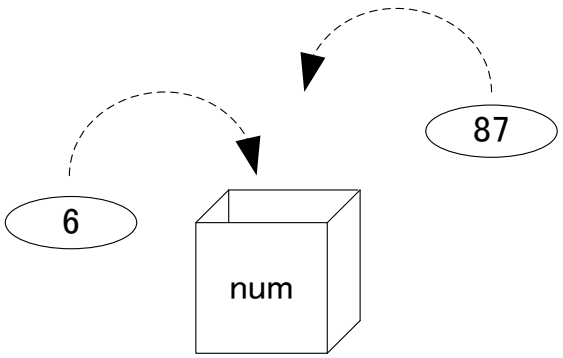
変数、入出力処理、演算子

変数の宣言



使用例
int num; num = 100;

使用できる変数名

規則
（1）最初の文字は「英字」か「下線 (_)」を使う
（2）2文字目以降は「英字」、「数字」、「下線」を使う
（3）英字の大文字と小文字は区別される
（4）表1に示す予約語（予約済み識別子）は使えない
（5）変数名の長さに制限はないが、最初の31文字で区別される。

表1　予約語

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void	volatile	while

※変数名の途中に英字、数字、下線以外があってはならず、空白も認められない。予約語そのものはだめだが、変数名の一部に使うことはできる（例えば、a_forなど）。

変数型の種類

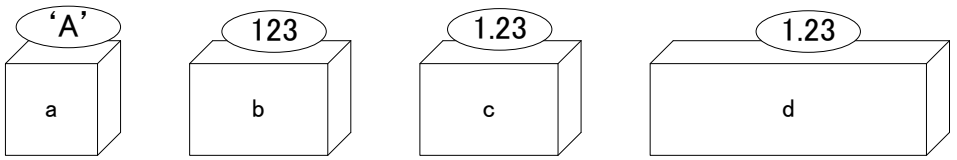
表2　基本型

型名	説明	一般的な使用ビット数
char	文字型（小さな整数）	8ビット
int	整数型	32
float	浮動小数点型	32
double	倍精度浮動小数点型	64

備考：

基本型の説明に示した「浮動小数点」とは数値を指数表現で表す方法である。例えば0.001は指数表現で書くと1.0×10^{-3} と表すことができる。これをC言語では1.0E-3というように表す。この1.0を仮数部、-3を指数部と呼ぶ。コンピュータではこの仮数部と指数部の値を変数の領域に分割して格納している。

```
// 変数の宣言
char a;    // 文字型
int b;     // 整数型
float c;   // 浮動小数点型
double d;  // 倍精度浮動小数点型
```



それぞれ用意される箱の大きさが異なる

変数の宣言方法と使用例

例1 文字型	char c; c = 'a';
例2 整数型	int i; i = 3;

例3
実数型
	float a;
	double b;
	
	a = 1.0;
	b = 3.0;

例4	float f, g;
	
	f = 3.0;
	g = f/4.0;

例5 初期化	int i = 3;

標準出力(printf)と変数型

変数方と変換仕様

%c	文字（char型）
%d	整数（int型）
%f	実数（float型）
%lf	より精度の高い実数（double型）

注意：
printf文の場合、double型の変換仕様%lfは%fとしても問題なく表示される。
ただし、scanf文ではdouble型の変換仕様は%lfだけが利用できる。%fを用いると入力に失敗する。

基本

例1	char t = 'a'; printf("%c", t);
例2	int i = 3; printf("%d", i);
例3	printf("%d %f", 3+4, 3.0/4.0);

例4	int i = 3;
	float f = 3.0;
	double d;
	d = f/4.0;
	printf("%d %f %lf", i, f, d);

応用(桁数調整)

例	float f = 3.567;
	
	printf("%.2f", f);

出力	3.57

例	float f = 13.2;
	
	printf("%7.3f", f);

出力	13.200

標準入力(scanf)と変数型

例1	int i; scanf(“%d”, &i);
例2	int i; float f; double d; scanf(″%d　%f　%lf″, &i, &f, &d);

※"%〇 … %〇"の中には「¥n」を書かないで使用すること

演算子

a = b + 10; において，「a」、「b」、「10」のように演算の対象となるものを「オペランド」という。また、「=」や「+」は「演算子」と呼ばれ、「=」は特に「代入演算子」、「+」は「算術演算子」というものに分類される。以下に算術演算子と代入演算子を示す。ただし、代入演算子はここに示したもの以外にもある。

算術演算子	
+	加算
−	減算
*	乗算
/	除算
%	余り（ただし整数での演算に限る）

代入演算子	
=	代入
+=	加算したものを代入する
−=	減算したものを代入する
*=	乗算したものを代入する
/=	除算したものを代入する
%=	剰余を求め、代入する

※　例えば、「a += b;」は「a = a + b;」と同じ処理となる。

演算子の優先順位

1	( )
2	*,/,%
3	+,−
4	=, +=, −=, *=, /=, %=