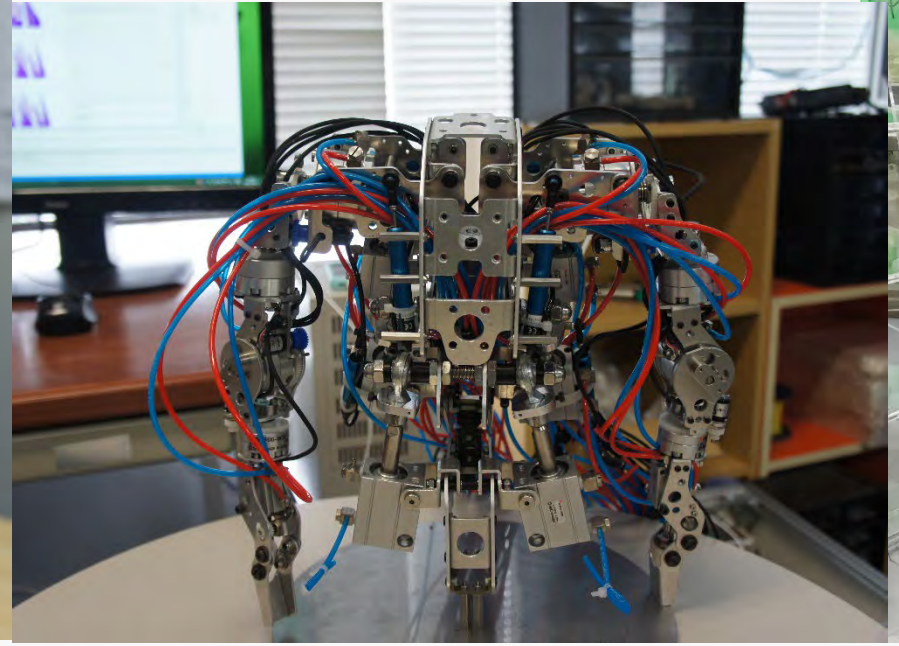




計算機とプログラミング

第４回　基本文法３

担当：石原尚
機械工学専攻　動的システム制御学領域　講師

同じ型の変数を並べてまとめたもの

利便性を高める「配列」について学びます

Pythonでも配列は扱いましたね

第１回
基礎を学ぶ準備
- 授業の進め方
- C言語の立ち位置
- C言語の環境整備

杉本先生

第２～４回
使い方の基礎
- 変数と制御構造
- ＋関数
- ＋配列

今日はここ！

石原

第５～８回以降
少し難しい使い方
- ポインタ
- データ構造

杉本先生＆平原先生

第９～１３回
応用と実践
- アルゴリズム
- 総合演習

中西先生＆杉本先生＆石川先生

C言語の配列で簡単に「できること」「できないこと」

Pythonにおけるリストに似たものですが，使い勝手は違います

○ できること

① 配列の**要素数**の定義
② 変数を用いた**要素指定**
③ 定義時の**初期化**

積極的に利用しましょう

× できないこと

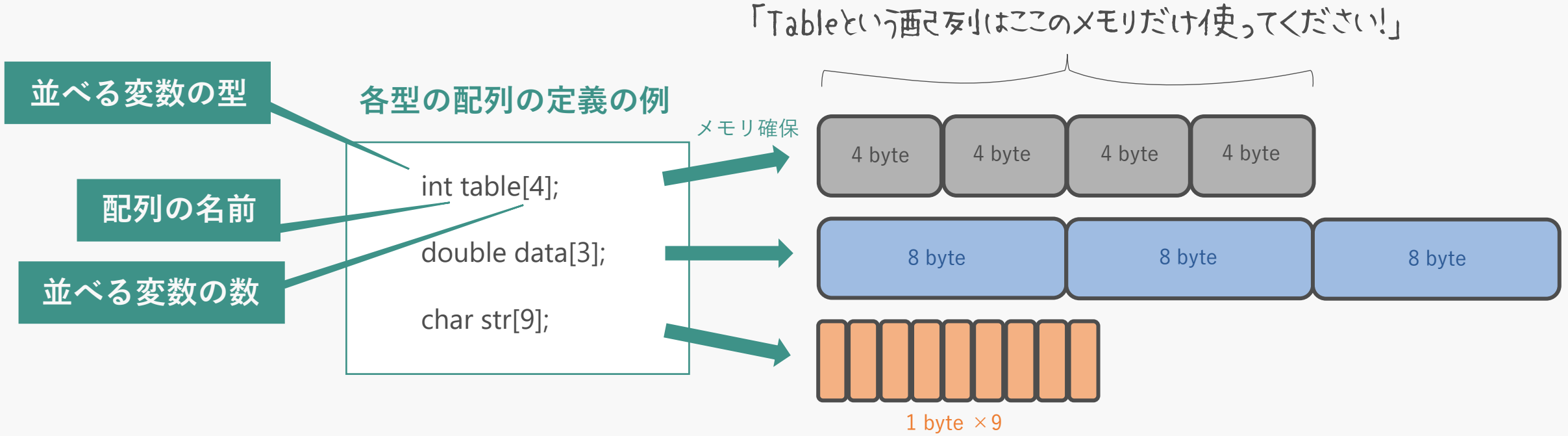
① 配列全体の**一括代入**
② **変数を用いた定義**
③ 仮引数への**値コピー**

コンパイラやインクルードするヘッダファイルによっては一部可能なので「基本」できないという認識でいてください

並べる変数の数

配列の定義の際に「配列要素の数」を確定できる

使用するコンピュータのメモリを厳密に管理できる



※Pythonでは自動的に（勝手に）要素数が変わります

配列の[]内の数値設定で狙った変数要素を指定できる

Pythonと同じ!

int型配列の定義と各要素への代入

```
int table[4];
table[0] = 1;
table[1] = 4;
table[3] = 6;
```

tableに割り当てられたメモリブロックに連番が降られる

0番目の要素に値が入る

1番目の要素に値が入る

3番目の要素に値が入る

0から始まるので
最後は「要素数-1」

ここには何が入っているかわからない

[]内に数値を入れると
普通のint型変数と同じように扱える

なので，例えば scanf("%d", &table[2]);
のような命令も可能

定義の際に「要素の数と値」を同時に設定できる

実施できる初期化の例

```
int a[3];

int b[2] = {3, 4};

int c[] = {10,20,30};

int d[4] = {100,200};
```

どの方法でも問題ないが，
どのように初期化されるかは
把握しておくべき

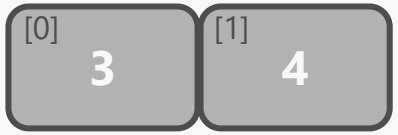
a

4 byte×3のメモリが確保されるだけ

値が入っていないわけではなく，
あくまで確定されていない状態

b

メモリ確保と共に全てのメモリブロックに値が入る



c

全てのメモリブロックに値が入るようにメモリが確保される

d

メモリ確保と共に番号の若い方から値が入る

値の代入は要素ごとに行わなければいけない

forを使う一括代入パターンを覚えましょう

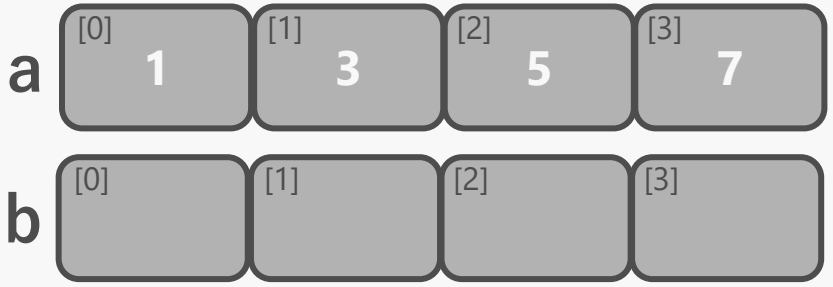
×一括代入できない例

配列名は変数でないので代入エラーになる

```
int a[] = {1,3,5,7};
int b[4];

b = a;
```

結果



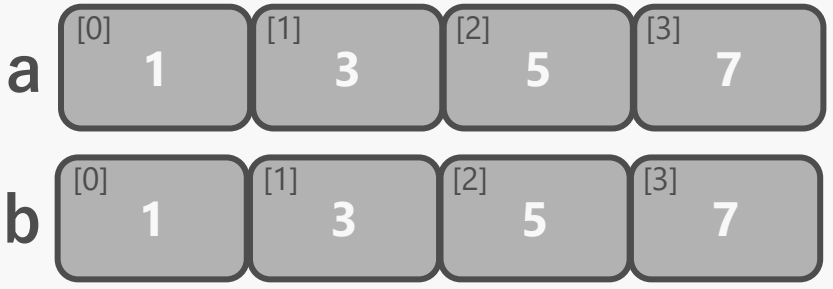
○一括代入できる例

```
int a[] = {1,3,5,7};
int b[4];

for(int i=0; i<4; i++){
    b[i] = a[i];
}
```

ここには配列要素数を書けばよい

iが0~3まで変わりながらbの配列要素に代入が行われる



(言語で少々やっかいな特徴

配列の要素数の定義には変数は使えない

複数の配列の要素数を後からまとめて変更しそうだから要素数は変数にしておきたい…

×これでは実現不可能

```
#include <stdio.h>

int main( ){
    int num = 4;

    int a[num];
    int b[num];
    int c[num];
}
```

numには4が入っているが，
num自体は変数なので不可

○こうしたい場合の代替案

```
#include <stdio.h>
#define SIZE 4

int main( ){

    int a[SIZE];
    int b[SIZE];
    int c[SIZE];
}
```

マクロというプリプロセッサ指令．
こう書くと，これ以降コンパイラは
SIZEを「4」として読み替える

```
int a[4];
int b[4];
int c[4];
```

と同じ意味になる

※前回演習課題で #define PI 3.14159265
を使うという工夫も可能だったというわけです

関数内でも書き換えられてしまうので注意

配列名を引数にした場合は本体が関数内に渡ってしまう

配列を関数の引数にした平均関数を実装した例

```
#include <stdio.h>
double average(int table[]);

int main( ){
    int a[] = {4,5};
    double ave;                 /*平均値格納用変数*/
    ave = average(a);
    printf("%f¥n",ave);          /*平均値の表示*/
    printf("%d", a[0]);         /*配列要素値のチェック用表示*/
}
double average(int table[]){
    double ret;
    ret = (double)(table[0] + table[1])/2;
    table[0] = 5;          /*平均計算には関係ない*/
    return ret;
}
```

実引数は配列の名前

これは単純で覚えやすい

仮の世界の変数のはずなのに，実際はa[0] = 5が実行される！

値のコピーが渡されているのではなく，配列自体がaverage関数の世界に渡ってきている!

仮引数は要素数指定なしの配列定義

他にも仮引数の設定の仕方はありますがこれが基本です