
Universidade do Minho, Portugal
ruipereira@di.uminho.pt

Marco Couto
HASLab/INESC TEC
Universidade do Minho, Portugal
marco.l.couto@inesctec.pt

Francisco Ribeiro, Rui Rua
HASLab/INESC TEC
Universidade do Minho, Portugal
fribeiro@di.uminho.pt
rrua@di.uminho.pt

Jácome Cunha
NOVA LINS, DI, FCT
Univ. Nova de Lisboa, Portugal
jacome@fct.unl.pt

João Paulo Fernandes
Release/LISP, CISUC
Universidade de Coimbra, Portugal
jpf@dei.uc.pt

João Saraiva
HASLab/INESC TEC
Universidade do Minho, Portugal
saraiva@di.uminho.pt

En ook



```
assembler_blinken - AtmelStudio
File Edit View VAssistX ASF Project Build Debug Tools Window Help
assembler_blinken.asm x
.include "m328Pdef.inc"

    ldi r16, 0b11111111 //alle output in Arbeitsregister 16 kopieren
    out DDRB, r16      //Arbeitsregister 16 in DDRB kopieren

    ldi r16, 0b00000000 //alle input in Arbeitsregister 16 kopieren(der alte Inhalt wird ueberschrieben)
    out DDRD, r16      //Arbeitsregister 16 in DDRD kopieren

Endlosschleife:
    in r16, PIND       //die anliegenden Spannungen des Ports D in Arbeitsregister 16 kopieren(uberschreiben)
    out PORTB, r16     //den Inhalt des Arbeitsregisters 16(Spannungen der Inputs) auf die Outputs kopieren
    rjmp Endlosschleife //Sprung im Programmcode zur Stelle Endlosschleife:.
```



En ook

Functionele taal

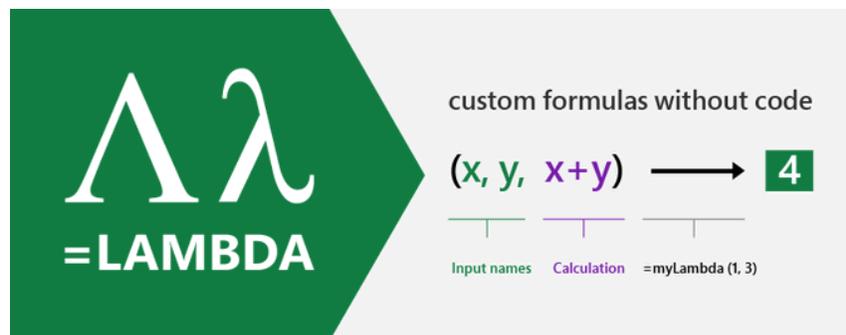
Turing compleet (pas)

Array taal



Eenvoudige maar verbale syntax

2 dimensioneel



Razend populair

Wrap-up

Conclusies

- Python is voor alles de tweede beste taal
- JavaScript is de grootste op het web, mobiel, desktop
- Traditionele systeemtaalen zoals Java, C, C++ en C# blijven sterk, met Rust als enorme stijger
- Focus op veiligheid, schaalbaarheid, onderhoudbaarheid: <blank> maar beter
- Het gaat over meer dan “de taal” alleen: arbeidsmarkt, documentatie, ecosysteem, community

Q & A

