

# Ruby



# O que é Ruby?

- Linguagem de 'scripting'
- Alto poder expressivo
- Sintaxe simples
- Ideal para desenvolvimento Web (On Rails)



# Classificação

- Dinâmica
- Orientada a Objetos
- Reflexiva
- Imperativa
- Funcional



# Influências

## Mother Tongues

Tracing the roots of computer languages through the ages

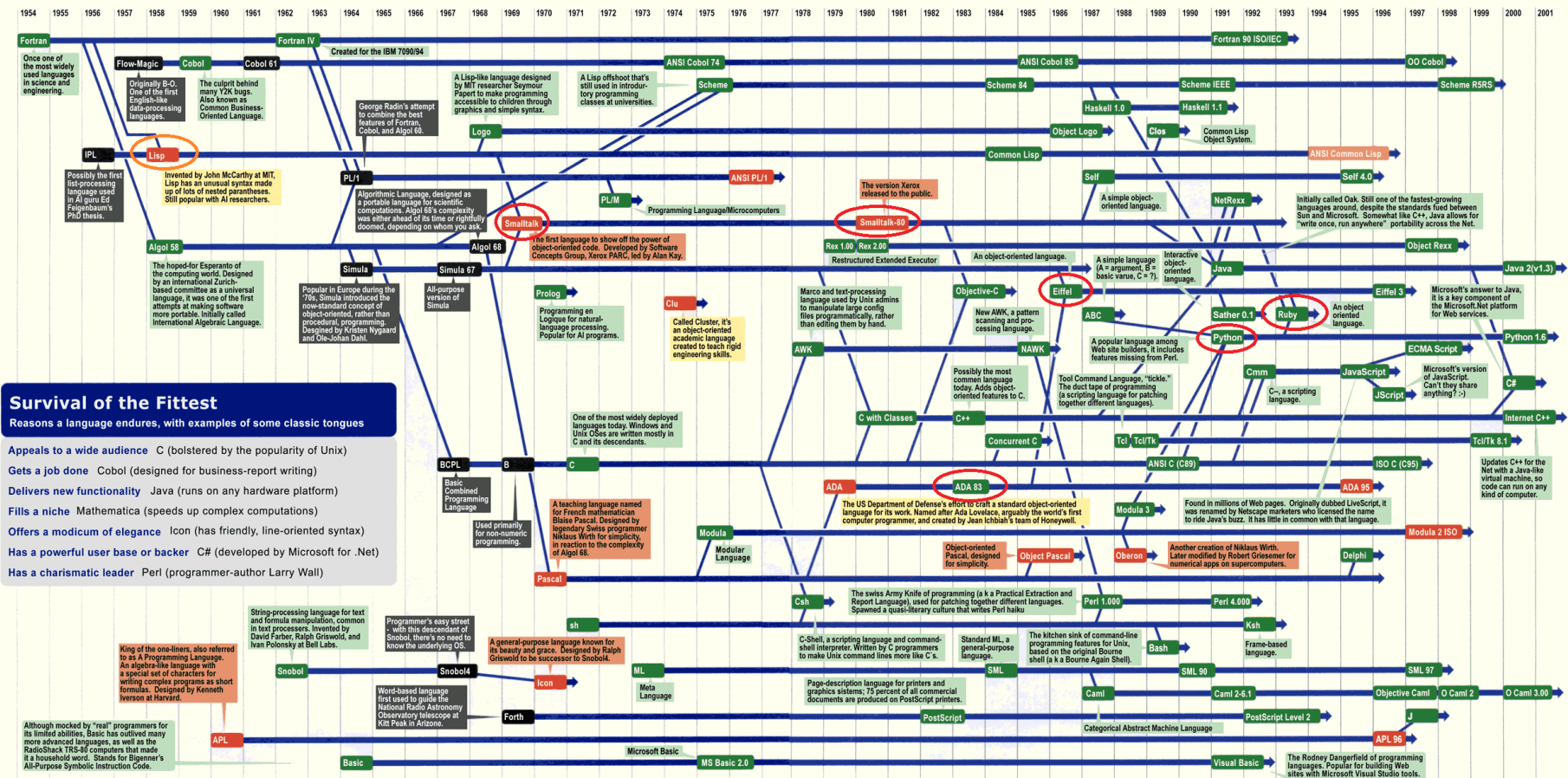
Just like half of the world's spoken tongues, most of the 2,300-plus computer programming languages are either endangered or extinct. As powerhouses C/C++, Visual Basic, Cobol, Java and other modern source codes dominate our systems, hundreds of older languages are running out of life.

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 frangas. Among the most endangered are Ada, APL, B (the predecessor of C), Lsp, 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 [HTTP://www.informatik.uni-freiburg.de/Java/misc/lang\\_list.html](http://www.informatik.uni-freiburg.de/Java/misc/lang_list.html). - Michael Mendeno

**Key**

- 1954 Year introduced
- Active: thousands of users
- Protected: taught at universities; compilers available
- Endangered: usage dropping off
- Extinct: no known active users or up-to-date compilers
- Lineage continues



Sources: Paul Boutin; Brent Haipern, associate director of computer science at IBM Research; The Retrocomputing Museum; Todd Proebsting, senior researcher at Microsoft; Gio Wiederhold, computer scientist, Stanford University

# Avaliação Comparativa

- Orientação a objetos “pura”
- `'=='`, `'eq1?'` e `'equals?'`
- `0 == true`
- Código simples/expressivo (Perl, Python)



# Exemplos de Código

Intervalos:

```
ano = 1987
puts case ano
      when 1970..1979: "Anos 70"
      when 1980..1989: "Anos 80"
      when 1990..1999: "Anos 90"
    end
```

Atribuição múltipla:

```
arg_array = [1, 2, 3, 4]
def my_method(*args)
  a, b, c, d = args
  puts "#{a}#{b}#{c}#{d}"
end
my_method(arg_array)
```

Saída: 1234

# Exemplos de Código

Remoção de diretório (FileUtils):

```
require 'fileutils'  
FileUtils.rm_r 'algum diretório'
```

Múltiplas instruções por linha ('and'):

```
queue = []  
%w{hello x world}.each do |word|  
  queue << word and puts "Added to queue" unless word.length < 2  
end  
puts queue.inspect
```