



Minicurso de C



Ramo Estudantil IEEE UEL

Minicurso de C

Módulo 2 - Variáveis, identificadores e tipos Comandos de Entrada e Saída

Fernando Morgado Pires Neto
Renan Ricoldi Fróis Pedro





Ensinar os comandos e paradigmas mais simples da linguagem C

O que são variáveis?



- Variáveis são endereços de memória que guardam valores e são mutáveis.

Declaração de uma variável



- Para declararmos uma variável primeiro dizemos o tipo dela e depois seu identificador. Ex: `tipo identificador;`

```
variavel.c

#include <stdio.h>

int main(){

    int idade = 19;

    return 0;

}
```



- O identificador das variáveis pode ser qualquer palavra que não seja uma palavra chave da linguagem.
- É possível conter um número na palavra: Casa1
- Não é aceitável iniciar com um número: 1casa (errado)
- É possível utilizar underline: Casa_da_ana
- Não pode utilizar: { (+ - * / ; . , ?
- Boas práticas



- Os tipos definem quais valores aquela variável pode armazenar.
- Os mais importantes são:
 - char
 - int
 - float
 - double
 - void



- Podemos inicializar um variável de duas formas:
 - `int numero = 10;`
 - `int numero; numero = 10;`
 - Ou então por comandos de entrada.



Para recebermos algo no nosso terminal, utilizamos uma função de entrada.
A principal delas é a `scanf()`;



- Podemos escanear vários tipos de valores, usamos o seguinte padrão:
 - Primeiro declaramos a variável: `int idade;`
 - Depois chamamos a `scanf()`: `scanf("%d", &idade);`
 - Ou seja `scanf("código de leitura", &nome_da_variavel);`



- Os códigos de leitura diferem para cada tipo de variável e são:
 - %c -> caractere
 - %d -> inteiro decimal
 - %f -> float
 - %s -> string
 - %lf -> double
- E seguimos o padrão:
 - `scanf("%d %s,%c", &var_1, var_2, &var_3);`



Para imprimirmos algo no nosso terminal, utilizamos uma função de saída.

A principal delas é a `printf()`;



- Podemos imprimir frases e caracteres diretamente.
 - Ex: `printf("Ola mundo!");`
 - `printf('a');`
- Podemos também imprimir o valor de variáveis.
 - Ex: `int idade = 19; printf("Sua idade e %d anos.", idade);`



- Para imprimirmos variáveis utilizamos alguns códigos de impressão dentro de uma String.
- Os códigos de impressão diferem para cada tipo de variável e são:
 - %c -> caractere
 - %d -> inteiro decimal
 - %f -> float
 - %s -> string
 - %lf -> double
- E seguimos o padrão:
 - `printf("palavras %d %s %c", var_1, var_2, var_3);`



- Ao utilizarmos o %f e o %lf podemos dizer quantas casas decimais queremos depois da vírgula, da forma %.{numero de casas}f:
 - float numero = 1.2; printf("%f", numero); -> resultado: 1.200000
 - float numero = 1.2; printf("%.1f", numero); -> resultado: 1.2



- Podemos utilizar a contra-barra para imprimir caracteres especiais
 - `\n` -> pula uma linha
 - `\t` -> aplica uma tabulação
 - `\"` -> imprime aspas duplas
 - `\\` -> imprime uma barra



```
variavel.c

#include <stdio.h>

int main(){

    float 2num = 1.1;
    double char = 4.2;
    double kilo = 73.231
    int va}r = 0;
    int var = 3.2;
    char palavra = "bom dia"

    return 0;

}
```



```
variavel.c

#include <stdio.h>

int main(){
    x = 1;
    int i = 3; z = 4;

    return 0;
}
```



```
variavel.c

#include <stdio.h>

int main(){
    int i = 32767;
    int j = 32767;
    int x = i + j;

    return 0;
}
```



- Corrija as funções se necessário:
 - `printf(bom dia);`
 - `printf('bom dia');`
 - `printf("\tbom dia\");`
 - `printf("prazer em te conhecer %s! \n", nome_do_usuario);`
 - `printf("o seu peso é de %.2f", peso_do_usuario);`
 - `int idade_do_usuario = 19;`
`printf("você tem %f anos de idade! \n", idade_do_usuario);`



- Corrija as funções se necessário:
 - `char inicial_do_usuario;`
`scanf("%f", &inicial_do_usuario);`
 - `char cidade_atual_do_usuario;`
`scanf("%s", &char);`



- Criar sistema de cadastro que leia, guarde e imprime:
 - Nome
 - Idade
 - Curso
 - Altura em metros