計算機プログラミング

第9回目

June 10, 2021

（10：30～12：00）

担当教員：Thi Thi Zin（ティティズイン）
<thithi@cc.miyazaki-u.ac.jp>

本日の内容

- **配列**
  - 一次元配列
  - 例題

変数

これまで変数は変数名のみによって区別されてきた。

変数が複数あった場合、「n 番目の変数」のような指定の方法はできない。

配列

同じような意味の変数を複数用意し、順番（番号）をつけて区別する。

レポート提出あり

提出〆切： 6月12日（土）20:00まで

- 配列 -

配列を使う

配列を利用したベクトル計算

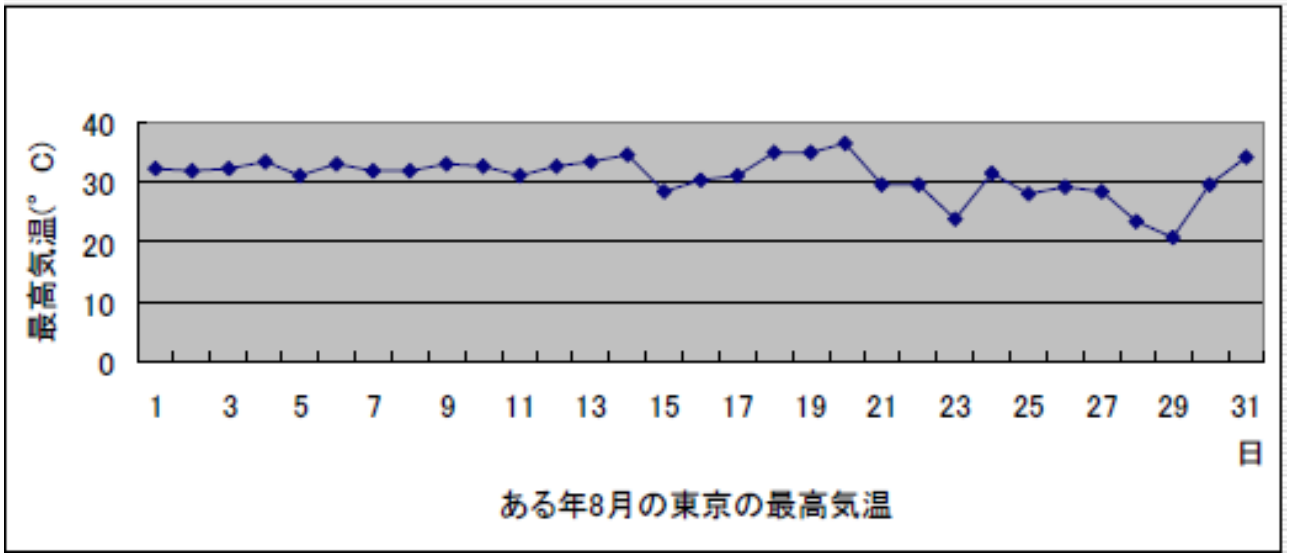
２次元配列

これまで変数は変数名のみによって区別されてきた。

変数ａ、変数ｂ、変数ｃとあった場合、「2番目の変数」のような指定の方法はできない。

データをまとめて保持しておきたい時にどうするか？

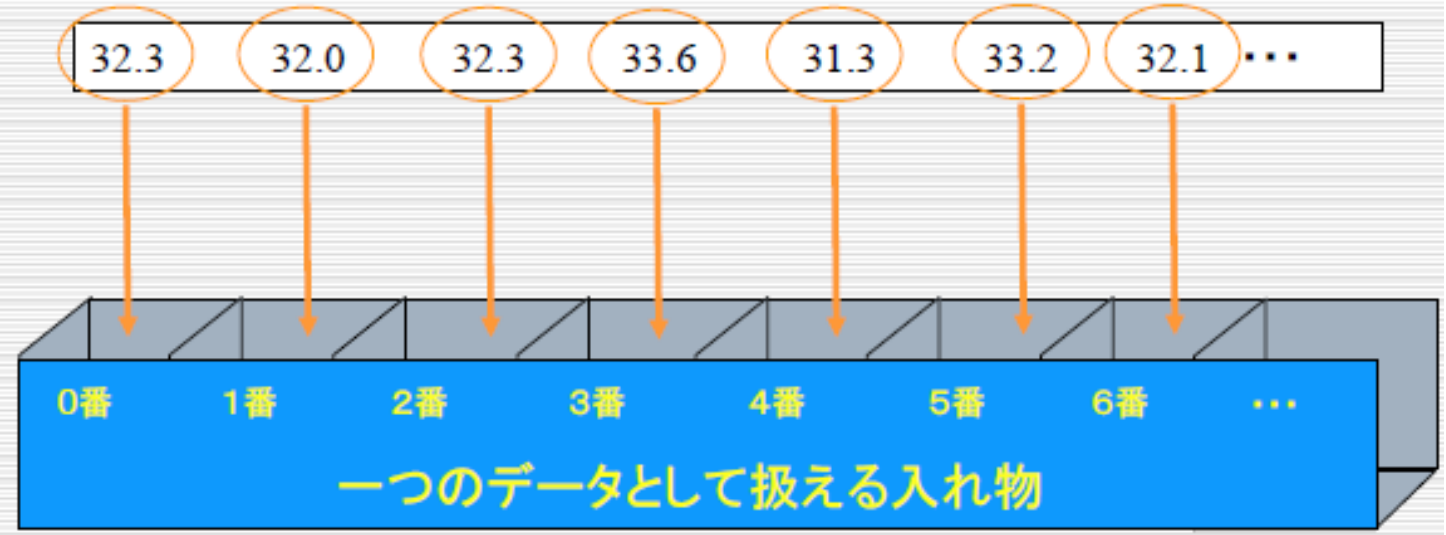
例えば、毎日の気温のデータをどうやってプログラムの中で扱えばいいでしょうか？



- 配列 -

データをまとめて扱いたい

32.3 32.0 32.3 33.6 31.3 33.2 32.1 31.9 33.0 32.6 31.1 32.6 33.5 34.7 28.5
30.3 31.0 35.1 34.9 36.5 29.8 29.8 23.8 31.7 27.9 29.2 28.3 23.4 20.9 29.7
34.4



- 全体でひとつのデータとして扱える
- そこから、個々のデータが取り出せる

- 配列の宣言 - 同じ型の変数をまとめて宣言

型　配列名 [要素数] ;

例 :　int　tana [5] ;

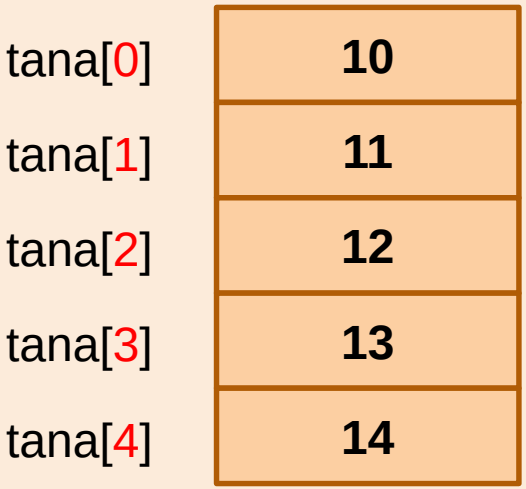
tana [1] = 10 ;

加算
tana [0]+ tana [1]

データが複数(5個の例）の場合 ⇨ 同じような意味の変数を複数用意し、順番（番号）をつけて区別する。

-　初期化　-

```
int   tana [5] = {10,11,12,13,14};
```



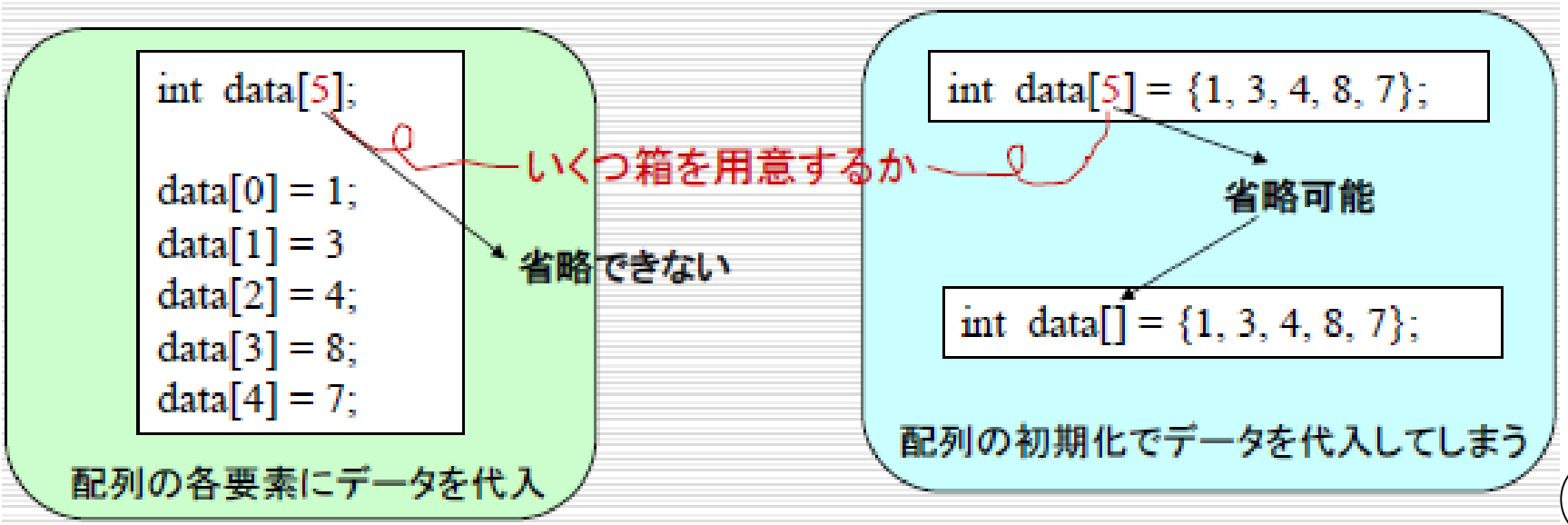
tana [0] から tana [4] までしか存在しない
tana [-1] や tana [5] は×

配列を使う

- 複数のデータをまとめて扱える入れ物があれば便利
- 数字のベクトルや行列と同じ考え方

1,3,4,8,7の5つのデータをint型の配列に入れる例

（普通の変数と同様に、配列であることを宣言する必要がある）



sample1

```
#include<stdio.h>
int main(void)
{
    int  tana_1 = 10;
    int  tana_2 = 11;
    int  tana_3 = 12;
    int  tana_4 = 13;
    int  tana_5 = 14;

    printf(“%d ¥n”, tana_1);
    printf(“%d ¥n”, tana_2);
    printf(“%d ¥n”, tana_3);
    printf(“%d ¥n”, tana_4);
    printf(“%d ¥n”, tana_5);

    return 0;
}
```

変数を使う場合

```
10
11
12
13
14
Press any key to continue
```

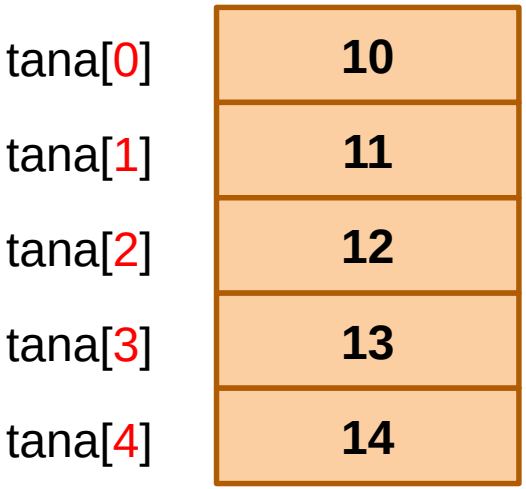
sample1

```
#include<stdio.h>
int main(void)
{
    int  tana[5] = {10,11,12,13,14};

    printf("%d ¥n", tana[0]);
    printf("%d ¥n", tana[1]);
    printf("%d ¥n", tana[2]);
    printf("%d ¥n", tana[3]);
    printf("%d ¥n", tana[4]);

    return 0;
}
```

配列を使う場合



sample1　を変更しましょう。

```
#include<stdio.h>
int main(void)
{
    int   i;
    int  tana[5] = [10,11,12,13,14];

    for ( i=0 ; i<5 ; i++ )
    {
        printf(“%d ¥n”, tana[i]);
    }

    return 0;
}
```

forを使う場合

考え方（説明）

i=0

printf(“%d ¥n”, tana[0]);　⇒10

i=1

printf(“%d ¥n”, tana[1]);　⇒ 11

:

Sample1 を変更しましょう。

```
#include<stdio.h>
int main(void)
{
    int   i;
    int   mark[5]={50,60,70,80,90};

    for ( i = 0 ; i < 5 ; i++ )  {
        if ( mark[ i ] >=  60 ) {
            printf("%d ¥n", mark[i]);
        }
    }

    return 0;
}
```

```
60
70
80
90
Press any key to continue
```

＜変数を使用＞

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

a = 10;
b = 11;

c = a + b;
d = a - b;
```

＜配列を使用＞

```
int  tana[4];

tana[0] = 10;
tana[1] = 11;

tana[2] = tana[0] + tana[1];
tana[3] = tana[0] -  tana[1];
```

※整数型の変数が使用可能

```
# include <stdio.h>
int  main (void)
{
        int   i, s = 0;
        int   tana[5] = {10,11,12,13,14};

        for (i = 0; i < 5; i++ )
        {
                s = s + tana[i] ;
        }

        printf(“ 合計 :  %d  ¥n ”, s);

        return 0;
}
```

本遠隔授業を受けるルール

- 課題の内容、小テスト、レポートの情報は評価に関わるので、他人に提供しないこと
- 遠隔授業の内容を勝手にSNS等で大学外第3者が観覧できるようにアップロードはしないでください。違法行為に当たる場合があります。

出席条件

- Webclassへのログインがあり、準備したテスト、アンケート、レポート等の提出があることで出席とする
- 締め切りまでに問題の解答をレポートとして提出する