

[5] いろいろな演算

1. 数値演算関数

- ・一般的な数学関数については、すでに関数定義されているので作る必要はない。
- ・以下のような関数を使用する場合には、プログラムの最初に `math.h` を追加する。

```
#include <math.h>      プログラムの最初に書く
```

- ・比較的良く使う関数. (xは倍精度実数. kは整数を表す) (テキストp.211参照)

| | | |
|-----------------------|------------------------------------|-------------|
| <code>sin(x)</code> | <code>sin(x)</code> . | (x はラジアン単位) |
| <code>cos(x)</code> | <code>cos(x)</code> . | " |
| <code>tan(x)</code> | <code>tan(x)</code> . | " |
| <code>atan(x)</code> | <code>tan⁻¹(x)</code> . | (結果はラジアン単位) |
| <code>abs(k)</code> | kの絶対値. (kは整数) | |
| <code>fabs(x)</code> | xの絶対値. (xは実数) | |
| <code>exp(x)</code> | e^x . 指数関数. | |
| <code>log(x)</code> | xの自然対数. | |
| <code>pow(x,y)</code> | x^y . べき乗 | |
| <code>sqrt(x)</code> | xの平方根. | |

2. C言語特有の演算子

- ・インクリメント演算子 `++`, デクリメント演算子 `--`

`a++`; 変数aの値を1ふやす. $a=a+1$;と同じ.

`b--`; 変数bの値を1へらす. $b=b-1$;と同じ.

- ・ `+=`, `-=`, `*=`, `/=`演算子

`a += 5`; $a=a+5$;

`b -= 2.0`; $b=b-2.0$;

`c *= 1.5`; $c=c*1.5$;

`d /= 0.5`; $d=d/0.5$;

- ・前置表記 (演算) と後置表記 (演算) `++`, `--`

`a++`; 後置表記 (演算).

`++b`; 前置表記 (演算).

後置表記では`++`が後に実行され, 前置表記では`++`が先に実行される.

`a=2`;

`x=a++`; xにaを代入した後aをインクリメント. xは2, aは3.

`y=++a`; aをインクリメントした後, yに代入. yもaも4.

3. 演算の優先順序

- ・演算子の優先順序を正しく理解すること. (テキストp.140参照)

★プログラムを書くだけでなく, プログラムを見てどのような結果になるかを, 頭の中で判断できるようになること, すなわち「プログラムを読めるようになる」ことが大事です.