

Prof. esp. Thalles Canela

- **Graduado:** Sistemas de Informação - Wyden Facimp
- **Pós-graduado:** Segurança em redes de computadores - Wyden Facimp
- **Professor:** Todo núcleo de T.I. (Graduação e Pós) - Wyden Facimp
- **Diretor:** SCS
- **Gerente de Projetos:** Motoca Systems

Redes sociais:

- **Linkedin:** <https://www.linkedin.com/in/thalles-canela/>
- **YouTube:** <https://www.youtube.com/aXR6CyberSecurity>
- **Facebook:** <https://www.facebook.com/axr6PenTest>
- **Instagram:** https://www.instagram.com/thalles_canela
- **Github:** <https://github.com/ThallesCanela>
- **Github:** <https://github.com/aXR6>
- **Twitter:** <https://twitter.com/Axr6S>

TIPOS DE FUNÇÕES PREDEFINIDAS NA LINGUAGEM C

A linguagem C possui várias bibliotecas padrão que contêm muitas funções predefinidas para realizar tarefas comuns.

Alguns exemplos:

- **printf** e **scanf** da biblioteca **stdio.h** para entrada e saída.
- **strlen**, **strcpy** e **strcmp** da biblioteca **string.h** para manipular strings.
- **malloc** e **free** da biblioteca **stdlib.h** para alocação de memória.

c

 Copy code

```
#include <stdio.h>
#include <string.h>

int main() {
    char str[50];
    strcpy(str, "OpenAI");
    printf("Comprimento da string: %d\n", strlen(str));
    return 0;
}
```

PASSAGEM DE PARÂMETROS, POR VALOR E POR REFERÊNCIA.

- **Por Valor:** O valor do argumento é passado para o parâmetro da função. Alterações feitas no parâmetro não afetam o argumento.

c

 Copy code

```
void modificarValor(int a) {  
    a = a + 10;  
}  
  
int main() {  
    int num = 5;  
    modificarValor(num);  
    printf("%d", num); // Imprime 5  
    return 0;  
}
```

PASSAGEM DE PARÂMETROS POR VALOR E POR REFERÊNCIA


- **Por Referência:** Um endereço de memória (referência) é passado como argumento. Alterações feitas no parâmetro afetam o argumento.

```
void modificarReferencia(int *a) {  
    *a = *a + 10;  
}  
  
int main() {  
    int num = 5;  
    modificarReferencia(&num);  
    printf("%d", num); // Imprime 15  
    return 0;  
}
```


ESCOPO DAS VARIÁVEIS LOCAIS E GLOBAIS

- **Variáveis Locais:** Existem apenas dentro da função ou bloco onde foram declaradas.

c

 Copy code

```
void funcaoExemplo() {  
    int variavelLocal = 10; // Apenas visível nesta função  
}
```

ESCOPO DAS VARIÁVEIS LOCAIS E GLOBAIS

- **Variáveis Globais:** Declaradas fora de todas as funções e são visíveis para qualquer parte do código.

c

 Copy code

```
int variavelGlobal = 20; // Visível em todas as funções

void funcaoExemplo() {
    printf("%d", variavelGlobal); // Acesso a variável global
}
```