

[2] データの型と演算

- ・プログラムでデータの計算 (演算) をする場合には, 変数と演算子を用いる
- 2つの値の和を表示するプログラム例

```
#include <stdio.h>
int main(void)
{
    int a,b,wa;           変数を準備する.
    a=10;                変数aに10を代入する.
    b=3;                 変数bに3を代入する.
    wa=a+b;              変数waにaとbの和を代入する.
    printf("%d ¥n",wa);  変数waの値を表示する.
    return 0;
}
```

1. 定数と変数

1.1 定数

- ・ 整定数 0, 10, 500, 9424, -10
- ・ 実定数 0.0, 12.3, -0.02
- ・ 浮動小数点定数 1.23e2 (1.23×10^2), 1.2e-2 (1.2×10^{-2})
- ・ 文字定数 ('で1文字をかこむ) 'A', '1'. 'AA'や'123'とはできない.
- ・ 文字列定数 ("で文字列をかこむ) "ABC", "123", "1"

1.2 データの型

型	サイズ	説明
char	1	文字型. (文字 1 文字を記憶する)
int	4	整数型
float	4	単精度実数型
double	8	倍精度実数型

1.3 変数と変数型宣言

★プログラムの最初で, どのような変数を使うかを示す (変数型宣言, 変数定義)

- ・ 変数名は, 英文字か_で始まり, A-Z, a-z, 0~9, および_が使用できる.
- ・ 予約語にあるものは使用できない. (void など)

型 変数名 ;

```
int a, sum;           整数型変数 aとsumを宣言する.
double x, y;         実数型変数 xとyを宣言する.
```

2. 演算

2.1 算術演算子 +, -, *, /, %

- ・演算の優先順序は、数学と同じ。 * , / , % > + , - . () 内を優先する.

★かけ算の"*"は省略できない.

★定数の型に注意する.

3/2と3.0/2.0は違う. 3/2 → 1 (整数演算), 3.0/2.0 → 1.5 (実数演算).

2.2 代入演算子 =

- ・左辺の変数に右辺の定数や変数の値, 計算結果を代入すること.

★代入式は等式ではない.

a=1.0;	変数aに1.0を代入する.
day=year*365+1;	変数yearに365を掛け, 1を足した結果をdayに代入.
n=n+1;	変数nの値に1を足し, その結果を変数nに代入. 結果としてnの値が1増える.

2.3 変数の初期化

- ・変数宣言した直後の変数の値は不定である (何が入っているかわからない).

int a, b;	整数型変数a, bを宣言.
a=b+10;	bの内容に10を足した結果をaに代入する. aの値は?

- ・変数に値を代入することを初期化するという.

int a;	int a=2; このように書くこともできる.
a=2;	a=a+10;
a=a+10;	この場合は, aの値は12になる.