

Laboratório de Fundamentos em TIC

Apresentação da Linguagem C

Prof. Gabriel Resende Machado



gabrielmachado@unifeso.edu.br



<https://www.linkedin.com/in/machadogabriel>



<https://github.com/UNIFESO-Gabriel/fundamentos-em-tic>

Conceitos de Linguagem de Programação

- **Comunicação com Computadores:**

- Elas permitem que programadores escrevam código em um formato legível para humanos, que é então traduzido em instruções legíveis para máquinas por compiladores ou interpretadores.

- **Abstração e Resolução de Problemas:**

- Linguagens de programação fornecem camadas de abstração que permitem aos desenvolvedores se concentrarem em resolver problemas em níveis mais altos de complexidade sem precisar focar em detalhes intrínsecos de *hardware* ou de operações de baixo nível.

- **Desenvolvimento de Software:**

- Linguagens de programação desempenham um papel crucial no desenvolvimento de *software*, permitindo a criação de uma ampla gama de aplicativos, desde *scripts* simples até sistemas e aplicativos complexos;
- Linguagens de programação também promovem a colaboração entre desenvolvedores, fornecendo sintaxe padronizada, bibliotecas e estruturas, que simplificam o processo de desenvolvimento e promovem a reutilização e manutenção de código.

Conceitos de Linguagem de Programação

Linguagens de Alto e Baixo Nível

<pre>Var1 = 0.5 if var1 > 1.3 or var1 < 0.9:</pre>	<pre>10100101010010101010101 01010101010010101010100</pre>
Linguagem de Alto Nível	Linguagem de Baixo Nível
Fácil para humanos lerem, escrevem e modificarem	Difícil para humanos lerem, escreverem e modificarem
Programas executam mais lentamente já que fazem pior uso da CPU e memória.	Controle direto sobre a CPU resulta em um uso de memória mais eficiente e, conseqüentemente, os programas executam mais rápido.
Necessário pouco conhecimento em como os componentes do hardware funcionam.	Necessário conhecimento avançado em como diferentes componentes de hardware funcionam.
Exemplos são Python, Java, C, C++, C#, etc.	Exemplos são as linguagens Assembly e de Máquina.

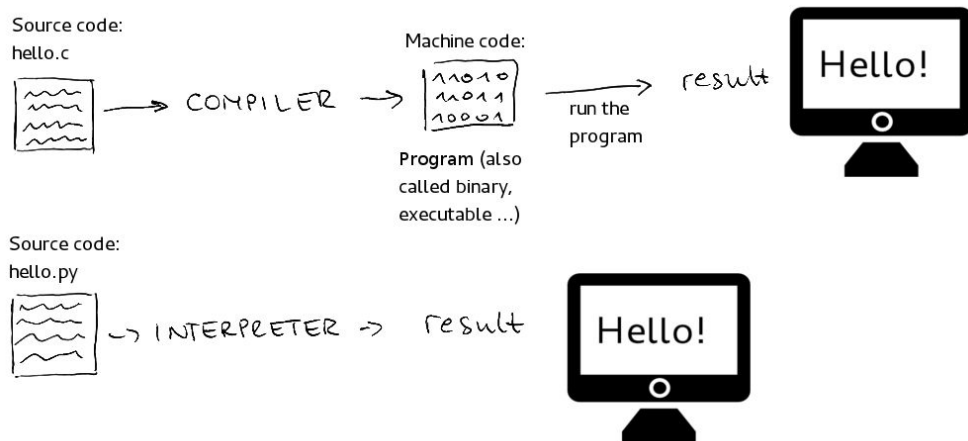
Linguagens Compiladas e Interpretadas

- **Linguagens Compiladas:**

- O código-fonte, a partir de um compilador, é traduzido para um código binário atrelado à arquitetura do Sistema Operacional;

- **Linguagens Interpretadas:**

- O código-fonte é executado linha por linha, diretamente por um interpretador, sem a geração de um código binário atrelado ao Sistema Operacional.



Linguagens Compiladas e Interpretadas

Compiladas		Interpretadas	
Vantagens	Desvantagens	Vantagens	Desvantagens
Pronta para execução após compilação	Não é multiplataforma	Multiplataforma	Necessita do interpretador
Execução mais rápida	Tempo de compilação pode ser demorado	Execução linha por linha	Execução mais lenta
Código-fonte pode ser omitido	Mais tendenciosas a erros	Menos tendenciosas a erros	Necessário a disponibilização do código-fonte

História da Linguagem C

- **Evolução do C a partir do BCPL e B:**

- O C originou-se de duas linguagens precursoras, BCPL e B, com o BCPL sendo desenvolvido por Martin Richards em 1967 para escrever sistemas operacionais e compiladores;
- Ken Thompson desenvolveu B, fortemente influenciado pelo BCPL, e o utilizou para criar as primeiras versões do sistema operacional UNIX em 1970;
- O C, evoluído a partir do B por Dennis Ritchie nos Laboratórios Bell em 1972, manteve conceitos-chave do BCPL e B, enquanto introduzia tipagem de dados e outros recursos poderosos.

- **Popularidade e Ubiquidade do C:**

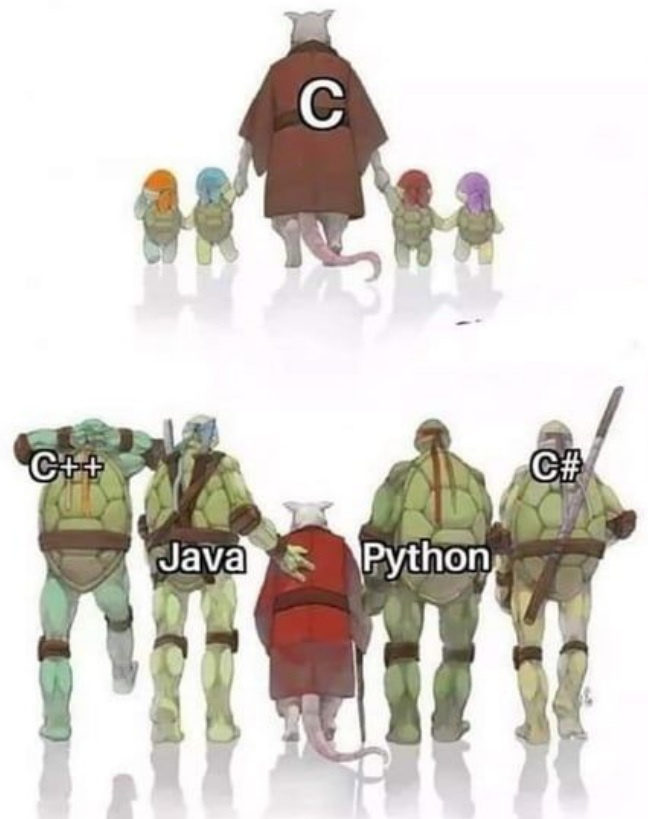
- Inicialmente conhecida como a linguagem de desenvolvimento para UNIX, o C ganhou ampla adoção e é agora utilizado na criação da maioria dos principais sistemas operacionais.

- **Padronização do C:**

- A expansão do C em várias plataformas de *hardware* levou a problemas de compatibilidade, necessitando de uma versão padrão da linguagem. O comitê técnico X3J11 foi formado em 1983 para fornecer uma definição independente de máquina do C, levando à aprovação do padrão em 1989 (atualizado em 1999), conhecido como INCITS/ISO/IEC 9899-1999.

Por que aprender C?

- C é considerado **“O Pai de Todas as Linguagens”**;
- Linguagem procedural, formada por sintaxe intuitiva e de fácil aprendizado;
- Ideal para iniciantes em programação, pois incentiva a adoção de boas práticas e proporciona maior conhecimento para a devida manipulação dos componentes do computador;
- Linguagem extremamente popular e eficiente, utilizada na implementação de sistemas operacionais, compiladores, sistemas de bancos de dados, entre outros.



O “Hello World” em C

```
// Pré-processador
#include <stdio.h>

int main() {
    // imprime no terminal a mensagem "Olá, Mundo!"
    printf("Olá, Mundo!");
    return 0;
}
```

Pré-processador

Comentários

main() - Começar aqui

Chaves definem blocos de código

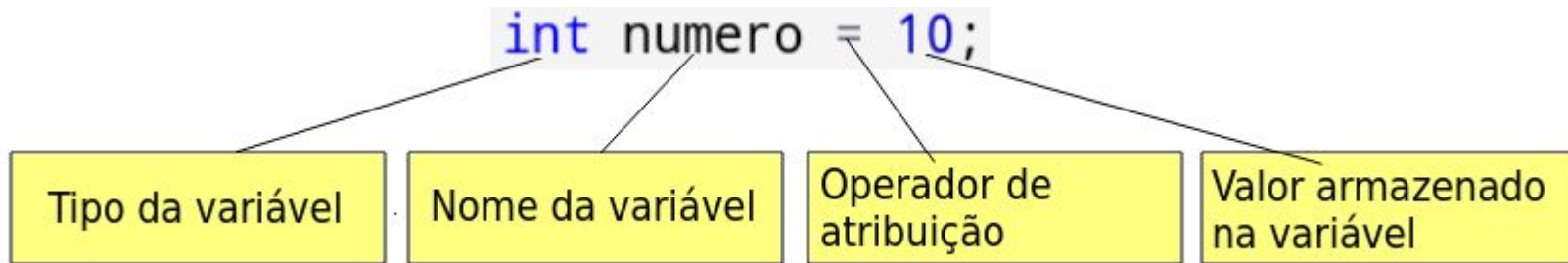
Função para exibição

Indica execução realizada sem erros

Executando o 1º Programa em C

1. Verifique se o compilador do C está instalado, digitando **gcc --version** no terminal;
2. Crie uma pasta vazia chamada **helloworld** para maior organização;
3. Dentro da pasta, crie um arquivo chamado `helloworld.c`;
4. Abra-o com um editor de texto e digite o programa do slide anterior;
5. Acesse o terminal dentro da pasta e digite **gcc -o helloworld helloworld.c**;
6. Para executar o programa compilado, digite **./helloworld**.

Variáveis em C



- Há variáveis especiais que não podem ter seus valores modificados, as constantes (**const**);
- **int** declara o tipo da variável: inteiro;
 - Há outros tipos, como **float**, **double**, **char**, **bool**, **short**, **long**, **unsigned**, etc.
- **int** também é conhecida como **palavra reservada**. Há 32 palavras reservadas em C (<https://shorturl.at/fIBLT>).
- Nomes de variáveis não podem ter (i) espaços, (ii) começarem com números e (iii) pertencerem ao conjunto das palavras reservadas;

Subrotinas em C

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

```
int somar_numeros(int num1, int num2) {  
    int soma = num1 + num2;  
    return soma;  
}
```

```
int main() {  
    // chama a subrotina 'somar_numeros' e retorna o valor  
    // para a variável 'resultado'.  
    int resultado = somar_numeros(10,20);  
    printf("%d", resultado);  
}
```

Tipo retornado

Nome da subrotina

Parâmetros

Variável retornada

Subrotina invocada

Argumentos

- Subrotinas podem ser conhecidas também como (i) procedimentos, (ii) funções ou (iii) métodos;
- Quando uma subrotina não retorna um valor em C, seu tipo retornado é definido como **void**;
- Argumentos podem ser passados **por valor** ou **por referência**.

Praticando em C

- Refazer todas as listas de exercícios da Aula 02 em linguagem C;
- Resolver problemas do portal *Beecrowd*: <https://judge.beecrowd.com>.



Laboratório de Fundamentos em TIC

Apresentação da Linguagem C

Prof. Gabriel Resende Machado



gabrielmachado@unifeso.edu.br



<https://www.linkedin.com/in/machadogabriel>



<https://github.com/UNIFESO-Gabriel/fundamentos-em-tic>