

【可変引数】

引数を「...」にすると「可変引数」になります。

可変引数は、引数の数が決まっていなく、複数の値を受け取ることができる引数で、printf 関数や scanf 関数等で使用されています。

<stdarg.h ヘッダーファイル>

可変引数を使用するには「stdarg.h」をインクルードする必要があります。

<可変引数の書き方>

```
戻り値のデータ型 関数名( 第 1 引数, ... ) {
```

第 2 引数を可変引数にしています。可変引数は第 2 引数以降でなければ設定することができません。

<例文>

```
#include <stdio.h>
#include <stdarg.h>          /* 可変引数用のマクロを使用するのに必要 */

int main(void) {
    void func(int, ...);    /* 関数プロトタイプ宣言 */

    /* 第 2 引数以降は可変引数 */
    func(1, 100);
    func(2, 100, 200);
    func(3, 100, 200, 300);

    return 0;
}

void func(int num, ...) {          /* 第 1 引数は num、第 2 引数は可変引数 */
    va_list list;                 /* 可変引数の実引数を格納する「va_list 型」 */
    int i;
    va_start(list, num);          /* 可変引数を初期化する「va_start マクロ」 */
    for(i = 0; i < num; i++){
        printf("%d\n", va_arg(list, int)); /* va_list から値を取得する「va_arg マクロ」 */
    }
    va_end(list);                 /* 可変引数を終了する「va_end マクロ」 */
}
```

<va_list 型>

```
va_list list;
```

可変引数の実引数を格納するデータ型です。

多くのコンパイラでは「typedef void* va_list;」で宣言されています。

<va_start マクロ>

```
va_start(list, num);
```

list に実引数値のアドレスを記録するマクロです。

第 2 引数には可変引数の 1 個前の引数名を指定します。例えば、関数定義が

「 void func(int num, char* str, ...) { 」であれば「str」になります。

<va_arg マクロ>

```
va_arg(list, int)
```

list に記録している値をひとつずつ取得するマクロです。

第 2 引数には取り出す値のデータ型を指定します。

ひとつ値を取得すると、次のマクロ実行時には次の値が取得されます。

なお、printf 関数や scanf 関数は、書式指定子から可変引数の個数とデータ型を把握しています。

<va_end マクロ>

```
va_end(list);
```

list のアドレスを NULL にリセットするマクロです。