



Programação I

</ Funções />

- Sintaxe
- Funções de retorno



2021

TekMind

Funções

Consiste em **dividir uma tarefa** complexa em **menores**, permitindo esconder detalhes de implementação. Assim **evitando a repetição** de um mesmo código.

Sintaxe:

```
tipo nome (lista de parâmetros){  
    bloco de comandos;  
}
```

- Podem ser do tipo **void**, conhecidas como **função sem retorno**.
- O nome pode ser uma palavra reservada da biblioteca do C, ou um nome dado pelo programador caso esta tenha sido estruturada durante a criação do programa.
- A declaração dos parâmetros/argumentos pode ser do tipo void (não receber argumento), ou receber qualquer outro tipo suportado pelo C.
- Podemos definir a função no início de um programa, antes ou depois da *main()*.
- A declaração da função consiste do cabeçalho da função seguido do ";".
- Toda função deve ser **declarada** antes de ser usada.

→ Funções de retorno (VOID)

Protótipo da função:

Protótipo é uma declaração de uma função que omite o corpo, mas especifica o seu nome, tipos de argumentos e tipos de retorno. Enquanto a **definição da função** especifica o que ela faz. Um protótipo de função pode ser entendido como a **especificação** da sua interface.

Exemplo (sem retorno):

```
#include <stdio.h>  
void led_on(); //protótipo da função ou declaração da função  
void led_on() {  
    printf("\n Led Ligado");  
}  
main() {  
    printf ("\nUsando funções\n");  
    led_on();  
}
```

Exemplo (com retorno):

```
#include <stdio.h>  
int led_on(); //declaração da função, não obrigatório.  
int led_on(){  
    return(1);  
}  
main() {
```

```

int retorno;
printf("\nUsando funções\n");
retorno = led_on();
printf("Led = %d",retorno);
}

```

Passagem de parâmetros:

Serve para **passar dados** para uma função, onde, uma cópia do argumento é passada para a função. O parâmetro se comporta como uma **variável local**.

Exemplo (sem retorno e parâmetro)

```

#include <stdio.h>
int led_on(int);
int led_on(int led){
    printf("Led %d ligado.",led);
}
main() {
    int led;
    printf("\nLed para ligar: (1-13)\n");
    scanf("%d",&led);
    led_on(led); //chamada de função com passagem de parâmetro
}

```

Exemplo (com retorno e parâmetro)

```

#include <stdio.h>
int led_on (int led);
int led_on (int led){
    if ((led>=1) || (led<=13))
        return (1);
    else return(0);
}
main() {
    int led, retorno;
    printf("\nInforme Led (1-13)\n");
    scanf("%d",&led);
    retorno = led_on(led); //chamada da função
    printf ("Led = %d Situacao: %d",led,retorno);
}

```

Variáveis locais:

Variáveis declaradas dentro de **uma função** são denominadas **locais** e somente podem ser usadas **dentro do próprio bloco**. São criadas apenas na entrada do bloco e destruídas na saída automaticamente.

Variáveis globais:

Variável declarada **externamente** podendo ser acessada por **qualquer função**.

Exemplo (com variáveis globais e local)

```
#include <stdio.h>
int retorno; //variável global
int soma(int a, int b){
    int res; //variável local
    res = a+b;
    return(res);
}
main() {
    int a,b;
    printf("\nInforme 2 valores:\n");
    scanf("%d %d",&a,&b);
    retorno = soma(a,b);
    printf("Soma = %d",retorno);
}
```