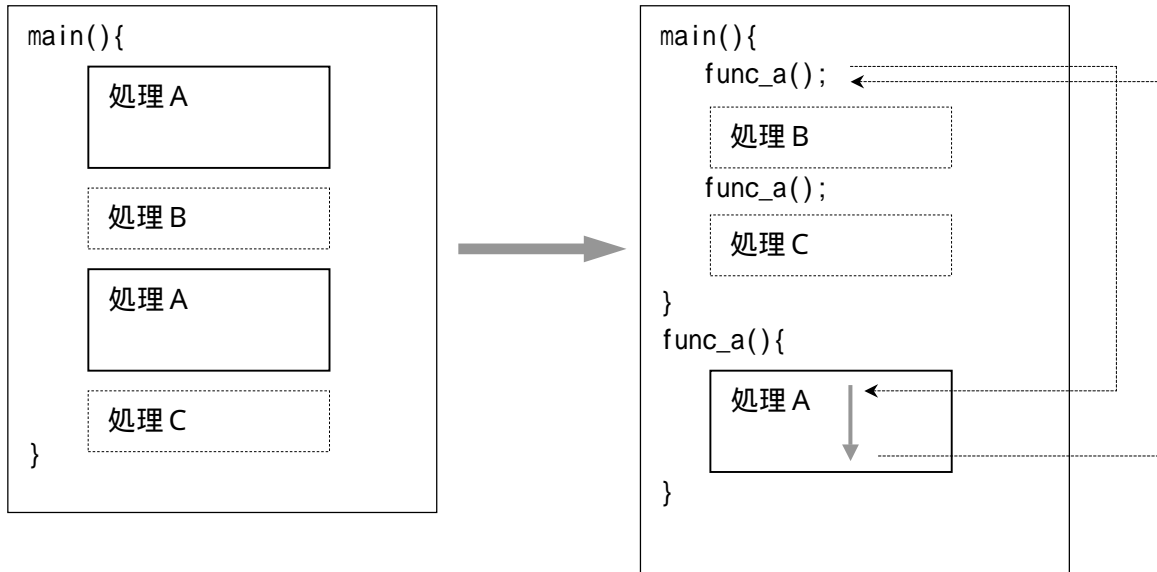


関数とは … 一連の処理（手続き、機能）をブロックとしてまとめたもの

ユーザ定義関数



関数利用の利点

- プログラム内の同一処理（機能）は 1 回記述すればよい
 （重複記述の回避、プログラムサイズの減少、メンテナンス（修正）の効率化）
- 大規模プログラムにおける機能の分散化
 （個別翻訳が可能、開発効率の向上）
- 機能の相互独立
 （外部の関数から変数や配列が独立）
- 機能の共同利用
 （標準ライブラリ関数の利用など）

関数の宣言と呼び出し

```
double func_a(int ,int);      関数のプロトタイプ宣言
main(){
    double kotae;
    :
    kotae = func_a(x, y);    関数の呼び出し（x と y を実引数という）
}

double func_a(int a, int b)  関数の実体（a と b を仮引数という）
{
    ...
    ... （処理）
    return (返却する値);    返却する必要がある場合は return を省略できる
}
```

プロトタイプ宣言の指定例

- int enzan(int ,int); 引数が 2 つ（int 型）で戻り値も int 型
- void hyouji(double); 引数が 1 つ（double 型）で、戻り値なし
- void sort_shori(void); 引数なし、戻り値なし