

初級C言語講座Vol.3

字下げ

- 見やすく書こう！

```
#include <stdio.h>
```

```
int main(void)
{
    int a;
    printf("値=>");
    scanf("%d", &a);
    if(a>10)
    {
        printf("10より大きい値です¥n");
    }
    else
    {
        printf("10以下の値です¥n");
    }
    return 0;
}
```

```
#include <stdio.h>
```

```
int main(void)
{
    int a;
    printf("値=>");
    scanf("%d", &a);
    if(a>10)
    {
        printf("10より大きい値です¥n");
    }
    else
    {
        printf("10以下の値です¥n");
    }
    return 0;
}
```

別種の表記

<code>i = i + 1</code>	<code>i ++</code>	<code>++ i</code>
<code>i = i - 1</code>	<code>i --</code>	<code>-- i</code>
<code>i = i + 2</code>	<code>i += 2</code>	
<code>i = i - 2</code>	<code>i -= 2</code>	
<code>i = i * 2</code>	<code>i *= 2</code>	
<code>i = i / 2</code>	<code>i /= 2</code>	

```
printf("同じです");  
puts("同じです");  
putchar('A');
```

```
scanf("%s");  
gets();  
getchar('A');
```

一部まだ教えていないことがあるので、
下の入力編は参考程度に。

++iとi++の違い

- 下記のプログラムの、「++i」を「i++」に書き換えて試してみよう。

```
#include <stdio.h>

int main()
{
    int i;
    i=0;
    if(++i == 0)
    {
        printf("評価時点で\n");
    }
    else
    {
        printf("評価時点で以外\n");
    }
    return 0;
}
```

キャスト

- 強制的に式の型を変換する方法。

```
int a, b;  
float c, d;
```

```
a=10; b=3;
```

```
c=a/b;
```

```
d=(float)a/b; //変換したい場所の前に、括弧で型名を囲って記述する。
```

配列

- 多数の変数を連続して扱う

```
#include <stdio.h>
```

```
int main(void)  
{
```

```
    int a[10], i;
```

```
    for (i=0; i<10; i++)  
    {
```

```
        printf("a[%d]=? ¥n=>", i);  
        scanf("%d", &a[i]);
```

```
    }
```

```
    for (i=0; i<10; i++)  
    {
```

```
        printf("a[%d]=%d¥n", i, a[i]);
```

```
    }
```

```
    return 0;
```

```
}
```

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]

文字

- C言語では、文字はそれぞれに番号を振り、その番号を変数で保持することで、文字を表している。

たとえば大文字のA

これは、「A」とシングルクォートで囲って書くことで、割り振られた番号を示すことができる。

これをchar型の変数で保持することで、いわゆる文字となる。

```
#include <stdio.h>
```

```
int main()  
{  
    printf("Aの番号は%d。16進数だと%X\n", 'A', 'A');  
    return 0;  
}
```

文字コード

上位3ビット→ ↓下位4ビット	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAC	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(8	H	X	h	x
9	HT	EM)	9	I	Y	i	y
A	LF/NL	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

一文字だけ入力

- 次の命令、getchar()を利用することで、一文字だけ入力することができる。
- ほかに、scanfの場合は、%dではなく%cとすると、一文字だけ入力できる。

```
#include <stdio.h>

int main()
{
    char moji1, moji2;
    moji1=getchar();           //一文字だけ入力
    scanf("%c", &moji2);      //同上
    printf("%c %c\n", moji1, moji2); //出力も%c
    return 0;
}
```

文字列

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]
c	h	i	b	a	-	u	\0

↓

chiba-u

端末文字
(バックスラッシュまたは円マーク)

```
char a[8]="chiba-u";
```

```
#include <stdio.h>
```

```
int main()  
{
```

```
    char str[100];  
    printf("文字を入力=>");  
    scanf("%s", &str[0]);  
    //scanf("%s", str);  
    printf("%s\n", str);  
    return 0;
```

```
}
```

//と書いてもよい。
//上と同じ。

文字列操作関数

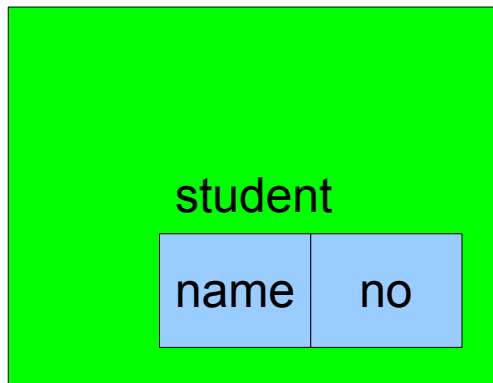
関数名	使い方	解説
strcpy	strcpy(文字列1, 文字列2);	文字列1に対し、文字列2をコピーする。
strncpy	strncpy(文字列1, 文字列2, 文字数);	文字列1に対し、文字列2を文字数分コピーする。
strcat	strcat(文字列1, 文字列2,);	文字列1の後ろに、文字列2をコピーする。
strncat	strncat(文字列1, 文字列2, 文字数);	文字列1の後ろに、文字列2を文字数分コピーする。
strcmp	返回值=strcmp(文字列1, 文字列2);	文字列1と文字列2を比較し、 文字列1>文字列2 なら正の値 文字列1=文字列2 なら0 を返す。
strlen	返回值=strlen(文字列);	与えられた文字列の終端文字を含まない長さを返す

記述例

```
char ccsName, str[100]="CCS";  
int flg;  
. . .  
strcpy(ccsName, str);  
flg=strcmp(ccsName, "ccs");
```

構造体

- 複数の変数をまとめて扱う



```
struct student{  
    char name[30];  
    int no;  
};
```

```
#include <stdio.h>  
  
struct student{  
    char name[30];  
    int no;  
};  
  
int main(void)  
{  
    struct student data[10];  
    int i;  
  
    for (i=0; i<10; i++)  
    {  
        data[i].no=i+1;  
        printf("名前を入力¥n=>");  
        scanf("%s", &(data[i].name[0]));  
    }  
    for (i=0; i<10; i++)  
    {  
        printf("No:%d¥n名前:%s¥n", data[i].no, data[i].name);  
    }  
    return 0;  
}
```

変数の初期値

- 変数を宣言した瞬間、初期値を入れる事ができる。

```
int a=0; //=で書いてやる。
int data[5]={10, 20, 12, 43, 165}, //このように、配列も初期値を入れられる。
    data2[]={10, 20, 12, 43, 165}; //要素数をしていないと、適切な要素数を勝手に確保する！
char name[]="CISC"; //文字列もこうやって初期設定可能。
struct student{
    char name[100];
    int age;
} ccsmember1={"CISC", 19}; //構造体もこうやる。
```

課題1

- n個の値x ($0 < n \leq 10, 0 < x < 10000$)を配列に入力し、その合計、平均、最大値、最小値を求めるプログラムを作れ。
 - 0が入力されたり、10個を超えたら、入力工程を終え計算する。
 - 入力のループと、計算のループは別にする事。
 - 最大値最小値を求めるアルゴリズムは、各自検討せよ。ただし、概要・ヒントは次のとおり。
 - 最大値(最小値)を保持する変数を設定。
 - この変数と調べたい変数を比較し、最大値(最小値)の候補になりうるかを調べる。
(最大値(最小値)の変数の初期値はどうすればよい?)
- さらに、入力された値を昇順に整列して出力せよ。
(任意課題)