C 言語 014 色々な変数

【自動変数と静的変数】

自動変数（auto 変数）と静的変数（static 変数）は、値の保存期間が異なります。

＜auto 変数＞

```
int main(void) {    /* 関数開始 */
    int var;        /* 本来は「auto int var;」と記述するが auto は省略可 */
    return 0;
}                   /* 関数終了 */
```

最も良く使用している、上記のように宣言した変数を「自動変数（auto 変数）」と称します。
関数開始時にメモリの中に領域が確保され、関数終了時にメモリから解放され、記録している値は消去される変数です。初期化しないと不定値になります。

＜static 変数＞

```
static データ型 変数名;
```

「static」を付けて宣言した変数を「静的変数（static 変数）」と称します。
関数開始時にメモリの中に領域が確保され、プログラムが終了するまで、値を保存し続ける変数です。
初期化は 1 度だけ行われ、初期化しないと値は 0（ゼロ）になります。

＜例文＞

```
int main(void) {
    void func(void);    /* 関数プロトタイプ宣言 */
    int i;
    for (i = 0; i < 10; i++) {
        func();         /* 関数を 10 回実行する */
    }
    return 0;
}

void func(void) {
    static int var = 0;     /* 初期化は 1 度だけ実行される */
    var++;                  /* var を+1 する */
    printf("%d¥n", var);    /* var の値を表示 */
}
```

実行結果は「1～10」が出力されます。

変数 var が自動変数であれば、func 関数を呼び出すたびに 0 で初期化され、+1 され、関数終了後に消去されますが、変数 var は静的変数です。
初期化は 1 度だけ行い、関数を呼び出されるたびに+1 され、プログラムが終了するまで値を記録し続けます。

【ローカル変数とグローバル変数】
ローカル変数とグローバル変数は、変数を使用できる範囲（スコープ）が異なります。

＜ローカル変数＞

```
int main(void) {
    int var;    /* 関数内に宣言 */
    return 0;
}
```

上記のように、関数内に記述した変数を「ローカル変数」と称します。
ローカル変数のスコープは、宣言した関数の中のみ使用できます。
static を付ければ静的変数、auto を付けるか何も付けなければ自動変数となります。

＜グローバル変数＞

```
int var;        /* グローバル領域に宣言 */

int main(void) {
    var = 10;    /* グローバル変数は全ての関数内で使用できる */
    return 0;
}
```

上記のように、関数の外（グローバル領域）で記述した変数を「グローバル変数」と称します。
グローバル変数のスコープは、記述した位置から下記にある全ての関数内で使用することができ、自動的に静的変数になります。

＜例文＞

```
int var;                    /* グローバル変数 */

void func(void) {
    var++;                  /* var を+1 する */
}

int main(void) {
    var++;                  /* var を+1 する */
    func();                 /* 関数実行 */
    printf("%d¥n", var);    /* var の値を表示 */
    return 0;
}
```

実行結果は「2」が出力されます。
グローバル変数 var はメイン関数と func 関数で使用できます。

＜関数プロトタイプ宣言も＞
関数プロトタイプ宣言もグローバル領域に記述してグローバル化できます。それにより宣言した位置から下記にある全ての関数内で関数プロトタイプ宣言が有効化されます。

＜変数名の重複について＞
変数名は同スコープ内で重複してはいけません。ローカル変数なら同関数内で、グローバル変数ならグローバル領域内で重複してはいけません。
ローカル変数とグローバル変数が同名だった場合はローカル変数が優先されます。

```
int var1;
double var1;        /* グローバル変数名が重複している */
char var3;

int main(void) {
    int var2;
    double var2;    /* ローカル変数名が重複している */
    char var3;

    var3 = '3'      /* ローカル変数の var3 になる */
```

【4 つの領域】

プログラムはメモリを 4 つの領域に分別しており、変数の種類によって記録される領域が異なります。

プログラム領域 （コード領域）	プログラムを実行するためのコードを記録する領域。
スタック領域	自動変数、実引数を記録する領域。関数実行時に確保され、関数終了後に自動的に解放される。容量は最大 1MB 程。
スタティック領域	静的変数を記録する領域。プログラム実行時に確保され、プログラム終了時に解放される。容量は最大 100MB 程。
ヒープ領域	動的確保用の領域。プログラムの中で自由に確保、解放を行うことができる。 容量は最大で 2GB 程。「malloc 関数」等を使用して確保を行う。

メモリの容量を使い切るとエラーが発生し、プログラムが中断されます。

最大容量数は OS やコンパイラや設定等によって異なります。