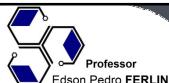


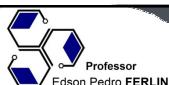
Introdução à Linguagem C

Prof. Edson Pedro Ferlin



Introdução

- Versatilidade e Poder;
- A linguagem C nasceu na década de 70;
- É uma linguagem de programação genérica que é utilizada para a criação de programas;
- O “C” é “Case Sensitive”, ou seja, maiúsculas e minúsculas fazem diferença;
- ANSI C;



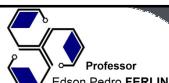
Programa #1

```
# include <stdio.h>
main ()           /* primeiro programa */
{
    printf ("Olá! Turma vamos começar!\n");
}
```

O compilador C desconsidera qualquer coisa que esteja começando com /* e terminando com */;

Os comando em C terminam com ";"

Toda função no C (inclusive a main()) retorna um inteiro.

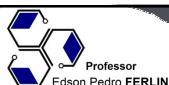


Programa #2

```
# include <stdio.h>
main ()
{
    int Dias;    /* Declaracao de Variaveis */
    float Anos;
    printf ("Entre com o numero de dias: ");      /* Entrada de Dados */
    scanf ("%d", &Dias);
    Anos = Dias/365.25;        /* Conversao Dias ->Anos */
    printf ("\n\n%d dias equivalem a %f anos.\n", Dias, Anos);
}
```

Quando temos mais de um parâmetro para uma função, eles serão separados por vírgula;

No C o "=" é o comando de atribuição.

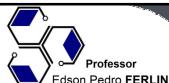


Introdução às Funções

```
# include <stdio.h>
mensagem ()           /* Declaracao da Funcao */
{
    printf ("Ola! ");
}

main ()
{
    mensagem ();      /* Chamada da Funcao */
    printf ("Estou comecando a entender! \n");
}
```

Uma função é um bloco de código de programa que pode ser usado diversas vezes em sua execução.



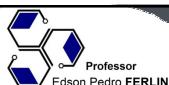
Introdução às Funções

Argumentos (1)

```
# include <stdio.h>
square (int x)          /* Declaracao da Funcao */
{
    printf ("O quadrado eh %d", (x*x));
}

main ()
{
    int num;
    printf ("Entre com um numero: ");
    scanf ("%d",&num);
    printf ("\n\n");
    square (num);        /* Chamada da Funcao */
}
```

Argumentos são as entradas que a função recebe.

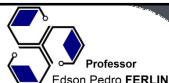


Introdução às Funções

Argumentos (2)

```
# include <stdio.h>
mult (float a, float b, float c)      /* Declaracao da Funcao */
{
    printf ("%f", a*b*c);
}

main ()
{
    float x,y;
    x=23.5;
    y=12.9;
    mult (x,y,3.87);    /* Chamada da Funcao */
}
```



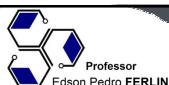
Introdução às Funções

Retornando valores (1)

```
# include <stdio.h>
prod (int x, int y)      /* Declaracao da Funcao */
{
    return (x*y);
}

main ()
{
    int saida;
    saida=prod (12, 7); /* Chamada da Funcao */
    printf ("A saida eh: %d \n", saida);
}
```

Utiliza-se a palavra reservada Return para informar o que será retornado.

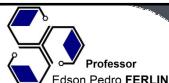


Introdução às Funções

Retornando valores (2)

```
# include <stdio.h>
float prod (float x, float y)          /* Declaracao da Funcao */
{
    return (x*y);
}

main ()
{
    float saida;
    saida=prod (45.2, 0.0067); /* Chamada da Funcao */
    printf ("A saida eh: %f \n", saida);
}
```



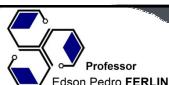
Introdução às Funções

Forma Geral

Tipo_de_retorno Nome_da_funcao (Lista_de_argumentos)

```
{
    Código_da_funcao;
```

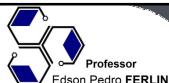
```
float prod (float x, float y)          /* Exemplo */
{
    return (x*y);
}
```



Introdução às Entradas e Saídas Caracteres (1)

```
# include <stdio.h>
main ()
{
    char Ch;
    Ch='D'; /* Inicializacao da Variavel */
    printf ("%c", Ch);
}
```

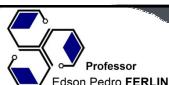
- ▶ Os caracteres são um tipo de dado:
o Char;
- ▶ O C trata os caracteres como
sendo variáveis de um byte (8 bits).



Introdução às Entradas e Saídas Caracteres (2)

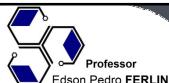
```
# include <stdio.h>      /* Leitura Caracter #1 */
# include <conio.h>
main ()
{
    char Ch;
    Ch=getch ();           /* Usando getch */
    printf ("Voce pressionou a tecla %c", Ch);
}
```

```
# include <stdio.h>      /* Leitura Caracter #2 */
main ()
{
    char Ch;
    scanf ("%c", &Ch); /* Usando scanf */
    printf ("Voce pressionou a tecla %c", Ch);
}
```



Introdução às Entradas e Saídas Strings (1)

- ▶ No C uma string é um vetor de caracteres terminado com um caractere nulo (\0);
- ▶ Char Nome_da_string[Tamanho_da_string];
- ▶ No mínimo uma caracter maior que a maior string;
- ▶ String com 7 posições: T-E-S-T-E-\0-?;
- ▶ O C não inicializa variáveis.



Introdução às Entradas e Saídas Strings (1)

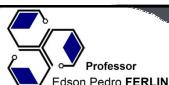
```
# include <stdio.h>
main ()
{
    char string [100];           /* Declaração da string */
    printf ("Digite uma string: ");
    gets (string);             /* Leitura da string */
    printf ("\n\nVoce digitou %s",string);
}
```

➤ Para acessar um determinado caractere de uma string, basta indexarmos, ou seja, usarmos um índice;

➤ str[1]='a'; /* segunda letra de str */

➤ Índice sempre começa em zero;

➤ Char str[10] = "Teste";



Introdução às Entradas e Saídas

Função printf (String_de_controle, Lista_de_argumentos)

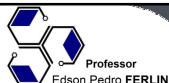
```
printf ("Teste #1: %%");
      → Teste #1: %
```

```
printf ("Teste #2: %f", 3.1415);
      → Teste #2: 3.1415
```

```
printf ("Teste #3: Um caracter %c e um inteiro %d", 'D', 120);
      → Teste #3: Um caracter D e um inteiro 120
```

```
printf ("Teste #4: %s uma string","Mostra");
      → Teste #4: Mostra uma string
```

```
printf ("Teste #5: %s de %d%%, "Juros", 10");
      → Teste #5: Juros de 10%
```



Introdução às Entradas e Saídas

Função scanf (String_de_controle, Lista_de_argumentos)

```
scanf("%d",&inteiro);           /* Leitura de um numero inteiro */
```

```
scanf("%f",&real);             /* Leitura de um numero real */
```

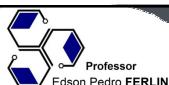
```
scanf("%c",&caracter);         /* Leitura de um caracter*/
```

```
scanf("%s",&string);           /* Leitura de uma string*/
```

```
scanf("%d%f",&inteiro, &real);   /* Leitura de um numero inteiro e um real */
```

Número de argumentos igual ao número de caracteres de controle na string;

Colocar sempre "&" antes das variáveis.

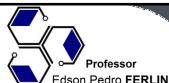


Introdução ao Controle Fluxo

Comando if (condição) declaração;

```
# include <stdio.h>
main ()                                /* Exemplo Comando If */
{
    int num;                            /* Operadores de comparacao: ==, >, <, >=, <= */
    printf ("Digite um numero: ");
    scanf ("%d", &num);
    if (num > 10) printf ("\n\nO numero eh maior que 10");
    if (num == 10)      /* num = 10 estah errado */
    {
        printf ("\n\nVoce acertou!\n");
        printf ("O numero he igual a 10");
    }
    if (num < 10) printf ("\n\nO numero eh menor que 10");
}
```

Se o resultado for diferente de zero a declaração será executada.



Introdução ao Controle Fluxo

Comando for (inicialização; condição; incremento) declaração;

```
# include <stdio.h>
main ()                                /* Exemplo Comando for */
{
    int count;
    for (count=1; count <= 100; count=count+1)
    {
        printf ("%d", count);
    }
}
```

O laço FOR é usado para repetir um comando (bloco) diversas vezes

Palavras Reservadas do C

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

Contato



eferlin@live.com



(BLOG) professorferlin.blogspot.com

(SITE) professorferlin.webnode.com.br

(YOUTUBE) [ProfEdsonPedroFerlin](https://www.youtube.com/ProfEdsonPedroFerlin)