

Funções

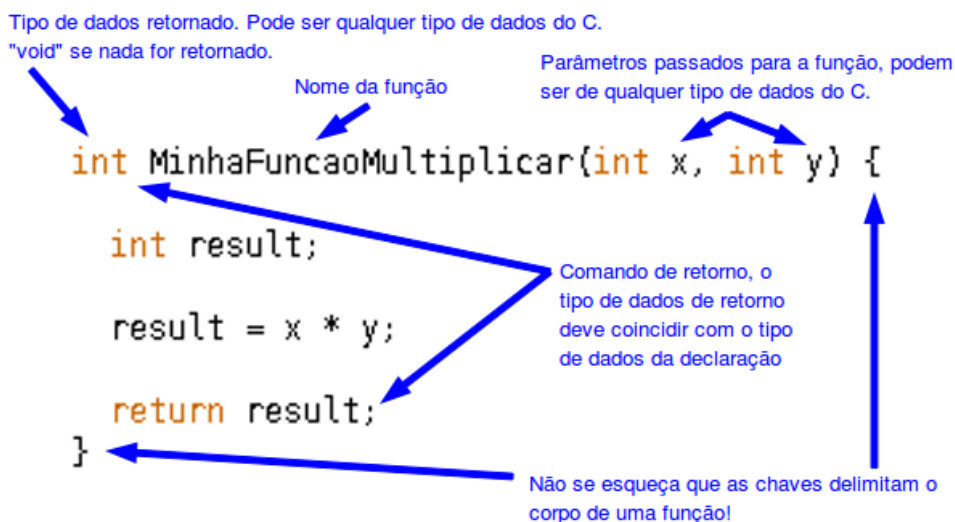
Separar partes do código de um programa em funções permite que um programador crie peças de código modulares que executem uma tarefa definida e, em seguida, retornem à área do código a partir da qual a função foi "chamada". O caso típico para criar uma função é quando é necessário executar a mesma ação várias vezes num programa.

Padronizar partes do código em funções tem várias vantagens:

- As funções ajudam o programador a se manter organizado. Muitas vezes isso ajuda a entender o programa.
- As funções codificam uma ação em um só lugar, de modo que a função só precisa ser pensada e depurada uma vez.
- Isso também reduz as chances de erros na modificação, se o código precisar ser alterado.
- As funções tornam o esboço inteiro menor e mais compacto porque seções de código são reutilizadas muitas vezes.
- Eles tornam mais fácil reutilizar o código em outros programas, tornando-o mais modular, e como um efeito agradável, o uso de funções também faz com que o código seja mais legível.
- Existem duas funções padrões num programa Arduino, *setup()* e *loop()*. Outras funções devem ser criadas fora do escopo dessas duas funções.

Por exemplo, veja a função abaixo:

Anatomia de uma Função em C



Essa função recebe dois parâmetros: x e y, dentro da função é realizado o processo de multiplicação entre elas e retornando o resultado.



Os argumentos são passados como valores, dentro da função são atribuídos nomes a esses valores, portanto fora da função as variáveis passadas não existirão.

Para chamar essa função devemos passar os 2 parâmetros esperados (os argumentos) e uma variável com o resultado:

```
void loop(){
  int i = 2;
  int j = 3;
  int k;

  k = MinhaFuncaoMultiplicar(i, j); // A função irá retornar o resultado da multiplicação
}
```

O programa completo ficaria assim:

```
void setup(){
  Serial.begin(9600);
}

void loop() {
  int i = 2;
  int j = 3;
  int k;

  k = MinhaFuncaoMultiplicar (i, j); // A função irá retornar o resultado da multiplicação
  Serial.println(k);
  delay(500);
}

int MinhaFuncaoMultiplicar (int x, int y){
  int result;
  result = x * y;
  return result;
}
```