

[8] 式の値

- c言語におけるいろいろな「式」.

代入式

```
x=10;
```

```
y=sin(3.14/6.0);
```

条件式 (関係演算子を用いた式)

```
x < 10
```

論理演算式

```
x && y
```

```
(x > 10) || (y < 0)
```

- c言語において「式」は値をもつ.
- 代入式は、代入した値と同じ値をもつ.
- 条件式, 論理演算式は,
成り立たない場合 0,
成り立つ場合 1 となる.

※if文やwhile文などにおいて, '式'が0なら成り立たない, 0以外の場合は成り立つと判断している.

○式の値の性質を利用すると, 以下のようなプログラムが書ける.

★複数の変数を同じ値に初期化する.

```
a=b=c=d=5;
```

- 式は, 右の項から順に処理される.
- d=5という代入式は 5 という値をもつので, cには5が代入される.
- したがって, すべての変数の値は 5 になる.

★if文を使わないで, 条件分けした計算を行う.

年齢(year)が 12才以上は料金(ryokin) 500円, それ以下は300円とする.

```
ryokin=(year >= 12)*500+(year < 12)*300;
```

[9] 制御構造 (3)

1. break文

- while文, do~while文, for文のループからの強制的な脱出に用いる.

例) int a,i;

```
for(i=1;i<100;i++){                -9999が入力されるとforループが終了する.
    scanf("%d",&a);
    printf("%d ¥n",a/2);
    if(a == -9999)
        break;
}
```

2. データの終わりの判定

- データの入力においてデータの個数がわかっている場合 (for文を使った例)

```
for(i=1;i<=5;i++){
    scanf("%d",&a);                // scanfを5回繰り返す.
    sum=sum+a;
}
```

- 入力するデータの個数がわかっていない場合には？

- (1) ある決めた値を入れると終了する. (例えば-9999を入力すると終了する)
- (2) ^zを入力する ([ctrl]と[z]を同時に押す). ^zは入力データの終わりを示す.

- scanf関数は、「^zが入力されるとEOFという値を返す」ようになっている.

```
int dat,sum=0;
while(scanf("%d",&dat)!=EOF)    scanfでdatに値を入力. scanf関数の値を比較.
    sum=sum+dat;                ^zでない場合(EOFでない)whileループをくり返す.
printf("Total= %d\n",sum);     ^zが入力される (EOFとなる) とwhileループが
                                終了する.
```

3. カウンタ

- ・ 上のようなプログラムで、「入力した回数」を知りたい場合、「カウンタ」を利用する。

```
int dat, sum=0, n=0;
while(scanf("%d",&dat)!=EOF){
    sum=sum+dat;
    n++;    /* n=n+1; でもよい */
}
printf("データの個数  %d\n",n);
printf("データの合計  %d\n",sum);
```

- ・ このプログラムの 変数nを「カウンタ (変数)」と呼ぶ。ある処理の実行毎に、変数の値を1増やすことで、その処理を行った回数を知る事ができる。
- ・ 専用の変数があるわけではなく、一般的には整数変数を用いる。

4. 多重ループ

- ・ for文やwhile文の繰り返し (ループ) では、ループの中にさらにループをいれた多重ループを作る事ができる。これをネスト (入れ子構造) ともいう。

5. 無限ループ (テキストP135)

- ・ 以下のfor文, およびwhile文は終わらないループ。終了するにはbreak文を使う。

```
for( ; ; )
```

終了する条件がない。

```
while(1)
```

条件が常に成立 (1) している。