



INTRODUÇÃO À PROGRAMAÇÃO

Aula 4.2 - Apresentação da Linguagem C

Prof. Carlos Alexandre Siqueira da Silva



Campus de Alegre



Histórico: Linguagem C

- Criada na década de 70 por Dennis Ritchie.
- Baseada na linguagem "B", que por sua vez é uma versão reduzida da linguagem "BCPL".
- Linguagem estruturada, compilada e de propósito geral.
- Influenciou diversas linguagens, como: C++, Java, PHP, C#.

Por que estudar a Linguagem C?

- Importância Histórica:
 - Base de linguagens modernas (C++, Java, PHP, C#).
 - Marco na evolução da programação estruturada.
- Clareza dos Fundamentos:
 - Declaração de variáveis e tipos de dados.
 - Estruturas de controle (sequência, condição, repetição).
 - Organização lógica explícita.
- Desempenho e Eficiência:
 - Linguagem próxima ao hardware.
 - Contato direto com memória.
 - Noção real de como o computador executa programas.
- Preparação para o Futuro:
 - Base sólida para aprender outras linguagens.
 - Facilita transição para alto nível.
 - Pensamento lógico fortalecido.

Tipos de Variáveis e seus Formatos

Na linguagem C, foram definidas as palavras reservadas da tabela abaixo para representar cada tipo de variável:

Tipo	Pseudocódigo	Linguagem C	Formato
Numérico inteiro	Inteiro	int	%d
Numérico real	Real	float	%f
Caractere	Caractere	char	%c
Texto	Cadeia	char	%s

Observação: Os **Formatos** apresentados na tabela acima serão utilizados nos comandos de Entrada e Saída de Dados.

Variáveis

- A utilização das variáveis na linguagem C segue as mesmas regras do pseudocódigo, sendo utilizadas as mesmas operações de **Declaração**, **Inicialização** e **Atribuição**.
- Sintaxe da Declaração: <tipo da variavel> <Nome da Variavel>;
 - int Idade;
 - float Desconto;
- Sintaxe da Inicialização: <tipo da variavel> <Nome da Variavel> = <Valor>;
 - int Idade = 18;
 - float Desconto = 10.50;
- Sintaxe da Atribuição: <Nome da Variavel> = <Valor>;
 - Idade = 20;
 - Desconto = 9.99;

Comando de Entrada de Dados

Para ler os dados digitados pelo usuário, na linguagem C usamos o comando **scanf**.

- Sintaxe: `scanf("Formatos", variáveis);`
- Para a leitura de variáveis no comando `scanf`, é necessário especificar o formato utilizando o caracter `"%"` e utilizar o caracter `"&"` nas variáveis.
- Exemplos:
 - `scanf("%d", &Idade);`
 - Atribuindo um valor digitado a uma variável numérica inteira **Idade**
 - `scanf("%f", &Nota1);`
 - Atribuindo um valor digitado a uma variável numérica real **Nota1**
 - `scanf("%d %d", &Valor1, &Valor2);`
 - Atribuindo dois valores digitados às variáveis inteiras **Valor1** e **Valor2**

Comando de Saída de Dados

Para exibir os dados para o usuário na tela, na linguagem C usamos o comando **printf**.

- Sintaxe: `printf("Texto e formatos", variavel1, variavel2, ...);`
- Para a exibição de variáveis no comando printf, é necessário especificar o formato utilizando o caracter "%".
- Exemplos:
 - `printf("Olá mundo!");`
 - Exibindo apenas texto, sem variáveis
 - `printf("A nota de Introdução à Programação é %f", Nota);`
 - Exibindo uma variável numérica real **Nota**
 - `printf("A idade do aluno %s é %d", Nome, Idade);`
 - Exibindo duas variáveis: **Nome** é textual e **Idade** é numérica inteira

Estrutura Básica de um Programa em C

Assim como no pseudocódigo, um programa na linguagem C tem a estrutura básica apresentada abaixo. Do lado direito, temos um exemplo de código na linguagem C.

1. Início do Programa.
2. Declaração de Variáveis.
3. Entrada de Dados.
4. Processamento.
5. Saída de Dados.
6. Fim do Programa.

```
int main() {  
    int DiaHoje, Amanha;  
    scanf("%d", &DiaHoje);  
    Amanha = DiaHoje + 1;  
    printf("Amanhã é dia %d", Amanha);  
    return 0;  
}
```

Observação: Nas próximas aulas, vamos ver com detalhes cada parte do código acima.

Ferramentas para Programação C

- É necessário apenas um editor de texto e um compilador da linguagem C, mas um ambiente integrado auxilia bastante, simplificando o trabalho.
- Existem diversos IDEs (Ambiente de Desenvolvimento Integrado):
 - Code::Blocks
 - VS Code
 - Online GDB

O que vem por aí...

Nas próximas aulas, vamos aprofundar o nosso conhecimento da Linguagem C através de atividades práticas e exercícios.