

Tipos

- **Funções sem argumento e sem valor de retorno:** Utilizadas para executar uma tarefa sem necessidade de entrada ou saída.;
- **Funções com argumento e sem valor de retorno:** Recebem dados de entrada, mas não retornam nenhum valor;
- **Funções sem argumento, mas com valor de retorno:** Não recebem dados, mas retornam um valor;
- **Funções com argumento e valor de retorno:** Recebem dados de entrada e retornam um valor.

Função Recursiva

São funções que se chamam a si mesmas durante sua execução, úteis para resolver problemas que podem ser divididos em subproblemas menores.

- Casos base
- Casos recursivos
- Exemplo: cálculo de fatorial, Fibonacci

Como são declaradas?

Em Python: Funções são declaradas com a palavra-chave **def**, seguida pelo nome da função e parâmetros entre parênteses.

Exemplo: **def teste(a,b):**
return a + b

- Nome da função;
- Tipo de retorno (se aplicável);
- Parâmetros (se aplicável).

Funções/ Procedimentos

Comparação:

- **C:** Estrita necessidade de declaração antes do uso, com foco na modularidade e clareza do código;
- **C++:** Permite mais flexibilidade com sobrecarga de funções e funções dentro de classes;
- **C#:** Integra funções dentro de um ambiente orientado a objetos, com suporte a namespaces e classes;
- **Python:** Oferece uma abordagem mais flexível e dinâmica, sem a necessidade de declaração de tipos de dados.

Como são utilizadas?

Funções são chamadas pelo nome, com argumentos passados entre parênteses, e podem ser métodos de objetos.

Exemplo: **soma = teste(5,4)**

- Chamada da função
- Passagem de parâmetros
- Captura de retorno (se aplicável)

Exemplos:

C:

```
int soma(int a, int b) {  
    return a + b;  
}
```

C++:

```
int soma(int a, int b)  
{ return a + b;  
}
```

C#:

```
static int Soma(int a, int b) {  
    return a + b;  
}
```

Python:

```
def soma(a, b):  
    return a + b
```