
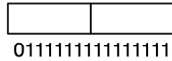
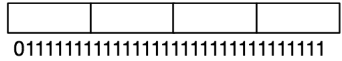


■ データのサイズ

★教科書 p.151, 表 9.3 参照

変数にせよ定数にせよ、C 言語が扱う値には型があり、型によって利用するメモリの量が異なります。

型 (bit幅)	正の最大値	バイト及びビットイメージ
char (8bit)	127	 01111111
short (16bit)	32767	 0111111111111111
long (32bit)	2147483647	 01111111111111111111111111111111

メモリ量の違いはそのまま表現できる値の幅に直結します。教科書の表 9.3 あるいは右図にあるように、例えば変数は割り当てられたデータの量によって表現できる値の限界が決まります。文字も内部的には数値で表現するため、数値同様の大小比較ができます。★教科書 p.154 参照。

■ 文字コード

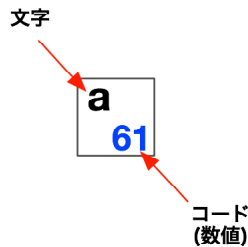
★教科書 p.150, 9.2.1 参照

押さえて欲しいポイント：

- ・メモリには値が入る（値しか入らない）
- ・文字は数値として入れる（そのメモリに文字があるとしたら、この値を「a」だとして）
- ・これをコードと呼ぶ
- ・文字と値との割り付け（マップ）表が必要（コード表）
- ・コードの体系は複数ある（参考：アスキー文字コード表）

ASCII 文字コード表

表の見方



ASCII文字は1バイトで表現されているが、その実体は数値である。つまり 'a' は番号 61 の文字。61 は 16 進数表記なので、10進数で表記すると 97 番文字となる。

- ・20番(16進)の sp は空白文字。
- ・\t はタブ。
- ・\n は改行文字。
- ・\0 はいわゆるヌル文字。(文字列の終端記号)

\0		sp	0	@	P	`	p
00	10	20	30	40	50	60	70
	!	1	A	Q	a	q	
01	11	21	31	41	51	61	71
	"	2	B	R	b	r	
02	12	22	32	42	52	62	72
	#	3	C	S	c	s	
03	13	23	33	43	53	63	73
	\$	4	D	T	d	t	
04	14	24	34	44	54	64	74
	%	5	E	U	e	u	
05	15	25	35	45	55	65	75
	&	6	F	V	f	v	
06	16	26	36	46	56	66	76
	'	7	G	W	g	w	
07	17	27	37	47	57	67	77
	(8	H	X	h	x	
08	18	28	38	48	58	68	78
\t)	9	I	Y	i	y	
09	19	29	39	49	59	69	79
\n	*	:	J	Z	j	z	
0a	1a	2a	3a	4a	5a	6a	7a
	+	;	K	[k	{	
0b	1b	2b	3b	4b	5b	6b	7b
	,	<	L	\	l		
0c	1c	2c	3c	4c	5c	6c	7c
	-	=	M]	m	}	
0d	1d	2d	3d	4d	5d	6d	7d
	.	>	N	^	n	~	
0e	1e	2e	3e	4e	5e	6e	7e
	/	?	O	_	o		
0f	1f	2f	3f	4f	5f	6f	7f

□ 参考：sizeof(), %x

sizeof 関数は引数に与えた変数やデータのバイト数を答えます。

printf 関数の変換文字として %x があります。データの中身を 16 進数で出力します。ともに変数の内部表現（コンピュータ内部でどのように実現されているか）を調べるのに便利です。

文字は 1 バイトで、A は 16 進数で 41 である（右表参照）。int は 4 バイトで 100 は 16 進数で 64 (6*16 + 4 = 100) である。

```

プログラム：
char a='A';
int i=100;
printf("%ld - %x\n", sizeof(a), a);
printf("%ld - %x\n", sizeof(i), i);

結果：
1 - 41
4 - 64
    
```

■ グラフィクス画面への文字の描画

右のサンプルプログラムはグラフィクス画面へ文字を描画するものです。

drawstr 関数の引数は左から

- win: 描画先 (いつもの)
- 100.0, 200.0 : 座標位置
- 16: 文字サイズ (高さ)
- 0.0: 常にゼロ
- string: 描画する文字列

です。詳しい仕様は EGGX のマニュアルを参照してください。

```
char string[100];
int win;

printf("input string >> ");
scanf("%s", string);

win=gopen(400,400); /* 描画ウィンドウを開く */
winname(win, "character"); /* 名前をつける */

/* (100,200) の位置に 16 のサイズで文字列を描画 */
drawstr(win, 100.0, 200.0, 16, 0.0, string);

ggetch(); /* キー入力进行待つ */
gclose(win); /* 描画ウィンドウを閉じる */
```

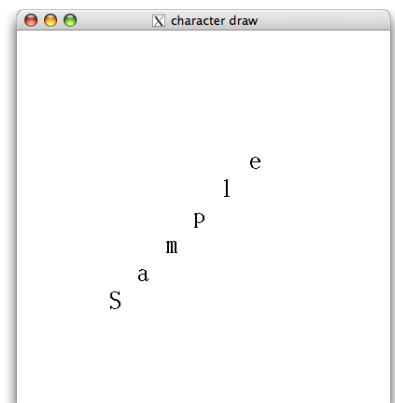
文字サイズは 1~24 で指定できますが、実際には以下の数種類の中から選ぶと良いでしょう。
(例えば 16 をサイズ指定すると、横 8 ドット、縦 16 ドットの文字が描画される)

指定	サイズ	指定	サイズ	指定	サイズ	指定	サイズ	指定	サイズ
7	5 x 7	8	5 x 8	9	6 x 9	10	6 x 10	12	6 x 12
13	7 x 13	14	7 x 14	16	8 x 16	20	10 x 20	24	12 x 24

□ 課題 1.

入力した文字を一文字ずつに分解して、斜めに描画するプログラムを作ってください。

(drawstr は文字型のデータを描画する機能がなく、文字列 (文字配列) であることが必要です。文字列 (文字配列) から一文字だけ取り出し、その文字だけの文字列 (長さ 2 の文字配列) を作って drawstr に与えると良いでしょう。)



□ 課題 2.

入力した文字列を使って、何かアニメーションを作ってください。例えば文字が徐々に上から落ちてくる、というのはどうでしょう。

