

## Diferença entre argumentos e parâmetros

A diferença entre **argumentos** e **parâmetros** está no contexto em que eles são usados na programação:

### 1. Parâmetros:

- **Definição:** São as variáveis que aparecem na definição de uma função. Eles servem como "espaços reservados" para os valores que serão passados quando a função for chamada.
- **Contexto:** Aparecem na **definição** da função.
- **Exemplo:**

```
// 'int a' e 'int b' são os parâmetros
int soma(int a, int b) {
    return a + b;
}
```

Aqui, a e b são **parâmetros** da função soma.

### 2. Argumentos:

- **Definição:** São os valores reais que são passados para a função quando ela é chamada. Esses valores são atribuídos aos parâmetros da função.
- **Contexto:** Aparecem na **chamada** da função.
- **Exemplo:**

```
int resultado = soma(5, 10); // '5' e '10'
são os argumentos
```

Neste exemplo, os valores 5 e 10 são **argumentos** que são passados para a função soma.

## Resumo:

- **Parâmetros:** São as variáveis definidas na assinatura da função e servem como "espaços reservados".
- **Argumentos:** São os valores reais que são passados para a função quando ela é chamada.

## Exemplo Completo:

```
#include <stdio.h>

// 'a' e 'b' são os parâmetros
int soma(int a, int b) {
    return a + b;
}

int main() {
    // '5' e '10' são os argumentos
    int resultado = soma(5, 10);
    printf("Resultado: %d\n", resultado);
    return 0;
}
```

Neste exemplo:

- **Parâmetros:** a e b (definidos na função soma).
- **Argumentos:** 5 e 10 (passados na chamada da função).

## Regra Geral:

- **Parâmetro:** O que está na definição da função.
- **Argumento:** O que é passado na chamada da função.