

Linguagem de Programação I - ESP201

Prof^a Ana Carolina Sokolonski

Bacharelado em Sistemas de Informação
Instituto Federal de Ciência e Tecnologia da Bahia
Campus de Feira de Santana

carolsoko@ifba.edu.br

May 27, 2025

Variáveis, Tipos de Dados e Entrada e Saída de Dados

1 Variáveis

- Declaração de Variáveis

2 Tipos de Dados da Linguagem C

- Tipos de Dados Básicos da Linguagem C
- Modificadores de Tipos
- Exemplos de Uso de Tipos

3 Instruções de E/S

4 Exercícios

5 Referências

Variáveis

Variáveis

- **Variável:** “objeto” que pode assumir diversos valores;
- Espaço de memória de um certo tipo de dado associado a um nome para referenciar seu conteúdo

```
1  #include <stdio.h>
2
3  void main ( ){
4      int idade;
5      idade = 30;
6      printf (" A idade é : %d", idade);
7  }
```

Regras para nomenclatura das variáveis

- Quantos caracteres quiser (limite de 32)
- Tem que começar com letras ou sublinhado (*underline*), seguidos de letras, números ou sublinhados

- C é *Case Sensitive*, logo:

peso <> Peso <> pEso

- Uma variável não pode ter o mesmo nome de uma palavra chave... que são as palavras reservadas da linguagem

Lembrando... Palavras Reservadas da Linguagem C

Nomes Reservados (42 comandos)

auto	double	if	static
break	else	int	struct
case	entry	long	switch
char	extern	register	typedef
continue	float	return	union
default	for	sizeof	unsigned
do	goto	short	while

Declaração de Variáveis

- Instrução para reservar uma quantidade de memória para um certo tipo de dado, indicando o **nome** pelo qual a área será referenciada no programa [Schildt e Mayer 1997]
- `>> tipo nome-da-variável; ou`
- `>> tipo nome1, nome2,...,nomeN;`

EX: `char nome;`

`int idade, num;`

Declaração de Variáveis

```
1  #include <stdio.h>
2
3  void main ( ){
4      int idade = 30;
5      printf (" A idade é : %d", idade);
6  }
7
```


Declaração de Variáveis

```
1  #include <stdio.h>
2
3  void main ( ){
4      int idade;
5      printf("Informe a idade:");
6      scanf("%d",&idade);
7      printf ("| A idade é : %d", idade);
8  }
```

Tipos de Dados Básicos da Linguagem C

Tipos de Dados Básicos da Linguagem C

Tipo	Tamanho (em bits)	Intervalo
Char	8	-128 a 127
Int	16	-32768 a 32767
Float	32	3,4E-38 a 3,4E+38
double	64	1,7E-308 a 1,7E+308
void	0	sem valor

Fortemente Tipada

*Como vimos anteriormente, a linguagem C é uma linguagem **FORTEMENTE TIPADA**, ou seja, uma vez que suas variáveis sejam declaradas como um determinado tipo, elas serão deste tipo do início ao fim da execução do programa. A variável se mantém fixa durante toda a execução.*

*A estrutura de tipos da linguagem de programação C é **ESTÁTICA**, ou seja, o tipo das suas variáveis é definido em tempo de compilação e não pode ser alterado depois disso.*

Modificadores de Tipos

- **long:** aumenta o espaço tomado pela variável
- **short:** diminui o espaço tomado pela variável
- **unsigned:** cria a variável sem sinal (apenas positivo)
- **signed:** cria a variável com sinal (padrão para a maioria)

obs: **float** e **char** não podem ter seus tamanhos modificados, assim não pode usar modificadores *long* e *short*

Modificadores de Tipos

Tipo	Tamanho (em bits)	Intervalo
char	8	-128 a 127
unsigned char	8	0 a 255
signed char	8	-128 a 127
int	16	-32768 a 32767
unsigned int	16	0 a 65535
signed int	16	-32768 a 32767
short int	16	-32768 a 32767
unsigned short int	16	0 a 65535
signed short int	16	-32768 a 32767
long int	32	-2147483648 a 2147483647
signed long int	32	-2147483648 a 2147483647
unsigned long int	32	0 a 4294967295
float	32	3,4E-38 a 3,4E+38
double	64	1,7E-308 a 1,7E+308
long double	80	3,4E-4932 a 1,1E+4932

Exemplos de Uso de Tipos

```
9  #include <stdio.h>
10
11 void main( ){
12
13     int soma=10;
14     float money=2.21;
15     char letra= 'A';
16     double pi=2.01E6;
17
18     printf ("valor da soma = %d\n", soma);
19     printf ("Valor de Money = %f\n", money);
20     printf("Valor de Letra = %c\n", letra);
21     printf("Valor de Pi = %e\n", pi);
22
23 }
```

Instruções de E/S

scanf / printf

scanf()

Leitura de dados tipados via teclado

Scanf (“string de controle”, lista de argumentos);

Exemplo:

```
scanf("%d",&idade);
```

OBS: *Para sequências de caracteres (%s), o caracter & não deverá ser usado.*

printf()

Apresentação de dados no monitor

```
printf("string de controle", lista de argumentos);
```

Exemplo:

```
printf ("Digite a sua idade:\n");  
scanf ("%d", &idade);  
printf("Sua idade é: %d", idade);  
|
```

Strings de controle scanf/printf

- `%c` → caracter
- `%d` → inteiro
- `%e` → número ou notação científica
- `%f` → ponto flutuante
- `%o` → octal
- `%x` → hexadecimal
- `%s` → string (cadeia de caracteres)
- `%lf` → double
- `%lu` → endereço

OBS.: Para imprimir o caracter `%` deve-se colocar `%%` para o compilador não confundir com a entrada de um dado externo. **Ex:**
`printf("A empresa cresceu 60%%");`

Exemplo do uso das Strings de controle

Diversos formatos:

```
#include <stdio.h>

void main (){
    printf("Esse é o meu primeiro programa:\n");
    printf("%c é uma letra\n", 'j');
    printf("%d é um número inteiro\n", 30);
    printf("%f é um número de ponto flutuante\n", 12.2);
    printf("%5.2f é um número de ponto flutuante formatado\n", 12.2);
    printf("%s é legal\n", "linguagem C");
    printf("%d + %d = %d\n", 10, 21, 10+21);
}
```

Exemplo do uso das Strings de controle

Decimal, Octal e Hexadecimal

```
#include <stdio.h>
void main ( ){
    char a ;
    printf ("digite um caracter: ");
    scanf ( "%c", &a );
    printf (" \n %c = %d em decimal, |", a, a);
    printf ("%o  em octal, e %x em hexadecimal", a, a);
}
```

Leitura de Strings - scanf

OBS.: Na leitura de sequências de caracteres, quando encontrado um espaço em branco, o **scanf** não lê mais os caracteres restantes, imprimindo apenas os caracteres que se encontram antes do espaço em branco. Como solucionar isto? Vejamos dois exemplos:

```
#include <stdio.h>
void main() {
    char str[20];
    printf("Digite uma string qualquer: ");
    scanf("%s",str);
    printf("A string digitada foi: %s\n", str);
}
```

Leitura de Strings - scanf

OBS.: Na leitura de sequências de caracteres, quando encontrado um espaço em branco, o **scanf** não lê mais os caracteres restantes, imprimindo apenas os caracteres que se encontram antes do espaço em branco. Como solucionar isto? Vejamos dois exemplos:

```
#include <stdio.h>
void main(){
    char str[20];
    printf("Digite uma string qualquer: ");
    scanf("%[A-Z a-z]",str);
    printf("A string digitada foi: %s\n", str);
}
```

Caracteres especiais no scanf

No **scanf** podemos exibir alguns caracteres especiais usando o barra. Vejamos exemplos:

- `\n` nova linha
- `\r` enter
- `\t` tabulação (tab)
- `\b` retrocesso
- `\"` aspas
- `\\` barra

Formatando as saídas usando o scanf

No **scanf** podemos exibir as informações usando formatações específicas. Vejamos exemplos:

```
#include <stdio.h>
void main ( )
{
    printf ("os alunos são %2d \n", 350);
    printf ("os alunos são %4d \n", 350);
    printf ("os alunos são %5d \n", 350);
}
```

```
Saída:   os alunos são 350
         os alunos são  350
         os alunos são   350
```

Formatando as saídas usando o scanf

No **scanf** podemos exibir as informações usando formatações específicas. Vejamos exemplos:

```
#include <stdio.h>
void main ( ){
    printf (" %3.1f \n", 3456.78);
    printf (" %10.3f \n", 3456.78);
}
```

Saída: 3456.8
 3456.780

Instruções de E/S

getchar / putchar

getchar() / putchar()

```
1 #include <stdio.h>
2
3 void main(){
4     int exemplo;
5
6     printf("Digite uma letra qualquer entre a-z: ");
7     exemplo = getchar();
8
9     printf("A letra digitada foi : ");
10    putchar(exemplo);
11 }
```

Exercícios

- 1 Faça um algoritmo em linguagem de programação C que pergunte, leia e mostre na tela o seu nome e idade.
- 2 Faça um algoritmo em linguagem de programação C que pergunte, leia e mostre na tela o nome, endereço, peso, altura, telefone e idade de uma pessoa.

Referências

Referências



SCHILDT, H.; MAYER, R. *C completo e total*. Pearson Universidades, 1997. ISBN 9788534605953. Disponível em: <<https://books.google.com.br/books?id=Pbl0AAAACAAJ>>.