Introdução

Programação I **2016.2017**

Teresa Gonçalves
tcg@uevora.pt

Departamento de Informática, ECT-UÉ

Sumário

O que é a Programação?

Linguagem de Programação

Como Programar?

Python

O que é a Programação?

Programação

O que é?

Concepção de métodos para resolução de problemas usando computadores

Criação de **programas** informáticos

Competências

Matemática

linguagens formais para especificar ideias

Engenharia

projectar, unir componentes para formar um sistema, avaliar prós/contras de alternativas

Ciências naturais

observar comportamento de sistemas complexos, tecer hipóteses, testar previsões

Programa

O que é?

Sequência de instruções (escritas numa linguagem de programação) para controlar o comportamento de um sistema

Objetivo

Executar uma computação

Fazer cálculos, controlar periféricos, cesenhar um gráficos, realizar ações

Input/Output

Input: dados necessários para executar a computação

Output: resultado da computação

Linguagem de Programação

Linguagem de programação

O que é?

Linguagem formal concebida para exprimir computações

Linguagem formal

Sintaxe: regras “gramaticais”

Semântica: associação de significados ou ações

Exemplos

- Expressões aritméticas: 3+3=6
- Estrutura molecular: H_2O

Linguagem natural vs Linguagem formal

Linguagem natural

- Utilizada pelas pessoas (Português, Inglês, …)
- Inclui ambiguidade
  - "O João viu a Maria no parque com os binóculos"
  - Propensa a erros/diferenças de interpretação

Linguagem formal

- Não permite ambiguidade*
  - Significado literal, claro, independente do contexto

** Por vezes aceita-se ambiguidade mas reduzida*

Linguagem de baixo nível

Código máquina

Linguagem nativa dos computadores

Exemplo: `100011 00011 01000 00000 00001 000100`

Características

Única linguagem diretamente executável pelo computador

Difícil compreensão

Específica para a arquitatura do computador

Assembly

Utiliza mnemónicas (texto) para representar código máquina

Exemplo: `addi $t0, $zero, 100`

Assemblador: programa que traduz assembly para código máquina

Linguagem de alto nível

Mais próxima da formulação matemática dos problemas

Facilita a escrita, a leitura, a resolução dos problemas

Exemplos

C, Java, Prolog, Python, ...

Características

Mais fácil de entender

Portável

Permite a execução em diferentes arquiteturas de computadores

Traduzida para código máquina por interpretadores ou compiladores

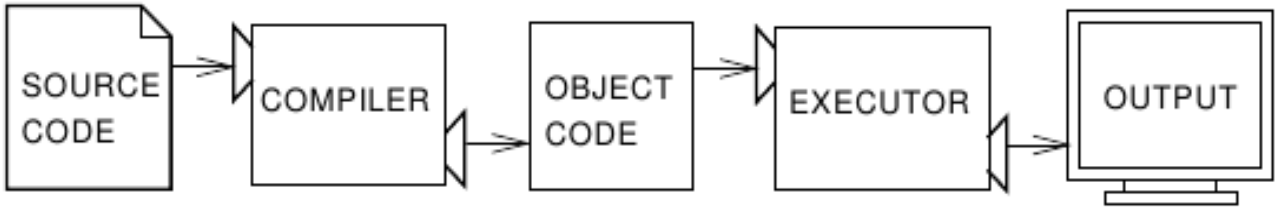
Interpretador vs compilador

Interpretador

Lê, interpreta e executa uma instrução de cada vez

Compilador

Traduz o programa para código máquina executável



Porquê tantas linguagens?

Diferentes nível de abstração

Alto nível: facilita a programação e a deteção e correção de erros

Baixo nível: possivelmente mais eficiente

Diferentes problemas

Cálculos numéricos: Fortran

Raciocínio: Prolog

Scripting: Perl, Python

Diferentes paradigmas

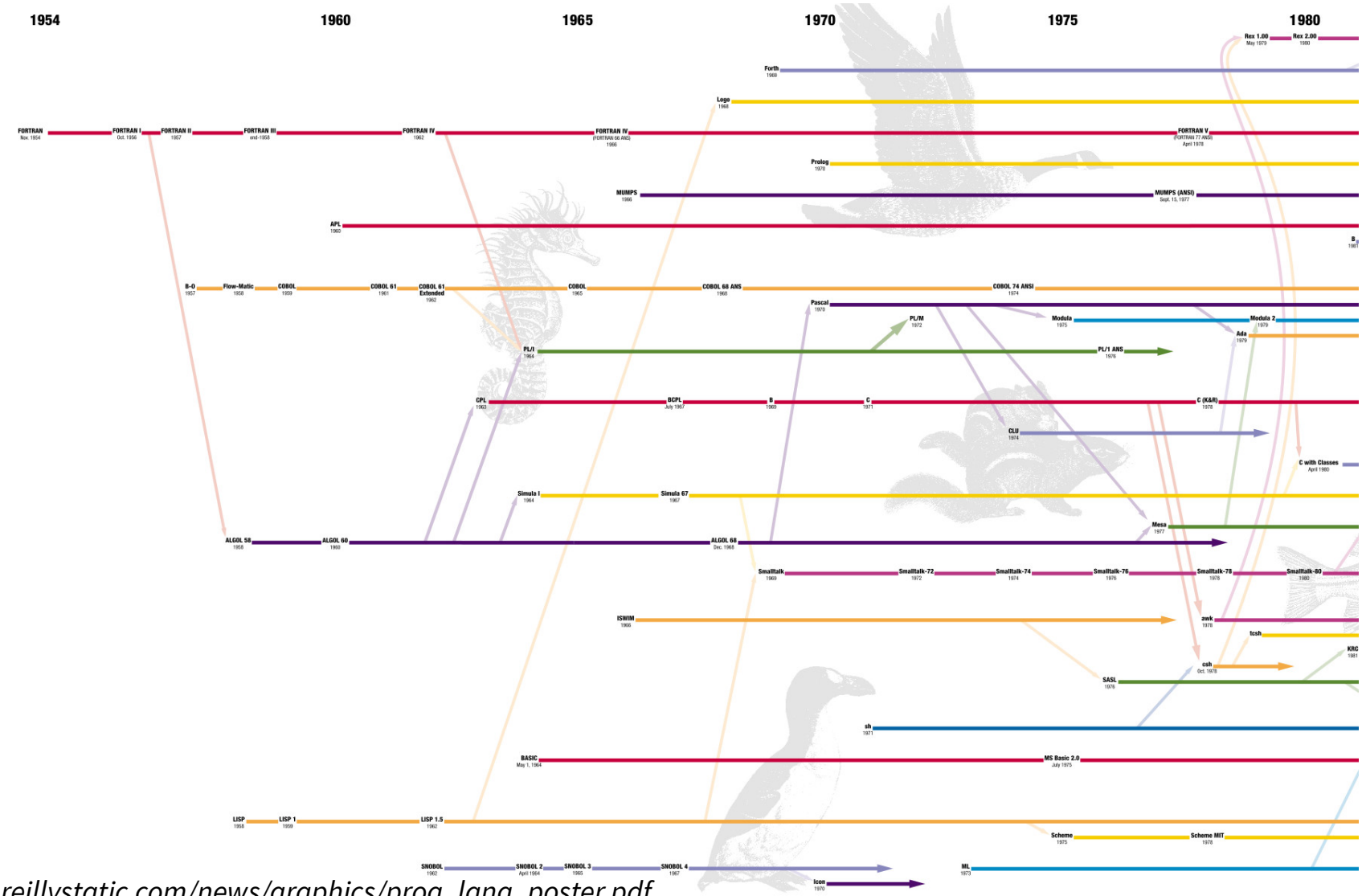
Imperativo: C, Pascal

Funcional: Haskell, Caml

Lógico: Prolog

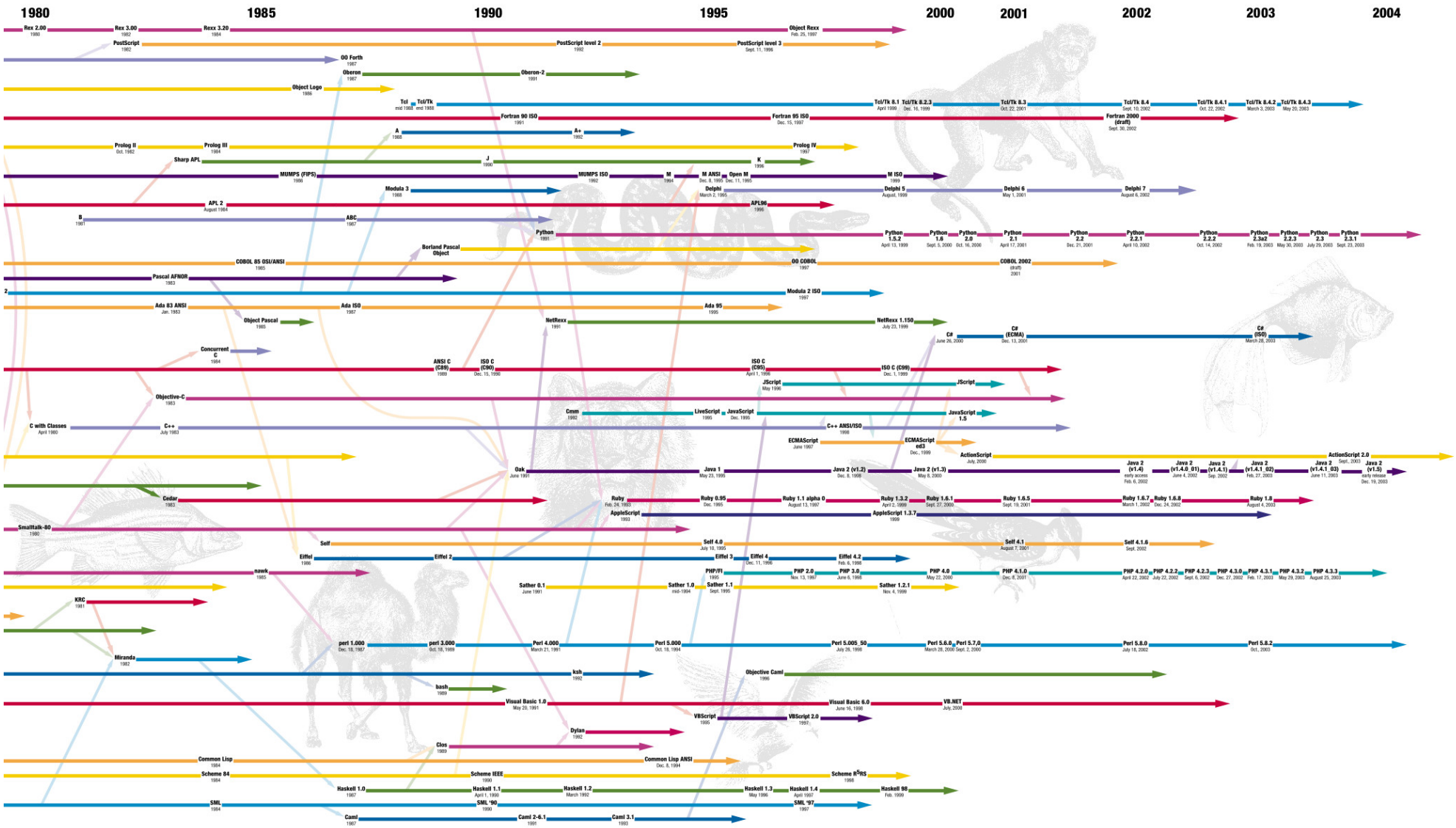
Orientado a objetos: Java, C++

História das linguagens de programação

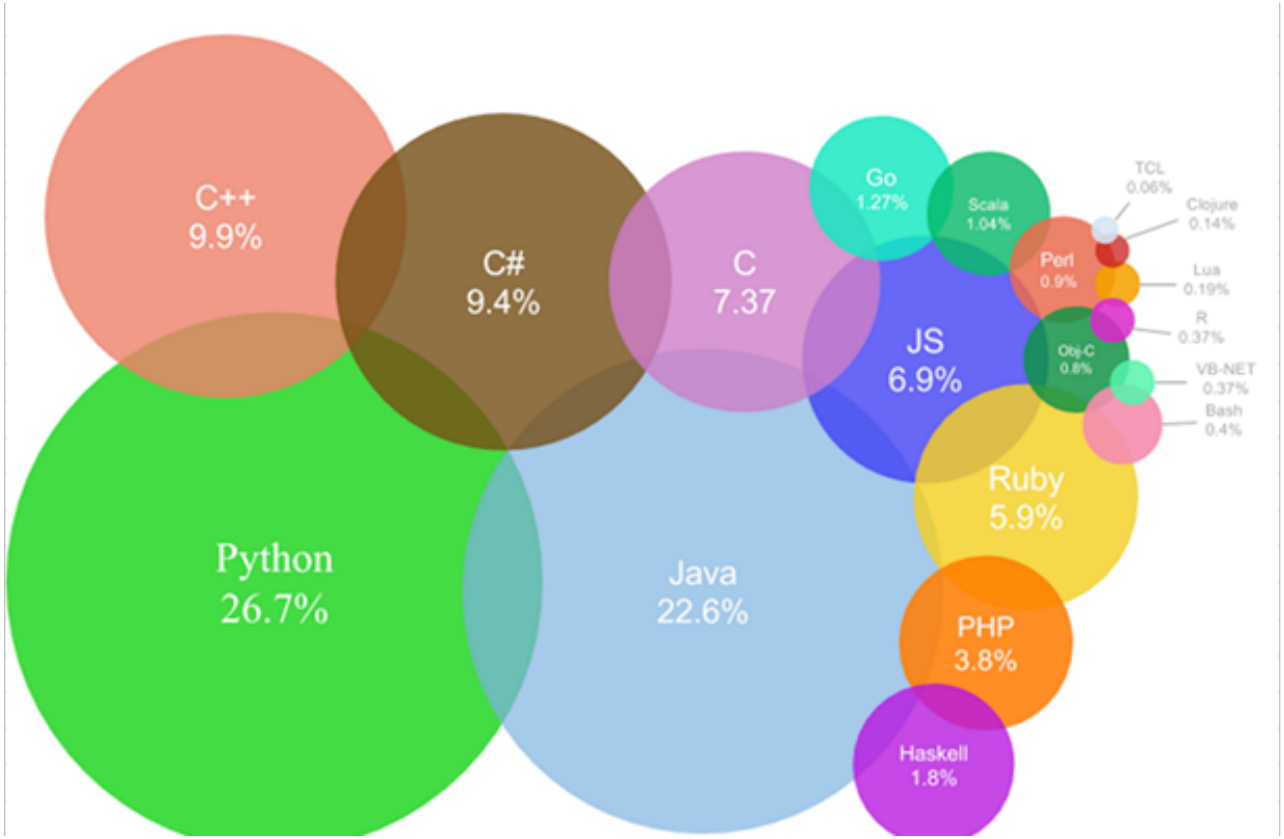


http://cdn.oreillystatic.com/news/graphics/prog_lang_poster.pdf

História das linguagens de programação



Linguagens mais populares



http://blog.codeeval.com/codeevalblog/2016/2/2/most-popular-coding-languages-of-2016

Linguagens mais populares

TIOBE Index

Sep 2016	Sep 2015	Change	Programming language	Ratings	Change
1	1		Java	18.236 %	-1.33 %
2	2		C	10.955 %	-4.67 %
3	3		C++	6.657 %	-0.13 %
4	4		C#	5.493 %	+0.58 %
5	5		Python	4.302 %	+0.64 %
6	7	↑	JavaScript	2.929 %	+0.59 %
7	6	↓	PHP	2.847 %	+0.32 %
8	11	↑	Assembly language	2.417 %	+0.61 %
9	8	↓	Visual Basic .NET	2.343 %	+0.28 %
10	9	↓	Perl	2.333 %	+0.43 %
11	13	↑	Delphi/Object Pascal	2.169 %	+0.42 %
12	12		Ruby	1.965 %	+0.18 %
13	16	↑	Swift	1.930 %	+0.74 %
14	10	↓↓	Objective-C	1.849 %	+0.03 %
15	17	↑	MATLAB	1.826 %	+0.65 %
16	34	↑↑	Groovy	1.818 %	+1.31 %
17	14	↓	Visual Basic	1.761 %	+0.23 %
18	19	↑	R	1.684 %	+0.64 %
19	44	↑↑	Go	1.625 %	+1.37 %
20	18	↓	PL/SQL	1.443 %	+0.36 %

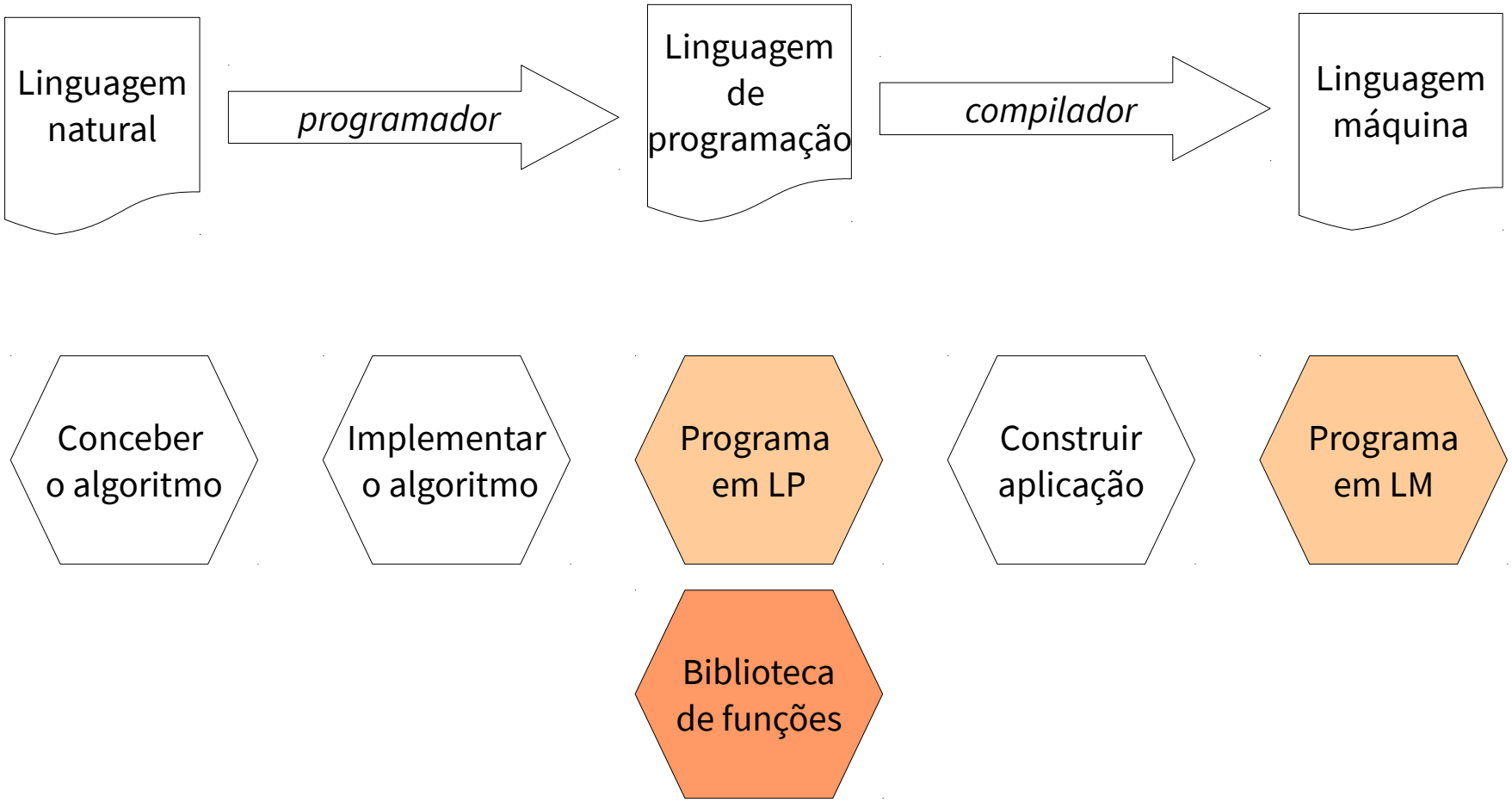
PYPL Index (Worldwide)

Sep 2016	Change	Programming language	Share	Trends
1		Java	23.6 %	-0.6 %
2	↑	Python	13.3 %	+2.4 %
3	↓	PHP	10.0 %	-0.8 %
4		C#	8.6 %	-0.3 %
5	↑↑	Javascript	7.6 %	+0.6 %
6	↓	C++	7.0 %	-0.6 %
7	↓	C	6.8 %	-0.7 %
8		Objective-C	4.5 %	-0.7 %
9	↑↑	R	3.3 %	+0.7 %
10		Swift	3.1 %	+0.4 %
11	↓↓	Matlab	2.8 %	+0.2 %
12		Ruby	2.2 %	-0.3 %
13	↑	VBA	1.6 %	+0.0 %
14	↓	Visual Basic	1.5 %	-0.5 %
15	↑	Scala	1.2 %	+0.3 %
16	↓	Perl	1.0 %	-0.2 %
17		lua	0.6 %	+0.1 %
18		Delphi	0.5 %	+0.0 %
19	↑	Go	0.4 %	+0.1 %
20	↓	Haskell	0.3 %	-0.1 %
21		Rust	0.3 %	+0.1 %

http://statisticstimes.com/tech/top-computer-languages.php

Como programar?

Programar



Baseado num slide de P1, FEUP

Passos da programação

Conceber o algoritmo

Linguagem natural / gráfica

Implementar o algoritmo

Linguagem de programação

Construir a aplicação

Linguagem máquina

Testar

Princípios a utilizar na programação

Programação estruturada

- Decompor um programa em pequenos módulos
  - Reutilização
  - Teste independente
  - Facilidade de modificação

Legibilidade

- Programas devem ser escritos para serem lidos por humanos!
  - Comentários, estrutura, nomes das "coisas", ...

Correção – simplicidade – eficiência

Como aprender?

Estudar, estudar, ...

Praticar, praticar, …

Cometer erros, cometer erros, …

Aprender com os erros, ...

Debugging

Bug

Erro de programação

Durante a programação surgem muitos erros!!!

Debugging

Processo de encontrar erros

Semelhante ao trabalho de um detetive

Suspeita de algo errado; altera o programa; faz um teste para confirmar a resolução

Tipos de bugs

Sintático

O código fonte não respeita a sintaxe da linguagem

```
>>> 1+2)
```

Semântico

Aparentemente executa bem mas não produz os resultados corretos!

Mais difícil de encontrar onde está o erro...

Runtime

Manifestam-se apenas durante a execução e sob circunstâncias especiais

Indicam que algo excecional (e normalmente mau) aconteceu!

Python

Python

https://www.python.org/

Características

- Alto nível
- Interpretada*
- Sintaxe simples
- Fácil aprendizagem
- Inclui mecanismos para cálculo científico
- Existente na maioria dos sistemas operativos
- É livre
- Utilizada com sucesso
  - https://www.python.org/about/success/

Python

Interpretação híbrida

compilador traduz Python para um código intermédio “byte-code”

extensão `.py` → extensão `.pyc`

execução feita por um interpretador de “byte-code”

Vantagens

Desenvolvimento rápido

Mais eficiente que um interpretador clássico

Desvantagens

Menos eficiente que linguagens compiladas (por ex. C)

Python

Comando `python`

Interativo

Executa uma instrução de cada vez e mostra o resultado

Batch

Executa todos os comandos existentes num ficheiro

Versões

Python 2

```
print ‘Hello World!’
```

Python 3

```
print( ‘Hello World!’ )
```