Linguagens Script

<Introdução>

Licenciatura em Engenharia Informática > LEI-PL > LEI-CE

Departamento de Engenharia Informática e de Sistemas

Cristiana Areias < cris@isec.pt >

Tipos de Linguagens

- Linguagens de Programação
  - Interpretação
  - Compilação
- Linguagens Script
  - Características

Linguagens de Programação

- Permitem escrever um programa para que um computador o possa executar
- Tipo de Linguagem
  - Linguagem Máquina
  - Linguagem *Assembly*

Linguagem Máquina

```
        mov ax, dseg
        mov ds, ax
        xor ax,ax
        mov cx,0
ciclo:  add ax,cx
        inc cx
        cmp cx,10
        jb ciclo
        mov al,0
        mov ah, 4ch
        int 21h
```

Linguagem Assemly 8086

- **Linguagens de Alto Nível**

- Aproximam-se mais da linguagem humana;
- Permitem reduzir o tempo despendido na implementação do código, detecção de erros, manutenção de código;
- **Reduzem Custos de Produção**

```
int main() {
  int i, soma=0;
  for (i = 0; i < 10; i++) {
     soma+=i;
     printf("i=%d - Soma=%d\n", i,soma);
  }
  printf("Soma Total = %d\n", soma);
  return 0;
}
```

Linguagem Alto Nível
Ex. ***Programação em C***

Linguagens Script

Linguagens de Programação

- *Code fonte (Source Code) é o* código que o programador escreve

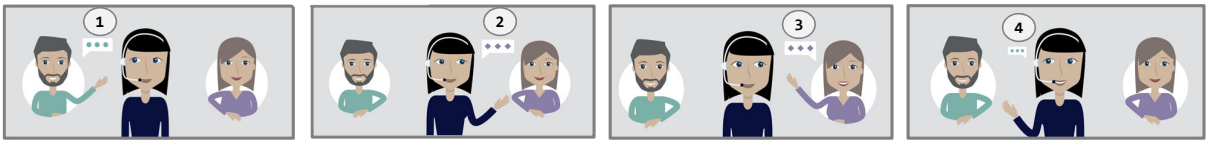
```
int main() {
  int i, soma=0;
  for (i = 0; i < 10; i++) {
     soma+=i;
     printf("i=%d - Soma=%d\n", i,soma);
  }
  printf("Soma Total = %d\n", soma);
  return 0;
}
```

Computadores não compreendem nem linguagens de alto nível, nem linguagem assembly. Apenas linguagem máquina! Logo, existe a necessidade conversão **destas linguagens para código maquina**

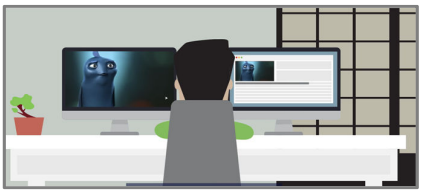
```
        mov ax, dseg
        mov ds, ax
        xor ax,ax
        mov cx,0
ciclo:  add ax,cx
        inc cx
        cmp cx,10
        jb ciclo
        mov al,0
        mov ah, 4ch
        int 21h
```

Linguagem de Programação

- Existem **duas técnicas principais** de como a tradução possa ser efetuada:
  - *Interpretação*
    - Intérprete  ?



  - Compilação
    - Tradução de um livro ?



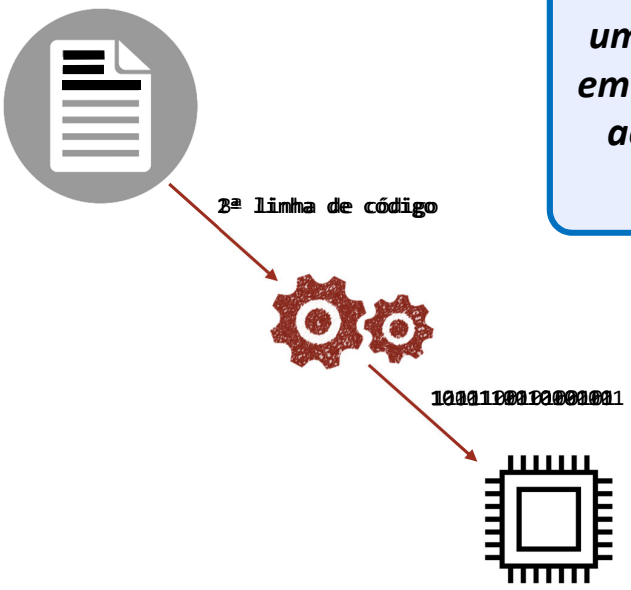
- Um compilador ou interpretador é um programa que permite efetuar a conversão de codigo fonte para codigo objeto.

https://player.vimeo.com/video/418207914?h=a6fa28cffd

Interpretação

Linguagem Interpretada

Um interpretador traduz uma linha do código fonte em código máquina e envia ao processador para ser executada



Compilação

Linguagem Compilada

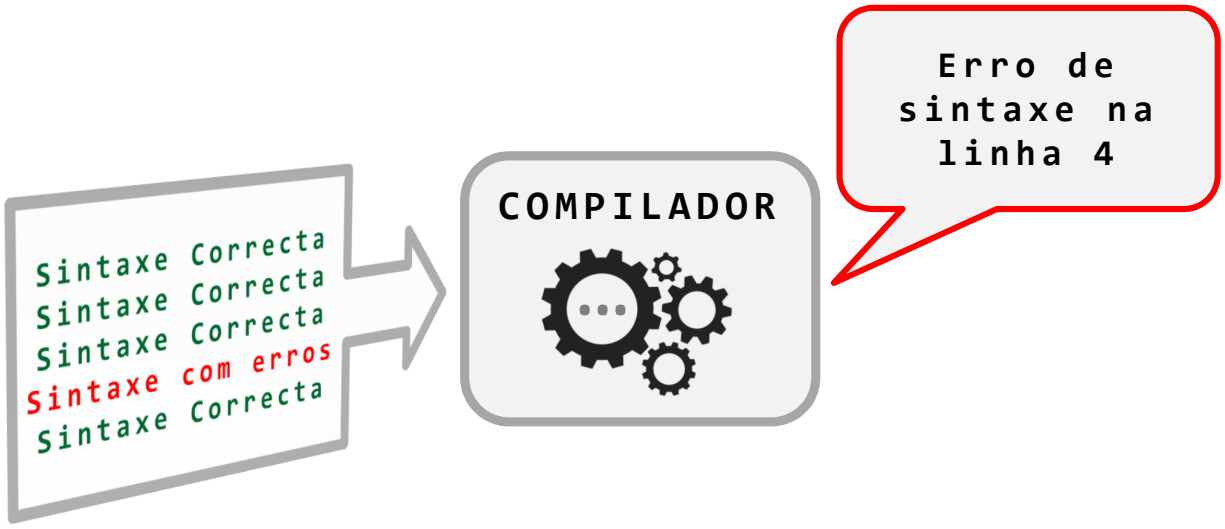
Um compilador traduz todo o código fonte e armazena o resultado do código máquina num ficheiro.

Compilador vs Interpretador

- O **interpretador** irá traduzir uma linha de cada vez e executá-la até encontrar um erro de sintaxe.

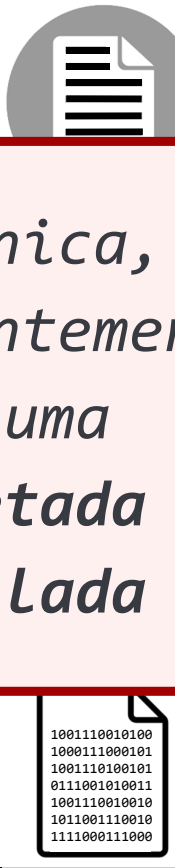
Compilador vs Interpretador

- O **compilador** não irá produzir qualquer *output* até que não existam erros de sintaxe.



Tipos de Linguagens

Dependendo da técnica, a linguagem é frequentemente referida como uma ***linguagem interpretada*** *ou* ***linguagem compilada***



Tipos de Linguagens

- Linguagens Compiladas
  - Linguagem de programação no qual o código fonte normalmente é compilado para linguagem máquina antes da sua execução.
    - C, C++,Rust, Java*, C#*…
  - Precisam de conter todas as informações para decidir sobre o fluxo de instruções de uma vez por todas.
- Linguagens Interpretadas
  - Linguagem de programação no qual o código fonte é lido e executado diretamente, linha por linha, sem compilação prévia para obter as instruções em linguagem máquina.
    - JavaScript, Phyton, PHP, Ruby, …

Tipos de Linguagens

- Existem, também, um grupo de linguagens que usam ambas as técnicas.
  - Aquando compilação do código fonte, não compilam diretamente em código máquina.
    - Existe um passo intermédio no qual compilam o código fonte em byte code, permitindo que este possa ser distribuído dessa forma para quem quiser executar o programa.
    - Posteriormente um interpretador, habitualmente, interpreta o *byte code* em código máquina de acordo com o corrente sistema que está a ser executado
    - Java, C# (.NET)
    - Recorre também ao *just-in-time* (JIT) *compilation*
- A execução de código em uma linguagem de programação é mais rápida, pois o código está disponível em um formato compreensível por máquina quando o programa é executado.

Paradigmas de Programação

- Um **paradigma de programação** refere-se a um estilo ou "forma" de programar.
- Existem vários paradigmas de programação, entre eles:

Funcional
(Paradigma Declarativo)
SQL, Haskell, Lisp,...

Procedimental
(Paradigma Imperativo)
C

Orientada a Objectos
(Paradigma Imperativo)
C++*, Java*

Lógica
(Paradigma Declarativo)
Prolog

Os paradigmas não foram feitos para ser mutuamente exclusivos; um único programa pode apresentar vários paradigmas!

Linguagens de Programação de Script

Paradigmas de Programação


```
const numeros = [1, 5, 9, 4, 10,2]
const resultado = []

for (let i = 0; i < numeros.length; i++) {
  if (numeros[i] > 5) resultado.push(nums[i])
}

console.log(resultado)
```

O fluxo de controle na programação imperativa é explícito.

```
const numeros = [1, 5, 9, 4, 10, 2]
function filtraNumeros() {
  const resultado = []
  for (let i = 0; i < numeros.length; i++) {
    if (numeros[i] > 5)
          resultado.push(numeros[i])
  }
  return resultado
}
console.log(filtraNumeros())
```

O fluxo de controle na programação declarativa é implícito.

```
select nome
from alunos
where numero = 202212345
```

Paradigmas de Programação

```
const numeros = [1, 5, 9, 4, 10, 2]
console.log(numeros.filter(num => num > 5))
```

Funções como filter, map, reduce, sort dos JavaScript são bons exemplos de código declarativo!

JSX usado pelo React, é declarativo

Linguagens Script

Paradigmas de Programação

As linguagens de programação nem sempre estão vinculadas a um paradigma específico. Existem linguagens que foram construídas com um certo paradigma em mente e possuem funcionalidades que facilitam esse tipo de programação mais do que outras.

Existem ainda linguagens "multi-paradigm" permitindo adaptar o código a um determinado paradigma ou outro – Ex. ***JavaScript***

Linguagens de Script

interpretada ou compilada?

Linguagens Script > Introdução

- O código-fonte ou bytecode de uma linguagem de script é frequentemente interpretado
  - *interpreter-based*;
- Habitualmente recorrem a uma codificação mais simplificada;
- Muito Populares no contexto web;
- Sistemas operativos usam linguagens script para executar algumas tarefas básicas, gerar e automatizar tarefas, ficheiros de inicialização,….
- Podem ser aplicados em diferentes domínios
  - Automatização de determinadas tarefas em programas grande dimensão, como chamadas API
  - Usados para *server-side scripting*, como exemplo recorrendo ao PHP, Python, Perl, etc.
  - Usados como *client-side scripting* - JavaScript, etc.
  - Usados na administração de sistemas recorrendo ao Perl, Python, etc.
- Todas as linguagens de script são linguagens de programação, mas o inverso nem sempre é verdadeiro.

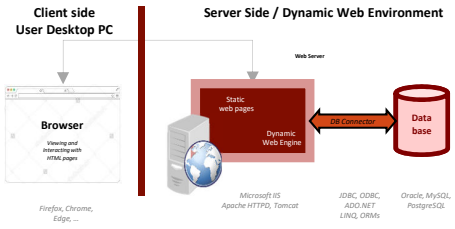
Linguagens Script> Exemplos

- JavaScript/ECMAScript.
- Python
- PHP
- Ruby
- Groovy
- Perl
- Lua
- Bash

https://www.atatus.com/blog/scripting-languages-that-you-should-learn-to-improve-yourself/

Linguagens Script

- Dependendo de onde o script é executado, as linguagens de script podem ser divididas em:
  - Linguagens de script do lado do servidor
    - Os scripts escritos nessas linguagens são executados no servidor.
      - Alguns exemplos de linguagens de script do lado do servidor são Perl, Python, PHP, JavaScript etc.
  - Linguagens de script do lado do cliente
    - Os scripts escritos nessas linguagens são executados no navegador do cliente.
      - Alguns exemplos de linguagens de script do lado do cliente são Javascript, VBScript, etc.



Qual a diferença principal entre **Linguagens Script** e **Linguagens de Programação?**

LS > Benefícios...

- Facilidade de uso
  - As linguagens de script são geralmente fáceis de aprender e usar. Não é necessário muito esforço ou tempo para dominar uma linguagem de script e usar a mesma.
- Área de uso
  - As áreas de uso de uma linguagem de script são bastante vastas e podem ser usadas como uma linguagem de domínio específico para uma linguagem de programação de uso geral.
- Sem Compilação
  - Estas linguagens geralmente não requerem que o programa seja compilado antes do tempo de execução;
- Portabilidade
  - podem ser usados facilmente em vários sistemas operativos

isec Engenharia

LS > Algumas Características

- Falta de declarações
- Regras de *scoping* simples
- *Tipagem dinâmica*
  - Não exigem declarações de tipos de dados
  - Capacidade da linguagem de programação em **escolher dinamicamente o tipo de dado de acordo com o valor atribuído à variável,** podendo alterá-lo durante a compilação ou a execução do programa.
  - O JavaScript consegue **converter o tipo dos dados** armazenados na variável em **tempo de execução**, oferecendo um **dinamismo** não encontrado em linguagens **estáticas de tipagem forte**

A ver nas próximas aulas…

LS > Algumas Características

- *Tipagem fraca*
  - Relacionado com a característica da linguagem em realizar conversões de forma automática entre tipos diferentes de dados

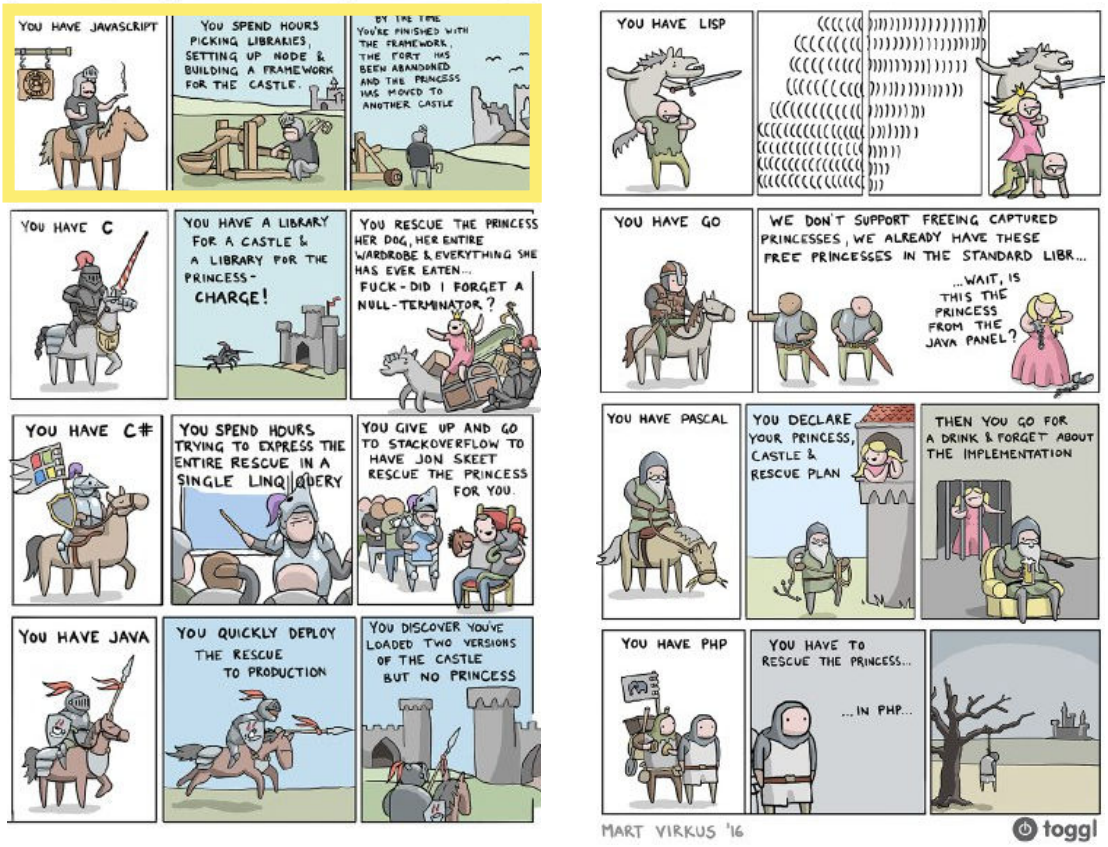
JavaScript

```
texto = "Isto é uma string...";
numero = 1;
console.log(texto);
console.log(numero);
console.log(texto + " " + numero);
```

Python

```
texto = "Isto é uma string...";
numero = 1;
print(texto);
printf(numero);
printf(texto + " " + numero);
```

- Muitas linguagens de programação não se enquadram exatamente entre **tipagem estática ou dinâmica** e **tipagem forte ou fraca**. Cada uma destas abordagens possuem vantagens e desvantagens. As linguagens mais modernas tendem a incorporar alguns aspetos de cada uma.



Linguagens Script

</Introdução>