

基本骨格

```
#include <ヘッダ 1>
```

```
#include <ヘッダ 2>
```

```
:
```

```
int main(void)
```

```
{
```

変数の宣言

実行すること

```
return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
int i;
```

```
printf("Hello world!¥n");
```

```
return 0;
```

```
}
```

変数の宣言

```
#include <ヘッダ>
```

```
int main(void)
```

```
{
```

変数の宣言1

変数の宣言2

```
:
```

```
return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

→ int i;

→ float x;

```
return 0;
```

```
}
```

変数の型と表記

型	扱える範囲(gcc-cygwin)	printf 中の表記
文字型 char	-127 ~ 128 (文字)	%c(文字), %d(値)
整数型 int	-2147483648 ~ 2147483647	%d
倍長整数型 long	- 9223372036854775808 ~ 9223372036854775807	%ld
単精度浮動小数点型 float	有効数字6桁の実数 3.4E-38 ~ 3.4E+38	%f
倍精度浮動小数点型 double	有効数字15桁の実数 1.7E-308 ~ 1.7E+308	%f

変数の名付け規則 :

- 1) 先頭文字は英字(a-z, A-Z)または下線(_)
- 2) 2文字目以降は英字, 下線, 数字
- 3) 大文字と小文字は区別
- 4) 先頭から31文字までが有効, 予約語(do, for, ...)などは不可

変数宣言と初期値代入

```
#include <ヘッダ>
```

```
int main(void)
```

```
{
```

変数の宣言、初期値代入1

変数の宣言、初期値代入2

```
return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
int i=0;
```

```
float x=1.3E3;
```

```
return 0;
```

```
}
```

変数宣言と後から代入

```
#include <ヘッダ>
```

```
int main(void)
```

```
{
```

変数の宣言1

変数の宣言2

変数の代入1

変数の代入2

```
return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
int i;
```

```
float x;
```

```
i=0;
```

```
x=1.3E3;
```

```
return 0;
```

```
}
```

画面に「数値」出力

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

変数の宣言と初期値代入

画面への出力命令

```
return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
int i=3;
```

```
printf("%d\n",i);
```

```
return 0;
```

```
}
```

画面に「文章」出力

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

画面への出力命令

```
return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main(void)
```

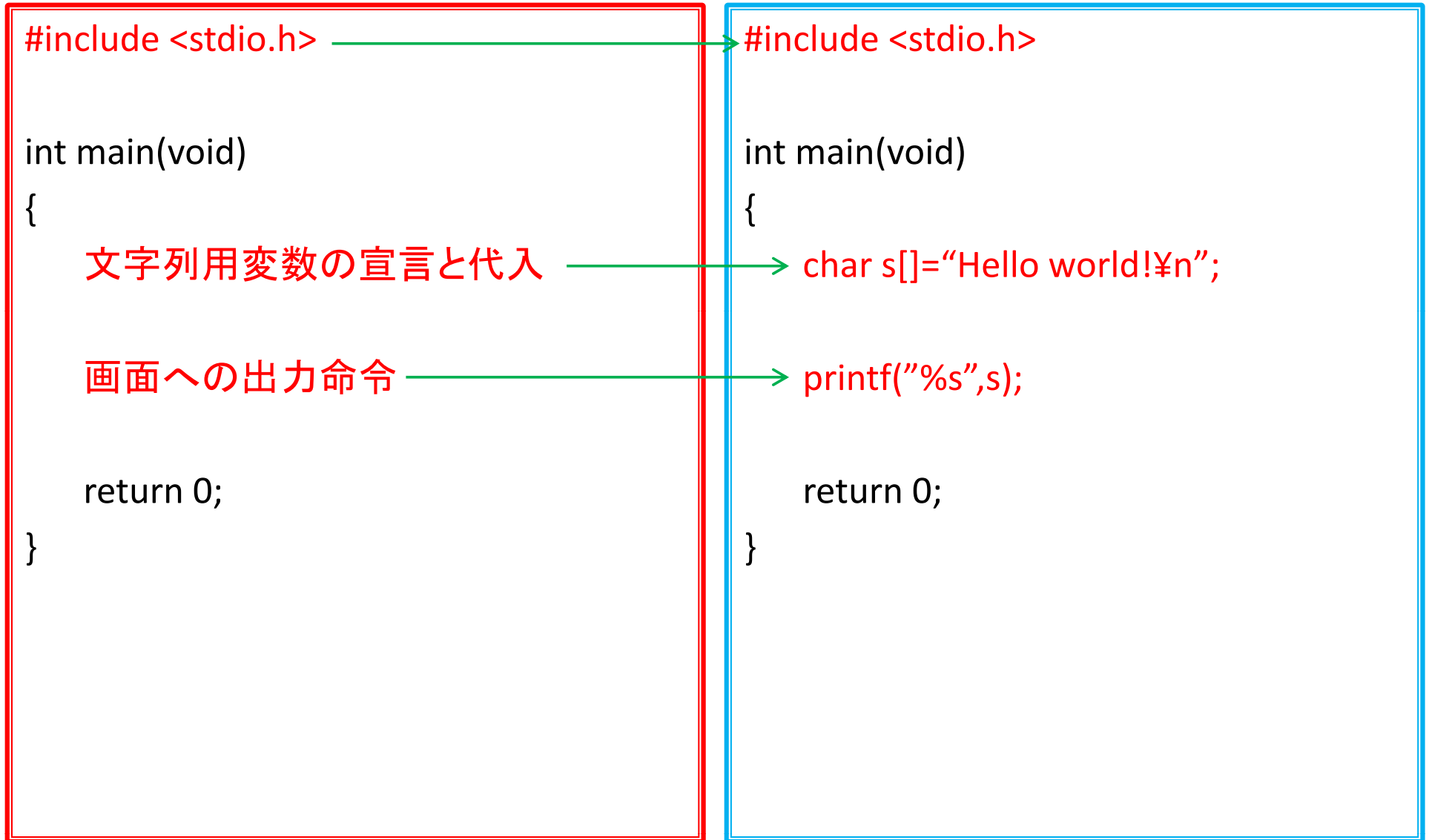
```
{
```

```
printf("Hello world!¥n");
```

```
return 0;
```

```
}
```

変数に記憶した「文章」を画面出力



キーボードから「数値」入力

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

変数の宣言

キーボードからの入力命令

```
return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
int a;
```

```
scanf("%d",&a);
```

```
return 0;
```

```
}
```

キーボードから「一文字」入力

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

変数の宣言

キーボードからの入力命令

```
return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
char a;
```

```
scanf("%c",&a);
```

```
return 0;
```

```
}
```

キーボードから「文章」入力

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

変数の宣言

キーボードからの入力命令

```
return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
char a[20];
```

```
scanf("%s",a);
```

```
return 0;
```

```
}
```

入出力の方法 printf(), scanf()

変換指定文字	意味	使われるデータ型
%c	1文字として出力	文字型
%d (%ld)	10進数 で出力 (long型のデータの場合)	整数型
%x	16進数で出力	整数型
%o	8進数で出力	整数型
%f (%lf)	[-]dddd.dddddの形式 で出力 (double型のデータの場合)	浮動小数点型
%e	指数形式 で出力	浮動小数点型
%s	文字列 として出力	文字型

出力幅の指定

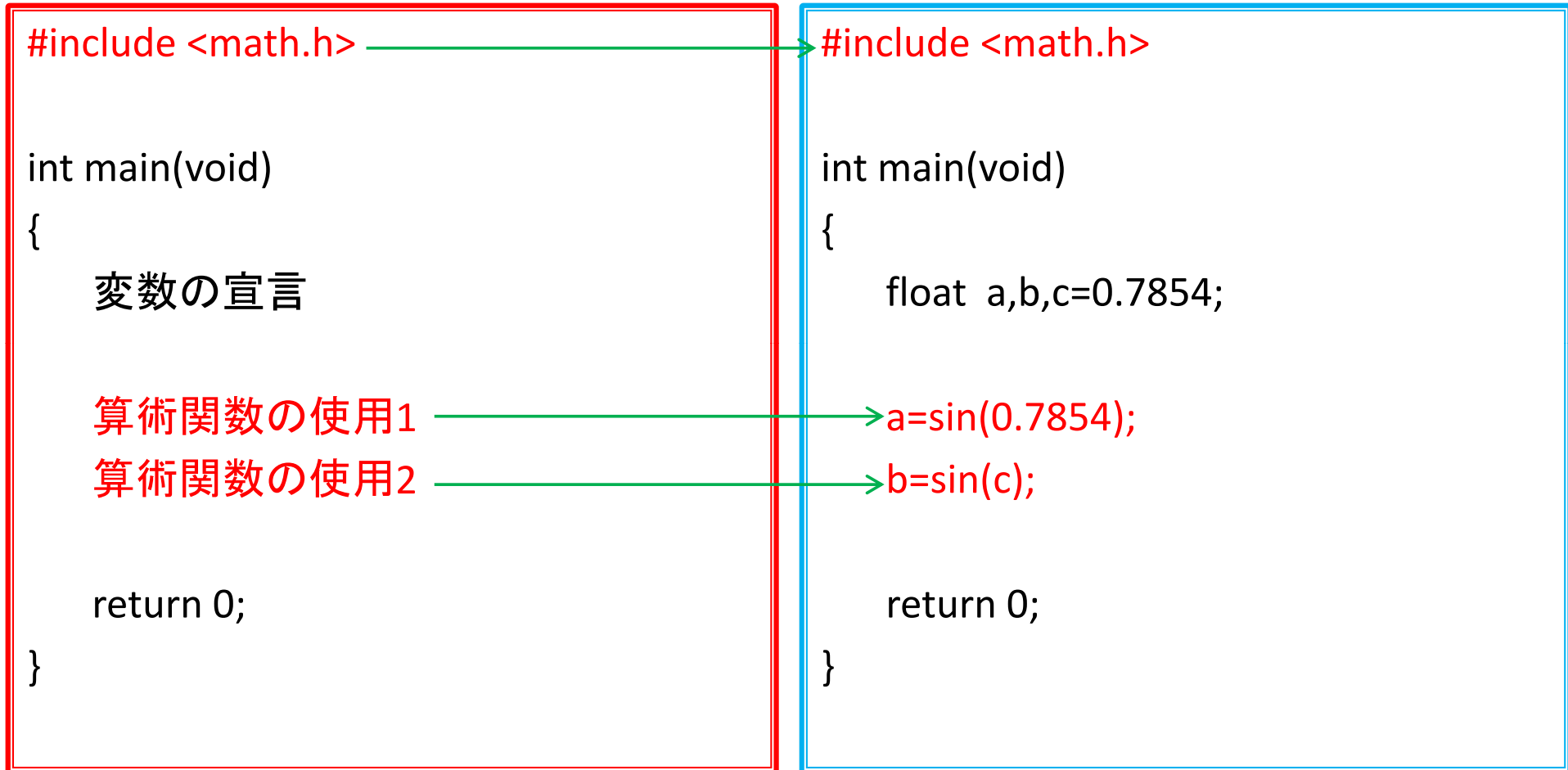
(例) int a=123; の場合

プログラム	実行結果	
<code>printf(“%d¥n”,a);</code>	123	(指定のない場合)
<code>printf(“%5d¥n”,a);</code>	..123	スペースを含めて5文字
<code>printf(“%10d¥n”,a);</code>123	スペースを含めて10文字

(例) float x=123.4567890; の場合

プログラム	実行結果	
<code>printf(“%f¥n”,x);</code>	123.456787	(指定のない場合)
<code>printf(“%12f¥n”,x);</code>	..123.456787	小数点を含めて12桁, 小数点以下は標準の桁数
<code>printf(“%9.2f¥n”,x);</code>	...123.46	小数点を含めて9桁,小数点以下2桁

算術関数の使用



コンパイル時に「-lm」をつける → `$ cc -lm ex-1.c`

算術演算子,算術関数

意味	数学記号	C言語での書き方
加算	+	+
減算	-	-
乗算	×	*
除算	÷	/
あまり	...	%

分類	「#include <math.h>」とコンパイル時「-lm」必要
三角関数、逆三角関数	sin(x) cos(x) tan(x) asin(x) acos(x) atan(x) atan2(x,y)
指数関数、対数関数	sqrt(x) exp(x) log(x) log10(x) ベキ乗: pow(x,y)
その他	切り上げ: ceil(x) 切り捨て: floor(x) 絶対値: fabs(x) ...

乱数(古い)

```
#include <stdlib.h>  
#include <time.h>
```

```
int main(void)
```

```
{
```

```
    int a,b;
```

乱数系列の初期化

乱数の使用1 (0~最大)

最大: RND_MAX=2147483647

乱数の使用2 (0~9)

```
    return 0;
```

```
}
```

```
#include <stdlib.h>  
#include <time.h>
```

```
int main(void)
```

```
{
```

```
    int a,b;
```

```
    srand((unsigned int)time(NULL));
```

```
    a=rand();
```

```
    b=rand()%10;
```

```
    return 0;
```

```
}
```


乱数 (新しい)

```
#include <stdlib.h>  
#include <time.h>
```

```
int main(void)
```

```
{
```

```
    int a,b;
```

乱数系列の初期化

乱数の使用1 (0~最大)

最大: RND_MAX=2147483647

乱数の使用2 (0~9)

```
    return 0;
```

```
}
```

```
#include <stdlib.h>  
#include <time.h>
```

```
int main(void)
```

```
{
```

```
    long a,b;
```

```
    srand((unsigned long)time(NULL));
```

```
    a=random();
```

```
    b=random()%10;
```

```
    return 0;
```

```
}
```

条件分岐 (2重)

```
#include <ヘッダ>
```

```
int main(void)
```

```
{
```

```
    条件式用の変数宣言
```

```
    if(条件式){
```

```
        条件が成り立つとき実行
```

```
    }
```

```
    else{
```

```
        条件が成り立たないとき実行
```

```
    }
```

```
    return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int i;
```

```
    if(i==0){
```

```
        printf("iは0です。¥n");
```

```
    }
```

```
    else{
```

```
        printf("iは0以外です。¥n");
```

```
    }
```

```
    return 0;
```

```
}
```

関係演算子

意味	数学記号	C言語
等しい	=	==
より大きい	>	>
より小さい	<	<
等しいか, より大きい	\geq	>=
等しいか, より小さい	\leq	<=
等しくない	\neq	!=

条件式の書き方

(Ex.1) $a+b = c$

→ $a+b == c$

(Ex.2) $a+b \geq c$

→ $a+b >= c$

論理演算

A, B のそれぞれを条件式として

意味	名前	C言語での表現
A, Bの両者とも成立すれば 真	論理積 AND	A && B
A, Bのどちらかが成立すれば 真	論理和 OR	A B
Aが成立しなければ 真	否定 NOT	!A

(Ex.1) “ $x+y = c$ ” と “ $z > 0$ ” の両方が成立すれば真

→ $x+y == c \ \&\& \ z > 0$

(Ex.2) “ $0 < x < 100$ ” → $0 < x \ \&\& \ x < 100$

(Ex.3) “ $x = y = 1$ ” → $x == 1 \ \&\& \ y == 1$

(注) || よりも && の方が優先順位は高い。 同じ順位の間では左から順に評価される。 また, () でくくってあればその中が先に評価される。

条件分岐(多重、数値)

```
#include <ヘッダ>
```

```
int main(void)
```

```
{
```

条件式用の変数宣言

```
switch(i){
```

case 0: i=0の時実行すること
ブレーク文(以後実行せず)

case 1: i=1の時実行すること
ブレーク文

case 2: i=2の時実行すること
ブレーク文

default: 上に当てはまらないとき

```
}
```

```
return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
int i;
```

```
switch(i){
```

case 0: printf("iは0です。¥n");
break;

case 1: printf("iは1です。¥n");
break;

case 2: printf("iは2です。¥n");
break;

default: printf("0,1,2以外です。¥n");

```
}
```

```
return 0;
```

```
}
```

条件分岐 (多重、文字)

```
#include <ヘッダ>
```

```
int main(void)
```

```
{
```

条件式用の変数宣言

```
switch(i){
```

```
case 'a': i=aの時実行すること  
         ブレーク文(以後実行せず)
```

```
case 'b': i=bの時実行すること  
         ブレーク文
```

```
case 'c': i=cの時実行すること  
         ブレーク文
```

```
default: 上に当てはまらないとき
```

```
}
```

```
return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
char i;
```

```
switch(i){
```

```
case 'a': printf("iはaです。¥n");  
         break;
```

```
case 'b': printf("iはbです。¥n");  
         break;
```

```
case 'c': printf("iはcです。¥n");  
         break;
```

```
default: printf("a,b,c以外です。¥n");
```

```
}
```

```
return 0;
```

```
}
```

繰り返し(for文)

```
#include <ヘッダ>
```

```
int main(void)
```

```
{
```

繰り返し用変数の宣言

```
for(初期値; 条件式; ループ後処理){
```

繰り返し処理

```
}
```

```
return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
int i;
```

```
for(i=0; i<10; i++){
```

```
printf("Hello world! No. %d¥n",i);
```

```
}
```

```
return 0;
```

```
}
```

繰り返し (while文)

```
#include <ヘッダ>
```

```
int main(void)
```

```
{
```

繰り返し用変数の宣言と初期化

```
while(条件式){
```

```
    繰り返し処理1
```

```
    繰り返し処理2
```

```
}
```

```
return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int i=0;
```

```
    while(i<10){
```

```
        printf("Hello world! No. %d¥n",i);
```

```
        i++;
```

```
    }
```

```
return 0;
```

```
}
```


繰り返し (do, while 文)

```
#include <ヘッダ>
```

```
int main(void)
```

```
{
```

繰り返し用変数の宣言と初期化

```
do{
```

繰り返し処理1

繰り返し処理2

```
}while(条件式);
```

```
return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
int i=0;
```

```
do{
```

```
printf("Hello world! No. %d¥n",i);
```

```
i++;
```

```
}while(i<10);
```

```
return 0;
```

```
}
```

C言語独特の表現

```
#include <ヘッダ>
```

```
int main(void)
```

```
{
```

変数の宣言

変数に値代入

値を計算して変数に代入し直し

値を1増やす

値を1減らす

値を一定値増やす;

```
return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
int i;
```

```
i=1;
```

```
i=i+1;
```

```
i++;
```

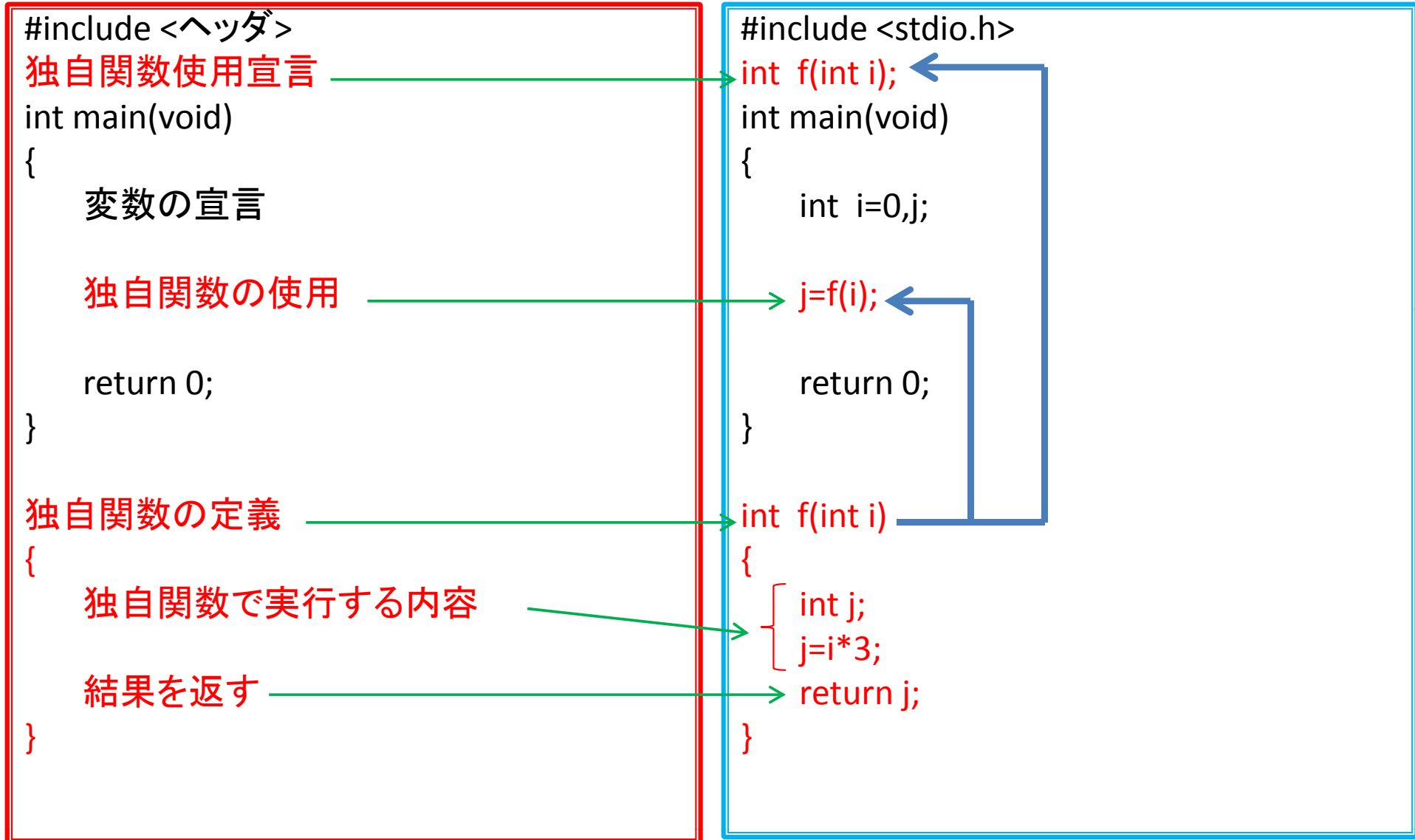
```
i--;
```

```
i+=2;
```

```
return 0;
```

```
}
```

独自関数の定義と使用



ファイルに出力

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main(void)
```

```
{
```

ファイル用変数(ファイルポインタ)の
宣言

ファイルをオープン

ファイルへの書き込み

ファイルをクローズ

```
return 0;
```

```
}
```

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main(void)
```

```
{
```

```
FILE *f;
```

```
if((f = fopen("test.txt","w"))==NULL){
    printf("オープン失敗¥n");
    exit(1);
}
```

```
fprintf(f, "Hello world!¥n");
```

```
fclose(f);
```

```
return 0;
```

```
}
```

ファイルから入力

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
```

```
{
```

ファイル用変数(ファイルポインタ)の
宣言

ファイルをオープン

ファイルから読み込み

ファイルをクローズ

```
return 0;
```

```
}
```

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
```

```
{
```

```
FILE *f;
```

```
int i;
```

```
if((f = fopen("test.txt","r"))==NULL){
    printf("オープン失敗\n");
    exit(1);
}
```

```
fscanf(f, "%d",&i);
```

```
fclose(f);
```

```
return 0;
```

```
}
```