

# Cinco linguagens para você aprender... e por quê?

Fernando Castor

Centro de Informática – Universidade Federal de Pernambuco

Alguns direitos reservados



# Linguagens de Programação

- Servem para **dizer ao computador o que fazer**
- **Independentemente da máquina** subjacente
- **Legíveis** para humanos



# Linguagens de Programação

- Servem para **dizer ao computador o que fazer**
- **Independentemente da máquina** subjacente
- **Legíveis** para humanos

Hello World!

```
C      Hello World in Fortran 77

      PROGRAM HELLO
      PRINT*, 'Hello, World!'
      END
```

# Linguagens de Programação são **Ferramentas**

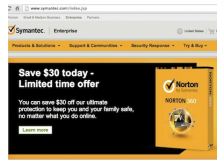
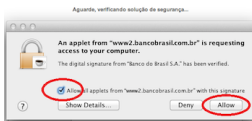
Cada uma se presta a um fim

- **Fortran:** processamento numérico
- **Cobol:** descrição de dados e aplicações comerciais
- **Perl:** processamento de cadeias de caracteres



# Esse fim **pode mudar** com o tempo

## Exemplo: **Java**





# Certo, são muitas linguagens, mas...

- 1 Quais são as linguagens de programação mais usadas na prática?
- 2 Se você quisesse aprender cinco linguagens para o futuro, quais deveriam ser?



# Parte 1

Quais são as linguagens de programação mais usadas na prática?





# Não há um censo oficial sobre uso de linguagens

- Resposta precisa vir de várias fontes
- Com diferentes contextos
- Dados são **difíceis de comparar**



# O que significam “ser usada” e “na prática”?

- Mais **linhas de código** escritas na linguagem?



# O que significam “ser usada” e “na prática”?

- Mais **linhas de código** escritas na linguagem?
- Número de **commits** de programas que usam a linguagem?



# O que significam “ser usada” e “na prática”?

- Mais **linhas de código** escritas na linguagem?
- Número de **commits** de programas que usam a linguagem?
- Mais **“projetos”** com linhas de código escritas na linguagem?



# O que significam “ser usada” e “na prática”?

- Mais **linhas de código** escritas na linguagem?
- Número de **commits** de programas que usam a linguagem?
- Mais **“projetos”** com linhas de código escritas na linguagem?
  - O que é um “projeto”?
  - **Forks** contam como projetos separados?
  - Se usar apenas **código de terceiros** conta?



# O que significam “ser usada” e “na prática”?

- Mais **linhas de código** escritas na linguagem?
- Número de **commits** de programas que usam a linguagem?
- Mais **“projetos”** com linhas de código escritas na linguagem?
  - O que é um “projeto”?
  - **Forks** contam como projetos separados?
  - Se usar apenas **código de terceiros** conta?
- Onde a **linha** da “prática” é **traçada**?



# O que significam “ser usada” e “na prática”?

- Mais **linhas de código** escritas na linguagem?
- Número de **commits** de programas que usam a linguagem?
- Mais **“projetos”** com linhas de código escritas na linguagem?
  - O que é um “projeto”?
  - **Forks** contam como projetos separados?
  - Se usar apenas **código de terceiros** conta?
- Onde a **linha** da “prática” é **traçada**?
- É relevante saber se desenvolvedores **falam** muito **sobre**?



Em resumo:  
é necessário adotar  
diferentes perspectivas.





# Felizmente...

- Alguém já fez **parte** do trabalho
- TIOBE Programming Community Index

*... is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. The popular search engines Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings.*



De acordo com a TIOBE, o ranking em outubro/2015 era este:

Oct 2015	Oct 2014	Change	Programming Language	Ratings	Change
1	2	▲	Java	19.543%	+6.04%
2	1	▼	C	16.190%	-1.47%
3	4	▲	C++	5.749%	+0.88%
4	5	▲	C#	4.825%	+0.08%
5	8	▲	Python	4.512%	+2.18%
6	7	▲	PHP	2.561%	-0.38%
7	13	▲▲	Visual Basic .NET	2.462%	+0.71%
8	12	▲▲	JavaScript	2.292%	+0.52%
9	9		Perl	2.247%	+0.13%
10	16	▲▲	Ruby	1.825%	+0.70%
11	11		Delphi/Object Pascal	1.637%	-0.18%
12	31	▲▲	Assembly language	1.573%	+1.16%
13	14	▲	Visual Basic	1.515%	-0.05%
14	3	▼▼	Objective-C	1.419%	-8.68%
15	19	▲▲	Swift	1.277%	+0.52%



De acordo com a TIOBE, o ranking em outubro/2015 era este:

Oct 2015	Oct 2014	Change	Programming Language	Ratings	Change
1	2	▲	Java	19.543%	+6.04%
2	1	▼	<u>C</u>	16.190%	-1.47%
3	4	▲	C++	5.749%	+0.88%
4	5	▲	C#	4.825%	+0.08%
5	8	▲	Python	4.512%	+2.18%
6	7	▲	PHP	2.561%	-0.38%
7	13	▲▲	Visual Basic .NET	2.462%	+0.71%
8	12	▲▲	JavaScript	2.292%	+0.52%
9	9		<u>Perl</u>	2.247%	+0.13%
10	16	▲▲	Ruby	1.825%	+0.70%
11	11		Delphi/Object Pascal	1.637%	-0.18%
12	31	▲▲	<u>Assembly language</u>	1.573%	+1.16%
13	14	▲	Visual Basic	1.515%	-0.05%
14	3	▼▼	Objective-C	1.419%	-8.68%
15	19	▲▲	Swift	1.277%	+0.52%



## Relembrando:

... is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the **number of skilled engineers** world-wide, **courses** and **third party vendors**. The **popular search engines** Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings.



## Relembrando:

... is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the **number of skilled engineers** world-wide, **courses** and **third party vendors**. The **popular search engines** Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings.

Problema do ranking do índice da TIOBE:  
não menciona diretamente **código** escrito nas linguagens ou **número de repositórios ou projetos** que as usam.

# Obtendo informações sobre código

- Repositórios de software de código aberto (*forges*)
  - Os maiores incluem dezenas de milhares de projetos
  - E já fornecem algumas estatísticas

Examinaremos os dados de dois desses repositórios:



# Dados do SourceForge

Programming Language ▾	St
Java (53,572)	
C++ (43,187)	
PHP (33,775)	
C (31,835)	
C# (17,050)	
Python (16,583)	
JavaScript (13,881)	
Perl (10,012)	
Unix Shell (4,775)	
Visual Basic .NET (4,094)	
Delphi/Kylix (4,050)	
Visual Basic (3,684)	
Assembly (2,747)	
JSP (2,173)	
Ruby (2,062)	

**Entre parênteses:** número de projetos que declaram usar a linguagem



# Dados do SourceForge

Programming Language ▾	St
Java (53,572)	
C++ (43,187)	
PHP (33,775)	
C (31,835)	
C# (17,050)	
Python (16,583)	
JavaScript (13,881)	
Perl (10,012)	
Unix Shell (4,775)	
Visual Basic .NET (4,094)	
Delphi/Kylix (4,050)	
Visual Basic (3,684)	
Assembly (2,747)	
JSP (2,173)	
Ruby (2,062)	

**Entre parênteses:** número de projetos que declaram usar a linguagem

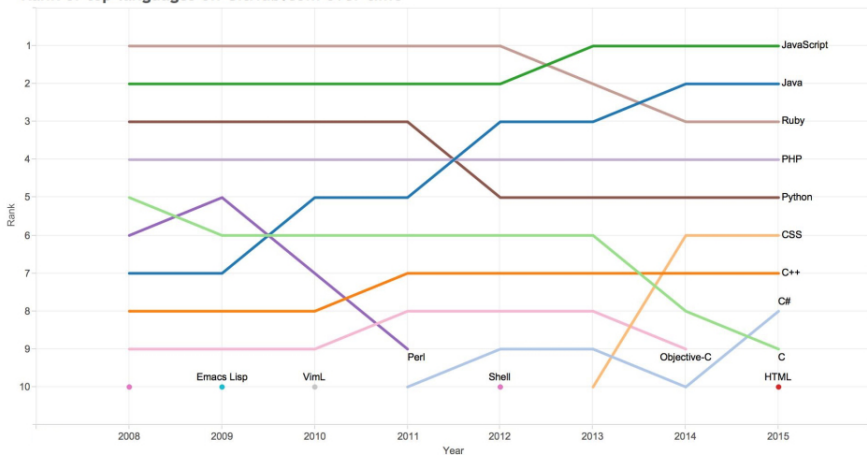
**Objective-C, Perl e Swift** aparecem no Top 15 da TIOBE mas **não aqui**





# Dados do Github

Rank of top languages on GitHub.com over time



Source: GitHub.com



# E sobre quais os programadores estão falando?

Tags do StackOverflow.com  
em outubro/2015:

- 1 JavaScript (963637)
- 2 Java (944061)
- 3 C# (860626)
- 4 PHP (819438)
- 5 Python (483959)
- 6 C++ (401005)
- 7 CSS (341154)
- 8 SQL (298376)
- 9 asp.net (270712)
- 10 Objective-C (242939)
- 11 C (195691)
- 12 Ruby (145367)
- 13 r (110405)
- 14 VB.NET (90188)

Entre parênteses: número de  
posts com aquela tag



# E sobre quais os programadores estão falando?

Tags do StackOverflow.com  
em outubro/2015:

- 1 JavaScript (963637)
- 2 Java (944061)
- 3 C# (860626)
- 4 PHP (819438)
- 5 Python (483959)
- 6 C++ (401005)
- 7 CSS (341154)
- 8 SQL (298376)
- 9 asp.net (270712)
- 10 Objective-C (242939)
- 11 C (195691)
- 12 Ruby (145367)
- 13 r (110405)
- 14 VB.NET (90188)

Entre parênteses: número de  
posts com aquela tag

Se levarmos JQuery, AJAX,  
node.js, AngularJS, Django e  
Rails em conta:

- 1 JavaScript (**1983090**)
- 6 Python (**589276**)
- 7 Ruby (**366600**)



# E sobre quais os programadores estão falando?

Tags do StackOverflow.com  
em outubro/2015:

- 1 JavaScript (963637)
- 2 Java (944061)
- 3 C# (860626)
- 4 PHP (819438)
- 5 Python (483959)
- 6 C++ (401005)
- 7 CSS (341154)
- 8 SQL (298376)
- 9 asp.net (270712)
- 10 Objective-C (242939)
- 11 C (195691)
- 12 Ruby (145367)
- 13 r (110405)
- 14 VB.NET (90188)

Entre parênteses: número de  
posts com aquela tag

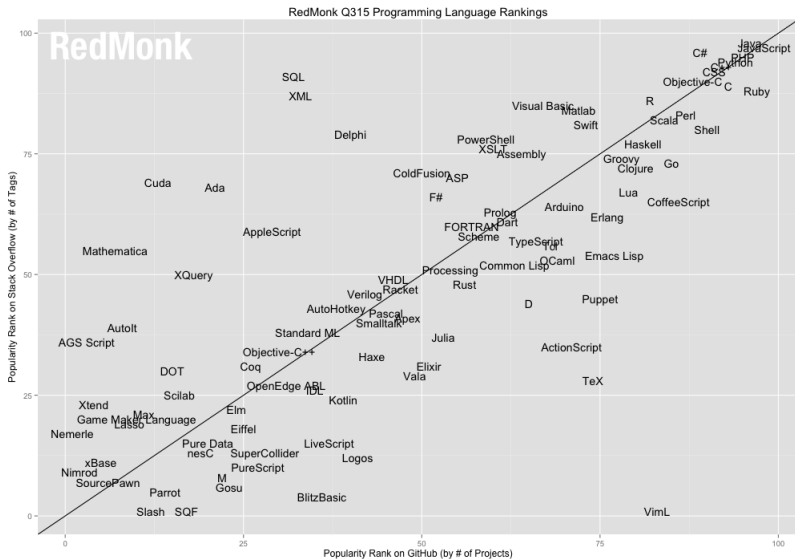
Se levarmos JQuery, AJAX,  
node.js, AngularJS, Django e  
Rails em conta:

- 1 JavaScript (**1983090**)
- 6 Python (**589276**)
- 7 Ruby (**366600**)

Se incluirmos Android (Java) e  
iPhone, iOS e Xcode1  
(Objective-C):

- 2 Java (**1683105**)
- 3 Objective-C (**920493**)

# The RedMonk Programming Language Rankings



# Top 10 a partir de diversas fontes

TIOBE	SourceForge	Github	StackOverflow
Java	Java	Javascript	Javascript
C	C++	Java	Java
C++	PHP	Ruby	Objective-C
C#	C	PHP	C#
Python	C#	Python	PHP
PHP	Python	CSS	Python
VB.NET	Javascript	C++	Ruby
Javascript	Perl	C#	C++
Perl	VB[.NET]	C	SQL
Ruby	UNIX Shell	HTML(????)	Asp.net

# Top 10 a partir de diversas fontes

TIOBE	SourceForge	Github	StackOverflow
Java	Java	Javascript	Javascript
C	C++	Java	Java
C++	PHP	Ruby	Objective-C
C#	C	PHP	C#
Python	C#	Python	PHP
PHP	Python	CSS	Python
VB.NET	Javascript	C++	Ruby
Javascript	Perl	C#	C++
Perl	VB[.NET]	C	SQL
Ruby	UNIX Shell	HTML(????)	Asp.net

**Importante:** não representam a indústria **como um todo**.

# Parte 2

Se você quisesse aprender cinco linguagens para o futuro, quais deveriam ser?





Cinco linguagens de programação para que? Quais os critérios para escolhê-las?



# Meus critérios

Nada de linguagens “acadêmicas”

Não devem servir apenas como prova de conceito para uma ideia



# Meus critérios

Nada de linguagens “acadêmicas”

Não devem servir apenas como prova de conceito para uma ideia

Linguagens que forneçam **lições amplas**

- linguagens que se possa empregar diretamente
- e linguagens que ajudem a usar melhor outras linguagens

# Meus critérios

Nada de linguagens “acadêmicas”

Não devem servir apenas como prova de conceito para uma ideia

Linguagens que forneçam **lições amplas**

- linguagens que se possa empregar diretamente
- e linguagens que ajudem a usar melhor outras linguagens

Linguagens de programação para você **aprender**



# Meus critérios

Levando em conta certos nichos de aplicação e cenários de uso

- 1 Primeira linguagem de programação para aprender
- 2 Desenvolvimento Móvel
- 3 Programação multi-núcleo e sistemas confiáveis
- 4 Desenvolvimento para a Web
- 5 Software de baixo nível e de alto desempenho

# 1. Primeira Linguagem de Programação para Aprender



# Olá, Mundo!

```
class HelloWorld {  
    static public void main( String args[] ) {  
        System.out.println(" Hello World!");  
    }  
}
```

```
#include <stdio.h>
```

```
int main(void) {  
    puts(" Hello World!");  
}
```

```
10 PRINT " Hello World!"
```

```
(defun helloworld ()  
  (print " Hello World!")  
)
```

```
print " Hello World!"
```

```
package main  
import "fmt"  
func main() {  
    fmt.Printf(" Hello World!")  
}
```

# Linguagem 1





# Linguagem 1





## Porquês

- Sintaxe amigável
- Dinamicamente tipificada
- Multiparadigma
- Muito popular na prática
- Usada como primeira linguagem em vários cursos de computação

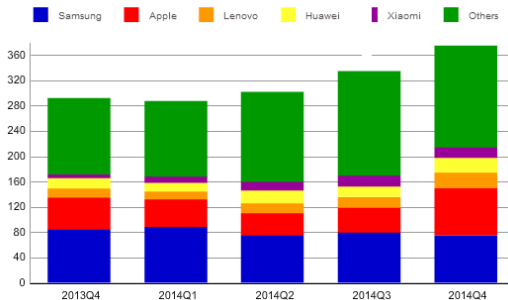
# Alternativas



## 2. Desenvolvimento Móvel

# Vendas de Smartphones no mundo

Top 5 WW Smartphone Vendors, 2014Q4  
Unit Shipments (Millions)



Fonte: SmartphoneMarketResearch.com

# Para Android: Java



# Para Android: Java



Também é possível usar JavaScript

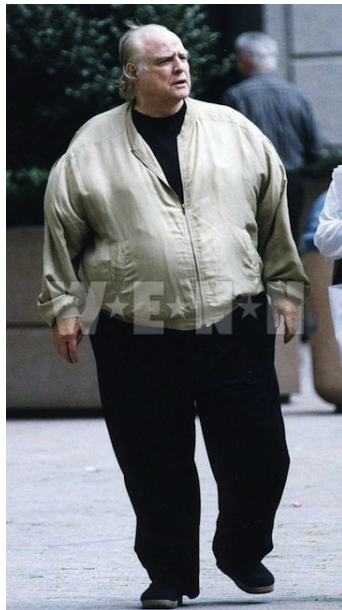


Mas...





Mas...



# Para iOS: Objective-C



**Objective-C**

# Para iOS: Objective-C

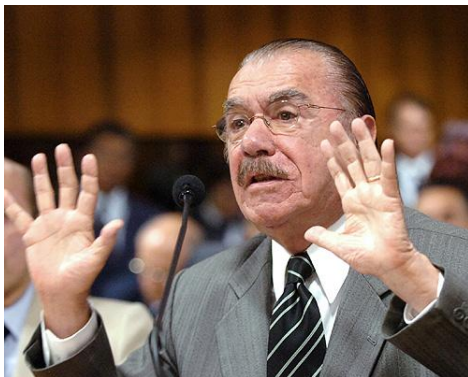


**Objective-C**

Também é possível usar JavaScript<sup>2</sup>

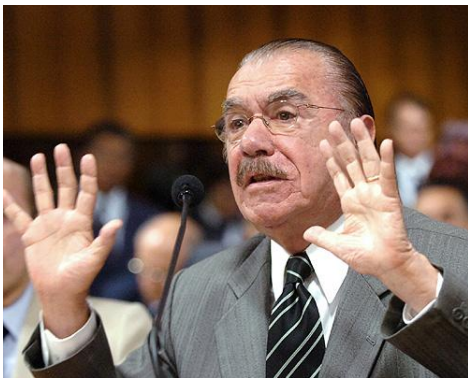


# Mas...



## Objective-C...

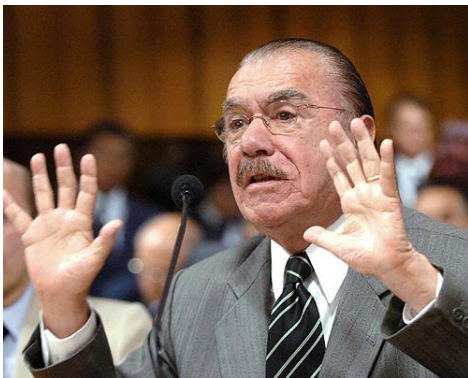
# Mas...



## Objective-C...

- tem uma aparência desagradável

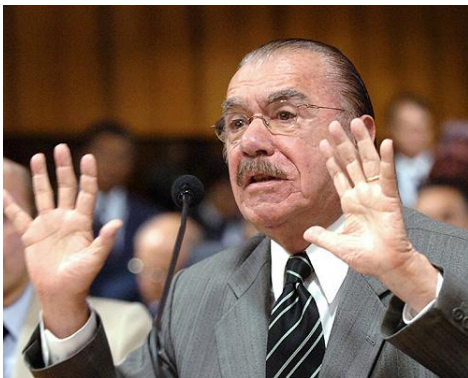
# Mas...



## Objective-C...

- tem uma aparência desagradável
- tem uma propensão a causar corrupção

# Mas...



## Objective-C...

- tem uma aparência desagradável
- tem uma propensão a causar corrupção
- está por aí há tempo demais (alguns dizem que é **imortal**)

# Linguagem 2





*The growth that Swift experienced is essentially unprecedented in the history of these rankings.*

<http://redmonk.com/sogradys/2015/01/14/language-rankings-1-15/>



# Por que Swift?

Orientada a Objetos + Funcional

Estaticamente tipificada, mas sem inchaço

*Playgrounds* interativos

Código aberto: ecossistema Apple + Linux



# Ainda não dá...

para fazer apps Android em Swift.



# Ainda não dá...

para fazer apps Android em Swift.

**AINDA.**



# Alternativas (obviamente)



# Alternativas (obviamente)



E tem também

JavaScript



### 3. Programação Multi-Núcleo e Sistemas Confiáveis



<http://is1.minnesotacomputers.net/images/productimages/M605.gif>

[http://www.ps3-systems.com/images/playstation\\_3.jpg](http://www.ps3-systems.com/images/playstation_3.jpg)

<http://www.zeldauniverse.net/wp-content/uploads/2010/07/oot3dsblack.png>

[http://www.tudocelular.com/new\\_files/images/global/Samsung-Galaxy-Note\\_53032\\_1.jpg](http://www.tudocelular.com/new_files/images/global/Samsung-Galaxy-Note_53032_1.jpg)

<http://maxcdn.liewcf.com/blog/wp-content/uploads/apple-ipad-2-wifi.jpg>



Multi-núcleo  $\Rightarrow$   
concorrência e paralelismo



# Problemas oriundos de programação multi-núcleo

- Condições de corrida
- Falhas de atomicidade
- Deadlocks, livelocks, starvation
- Morte prematura de threads
- Custo de criação e chaveamento entre threads
- Quebra de trabalho entre threads, sincronização



## Algumas causas fundamentais

- **Estado compartilhado mutável** é a raiz do mal.
- **Falta** de mecanismos para **tratamento de erros** em cenários concorrentes e distribuídos.
- **Threads do sistema operacional** como unidades de quebra de trabalho.

# Linguagem 3



# Linguagem 3





## Porquês

- **Filosofia Let It Crash**
- Puramente funcional e sem estado compartilhado
- Processos leves – unidades finas de quebra de trabalho
- Concorrência e distribuição são **inerentes** à linguagem
  - É impossível construir sistemas não-triviais ignorando isso

Nenhuma.



# 4. Desenvolvimento para a Web



# Qual o atual estado da prática de desenvolvimento para a Web?

(de acordo com a Wikipédia)



**Lado servidor:** ASP, CSP, Server-Side ANSI C, ColdFusion, CGI, Groovy (Grails), Java, Lotus Domino, Perl, PHP, Python (Django), Real Studio Web Edition, Ruby (Rails), Node.js, Server-Side JavaScript, Mozilla Rhino, Websphere, .NET, (...)

**Lado cliente:** Ajax, Flash, JavaScript, jQuery, Silverlight, HTML5



# JavaScript é um padrão de fato, mas...

- `[] + [] =`



# JavaScript é um padrão de fato, mas...

- `[] + [] = string vazio!`
- `[] + {} =`



# JavaScript é um padrão de fato, mas...

- `[] + [] = string vazio!`
- `[] + {} = [object Object]?!`
- `{ } + [] =`

# JavaScript é um padrão de fato, mas...

- `[] + [] = string vazio!`
- `[] + {} = [object Object]?!`
- `{ } + [] = 0!!!`
- `{ } + { } =`

# JavaScript é um padrão de fato, mas...

- $[] + [] = \text{string vazio!}$
- $[] + \{\} = [\text{object Object}]?!$
- $\{\} + [] = 0!!!$
- $\{\} + \{\} = \text{NaN!}$
- $\{\} - \{\} =$

# JavaScript é um padrão de fato, mas...

- $[] + [] = \text{string vazio!}$
- $[] + \{\} = [\text{object Object}]?!$
- $\{\} + [] = 0!!!$
- $\{\} + \{\} = \text{NaN!}$
- $\{\} - \{\} = \text{NaN!}$
- $\{\} * \{\} =$



# JavaScript é um padrão de fato, mas...

- $[] + [] = \text{string vazio!}$
- $[] + \{\} = [\text{object Object}]?!$
- $\{\} + [] = 0!!!$
- $\{\} + \{\} = \text{NaN!}$
- $\{\} - \{\} = \text{NaN!}$
- $\{\} * \{\} = \text{syntax error!}$
- $\text{Array}(16) + 5 =$

# JavaScript é um padrão de fato, mas...

- $[] + [] = \text{string vazio!}$
- $[] + \{\} = [\text{object Object}]?!$
- $\{\} + [] = 0!!!$
- $\{\} + \{\} = \text{NaN!}$
- $\{\} - \{\} = \text{NaN!}$
- $\{\} * \{\} = \text{syntax error!}$
- $\text{Array}(16) + 5 = \text{,,,,,,,,,,,,,5}$
- $\text{Array}(16) - 5 =$

# JavaScript é um padrão de fato, mas...

- $[] + [] = \text{string vazio!}$
- $[] + \{\} = [\text{object Object}]?!$
- $\{\} + [] = 0!!!$
- $\{\} + \{\} = \text{NaN!}$
- $\{\} - \{\} = \text{NaN!}$
- $\{\} * \{\} = \text{syntax error!}$
- $\text{Array}(16) + 5 = \text{,,,,,,,,,,,,,5}$
- $\text{Array}(16) - 5 = \text{NaN!}$

# E o escopo de variáveis?

*“Javascript programmers are practically ranked by how well they understand scope.”*

<http://stackoverflow.com/questions/500431/javascript-variable-scope>

- Cuidado com o significado de **this**!



# Linguagem 4







# DART

## Porquês

- Menos caos
  - sem objeto global ou atributos adicionados dinamicamente
- Possibilidade de usar **anotações de tipo**
- Escopos intuitivos, baseados em blocos
- Compilável para JavaScript
- Desenvolvida pelo Google

# Alternativas





## 5. Software de Baixo Nível e de Alto Desempenho



Que linguagem você acha que  
seria boa para construir um  
*shooter* 3d?



E um escalonador de processos?



# Um programa para processar imagens de satélites?



# E um driver de dispositivo?

# Sendo um pouco mais focado

Tem algo que nenhuma linguagem discutida anteriormente lhe dá  
**Controle** sobre a máquina



# Sendo um pouco mais focado

Tem algo que nenhuma linguagem discutida anteriormente lhe dá  
**Controle** sobre a máquina

Que linguagem usar se eu quiser...

- acesso direto à RAM
  - inclusive para gerenciar memória manualmente
- guardar valores explicitamente em registradores
  - em alguns casos, especificando o registrador
- produzir código extremamente eficiente
- e (possivelmente) portátil?

# Linguagem 5





# Linguagem 5



# Linguagem 5



THE

C



PROGRAMMING  
LANGUAGE

BRIAN W. KERNIGHAN  
DENNIS M. RITCHIE





THE

C



PROGRAMMING  
LANGUAGE

BRIAN W. KERNIGHAN  
DENNIS M. RITCHIE

## Porquês

- Porque permite **tudo mencionado no slide anterior**
- Porque é **menos complexa** que C++
- Porque Assembly só é uma opção quando é a **única opção**

# Mas há outras razões para aprender C

## Outros porquês

- Gerenciamento de memória manual
- Tipificação fraca
- Manipulação de ponteiros é
  - fundamental mesmo para as operações mais básicas
  - complexa
  - insegura
- difícil para novatos

**Resumindo:** é importante aprender C porque:



**Resumindo:** é importante aprender C porque:

- É uma linguagem muito usada na prática



**Resumindo:** é importante aprender C porque:

- É uma linguagem **muito usada na prática**
- Para vários tipos de aplicação, é uma **boa ferramenta**



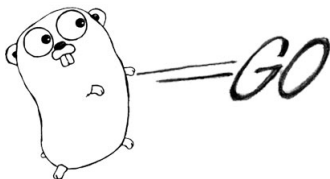
**Resumindo:** é importante aprender C porque:

- É uma linguagem **muito usada na prática**
- Para vários tipos de aplicação, é uma **boa ferramenta**
- As **características negativas** da linguagem fornecem lições importantes





# Alternativas



E sempre haverá **C++**, claro. E Assembly.





Obrigado!

**Contato:** castor arroba cin.ufpe.br

**Slides:** <http://github.com/fernandocastor>

**Vídeo:** <http://www.youtube.com/watch?v=MsWmSCCVj5Y>

(Busca no YouTube ou Google: **cpre1 linguagens**)