



BC1424

Algoritmos e Estruturas de Dados I

Aula 01:
Linguagens de Programação
Introdução à linguagem C

Prof. Jesús P. Mena-Chalco

jesus.mena@ufabc.edu.br

1Q-2015



Linguagens de programação

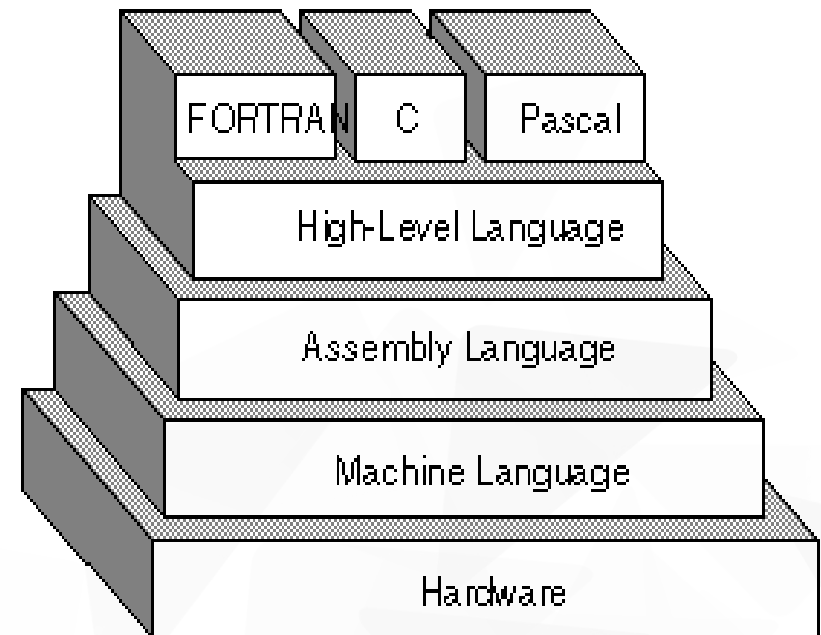
Linguagem de programação

É um conjunto limitado de:

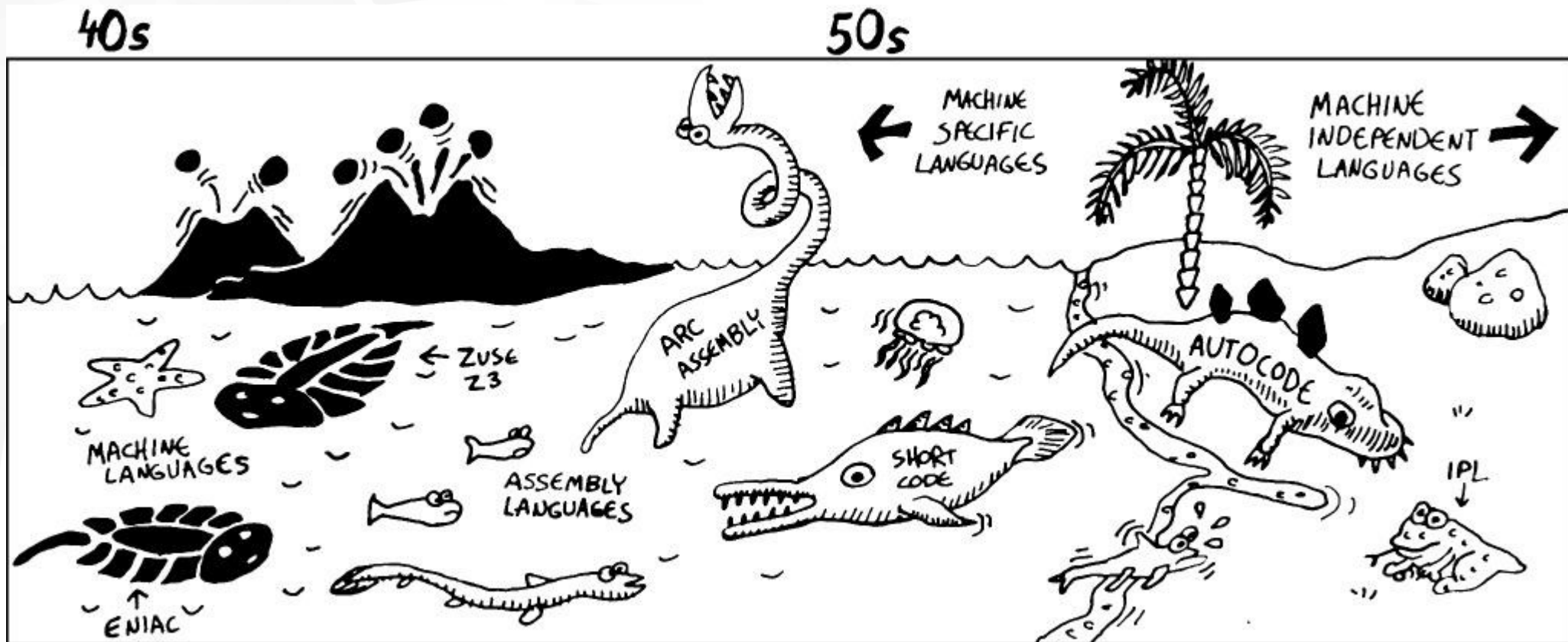
- **Símbolos** (comandos, identificadores, caracteres, etc)
- **Regras de sintaxe** (descrevem de forma precisa ações)

Tipos de linguagem de programação

Linguagem de máquina	Compreendida pelo computador. Dependente da arquitetura do computador
Linguagem de baixo nível	Utiliza mnemonicos para a representação de ações elementares Ex. Assembler
Linguagem de alto nível	Utiliza instruções próximas da linguagem humana Ex. C, Java, Python, PHP



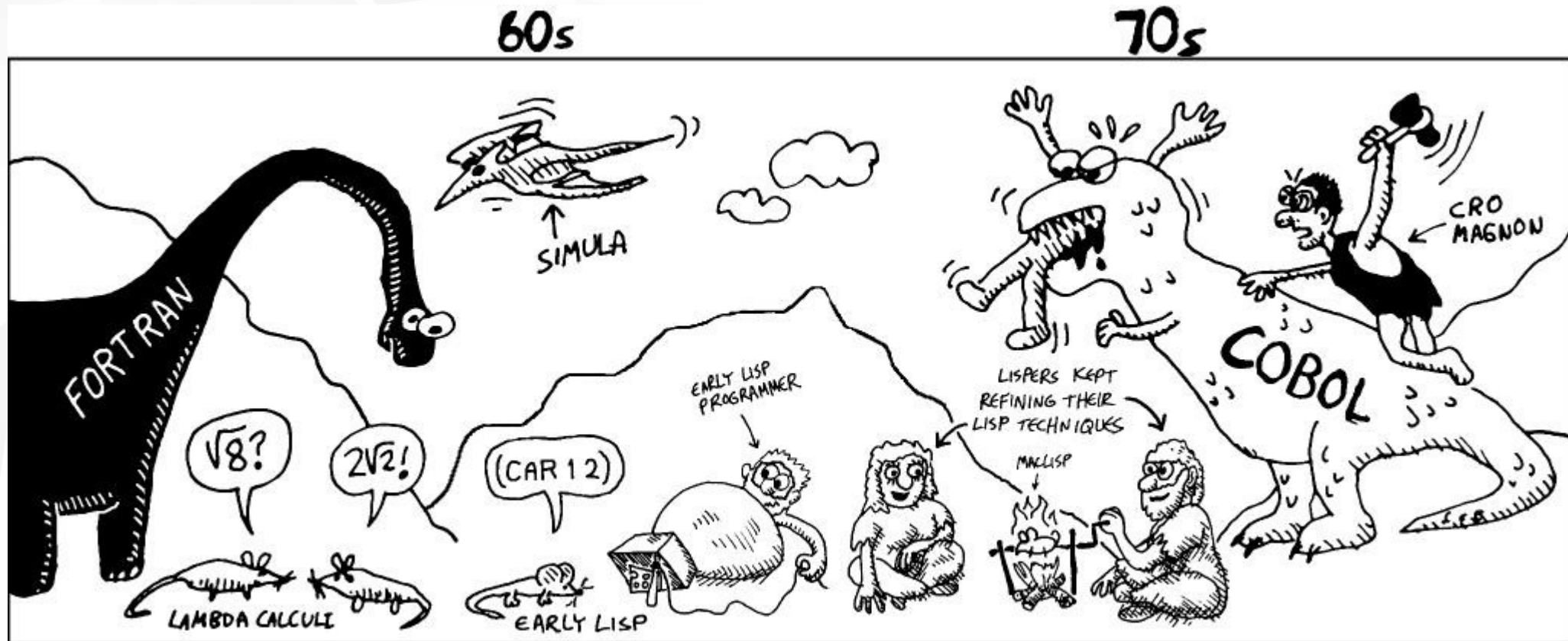
Linguagens de programação



Fonte: “Land of Lisp: Learn to Program in List, One Game at a time”. Autor: Conrad Barski.

http://www.amazon.com/gp/product/1593272812/ref=as_li_tf_il?ie=UTF8&camp=1789&creative=9325&creativeASIN=1593272812&linkCode=as2&tag=onionrealit-20

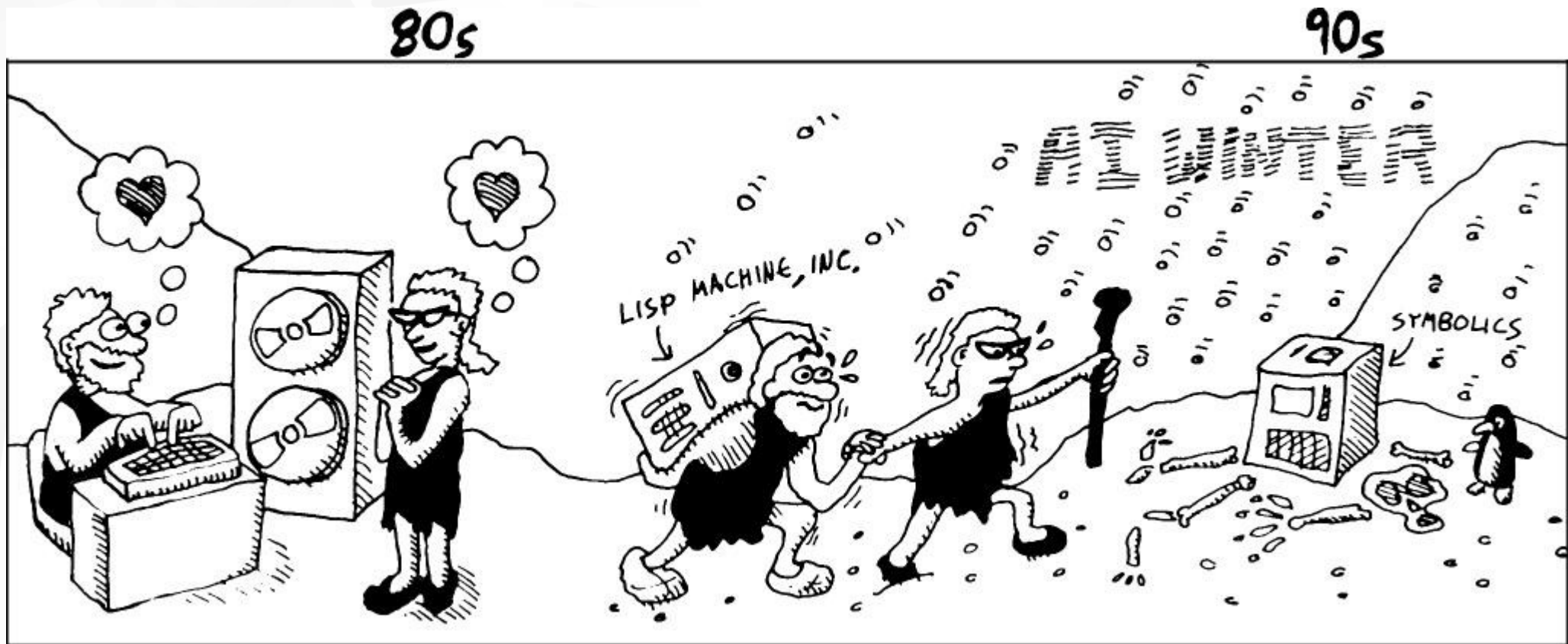
Linguagens de programação



Fonte: “Land of Lisp: Learn to Program in List, One Game at a time”. Autor: Conrad Barski.

http://www.amazon.com/gp/product/1593272812/ref=as_li_tf_il?ie=UTF8&camp=1789&creative=9325&creativeASIN=1593272812&linkCode=as2&tag=onionrealit-20

Linguagens de programação

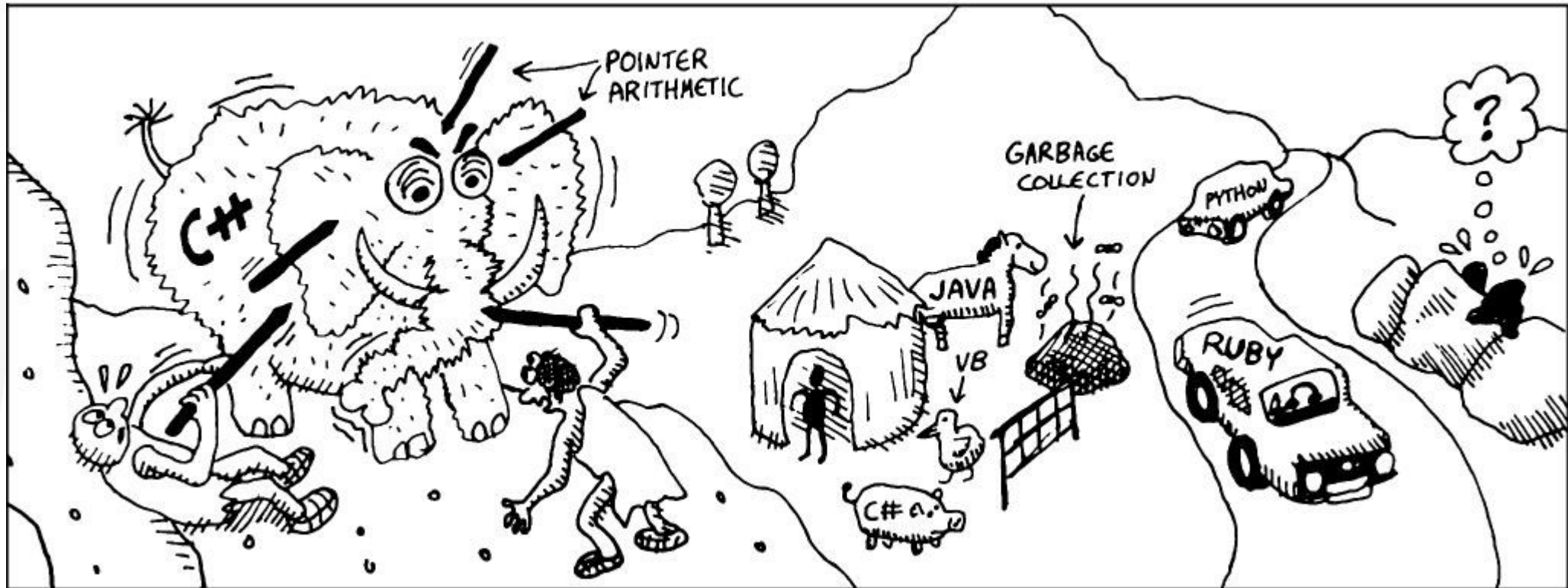


Fonte: "Land of Lisp: Learn to Program in List, One Game at a time". Autor: Conrad Barski.

http://www.amazon.com/gp/product/1593272812/ref=as_li_tf_il?ie=UTF8&camp=1789&creative=9325&creativeASIN=1593272812&linkCode=as2&tag=onionrealit-20

Linguagens de programação

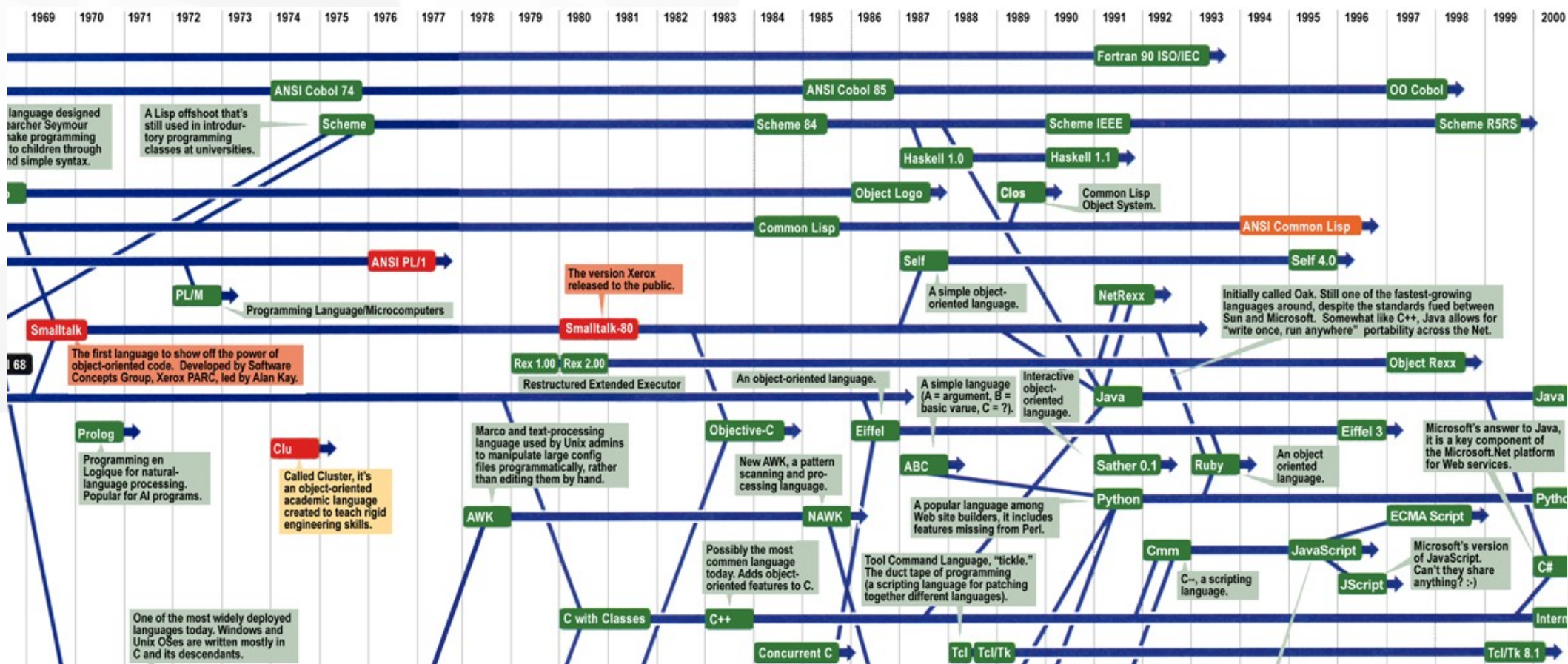
2000



Fonte: "Land of Lisp: Learn to Program in List, One Game at a time". Autor: Conrad Barski.

http://www.amazon.com/gp/product/1593272812/ref=as_li_tf_il?ie=UTF8&camp=1789&creative=9325&creativeASIN=1593272812&linkCode=as2&tag=onionrealit-20

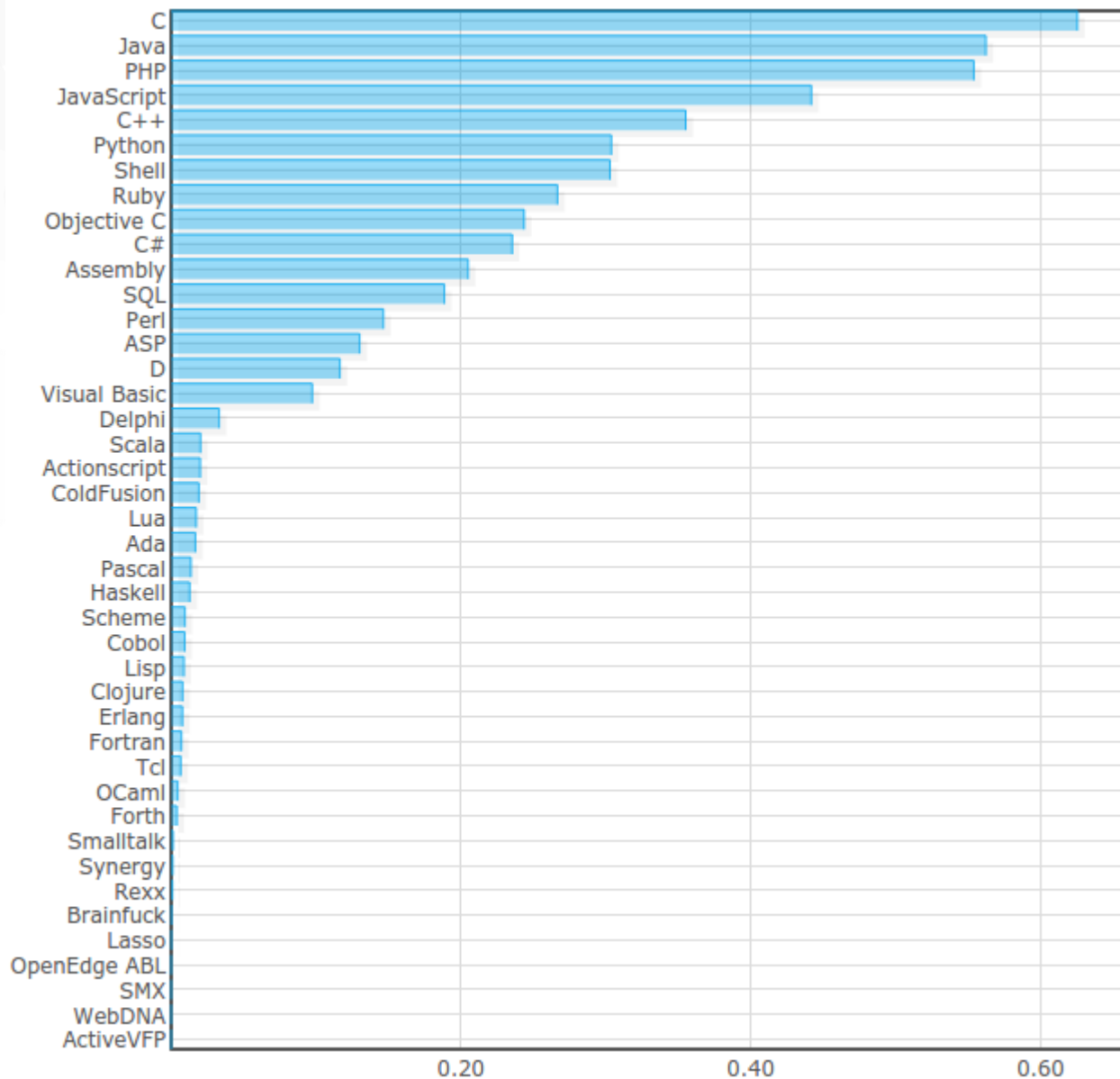
Linguagens de programação



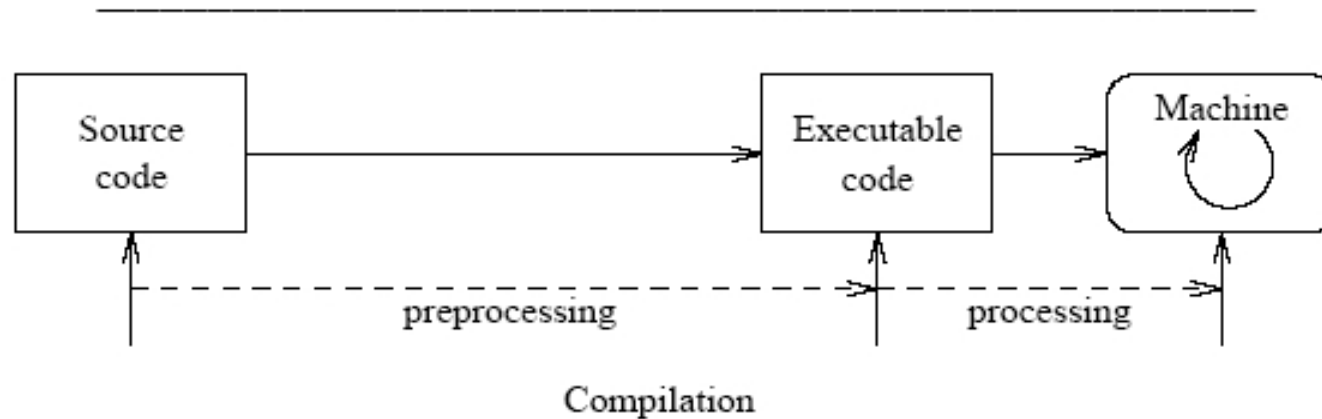
Linguagens de programação



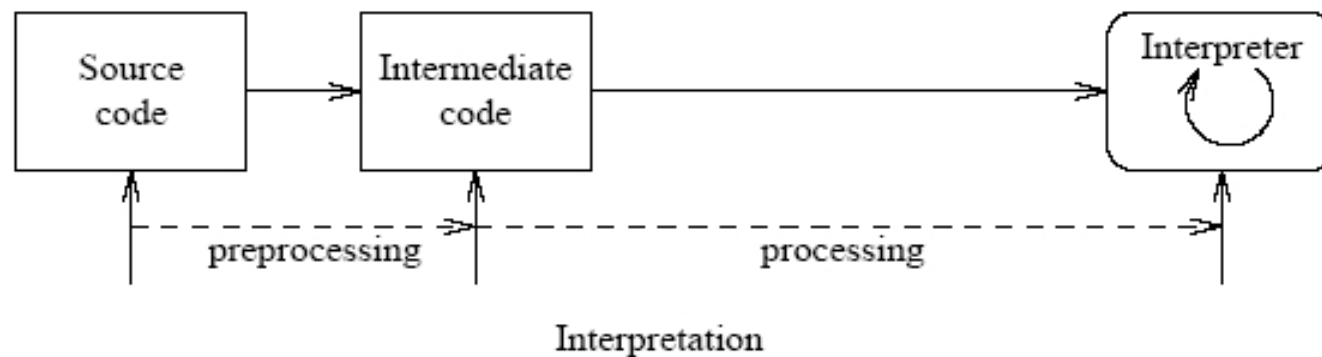
Linguagens de programação



Compiler Vs Interpreter



C
C++
Java



Python
Scilab
PHP

Sobre a linguagem de programação

- Atualmente existem várias linguagens que são consideradas para este tipo de disciplinas...
(Python, C, C++, Java, Haskell, Ruby)
- Também vários paradigmas de programação (e.g. procedural, orientado a objetos,) podem ser consideradas...

Todo programador competente deve saber/entender a linguagem C/C++.

Tradicionalmente é utilizada a linguagem C.

Nessa disciplina usaremos C.



A linguagem de programação C

Sobre a linguagem de programação C

- Permite escrever programas de alta performance.
- **C** funciona em um nível mais baixo que outras linguagens (da uma ideia melhor do que realmente está acontecendo).
- É preferida no mundo científico:
 - Poucas palavras reservadas.
 - Bom controle da máquina.
 - SO livres (Linux/UNIX) são feitos em C.
 - Base de outras linguagens: C++, Java, C#

Sobre a linguagem de programação C

- C é uma linguagem estruturada, desenvolvida nos Laboratórios BELL (1969-1972), por Dennis Ritchie.
- **Dialetos:**
 - K&R C (1978)
 - ANSI C
 - ISO C
 - C99
 - C11 (Dez. 2011)



Kenneth L. Thompson (ling. B)
Dennis M. Ritchie (ling. C)

The National Medal of Technology and Innovation 1998 Laureates

Kenneth L. Thompson (1943-)

Dennis M. Ritchie (1941-2011)



(*) Fonte: <http://www.uspto.gov/about/nmti/recipients/1998.jsp>

Sobre a linguagem de programação C

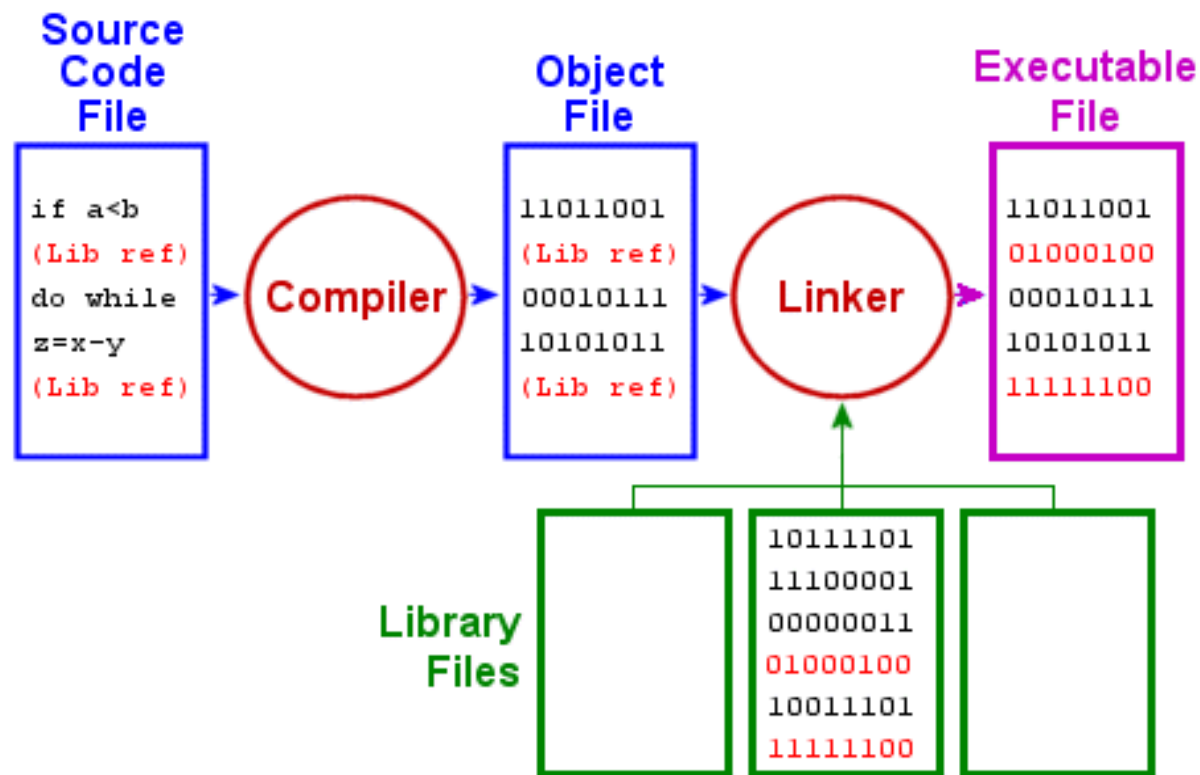
- **Influenciada por:**

- ALGOL 68
- Assembly
- B
- BCPL
- CPL
- Fortran
- PL/I

- **Influenciou:**

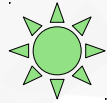
- AWK, BitC, C++, C#, C Shell, D, Euphoria, Go, Java, JavaScript, Limbo, Logic Basic, Objective-C, Perl, PHP, Python, ...
















Sobre a linguagem de programação C



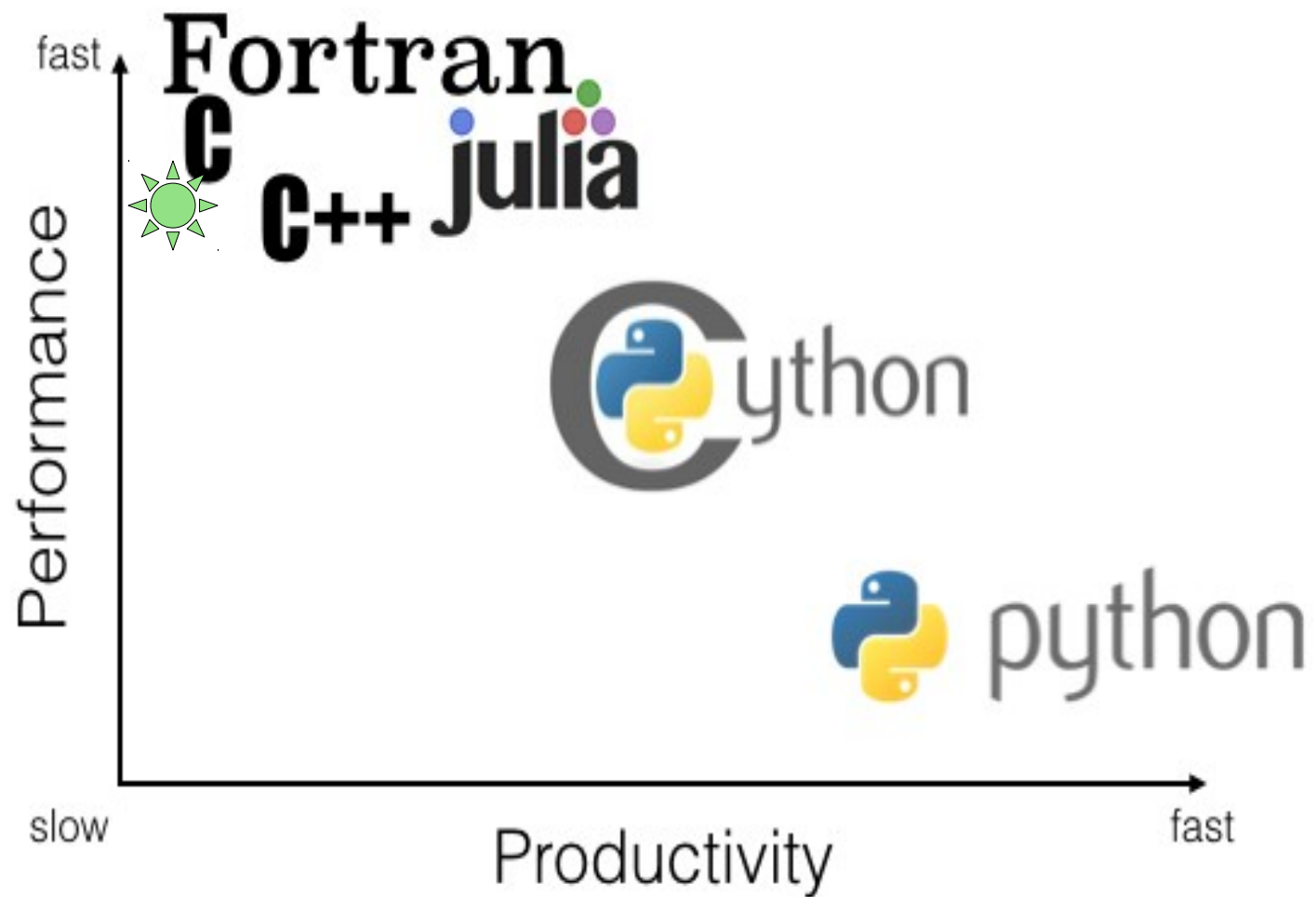
C x Java

- Não possui suporte a orientação a objetos
- Linguagem de nível intermediário:
 - controle mais direto do hardware, porém
 - também suporta estruturas complexas
- Gerenciamento de memória explícito
- Detecção de erro explícita (sem try/catch)
- Maior performance do programa final
- Maior dificuldade de manutenção

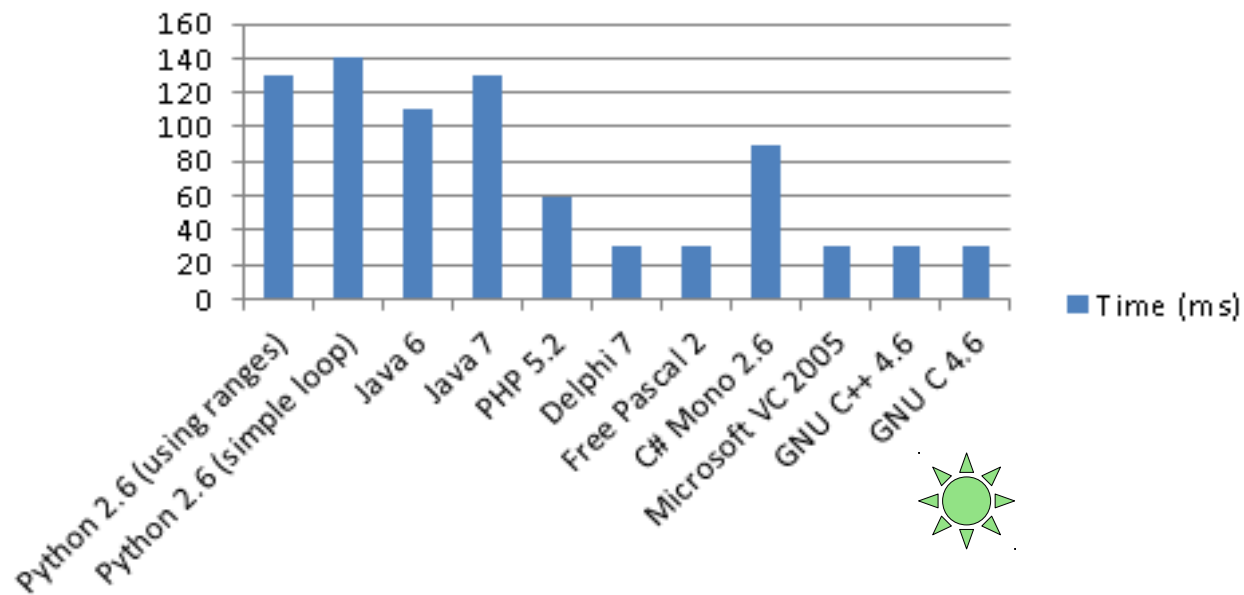


 C++	 JavaScript
 Java/C#	 PHP(Without MySQL)
 Ruby	 Pascal
 Perl	 Lisp
 Visual Basic	 Haskell
 Python	 C
 Assembly	 Cobra
	 Delphi

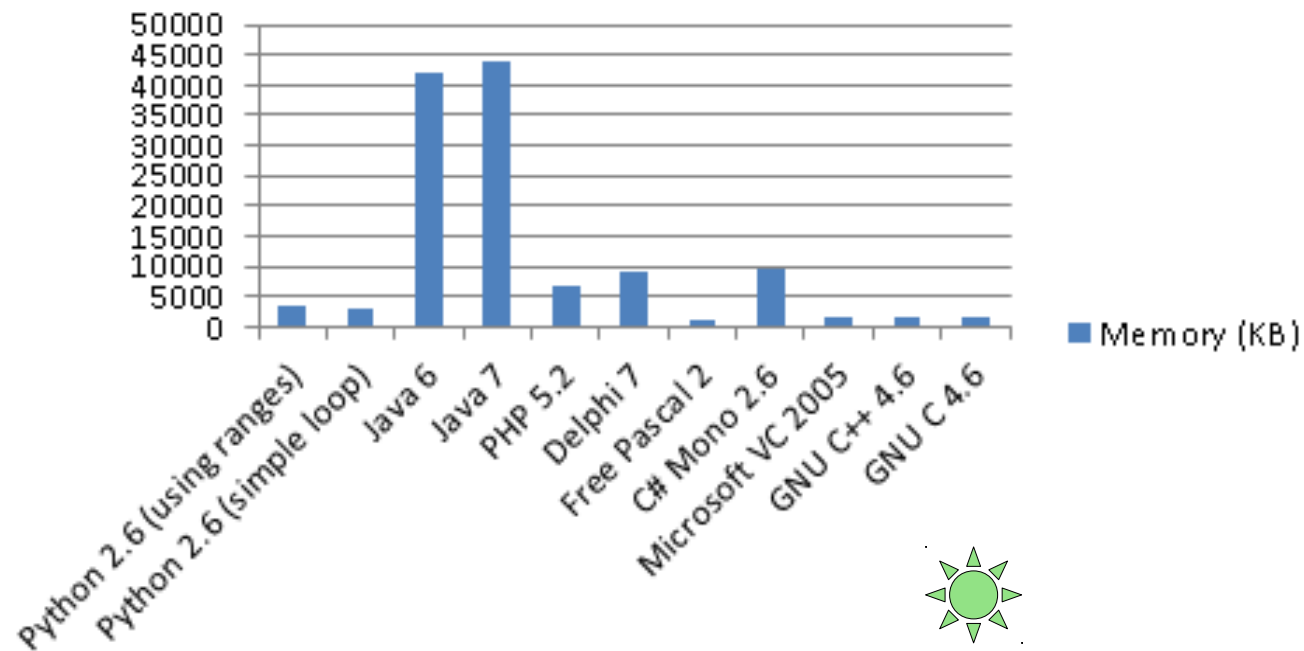




Time (ms)



Memory (KB)



Sobre a linguagem de programação C

Precisa instalar um compilador no seu computador:

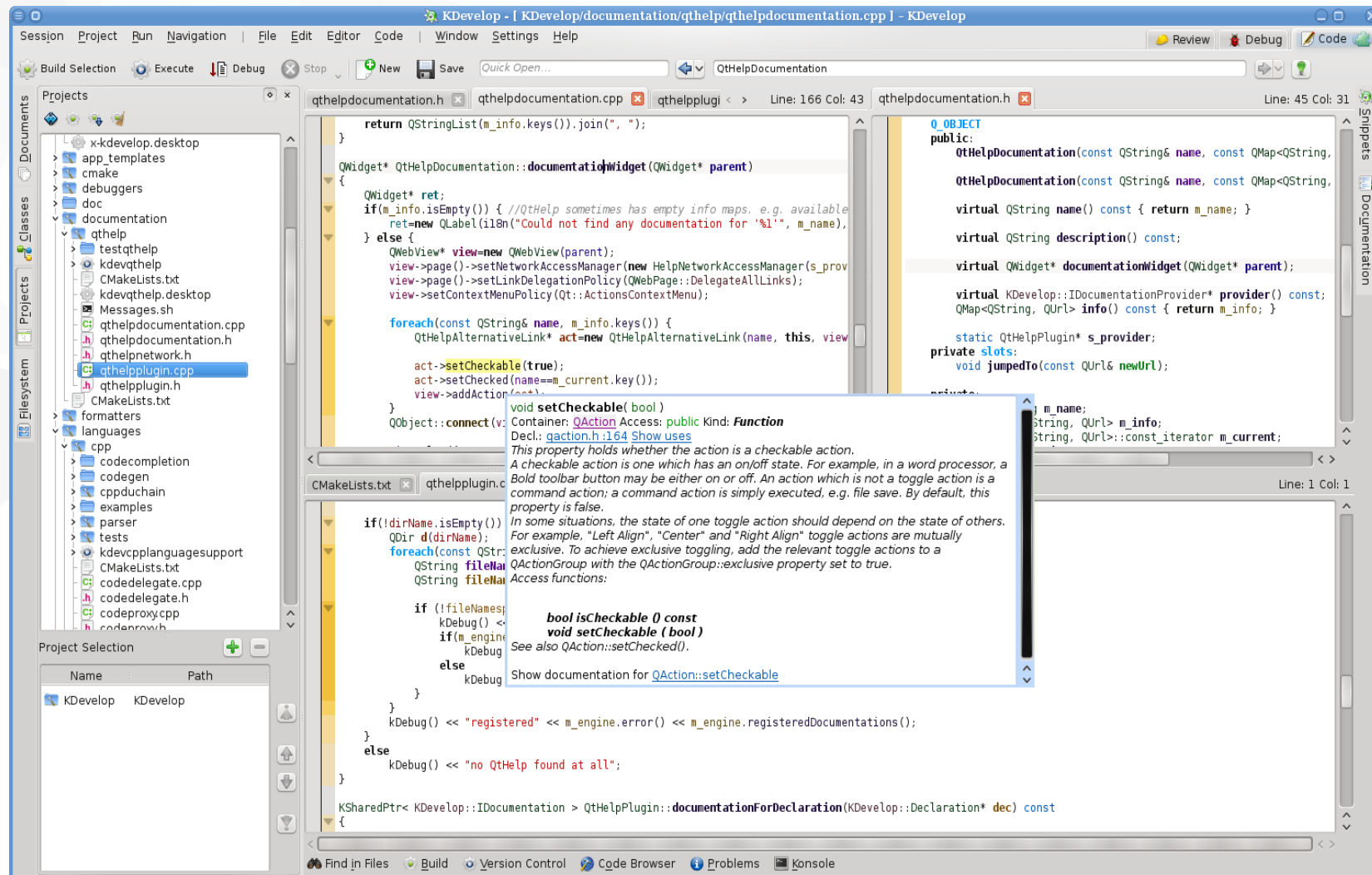
Usaremos o GNU Compiler Collection (gcc)

- **No Linux:** é padrão.
- **No Mac/OS:** terá que instalar as ferramentas Xcode/Developer.
- **No MS-Windows:**
 - Cygwin (simulação completa de um ambiente UNIX).
 - Ming-W (Minimalist GNU for Windows).

Sobre o IDE (Integrated development environment)

Ambiente de desenvolvimento integrado:

- Kdevelop
- Code Blocks
- Netbeans C++
- Eclipse CDT
- Dev C++
- C-Free
- Vi
- Emacs



The image shows a web browser window displaying the Cloud9 website. The browser's address bar shows the URL `https://c9.io`. The website has a blue header with the Cloud9 logo, navigation links for WEB, BLOG, and PRICING, and a SIGN IN button. The main content area features the headline "Your development environment, in the cloud" and a sub-headline "Cloud9 combines a powerful online code editor with a full Ubuntu workspace in the cloud". Below this are two buttons: SIGN UP and LEARN MORE.

Below the website content, a screenshot of the Cloud9 code editor interface is shown. The browser's address bar displays `https://c9.io/fjakobs/example`. The code editor has a menu bar with options: Cloud9, File, Edit, Find, View, Goto, Run, Tools, Window, and Help. The interface includes a sidebar with "Navigate" and "Workspace" views. The main editor area shows two files: `edit_session.js` and `hello.js`. The `edit_session.js` file contains JavaScript code for a document object model (DOM) manipulation, including methods like `getValue`, `toString`, and `getSelection`. The `hello.js` file contains JavaScript code for a simple HTTP server using the `http` module, which listens on port 8000 and responds with "Hello World". The bottom of the editor shows a file explorer with `README.md` and `package.json`.

Exemplo 0 (arquivo: olaMundo.c)

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Ola Mundo!\n");
5     return 0;
6 }
```

Exemplo 0 (arquivo: olaMundo.c)

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Ola Mundo!\n");
5     return 0;
6 }
```

```
jmenac@aed1:~/workspace/aed1-01 $ gcc olaMundo.c -o ola
jmenac@aed1:~/workspace/aed1-01 $ ./ola
Ola Mundo!
```

```
jmenac@aed1:~/workspace/aed1-01 $ gcc olaMundo.c -o ola && ./ola
Ola Mundo!
```


Exemplo 0 (arquivo: olaMundo.c)

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Ola Mundo!\n");
5     return 0;
6 }
```

```
jmenac@aed1:~/workspace/aed1-01 $ echo $?
```

```
0
```

Exemplo 0 (arquivo: olaMundo.c)

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Ola Mundo!\n");
5     return 110;
6 }
```

```
jmenac@aed1:~/workspace/aed1-01 $ gcc olaMundo.c -o ola && ./ola
Ola Mundo!
jmenac@aed1:~/workspace/aed1-01 $ echo $?
110
```

Exemplo 1 (arquivo: olaMundo2.c)

```
1 #include <stdio.h>
2
3 int main() {
4     char mensagem[] = "Ola Mundo!\n";
5     printf("%s",mensagem);
6     return 0;
7 }
```

```
jmenac@aed1:~/workspace/aed1-01 $ gcc olaMundo2.c -o olaMundo2 && ./olaMundo2
Ola Mundo!
```

Linguagem C

- char tem 8 bits (não é 16 bits como em Java)

- não existe tipo booleano (usar int ou char):

- 0 falso
- $\neq 0$ verdadeiro

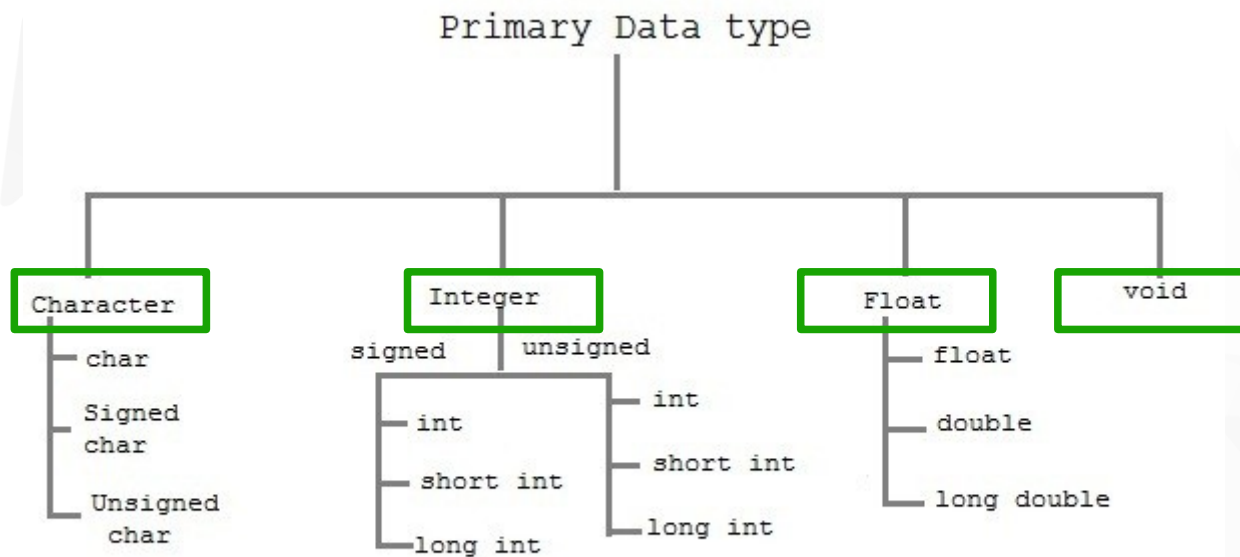
Na verdade C99 permite o uso de palavras true/false nos programas (mas o compilador os trata como os valores 1 e 0 de qualquer forma)

- tipos inteiros podem ser **signed** ou **unsigned**

- não tem tipo string: **usa-se vetor de char**

Linguagem C: Tipos de dados

- **Tipos de dados primários.**
- Tipos de dados derivados.
- Tipos definidos pelo usuário.



Linguagem C: Números inteiros

Size and range of Integer type on 16-bit machine

Type	Size(bytes)	Range
int or signed int	2	-32,768 to 32767
unsigned int	2	0 to 65535
short int or signed short int	1	-128 to 127
long int or signed long int	4	-2,147,483,648 to 2,147,483,647
unsigned long int	4	0 to 4,294,967,295

Linguagem C: Números reais

Size and range of Integer type on 16-bit machine

Type	Size(bytes)	Range
Float	4	3.4E-38 to 3.4E+38
double	8	1.7E-308 to 1.7E+308
long double	10	3.4E-4932 to 1.1E+4932

Linguagem C: Caracteres

Size and range of Integer type on 16-bit machine

Type	Size(bytes)	Range
char or signed char	1	-128 to 127
unsigned char	1	0 to 255

Linguagem C: void

void type

void type means no value. This is usually used to specify the type of functions.

Tipos de dados

```
1 #include<stdio.h>
2
3 int main() {
4     printf("Size of char is %ld bytes\n",sizeof(char));
5     printf("Size of short is %ld bytes\n",sizeof(short));
6     printf("Size of int is %ld bytes\n",sizeof(int));
7     printf("Size of long is %ld bytes\n",sizeof(long));
8     printf("Size of float is %ld bytes\n",sizeof(float));
9     printf("Size of double is %ld bytes\n",sizeof(double));
10    printf("Size of long double is %ld bytes\n",sizeof(long double));
11    return 0;
12 }
```


Tipos de dados

```
1 #include<stdio.h>
2
3 int main() {
4     printf("Size of char is      %ld bytes\n",sizeof(char));
5     printf("Size of short is     %ld bytes\n",sizeof(short));
6     printf("Size of int is       %ld bytes\n",sizeof(int));
7     printf("Size of long is      %ld bytes\n",sizeof(long));
8     printf("Size of float is     %ld bytes\n",sizeof(float));
9     printf("Size of double is    %ld bytes\n",sizeof(double));
10    printf("Size of long double is %ld bytes\n",sizeof(long double));
11    return 0;
12 }
```

```
jmenac@aed1:~/workspace/aed1-01 $ gcc tiposDeDados.c -o tiposDeDados && ./tiposDeDados
```

```
Size of char is      1 bytes
Size of short is     2 bytes
Size of int is       4 bytes
Size of long is      8 bytes
Size of float is     4 bytes
Size of double is    8 bytes
Size of long double is 16 bytes
```

```
jmenac@aed1:~/workspace/aed1-01 $ uname -a
```

```
Linux jmenac-aed1-1261073 3.14.13-c9 #1 SMP Tue Aug 19 16:13:44 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
```

Conversão de tipo

```
int a = 20;  
long b;
```

```
b = (long) a;
```

→ Valor preservado, só o tipo é alterado

Conversão de tipo

```
int a;  
long b = 32;
```

```
a = (int) b;
```

→ Perigoso: se **b** não “cabe” num int, seu valor será truncado

Conversão de tipo

```
int a = 17,  
b = 3;
```

```
double x = a / b;
```

→ Neste caso x contém o valor 5

Conversão de tipo

```
int a = 17,  
b = 3;
```

```
double x = a / (double) b;
```

→ Neste caso x contém o valor 5.66667

Conversão de tipo

```
double x = 5.93487;  
int j = (int) x;
```

→ Neste caso j contém o valor 5

Vetores

- Para declarar um vetor de inteiros de comprimento 100 faça:
int v[100];
→ Note que os índices vão de **0 a 99**
- Para inicializar o vetor com zeros faça:
int v[100] = {0};
- Para inicializar com outra constante faça manualmente (usando um laço)

Vetores

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int vetor[7] = {1,2,3,4,3,2,1};
6     int i;
7
8     for (i=0; i<7; i++)
9         printf("%d : %d\n", i, vetor[i]);
10
11     return 0;
12 }
```

Vetores

vetor

```
Running /home/ubuntu/workspace/aed1-01/exemploVetor.c
```

```
0 : 1
```

```
1 : 2
```

```
2 : 3
```

```
3 : 4
```

```
4 : 3
```

```
5 : 2
```

```
6 : 1
```

```
7 : 0
```

```
8 : 0
```

```
9 : 0
```

```
10 : 1382481605
```

```
11 : 32592
```

```
12 : 0
```

```
13 : 0
```

```
14 : -1635315752
```

```
15 : 32767
```

```
16 : 0
```

Vetores: Exercício

SOCORRAM-ME SUBI NO ONIBUS EM MARROCOS

Vetores: Exercício

- Dado um vetor de inteiros, $v[0..6]$, crie um programa (pseudocódigo) para determinar se a sequência de números corresponde a um palindromo.
 - $[1,2,3,4,3,2,1]$: vetor palindromo
 - $[1,2,3,4,5,6,7]$: não é vetor palindroma

Qual é o número mínimo de comparações necessárias para verificar se um vetor de tamanho N é ou não palidromo?

Vetores: Exercício

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int vetor[7] = {1,2,3,4,3,2,1};
6     int N = 7;
7     int i;
8
9     for (i=0; i<N/2; i++) {
10         if (vetor[i] != vetor[N-i-1]) {
11             printf("Vetor nao palindromo\n");
12             return 0;
13         }
14     }
15
16     printf("Vetor palindromo\n");
17     return 0;
18 }
```

Difícil?

Experimente a linguagem whitespace

The following is a commented Whitespace program that simply prints "Hello, world!", where each Space, Tab, or Linefeed character is preceded by the identifying comment "S", "T", or "L", respectively:

```
S S S T S S T S S S L
T L
S S S S S T T S S T S T L
T L
S S S S S T T S T T S S L
T L
S S S S S T T S T T S S L
T L
S S S S S T T S T T T T L
T L
S S S S S T S T T S S L
T L
S S S S S T S S S S S L
T L
S S S S S T T T S T T T L
T L
S S S S S T T S T T T T L
T L
S S S S S T T T S S T S L
T L
S S S S S T T S T T S S L
T L
S S S S S T T S S T S S L
T L
S S S S S T S S S S T L
T L
S S L
L
L
```

Atividades para casa

- **Teste todos os exercícios na plataforma cloud9**
 - **crie uma conta nova nessa plataforma**