変数(variable)

変数っていうテーマで今回行います．変数って名前，いかにも数学的ですが，変数があればもちろん定数もあることだけ覚えておいてもらえれば…

変数とは，データを入れる箱であるとよく言われます．C/C++言語では実際そのものぴったりで，本当に箱です．

次のコード例を見てください．

```
#include <stdio.h>

int main(void) {
  int i = 10;
  printf("%d¥n", i);
  return 0;
}
```

int っていうのは変数の型です．またあれな例なんですが，「コンピュータは 01 で動いてる!」っていうのを実はここで感じることができたりしちゃったりします．データは蓋を開けたらほんとに 01 しかなくて，コッチ側はある一定の決まりをもったある一定の大きさのデータをひとかたまりにして，決まり通りに扱うことで，さも様々なデータが存在するかのように扱うわけです．

ここでは，integer，つまり整数のデータの入れ物を宣言しています．宣言とは，「今から使うよ!」とコンパイラに向かってまさに「宣言」しているわけです．そして，このデータは整数として扱うよというのもコンパイラに伝えています．

こうして確保した領域に，10 という値を代入しています．これより int 型の変数 i の中には 10 というデータが入っているわけです．それを printf で%d(decimal)で表示します．

また，ある関数で使う変数は，関数のはじめで宣言しなければいけません．

```
#include <stdio.h>

int main(void) {
  printf("%d¥n", 10);
  int i = 10;
  printf("%d¥n", i);
  return 0;
```

```
}
```

はダメだということです. これは現在の gcc ではまあこれなのですが, C99 の仕様的には途中宣言可能になっています. ただ, オプションが必要なので, この場合は移植性も考えこの形式で行きます.

変数名(identifier)とキーワード(keyword)

今さっきの例では int 型変数 i としました. この i, 数学とかでは a-z やギリシャ文字を使って変数を表したりしますが, C ではある程度好きな長さが使えます. 昔のものだと 31 文字までとかありますが…

```
#include <stdio.h>

int main(void) {
  int this_is_integer = 10;
  return 0;
}
```

みたいに. 変数の名前を識別子(identifier)といいます. これは豆知識的なので覚える必要はほぼないです.

これには使えないものがあって, 数字及び英文字(大文字小文字区別する)と_(アンダースコア)が使えます.

```
this_is_integer
array2pointer
ThisIsInteger
_KOREMO_daijoubu
```

みたいな感じです. しかし, 先頭文字は数字を使えません.

```
02_is_integer_janai
```

はダメと言うことです. これは, 簡単に言えば数字のリテラルと区別がつきにくいからで, コンパイラの都合上丁度いいからです. コンパイラとしては, 最初の一文字を見た時点で identifier か数値リテラルか区別したいのでこのようになっています.

また, 変数に int なんて名前をつけられると困ってしまいます. そこで, コンパイラが特別扱いする名前のことをキーワードといい, これも変数名には使えなくなっています.

型(type)

int 以外にもたくさんの型があります．ここでは特に使うであろう物を…

型名	大きさ	説明
int	4	整数
long	4	大きな整数
short	2	小さな整数
char	1	文字
float	4	浮動少数点数
double	8	倍精度浮動少数点数

C 言語の特徴であり，そして強みであり，またハマリどころでもあるのですが，これらの型というのは，コンパイラの決めた決まりごとに従って私たちが扱うデータであり，実態は結局 01 です．

それを今から表示することができます．以下の program を作って実行してみてください．

```
#include <stdio.h>

int main(void) {
  char ch = 'a';
  printf("%c = %d¥n", ch, ch);
  return 0;
}
```

'a'はシングルクォーテーションに注意です．

```
a = 97
```

と出たら大丈夫です．%c は文字，%d は数値として引数を考え，それを印字するのでした．

char ch = 'a';で ch の中には文字 a の情報が入ります．そしてもちろん，これは 01 のデータです．これを私たちが「文字だ!」と考えたら a で，この同じ 01 のデータを「数字だ!」と考えたら 97 だったということです．

01 の中で文字を表すには文字に数字を割り当てて良くやり方を取ります．これが俗に言う文字コードというものです．

この決まりの中で，文字 a は数字 97 で表されていたということです．

数値 97 と文字 a のデータは大きさ以外は同じデータです. このように同じデータでも, こちらの受け取る決まり次第で全く違った意味に取れます.

今はあんまりわかんなくても大丈夫です. ただ, 結局はデータであるということ, 型は私たちがこのデータをどういうふうに考えているかをコンパイラに伝えているのだということが伝われば.

算術演算

おなじみの演算子を使って、変数、または定数の計算をします。

演算子	効果
+	足す
-	引く
*	掛ける
/	割る
%	余り

```
#include <stdio.h>
int main(void) {
  int a = 10;
  int b = 20;
  int c, d;

  c = a * b;
  d = 0.1 * 10;

  printf("a  *  b = %d¥n", c);
  printf("0.1 * 10 = %f¥n", d);

  return 0;
}
```

0.1 * 10 の計算の format 指定子に%f を使っています. これは浮動点少数を表示するものです. さて, 0.1 は float ですが, 10 は int ですね.

このとき, int と float を掛けるわけですが, 一般に float の方が表現できる値が広いです. このため, この広い方に片方が変換されてから計算されます.

このとき, 10 は一度 float に変換されて, それから掛け, その結果は float になります. これを通常の算術型変換といいます. 型 int から float に変わっているわけです.

入力(input)

scanf をやります. これは入力を scan し, format どおりに解釈するという関数です.

```
#include <stdio.h>

int main(void) {
  int d;
  scanf("%d", &d);
  printf("%d¥n", d);
  return 0;
}
```

まず, 変数を宣言しています. こうしてこれから取ってくる値のための箱を作っておく必要があるわけです.

scanf の第一引数は format 文字列です. これは printf のものと酷似していますが, ちょこちょこ違うので注意してください.

この場合は入力を 10 進の数値とみなして, その値と解釈できる値を d に代入しています.

注意してもらいたいのは, d の前に**&の記号**のついているところです. K&R でも

非常に多いエラーは scanf("%d", &n); と書かずに, 次のような書き方をすることである。 scanf("%d", n); このエラーはコンパイル時には通常検出されない。

と注意書きがされています. &の記号はアドレス演算子と呼ばれるもので, これの解説はかなり後になりますが, データ実態へのアドレスをデータのポインタ型として返すという今の時点では非常に意味不明な解説しかできません.

符号の有無(signed, unsigned)

計算機基礎などの授業を受けた方はわかることですが，受けてない方にもわかるように説明します．

int は 4byte，つまり 0/1 が 8*4 の 32 こで出来ています．これで表せるパターンは 2 の 32 乗ですね．

もしあなたが負の数も考えたいとします．すると，その数が+であるか−であるかを表すために 1bit 欲しいですね．この bit を一般には左端の bit をとり，これを MSB(Most Significant Bit)といいます．

で，それを考えると，数字に使えるのは 31bit であるということになります．この範囲によって，int の表せる数字の範囲が分かります．

- 2 の 31 乗 => 2 の 31 乗-1 までです．-1 なのは 0 があるからです．

しかし，もしあなたが負数を考える必要がなければ，0 から 2 の 32 乗-1 までで+の数を 2 倍大きな数まで考えることができます．

C 言語は非常に primitive なので，これをこちら側で選べます．それが signed or unsigned です．

ちなみに，通常 int とかくと，singed int のことです．

```
int main(void) {
  // 符号なし(unsigned)の整数(int)
  unsigned int ui = 1000000;
  // 符号あり(signed)の整数(int)
  signed int si = -1000000;
  return 0;
}
```

このように選ぶことができます．