

情報研究会 CACTUS

第一回 C言語講習

今回の内容

- 変数
- 標準入出力
- 演算子
- if文

サンプルプログラム

キーボードから入力
された整数を表示す
るプログラム

ソースコード

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int n;
6
7     printf("整数を入力してください:");
8     scanf("%d", &n);
9
10    printf("あなたが入力した値は%dです\n", n);
11
12    return 0;
13 }
```

実行結果

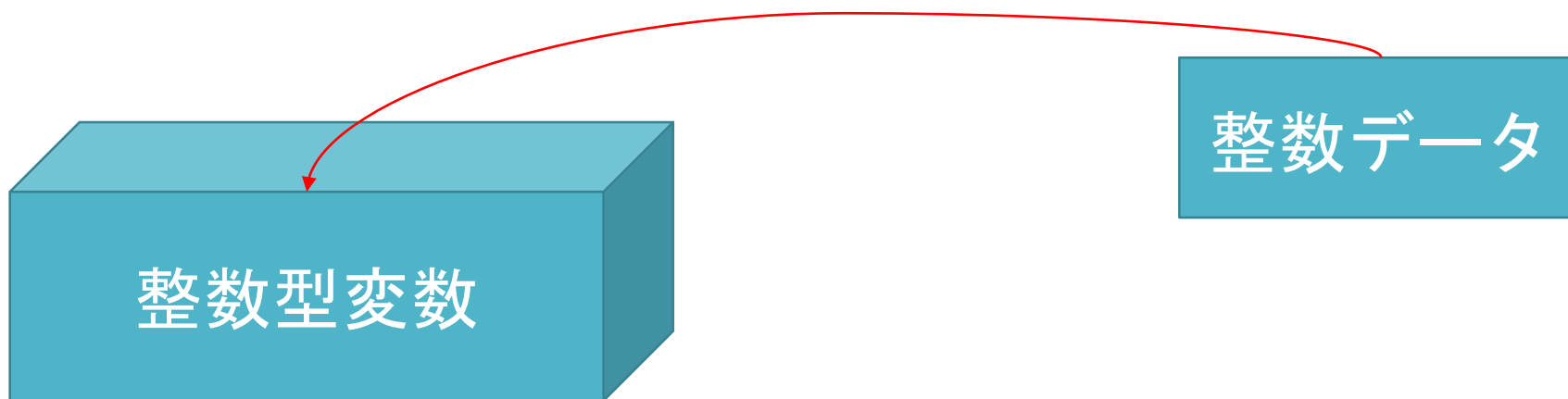
```
整数を入力してください:10
あなたが入力した値は10です
```

int main(void) とは何か

- ・ プログラムの大本を記載する場所（main関数）
- ・ { }内を上から順番に処理を行う（逐次処理）
- ・ return 0; はプログラムの終了を意味するようなもの

変数

- 変数とはデータを保存しておく箱のようなもの
- 箱の形(型)の種類によって保存できるデータは決められている



データ型の種類

名称	データ型	格納できる値	データサイズ
整数型	int	整数	4 バイト
浮動小数点型	float	実数	4 バイト
	double		8 バイト
文字型	char	文字	1 バイト

変数の宣言

- ・ 変数を使用するにはデータ型と変数名を宣言する必要がある
- ・ 変数名は自分の好きなように決められる

例) 整数型変数の宣言

```
int n ;
```

型名

変数名

標準入出力

`printf()` ... データを画面に表示する

`scanf()` ... キーボードから入力した値を変数に格納する

※ `printf()`, `scanf()` は `#include<stdio.h>` をソースコードの最初の部分に記述しないと使用できない

#include<stdio.h>とは何か

- `stdio.h`とはヘッダファイル（辞書みたいなもの）の一種
- `#include` と書くことで< >内のヘッダファイルを読み込み、ヘッダファイルの機能を使用できるようにする
- C言語ではほとんどの場合`stdio.h`をインクルードする

書式指定子

- ・ 入出力するデータの種別を指定するもの
- ・ 表示するデータが変数かどうかを識別するために必要

覚えておくべき書式指定子

書式指定子	データ型
%d	整数
%f	実数
%c	文字
%s	文字列

scanf()

```
scanf ( “%d” , &n) ;
```

- “ ” の中にキーボードから入力するデータ型に合わせた書式指定子を書く
- 入力した値を格納する変数をカンマの後に&をつけて書く

※現段階で&の意味を理解する必要はない

printf()

```
printf( “変数nに格納された値は%d¥n” , n);
```

- “ ” の中身が画面に表示される
- 変数に格納されているデータを表示するときは、“ ” の中に変数と同じ型の書式指定子を書き、カンマの後に変数名を書く
- ¥nを書くと前文を画面に表示した後に改行される

算術演算子

演算子	種類	例	意味
+	加算	$x + y$	和を求める
-	減算	$x - y$	差を求める
*	乗算	$x * y$	積を求める
/	除算	x / y	商を求める
%	剰余算	$x \% y$	剰余を求める

代入演算子 =

- 変数に値を代入するときに使用する
- 変数は左辺におく
- 数学のイコールと同義でないことに注意

例) int型変数 n に1と2の和を代入

```
n = 1 + 2 ;
```

初期化

- ・ 変数を宣言すると同時に値を代入すること
- ・ 変数宣言時に格納する値が決まっていない場合は0で初期化しておいたほうがよい

例) 0で初期化

```
int n = 0 ;
```

サンプルプログラム

キーボードから入力
された2つの整数の
大小を調べるプログ
ラム

ソースコード

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int n1, n2;
6
7     printf("整数を2つ入力してください\n");
8     scanf("%d%d", &n1, &n2);
9
10    if(n1>n2)
11    {
12        printf("%dは%dより大きいです\n", n1, n2);
13    }
14    else if(n1<n2)
15    {
16        printf("%dは%dより小さいです\n", n1, n2);
17    }
18    else
19    {
20        printf("2つの値は等しいです\n");
21    }
22
23    return 0;
24 }
```


if文

条件分岐をさせたい
ときに使用する文

```
if(条件1)
{
    条件1が成立するときに実行する処理
}
else if(条件2)
{
    条件2が成立するときに実行する処理
}
else
{
    どの条件も成立しないときに実行する
    処理
}
```

比較演算子

演算子	例	意味
>	$x > y$	xはyより大きい
>=	$x \geq y$	xはy以上
<	$x < y$	xはyより小さい
<=	$x \leq y$	xはy以下
==	$x == y$	xとyは等しい
!=	$x != y$	xとyは異なる

論理演算子

演算子	例	意味
& &	x && y	xが真かつyが真ならば真
	x y	xが真またはyが真ならば真
!	! n	nが真ならば偽、nが偽ならば真

- ・ 比較演算を組み合わせた条件式を定義するときに使用する

例) 変数 x が変数 y より大きいかつ変数 a と変数 b が等しいとき真

```
if (x > y && a == b)
```

条件式クイズ

$a = 1$, $b = 2$, $c = 3$ のとき以下の条件式の真偽を求めよ

1 `if(a < b && b < c)`

2 `if(a > b || b != c)`

3 `if(!(a == b))`

練習問題 1

キーボードから整数を2つ入力し、それらの和、差、積を表示するプログラム

例) 1と2を入力した場合

```
整数を2つ入力してください  
1 2  
和:3 差:-1 積:2
```

練習問題 2

キーボードから整数を入力し、それが3の倍数かチェックするプログラム

例) 12を入力した場合

```
整数を入力してください:12  
12は3の倍数です
```

7を入力した場合

```
整数を入力してください:7  
7は3の倍数ではありません
```

練習問題 3

西暦で年をキーボードから入力し、その年が閏年かどうかを確認するプログラム

(参考) 閏年判定方法

西暦が400で割り切れる年はうるう年である

400で割り切れない場合、西暦が100で割り切れる年はうるう年ではない

100で割り切れない場合、西暦が4で割り切れる年はうるう年である

4で割り切れない場合、うるう年ではない

例) 2016を入力した場合

西暦を入力:2016
2016年は閏年である