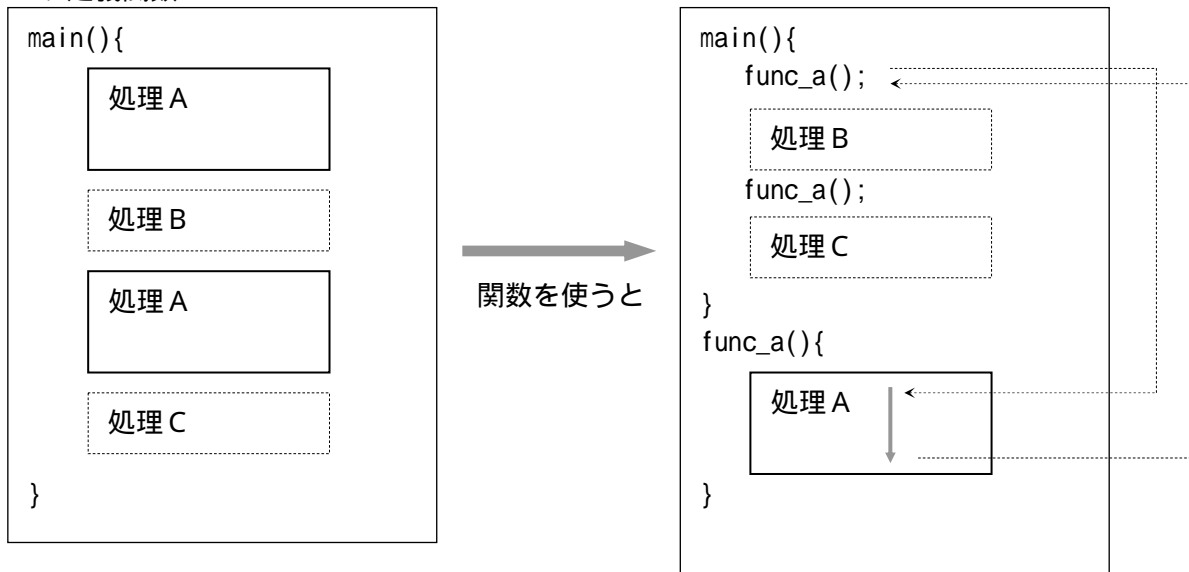


関数とは … 一連の処理（手続き、機能）をブロックとしてまとめたもの

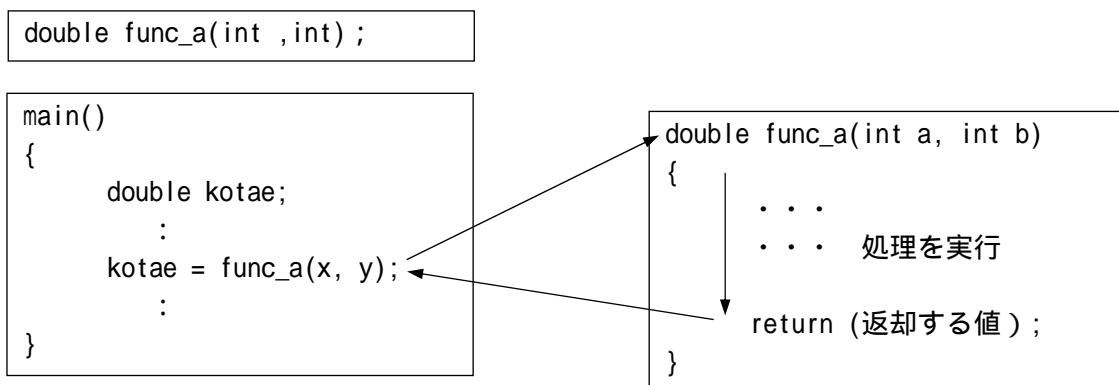
ユーザ定義関数



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



- 関数のプロトタイプ宣言（mainの前に書く）
- 関数の呼び出し（xとyを実引数という）
- 関数の本体を実行（aとbを仮引数という）
- returnでmain関数に戻る（値を返却する必要がない場合はreturn文を省略できる）

プロトタイプ宣言の例 戻り値の型 関数名(引数,...)

- 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