



# Processamento da Informação

## Algoritmos Sequenciais

Prof. Jesús P. Mena-Chalco  
CMCC/UFABC

Q1/2020

# Construção de programas

Para a construção de programas é necessário um **conjunto de instruções** colocadas em ordem sequencial lógica:  
**Algoritmo.**



# Algoritmos

# Algumas definições de algoritmo

- **Sequência ordenada de passos** que deve ser seguida para a realização de uma tarefa (ASCENCIO, 1999).
- **Regras formais** para a obtenção de um resultado ou **solução de um problema**, englobando formulas de expressões aritméticas (MANZANO, 1997).
- **Sequência finita de instruções ou operações** cuja execução, em tempo finito, resolve um problema computacional, qualquer que seja sua instância (SALVETTI, 1999)

# Algumas definições de algoritmo

## **Algoritmo:**

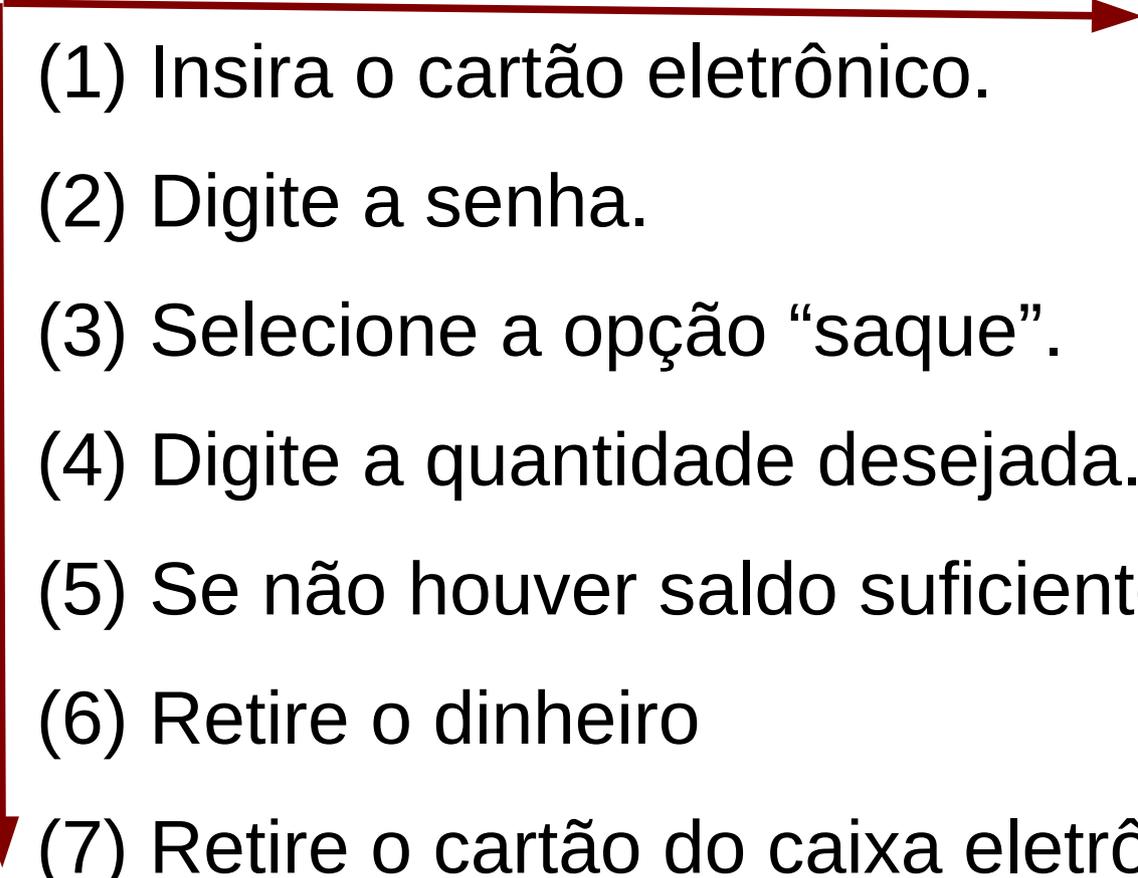
Sequência lógica são passos executados até atingir um objetivo ou solução de um problema.

## **Exemplos de situações onde uma sequência lógica de passos é necessária:**

- Fazer um bolo
- Construir um robô para explorar um local desconhecido
- Trocar uma lâmpada

# Exemplo de algoritmo

## Sacar dinheiro de um caixa eletrônico (2012):

- 
- (1) Insira o cartão eletrônico.
  - (2) Digite a senha.
  - (3) Selecione a opção “saque”.
  - (4) Digite a quantidade desejada.
  - (5) Se não houver saldo suficiente, continue no passo (7).
  - (6) Retire o dinheiro
  - (7) Retire o cartão do caixa eletrônico

# Exemplo de algoritmo

## Sacar dinheiro de um caixa eletrônico (2020):

- (1) ~~Insira o cartão.~~ Use o sensor biométrico (iris, face, digitais).
- (2) ~~Digite a senha.~~
- (3) Selecione a opção “saque”.
- (4) Digite a quantidade desejada.
- (5) Se não houver saldo suficiente, continue no passo (7).
- (6) Retire o dinheiro
- (7) ~~Retire o cartão.~~ Transação finalizada.

# A relação com a tecnologia?

A sequência de passos **depende do tipo do tecnologia utilizada.**

A sequência de passos **é limitada pela quantidade finita de possíveis operações.**

# Al-Khorezmi: Um Matemático pouco conhecido



Matemático, astrônomo, astrólogo, geógrafo e autor Persa (Bagdad, c.850)

Descreveu o sistema numérico atual a um nível entendível.

*al-Khwarizmi, Al-Khwarizmi, Al-Khawaritzmi ou al-Khowarizmi*

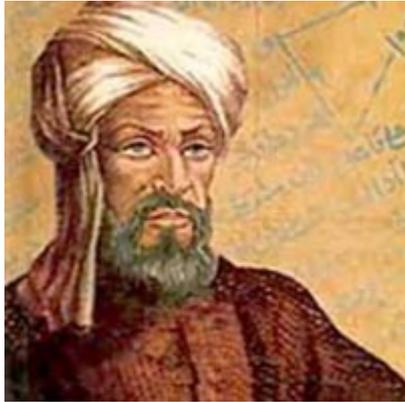
*O primeiro pensador algoritmico.*



*Selo postal (Rusia, 1983)  
Comemorando o aniversário 1200*

Quando em uma subtração nada queda, então escreva um pequeno círculo para que esse lugar não permaneça vacío (Al-Khorezmi explicando o zero, Século IX)

# Al-Khorezmi: Um Matemático pouco conhecido



Os termos:

- **Algoritmo**
- **Algarismo** (número/digito)
- **Algol** (linguagem de programação)  
provém de seu nome.

- Al-Khorezmi = “de Khorezm”
- Algoritmos datam dos gregos (por exemplo, algoritmo de Euclides para **calcular o máximo divisor comum**).
- Al-Khorezmi foi o primeiro em projetar algoritmos pensando na sua eficiência para o calculo raizes de equações.
- Usou um tipo mecânico similar a um ábaco.

# Algoritmos e programas

Para que um computador desempenhe uma tarefa é necessário que uma sequência de ações (**algoritmo**) seja especificada de uma forma compreensível pela máquina.

Um programa de computador nada mais é que um algoritmo escrito de forma compreensível pelo computador.

Ações especificadas de maneira “**formal**”.

# Algoritmos e programas

Um algoritmo computacional tem que **terminar em tempo finito**.

Se o algoritmo não terminar em tempo finito, alguns pesquisadores classificam essa sequência de passos como **método computacional**.



# Linguagem de programação

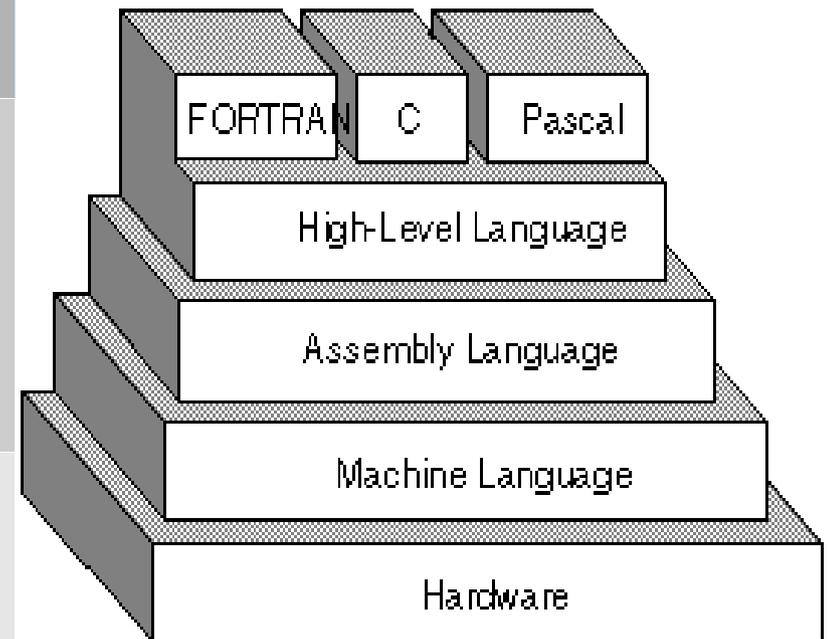
# O que é uma linguagem de programação?

É um conjunto limitado de:

- **símbolos** (comandos, identificadores, caracteres, etc)
- **regras de sintaxe** (descrevem de forma precisa ações)

# O que é uma linguagem de programação?

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







# Programa na linguagem Basic

```
10  REM MOSTRA SEQUENCIA DE NUMEROS de 1 A N
20  INPUT N
30  A = 1
40  PRINT A
50  A = A + 1
60  IF A <= N THEN GOTO 40
```

# Programa na linguagem C

```
/* Mostra uma sequencia de numeros de 1 a n */  
Main() {  
    int a, n;  
  
    scanf("%d", &n);  
  
    for(a = 1; a <= n; a++) {  
        printf("%d", a);  
    }  
}
```

# Programa na linguagem Java

```
import java.util.*;

class MeuPrograma
{
    public static void main (String[] args) {
        Scanner in = new Scanner(System.in);

        int n = in.nextInt();

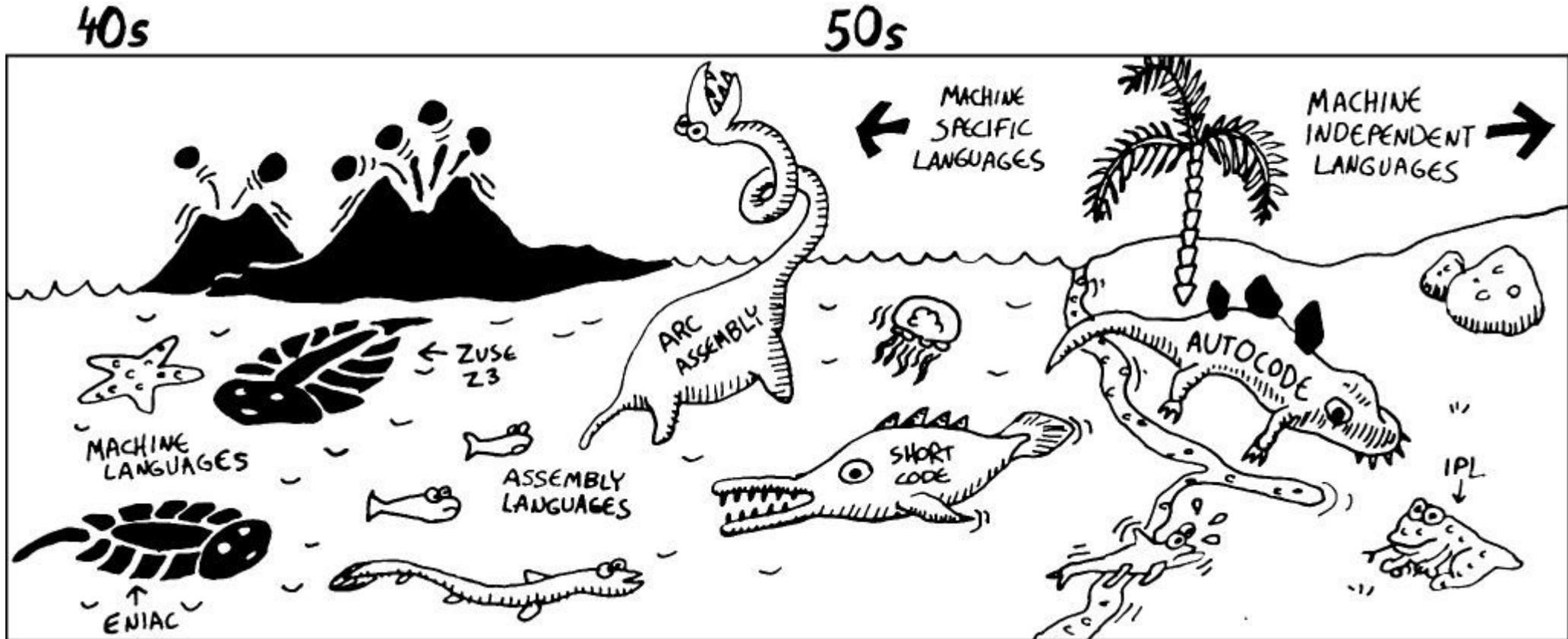
        for(int a = 1; a <= n; a++) {
            System.out.println(a);
        }
    }
}
```

# Programa na linguagem Python

```
# Mostra uma sequencia de numeros de 1 a n
n = int( input() )

for a in range(1, n+1):
    print(a)
```

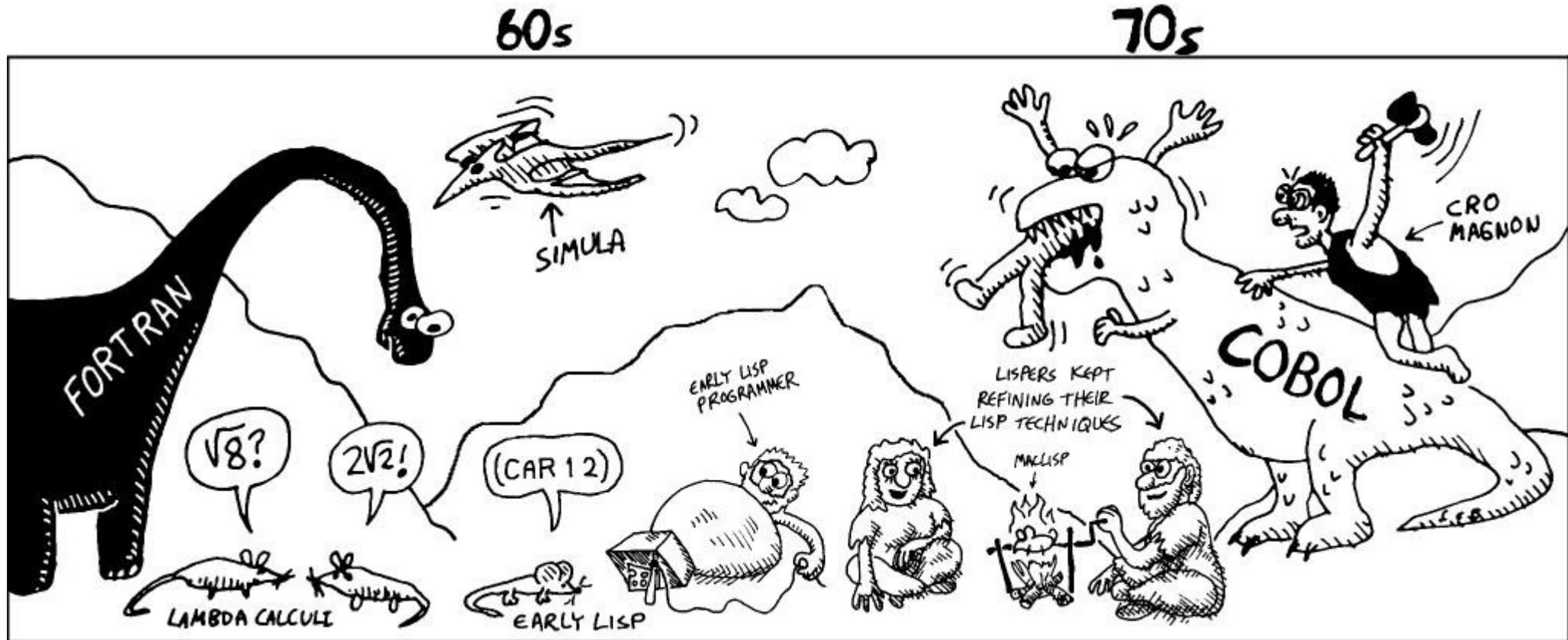
# Linguagens de programação



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](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)

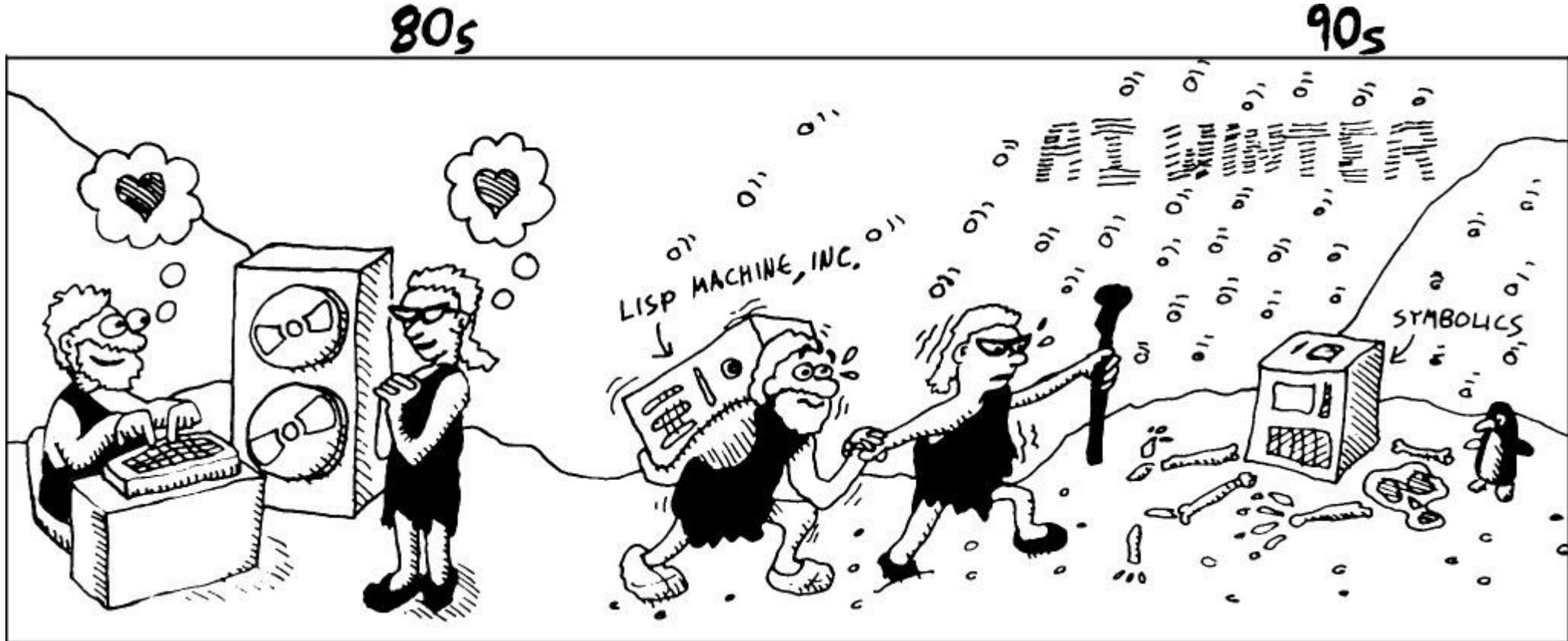
# Linguagens de programação



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](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

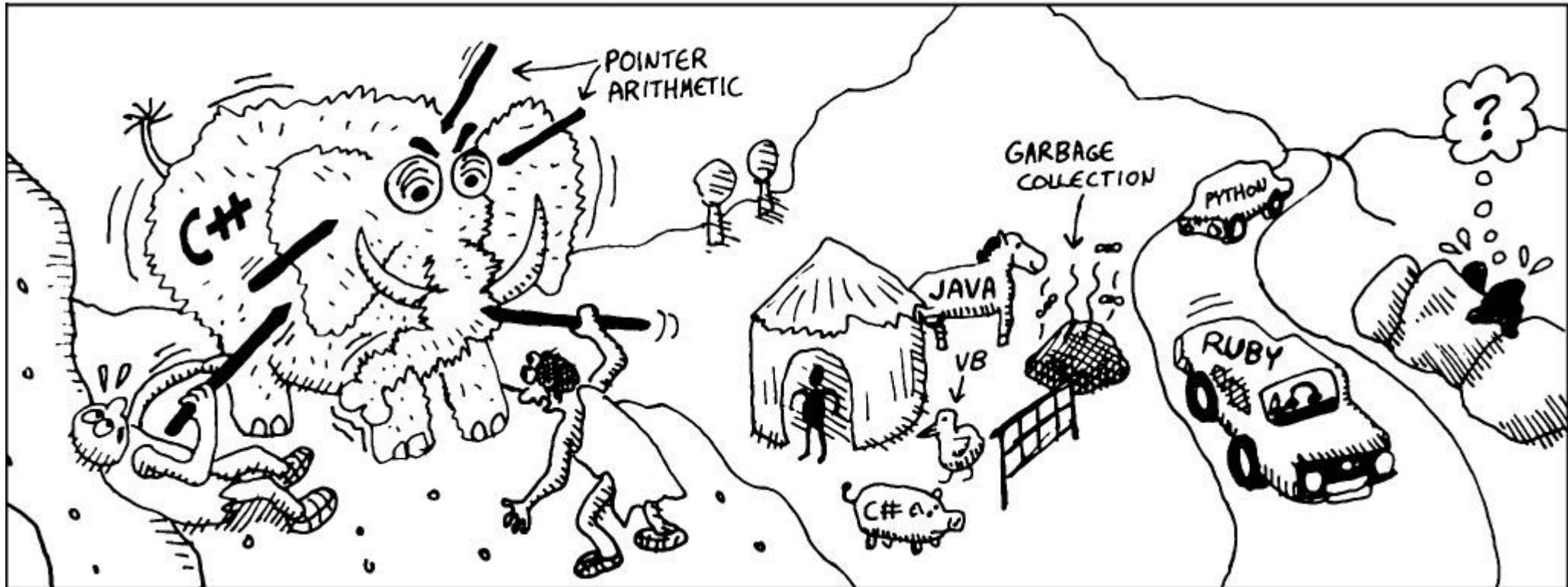


Fonte: "Land of Lisp: Learn to Program in List, One Game at a time". Autor: Conrad Barski.

[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](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

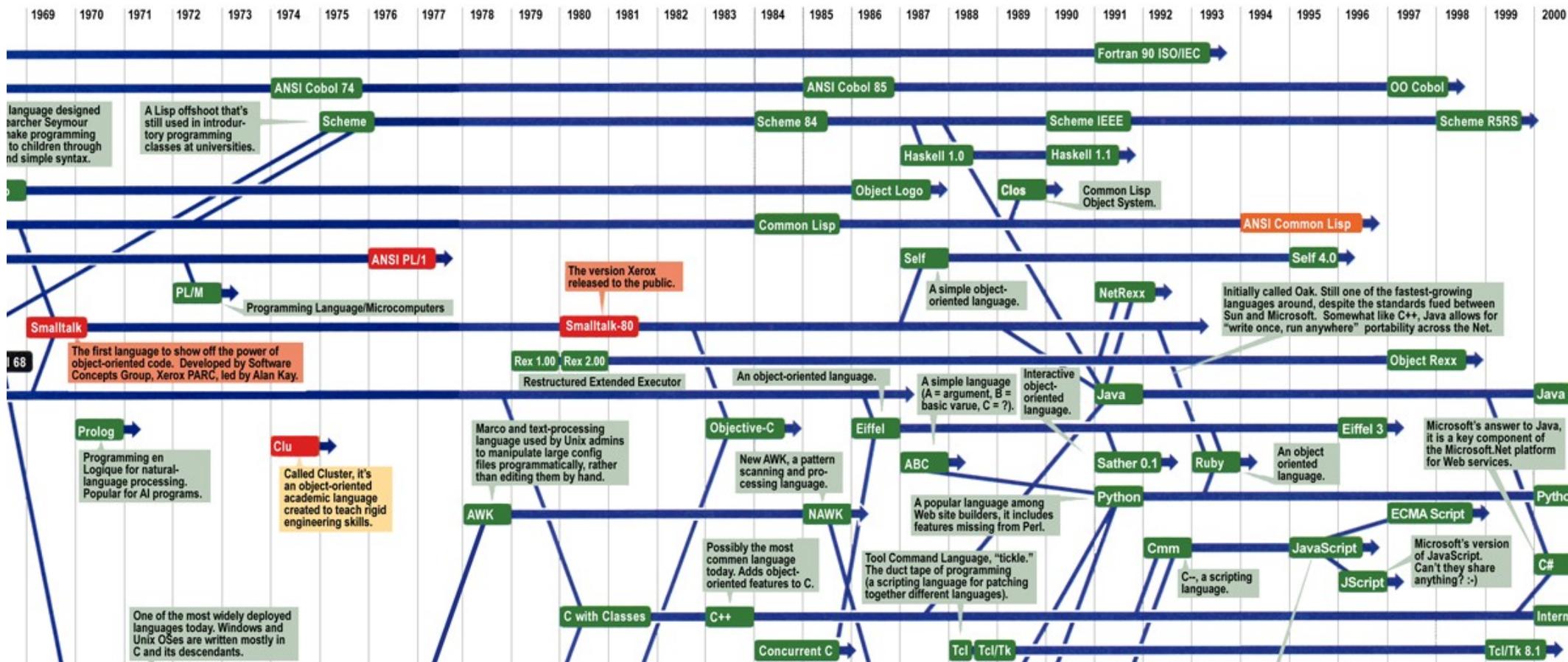
2000



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](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



(\*) <http://www.digibarn.com/collections/posters/tongues/>

# Linguagens de programação

## The Evolution Of Computer Programming Languages

```
00 01 02 03 04 05 06 07 08 09
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49
50 51 52 53 54 55 56 57 58 59
60 61 62 63 64 65 66 67 68 69
70 71 72 73 74 75 76 77 78 79
80 81 82 83 84 85 86 87 88 89
90 91 92 93 94 95 96 97 98 99
```



Hex

```
0000 0000 0000 0000
0001 0000 0000 0000
0002 0000 0000 0000
0003 0000 0000 0000
```



Assembler

```
Module "main.c"
Module "lib.c"
Module "utils.c"
Module "main.c"
Module "lib.c"
```



C

```
PROGRAM NAME, DATE, TIME
PROGRAM NAME, DATE, TIME
PROGRAM NAME, DATE, TIME
PROGRAM NAME, DATE, TIME
```



Fortran

```
Module "main.c"
Module "lib.c"
Module "utils.c"
Module "main.c"
Module "lib.c"
```



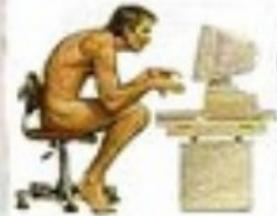
C++

```
Module "main.c"
Module "lib.c"
Module "utils.c"
Module "main.c"
Module "lib.c"
```



Java

```
Module "main.c"
Module "lib.c"
Module "utils.c"
Module "main.c"
Module "lib.c"
```



# Linguagens mais utilizadas

Rank	Language	Type	Score
1	Python	  	100.0
2	Java	  	96.3
3	C	  	94.4
4	C++	  	87.5
5	R		81.5
6	JavaScript		79.4
7	C#	   	74.5
8	Matlab		70.6
9	Swift	 	69.1
10	Go	 	68.0

# Quem usa python?

- Wikipedia
- Google
- Yahoo!
- CERN
- NASA
- Facebook
- Amazon
- Instagram
- Spotify



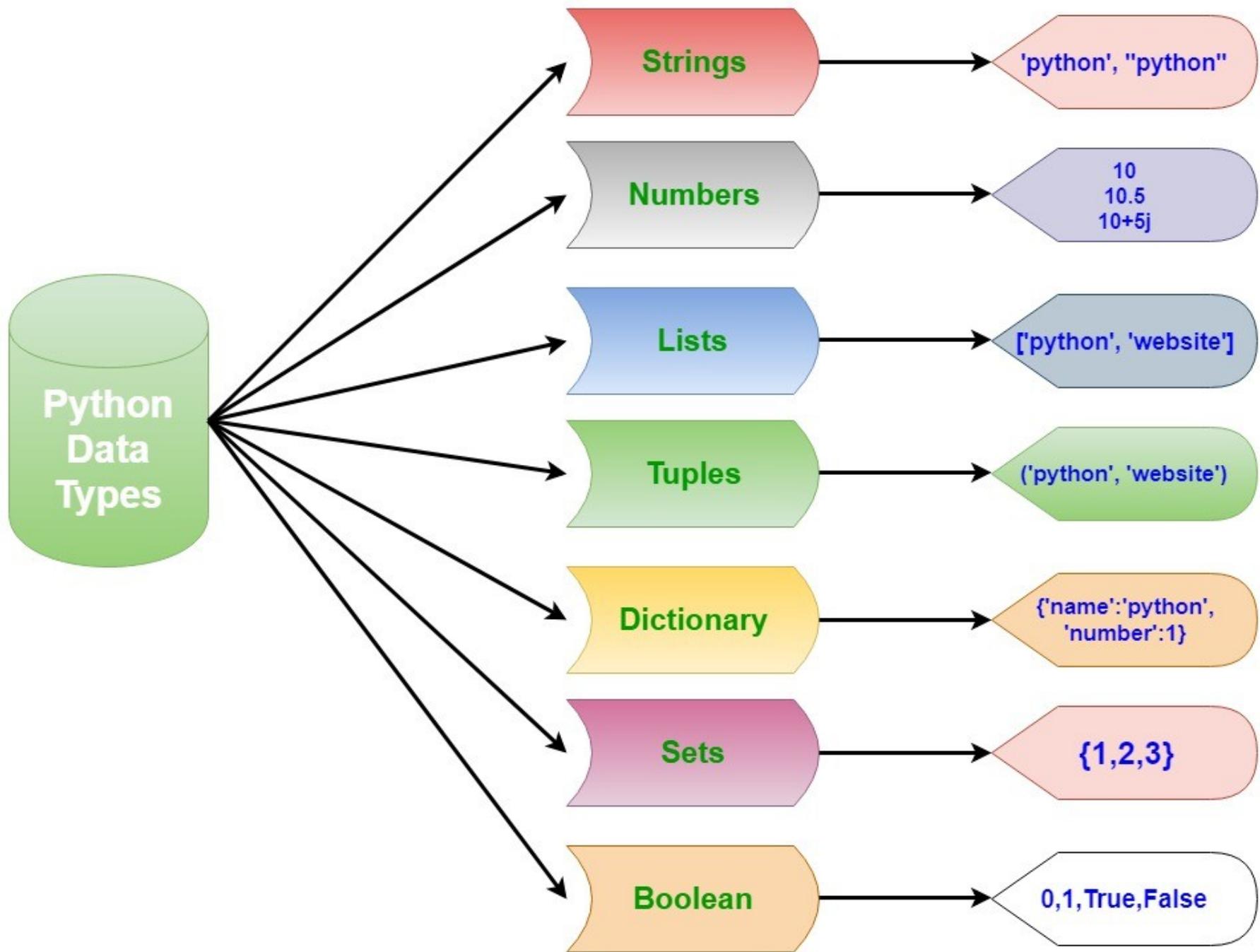


# Tipos de dados

# Tipos de dados em Python

Em geral os dados são classificados em três tipos: **numéricos**, **literais**, e **lógicos**.

Mas existem outros tipos de dados ...



# Tipos de dados em Python

Os tipos de dados que usaremos são os **primitivos**:

Name	Type	Description
Integers	int	Whole numbers, such as: 3 300 200
Floating point	float	Numbers with a decimal point: 2.3 4.6 100.0
Strings	str	Ordered sequence of characters: "hello" 'Sammy' "2000" "楽しい"
Lists	list	Ordered sequence of objects: [10,"hello",200.3]
Booleans	bool	Logical value indicating True or False

The term **floating point** is derived from the fact that there is no fixed number of digits before and after the decimal **point**; that is, the decimal **point** can **float**. There are also representations in which the number of digits before and after the decimal **point** is set, **called** fixed-point representations.

[www.webopedia.com](http://www.webopedia.com) > TERM > floating\_point\_number

[What is Floating-Point Number? Webopedia Definition](http://www.webopedia.com)







# Precisão finita

## Em Python:

```
>>> 5/3
```

```
1.66666666666666666667
```

```
>>> 2/3+1
```

```
1.66666666666666666665
```

Note que operar com aproximações pode gerar resultados diferentes.

No exemplo, vemos que  $5/3$  é ligeiramente diferente a  $2/3+1$

# Precisão finita

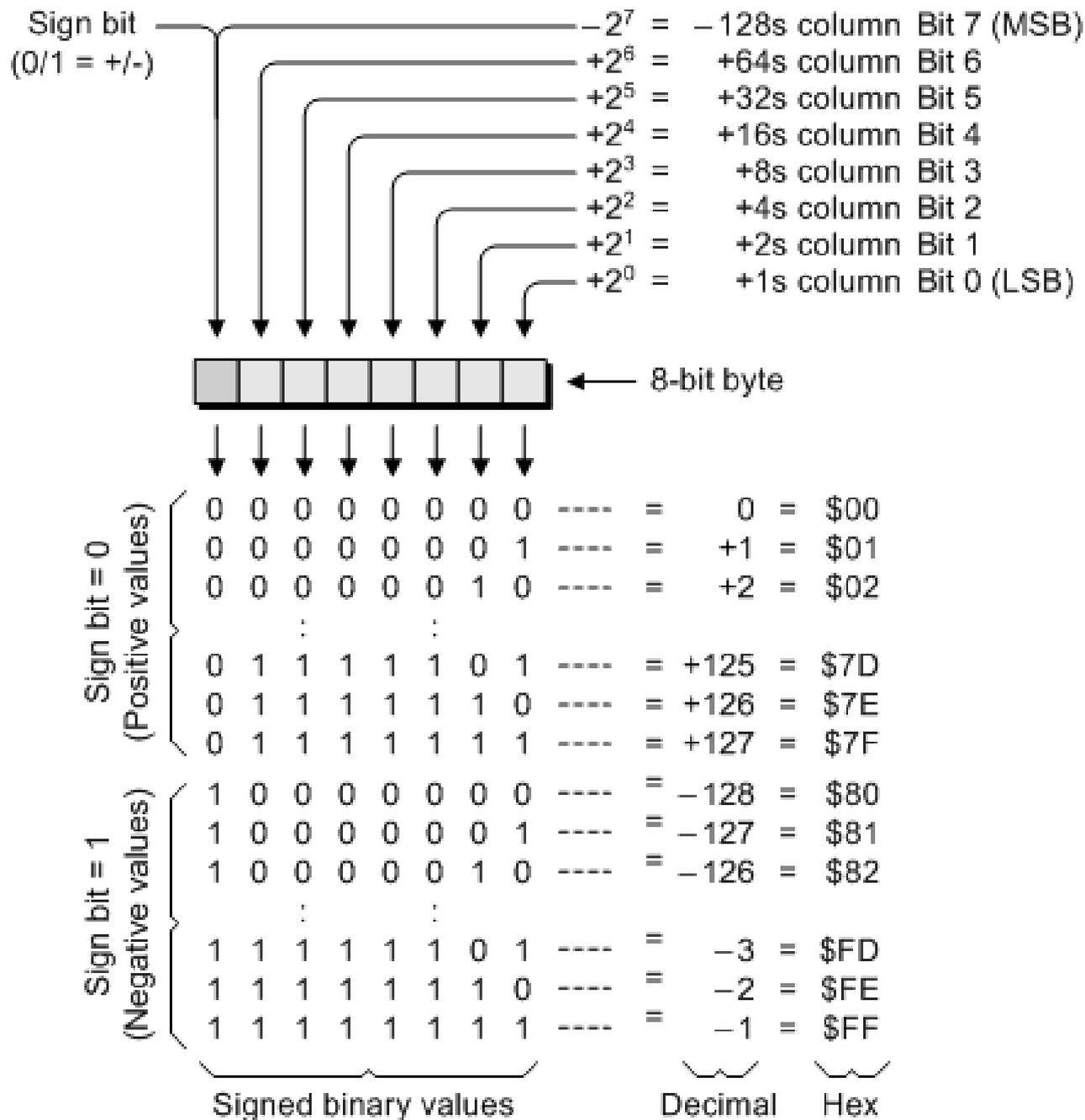
## Em Python:

```
>>> 100000000000 + 0.000000000001  
100000000000.0
```

Veja o exemplo que combina um número muito grande com um número muito pequeno.

### Geralmente

Os números reais (float) são representados  
Usando 64 bits (8 bytes)  
 **$[5.0 \times 10^{-324}, 1.8 \times 10^{308}]$**



Using 8-bits to represent signed binary numbers



# Precedência de operadores

# Precedência de operadores

Qual seria o resultado da execução das seguintes instruções?

- $4 + 5 + 6 / 3$

- $3 ** 2 + 2$

- $-2 ** 4$

# Precedência de (alguns) operadores

Operador	Operação
**	Exponenciação
-	Negação
*	Multiplicação
/	Divisão
//	Divisão inteira
%	Módulo
+	Soma
-	subtração



# **Atividade em sala**

# Precedência de operadores

3. Qual seria o resultado da execução das seguintes operações?

- $9 \% 2$  → 1
- $4 + 3 * 5$  → 19
- $(4 + 3) * 5$  → 35
- $5 + 3 / 4$  → 5.75
- $5 - 2 * 3 ** 4$  → -157

# Precedência de operadores

4. Quais das seguintes instruções dariam erro de sintaxe?

- $6 * \text{-----}8$
- **8 = alunos**
- $(((((4 ** 3)))))$
- $(-(-(-(-5))))$
- **4 += 7 / 2**