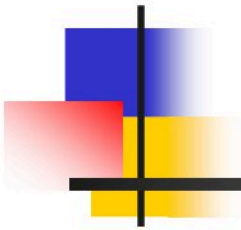


A LINGUAGEM C

Parte 1

© Prof. Engº esp Luiz Antonio Vargas Pinto
www.vargasp.com

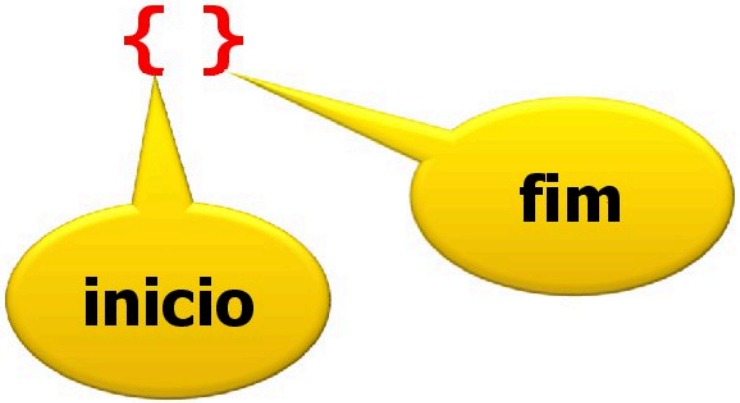
Origem

- Linguagem desenvolvida na Bell Laboratories.
- 1ª versão escrita e implementada por Denis Ritchie.
- 1ª utilização importante = reescrita do SO UNIX (original em Assembly)
- 1º Livro "The C Programming Language", por Denis Ritchie & Brian Kernighan - 1978.
- Diversas versões de C, incompatíveis, foram criadas.
  - Estas versões funcionavam somente com um determinado compilador, restritos a condições especiais.
- 1983, **ANSI** - **A**merican **N**ational **S**tandards **I**nstitute padronizou a linguagem C - ANSI C.

© Prof. Vargas

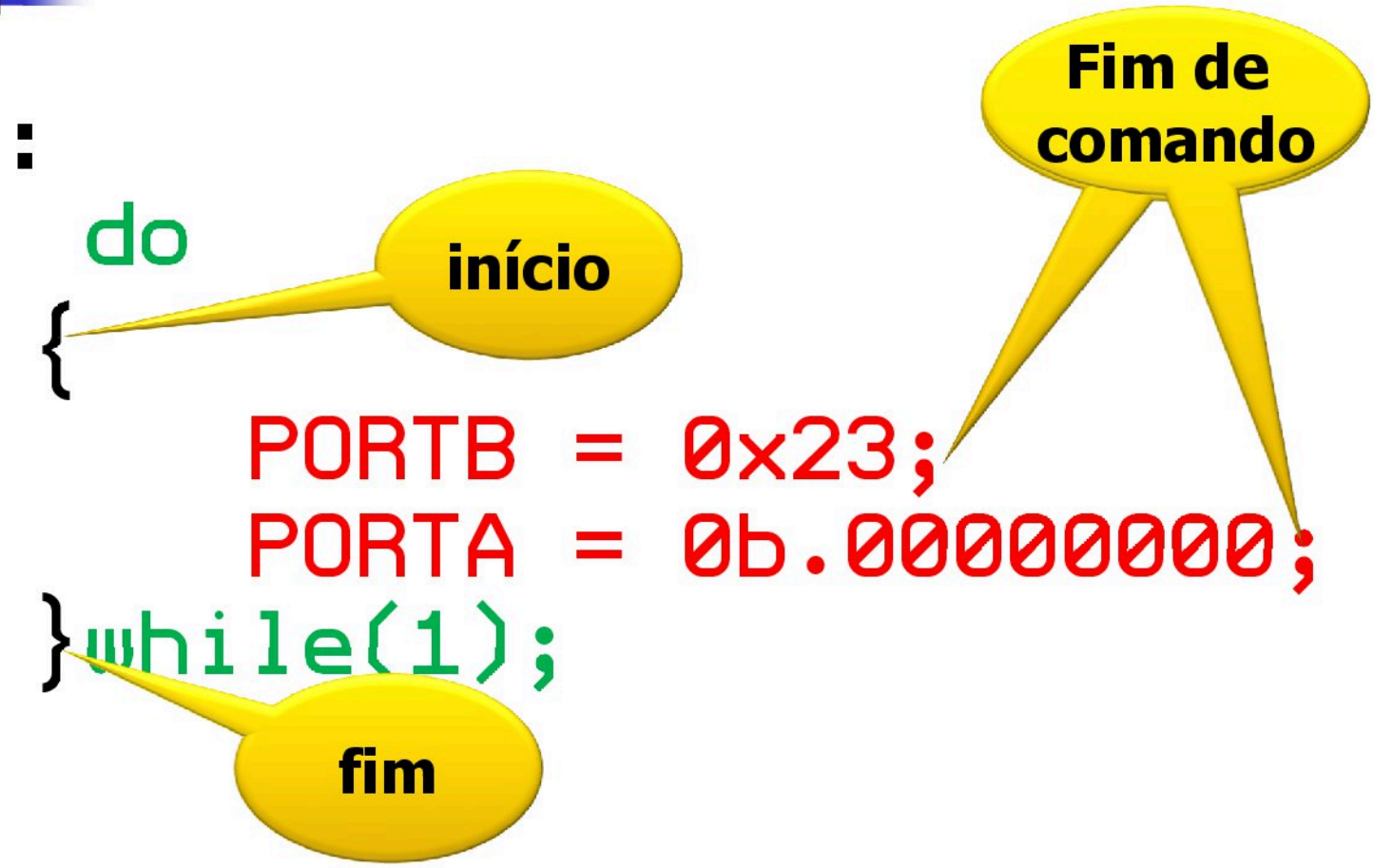
Características

- Case sensitive
- Separa parâmetros por vírgula
- Termina comando com ponto-e-vírgula.
- Blocos de código são delimitados por chaves

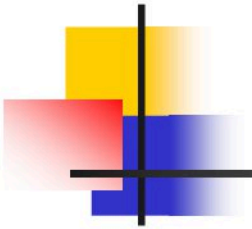


© Prof. Vargas

Exemplo



© Prof. Vargas



Comentários

- /* Comentário
  em mais de
  uma linha */

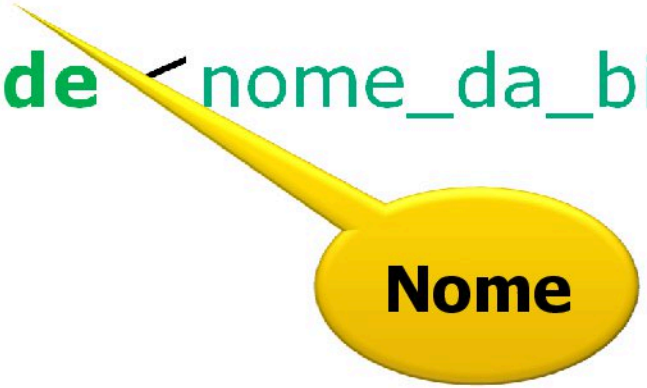
- // Comentário em uma linha

© Prof. Vargas

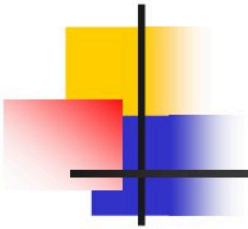
Bibliotecas

- Basicamente contém rotinas (funções) as quais complementam as funções internas
- Tem extensão **.h** de **header**
  - cabeçalho em inglês
  - **stdio.h** **st**and**d**dart **i**nput/**o**utput

#include <nome_da_biblioteca>



© Prof. Vargas



Função main()

- A linha **main()** indica que estamos definindo uma função de nome **main**.
- Todos os programas em C têm que ter uma função **main()**.
- É a primeira função a ser executada em um programa.

© Prof. Vargas

Tipos de Dados

Usuais

Tipo	Bytes	Formato para scanf	Intervalo	
			Início	Fim
	1	%c	-128	127
unsigned char	1	%c	0	255
signed char	1	%c	-128	127
int	2	%i	-32.768	32.767
unsigned int	2	%u	0	65.535
signed int	2	%i	-32.768	32.767
short int	2	%hi	-32.768	32.767
unsigned short int	2	%hu	0	65.535
signed short int	2	%hi	-32.768	32.767
long int	4	%li	-2.147.483.648	2.147.483.647
signed long int	4	%li	-2.147.483.648	2.147.483.647
unsigned long int	4	%lu	0	4.294.967.295
	4	%f	3,4E-38	3.4E+38
double	8	%lf	1,7E-308	1,7E+308
long double	10	%Lf	3,4E-4932	3,4E+4932

© Prof. Vargas

Declaração de Variáveis

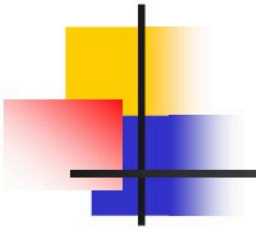
- As variáveis devem ser declaradas antes de serem usadas.
- A forma geral da declaração de variáveis é:

  tipo nome1,nome2;
- Exemplos:
  - **int Dias;**
  - **float x,y;**
  - **char dado[100]; // String, até 99 caracteres**
    - char nome_da_string[tamanho];
    - dado é um vetor de caracteres
- Podemos inicializar variáveis no momento de sua declaração:

  tipo numero = 12;

Quando C cria uma variável *não* atribui valor inicial ficando este *indefinido*. ***Nunca* presuma que uma variável declarada vale zero ou qualquer outro valor**.

© Prof. Vargas



Tipos char e void

- São caracteres/símbolos como a, b, etc..
- Devem vir entre apóstrofes: 'a', 'b', etc..
- **char** também armazena valores numéricos inteiros de um byte (8 bits, isto é, até 255).
- **void** é o tipo vazio, ou um "tipo sem tipo".
  - Aplicação deste "tipo" será vista oportunamente.

© Prof. Vargas

Escopo de Variáveis

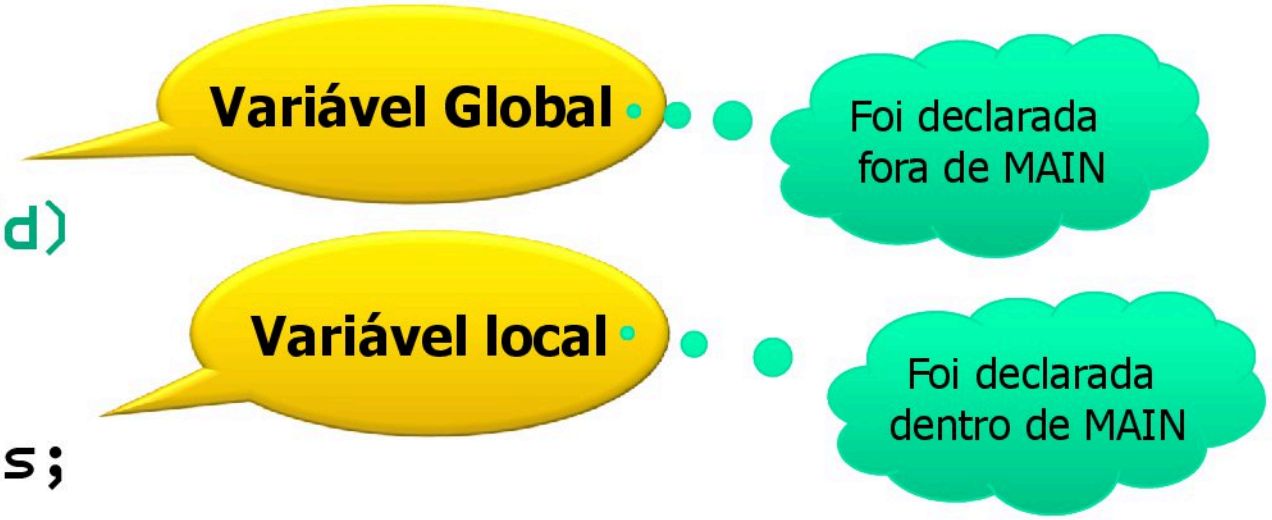
- Há três lugares nos quais podemos declarar variáveis:

1. Fora de todas as funções do programa → variável **global**
   - Todos podem utilizar – é o mesmo endereço de memória
2. No início de um bloco de código → variável **local**
   - só vale no bloco no qual são declaradas
3. Na lista de parâmetros de uma função
   - **Apesar de receberem valores externos, estas variáveis são conhecidas apenas pela função onde são declaradas.**

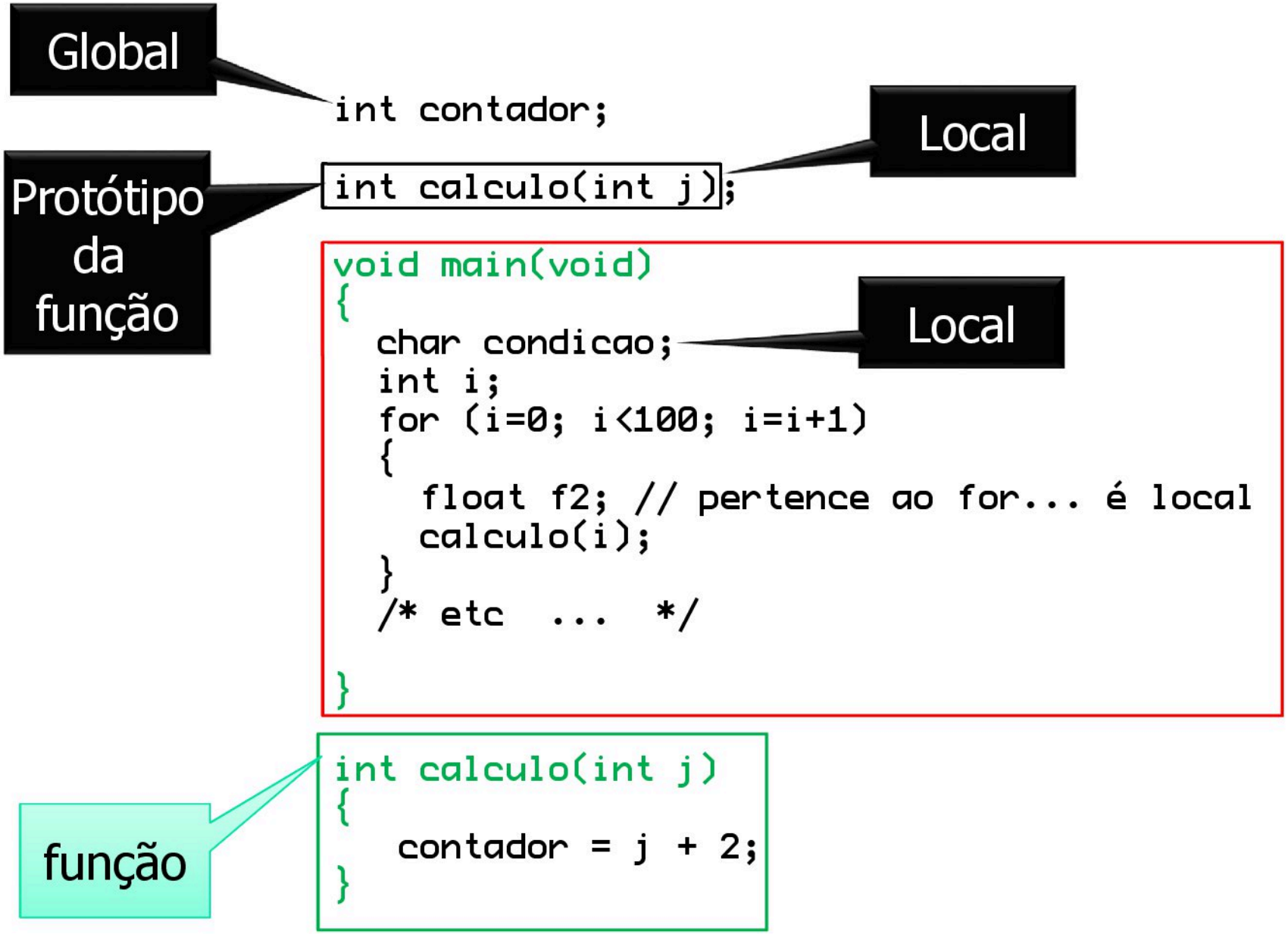
© Prof. Vargas

Exemplo:

```
float numero;
void main(void)
{
     int Dias;
     float Anos;
}
```



© Prof. Vargas


Global
int contador;
Local
Protótipo da função
int calculo(int j);
void main(void)
{
char condicao;
Local
int i;
for (i=0; i<100; i=i+1)
{
float f2; // pertence ao for... é local
calculo(i);
}
/* etc ... */
}
int calculo(int j)
{
contador = j + 2;
}
função

Atribuição

- Ch='D';
- x=23.5;
- Anos=Dias/365.25;
- str[1] = 'a';

© Prof. Vargas

A função if

Forma:

```
if (condição)
{
```

ou:

```
if (condição)
{
    comando1;
}
else
{
    comando2;
}
```

Opção do IF

© Prof. Vargas

Usando a função if

```
void main (void)
{
    int num;
    num = 23;
    if (num>10) RA0 = 1;
    if (num==10)
    {
       RA1 = 1;
    }else
    {
      RA2 = 1;
    }
}
```

© Prof. Vargas

Condicionais

- **==** igual
- **!=** diferente de
- **>** maior que
- **<** menor que
- **>=** maior ou igual
- **<=** menor ou igual.

© Prof. Vargas

Operadores lógicos e aritméticos

Operador	Ação
&&	and
||	or
!	not

Operador	Ação
+	Soma inteira e fracionada
-	Subtração e troca de sinais
*	Multiplicação inteira ou fracionada
/	Divisão inteira e fracionada
%	Resto da divisão
++	Incremento
--	decremento

© Prof. Vargas

Referências

- Material baseado no original da IFET Sul de Minas - Campus Muzambinho