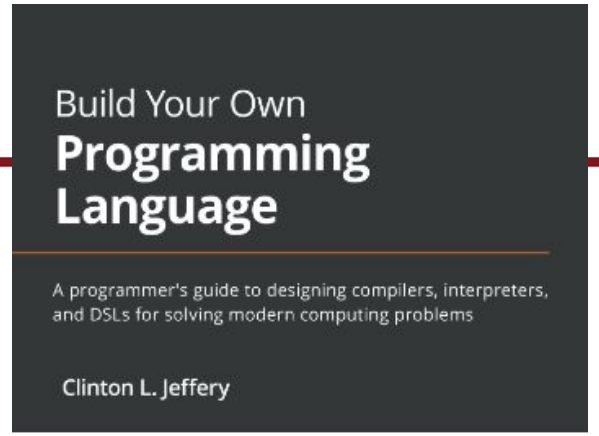


Compilers

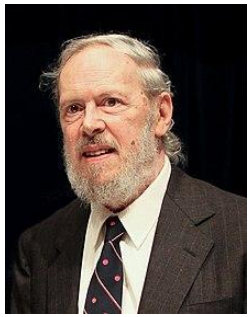
¿Why to build a new one?

Just for fun!

- Sometimes building a library is enough to solve the problem you are working on.

- And, just “rock stars of computer science” do it as simple as it is

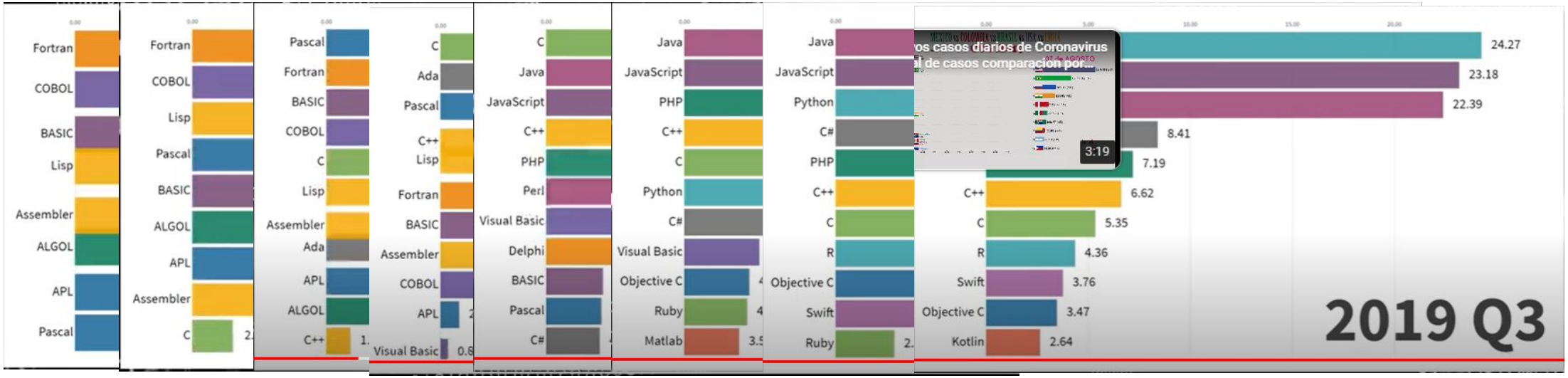
Guido van Rossum



Dennis Ritchie

Besides…

- The best languages have been invented



https://www.youtube.com/watch?v=4Vy73gEjQ1w

The programming languages you need

https://www.tiobe.com/tiobe-index/

Jan 2024	Jan 2023	Change	Programming Language	Ratings	Change
9	8	↓	SQL	1.46%	-1.04%
10	20	⇈	Scratch	1.44%	+0.86%
11	12	↑	Go	1.38%	+0.23%
12	27	⇈	Fortran	1.09%	+0.64%
13	17	⇈	Delphi/Object Pascal	1.09%	+0.36%
14	15	↑	MATLAB	0.97%	+0.06%
19	18	↓	Rust	0.79%	+0.18%
20	31	⇈	COBOL	0.78%	+0.45%

TIOBE Programming Community Index

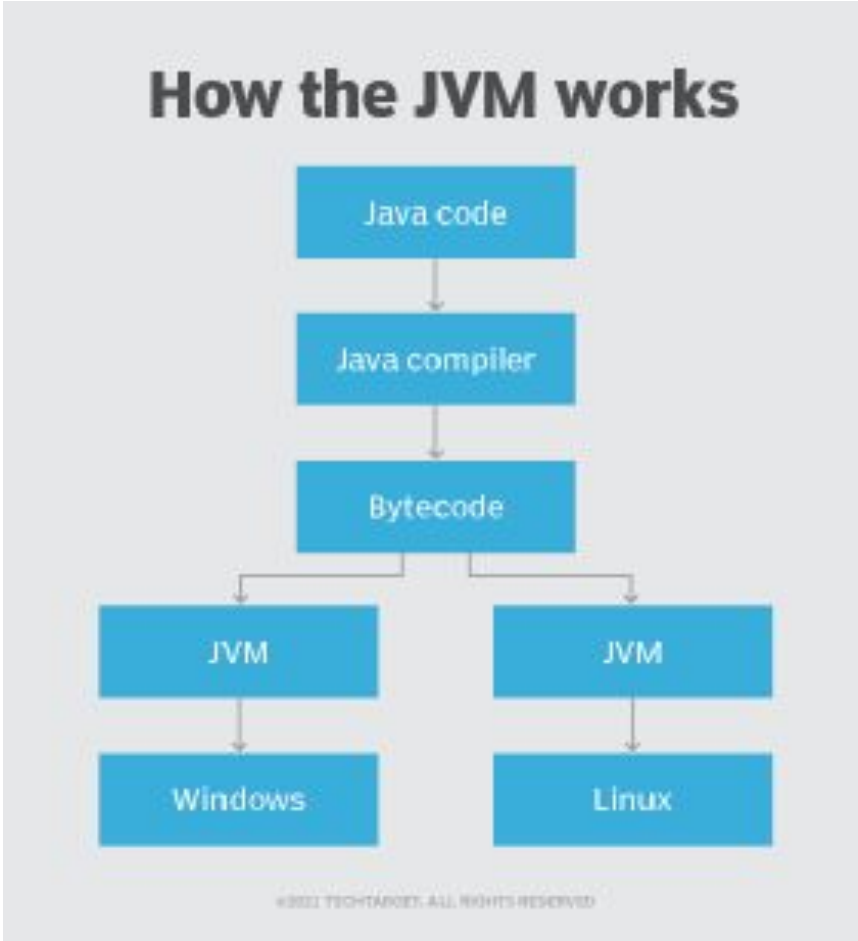
Source: www.tiobe.com

Types of implementations

https://www.freecodecamp.org/espanol/news/lenguajes-compilados-vs-interpretados/

- Interpreter
  - PHP, Ruby, Python, and JavaScript.
- Compiler
  - C, C++, Erlang, Haskell, Rust, and Go.
- Transpiler
  - Source to source converters
- Bytecode compiler

  PHP, Prolog, Raku, Scala, Unicon, Lisp



https://www.techtarget.com/whatis/definition/bytecode

JVM opcodes

Constants

00 (0x00)	nop
01 (0x01)	aconst_null
02 (0x02)	iconst_m1
03 (0x03)	iconst_0
04 (0x04)	iconst_1
05 (0x05)	iconst_2
06 (0x06)	iconst_3
07 (0x07)	iconst_4
08 (0x08)	iconst_5
09 (0x09)	lconst_0
10 (0x0a)	lconst_1
11 (0x0b)	fconst_0
12 (0x0c)	fconst_1
13 (0x0d)	fconst_2
14 (0x0e)	dconst_0
15 (0x0f)	dconst_1
16 (0x10)	bipush
17 (0x11)	sipush
18 (0x12)	ldc
19 (0x13)	ldc_w
20 (0x14)	ldc2_w

Loads

21 (0x15)	iload
22 (0x16)	lload
23 (0x17)	fload
24 (0x18)	dload
25 (0x19)	aload
26 (0x1a)	iload_0
27 (0x1b)	iload_1
28 (0x1c)	iload_2
29 (0x1d)	iload_3
30 (0x1e)	lload_0
31 (0x1f)	lload_1
32 (0x20)	lload_2
33 (0x21)	lload_3
34 (0x22)	fload_0
35 (0x23)	fload_1
36 (0x24)	fload_2
37 (0x25)	fload_3
38 (0x26)	dload_0
39 (0x27)	dload_1
40 (0x28)	dload_2
41 (0x29)	dload_3
42 (0x2a)	aload_0
43 (0x2b)	aload_1
44 (0x2c)	aload_2
45 (0x2d)	aload_3
46 (0x2e)	iaload
47 (0x2f)	laload
48 (0x30)	faload
49 (0x31)	daload
50 (0x32)	aaload
51 (0x33)	baload
52 (0x34)	caload
53 (0x35)	saload

Stores

54 (0x36)	istore
55 (0x37)	lstore
56 (0x38)	fstore
57 (0x39)	dstore
58 (0x3a)	astore
59 (0x3b)	istore_0
60 (0x3c)	istore_1
61 (0x3d)	istore_2
62 (0x3e)	istore_3
63 (0x3f)	lstore_0
64 (0x40)	lstore_1
65 (0x41)	lstore_2
66 (0x42)	lstore_3
67 (0x43)	fstore_0
68 (0x44)	fstore_1
69 (0x45)	fstore_2
70 (0x46)	fstore_3
71 (0x47)	dstore_0
72 (0x48)	dstore_1
73 (0x49)	dstore_2
74 (0x4a)	dstore_3
75 (0x4b)	astore_0
76 (0x4c)	astore_1
77 (0x4d)	astore_2
78 (0x4e)	astore_3
79 (0x4f)	iastore
80 (0x50)	lastore
81 (0x51)	fastore
82 (0x52)	dastore
83 (0x53)	aastore
84 (0x54)	bastore
85 (0x55)	castore
86 (0x56)	sastore

Comparisons

148 (0x94)	lcmp
149 (0x95)	fcmpl
150 (0x96)	fcmpg
151 (0x97)	dcmpl
152 (0x98)	dcmpg
153 (0x99)	ifeq
154 (0x9a)	ifne
155 (0x9b)	iflt
156 (0x9c)	ifge
157 (0x9d)	ifgt
158 (0x9e)	ifle
159 (0x9f)	if_icmpeq
160 (0xa0)	if_icmpne
161 (0xa1)	if_icmplt
162 (0xa2)	if_icmpge
163 (0xa3)	if_icmpgt
164 (0xa4)	if_icmple
165 (0xa5)	if_acmpeq
166 (0xa6)	if_acmpne

Control

167 (0xa7)	goto
168 (0xa8)	jsr
169 (0xa9)	ret
170 (0xaa)	tableswitch
171 (0xab)	lookupswitch
172 (0xac)	ireturn
173 (0xad)	lreturn
174 (0xae)	freturn
175 (0xaf)	dreturn
176 (0xb0)	areturn
177 (0xb1)	return

References

178 (0xb2)	getstatic
179 (0xb3)	putstatic
180 (0xb4)	getfield
181 (0xb5)	putfield
182 (0xb6)	invokevirtual
183 (0xb7)	invokespecial
184 (0xb8)	invokestatic
185 (0xb9)	invokeinterface
186 (0xba)	invokedynamic
187 (0xbb)	new
188 (0xbc)	newarray
189 (0xbd)	anewarray
190 (0xbe)	arraylength
191 (0xbf)	athrow
192 (0xc0)	checkcast
193 (0xc1)	instanceof
194 (0xc2)	monitorenter
195 (0xc3)	monitorexit

Extended

196 (0xc4)	wide
197 (0xc5)	multianewarray
198 (0xc6)	ifnull
199 (0xc7)	ifnonnull
200 (0xc8)	goto_w
201 (0xc9)	jsr_w

Reserved

202 (0xca)	breakpoint
254 (0xfe)	impdep1
255 (0xff)	impdep2

Stack

87 (0x57)	pop
88 (0x58)	pop2
89 (0x59)	dup
90 (0x5a)	dup_x1
91 (0x5b)	dup_x2
92 (0x5c)	dup2
93 (0x5d)	dup2_x1
94 (0x5e)	dup2_x2
95 (0x5f)	swap

Math

96 (0x60)	iadd
97 (0x61)	ladd
98 (0x62)	fadd
99 (0x63)	dadd
100 (0x64)	isub
101 (0x65)	lsub
102 (0x66)	fsub
103 (0x67)	dsub
104 (0x68)	imul
105 (0x69)	lmul
106 (0x6a)	fmul
107 (0x6b)	dmul
108 (0x6c)	idiv
109 (0x6d)	ldiv
110 (0x6e)	fdiv
111 (0x6f)	ddiv
112 (0x70)	irem
113 (0x71)	lrem
114 (0x72)	frem
115 (0x73)	drem
116 (0x74)	ineg
117 (0x75)	lneg
118 (0x76)	fneg
119 (0x77)	dneg
120 (0x78)	ishl
121 (0x79)	lshl
122 (0x7a)	ishr
123 (0x7b)	lshr
124 (0x7c)	iushr
125 (0x7d)	lushr
126 (0x7e)	iand
127 (0x7f)	land
128 (0x80)	ior
129 (0x81)	lor
130 (0x82)	ixor
131 (0x83)	lxor
132 (0x84)	iinc

Conversions

133 (0x85)	i2l
134 (0x86)	i2f
135 (0x87)	i2d
136 (0x88)	l2i
137 (0x89)	l2f
138 (0x8a)	l2d
139 (0x8b)	f2i
140 (0x8c)	f2l
141 (0x8d)	f2d
142 (0x8e)	d2i
143 (0x8f)	d2l
144 (0x90)	d2f
145 (0x91)	i2b
146 (0x92)	i2c
147 (0x93)	i2s

https://docs.oracle.com/javase/specs/jvms/se8/html/jvms-7.html

Referencias

- Jeffery, C. (2021). Build your own programming language. Kindle.