C 言語 007 型変換

【演算結果のデータ型】

var1 + var2

var1 も var2 も同じデータ型の場合、演算結果も同データ型になります。

var1 と var2 が異なるデータ型の場合、演算結果は「変換順位」の高い方のデータ型になります。

（ただし short 型 + short 型の演算結果が、short 型の範囲を超えた場合は long 型になる）

＜変換順位＞

long double > double > float > unsigned long > long > unsigned short > short

（MinGW では int 型は long 型になります）

＜例文 1＞

```
double var = 5 / 2;
printf("%f", var);
```

＜実行結果＞

2.000000

整数リテラルは int 型です。「5 / 2」は int 型同士の演算なので、演算結果も int 型になります。

「5 / 2 = 2.5」ですが、演算結果が int 型になるため小数は切り捨てられ「2」になってしまい、それが変数 var に代入されてしまいました。

演算結果を「2.5」にしたいなら「キャスト」を行います。

【キャスト】

「キャスト」は一時的にデータ型を変換する演算子です。

＜キャストの書き方＞

```
( データ型 )値・変数
```

＜例文 2＞

```
double var = (double)5 / (double)2;
printf("%f", var);
```

＜実行結果＞

2.500000

例文 2 では整数リテラルを double 型に変換して演算したため、演算結果も double 型になりました。

【代入時の自動変換】

var1 = var2

var2 のデータ型に関係無く、var1 のデータ型に変換されます。

ただし

- var2 の値が var1 のデータ型の範囲を超えたらオーバーフローになる。
- var1 が整数型で var2 が実数型の場合、コンパイラによって処理が異なるが、MinGW では小数は切り捨てられる。

【文字列を数値型に変換】

```
int var1 = atoi("10");          /* 文字列「10」を int 型に変換して代入 */
double var2 = atof("10.5");     /* 文字列「10.5」を double 型に変換して代入 */
```

「atoi 関数」は引数の文字列を整数に、「atof 関数」は引数の文字列を実数に変換する関数です。

＜atoi 関数＞

定　　義	int atoi(const char* str);
ﾍｯﾀﾞｰﾌｧｲﾙ	stdlib.h
説　　明	引数に指定した文字列を int 型に変換する。
戻 り 値	正常：変換された整数／異常：0

＜atol 関数＞

定　　義	long atol(const char* str);
ﾍｯﾀﾞｰﾌｧｲﾙ	stdlib.h
説　　明	引数に指定した文字列を long 型に変換する。
戻 り 値	正常：変換された整数／異常：0

＜atof 関数＞

定　　義	double atof(const char* str);
ﾍｯﾀﾞｰﾌｧｲﾙ	stdlib.h
説　　明	引数に指定した文字列を double 型に変換する。
戻 り 値	正常：変換された実数／異常：0

文字列が変換できなかった場合の処理はコンパイラによって処理が異なりますが、MinGW では戻り値に 0 が返されます。

【型のチェック】

＜isdigit 関数＞

定　　義	int isdigit(int val);
ﾍｯﾀﾞｰﾌｧｲﾙ	ctype.h
説　　明	引数が 10 進数字か判定する。
戻 り 値	判定が真であれば 0 以外、偽であれば 0 を返す。

＜例文＞

```
char var = 'a';
if (isdigit(var) == 0) {
    printf("数値ではない");
}
```

実行結果は「数値ではない」になります。

ctype.h には isdigit 以外にも下記のような関数が用意されています。

関数名	説明
isalnum	引数がアルファベットか数字ならば真
isalpha	引数がアルファベットなら真
islower	引数が小文字なら真
isupper	引数が大文字なら真
isspace	引数がタブ、スペースなら真