

第4回C言語勉強会

配列と繰り返し

コンピュータで扱われるもの

- 前回までの勉強会で学習したのは、コンピュータをどのように動作させるか、でした。
- コンピュータで扱われるものは2種類あります。それは“プログラム”と“データ”です。
- 前回までの内容は“プログラム”に当たります。

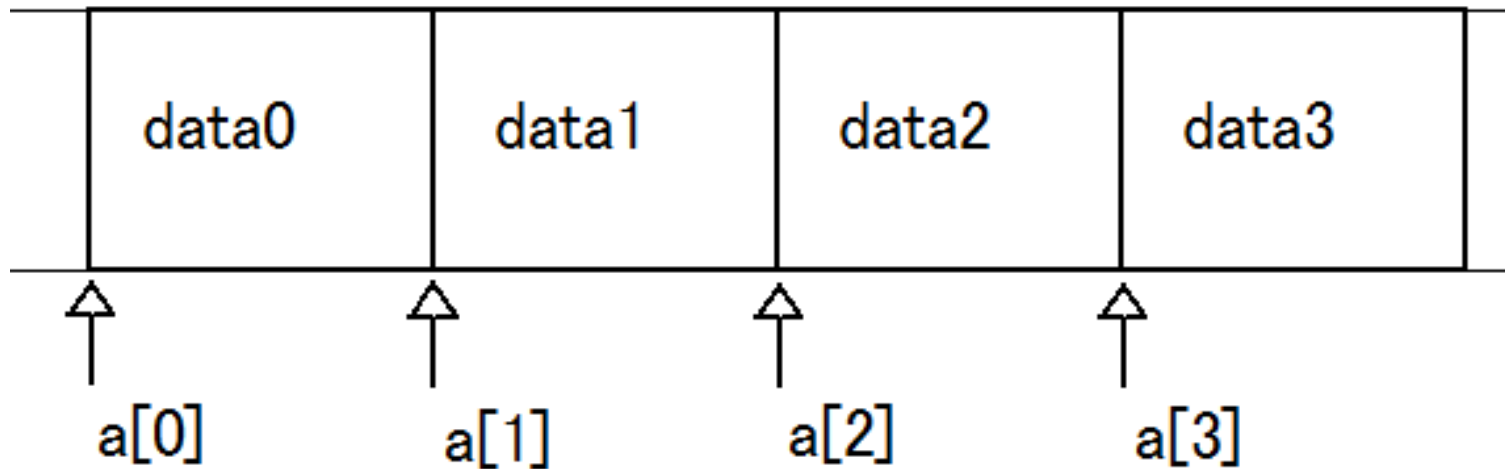
データ

- データとはコンピュータが処理の対象とする情報のことです。
- 様々な情報を表現するために、データの表現の方法にも色々な種類があります。
- **配列**はほぼ全ての言語でサポートされているメジャーなデータの表現方法です。

配列

- C言語では配列は変数と同じように型を持ちます。
- 実際は `int a[4];` などと宣言して使用しますが、この時にコンピュータの内部ではどのような物が用意されるのでしょうか。

配列a[4]の表現



コンピュータのメモリ上に四つの連続した領域が確保されます。それぞれの大きさは型によります。

例1

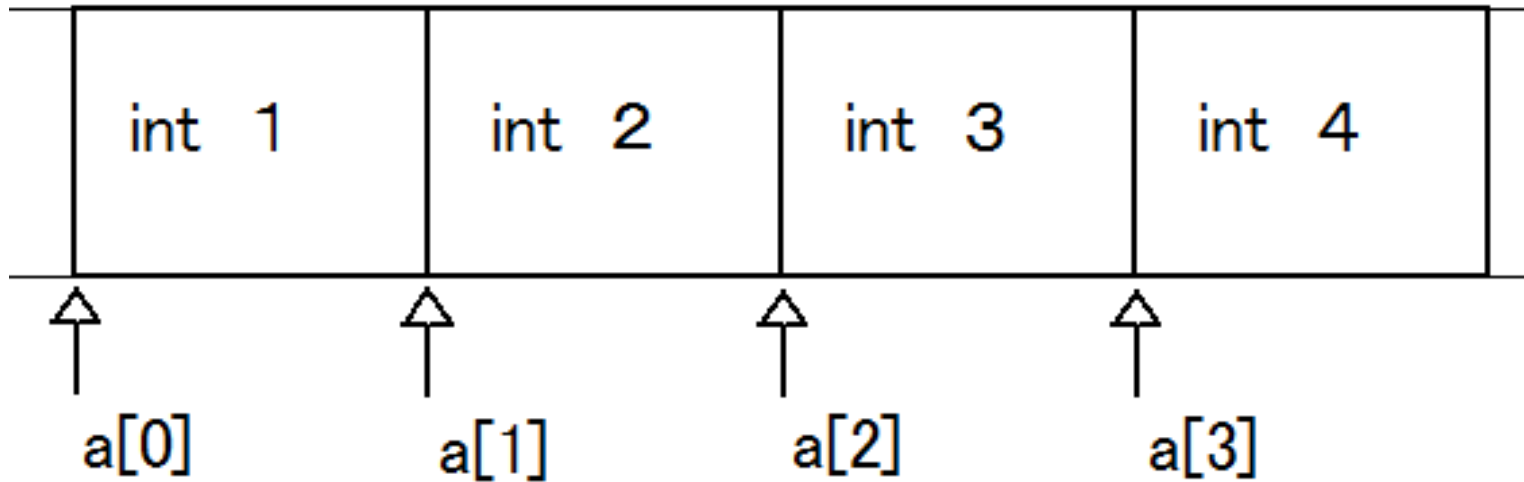
```
#include <stdio.h>
```

```
int main(void)
```

```
{  
    int a[4] = {1, 2, 3, 4}, i;  
  
    for (i = 0; i < 4 ; i++)  
    {  
        printf("%d¥n", a[i]);  
    }  
  
    return 0;  
}
```

a[i]と書くことで、代入したり値を得たりと、配列aのi番目の要素を変数と同じように扱えます。

```
int a[4] = {1,2,3,4};
```



こんな感じ。

問題

- 5個の数を入力として受け取り、その平均を出すプログラムを書いてみましょう。

ヒント

- まず配列を宣言してみましよう。
- 入力の受け取りには繰り返しを使うとすっきり表現できると思います。
- 割り算をする場合は型に注意してください。

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

```
int main() {
```

```
    int i, a[5];
```

```
    double ans;
```

```
    printf("五つの値の平均を算出します。¥n");
```

```
    for (i=0 ; i<5 ; i++)
```

```
    {
```

```
        printf("%d回目の入力 :", i+1);
```

```
        scanf("%d", &a[i]);
```

```
    }
```

```
    ans = (a[0] + a[1] + a[2] + a[3] + a[4]) / 5.0;
```

```
    printf("平均値 %.2f¥n", ans);
```

```
    return 0;
```

```
}
```

Tips

配列の宣言の方法はいくつかあります。

- `a[m]={}`; <-これは0で初期化されます。
- `a[]={1,2,3,4,...,n}`;
- `a[n]={1,2,3,4,...,n}`;

などなど

また、配列を初期化せずに使用するのは大変危険です。

多次元配列

- 配列は入れ子にして使う事ができます。
- このような配列を多次元配列と呼びます。
- `a[m][n];` のように宣言します。この場合は、ちょうど $m \times n$ の行列の表現と一致します。

ここで少し復習

- 二次元配列に掛け算の九九を収納してみよう！
- 先週習った繰り返しを入れ子にして、二次元配列を埋めていきましょう。
- 表示を工夫して

1 2 3 4 5 6 7 8 9

2 4 6 8 10 12 14 16 18 のようにしてください。

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

```
int main()
```

```
{
```

```
    int a[9][9];
```

```
    int i, j;
```

```
    for (i=1; i<=9; i++)
```

```
    {
```

```
        for (j=1; j<=9; j++)
```

```
        {
```

```
            a[i][j] = i*j;
```

```
        }
```

```
    }
```

```
    for (i=1; i<=9; i++)
```

```
    {
```

```
        for (j=1; j<=9; j++)
```

```
        {
```

```
            printf("%2d ", a[i][j]);
```

```
        }
```

```
        printf("\n");
```

```
    }
```

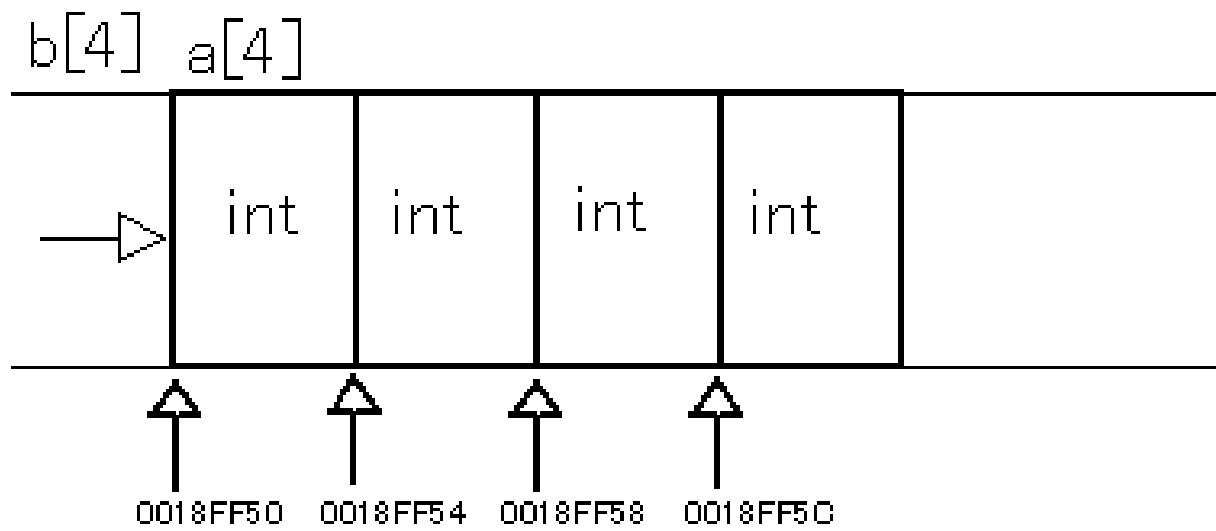
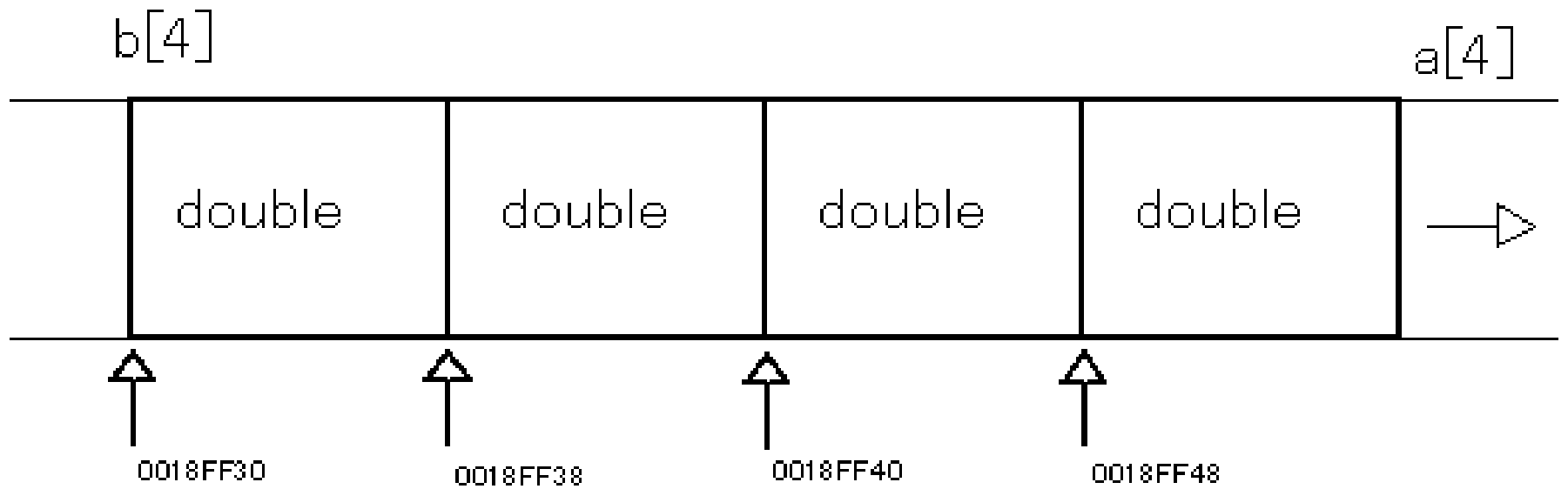
```
    return 0;
```

```
}
```

Plus α

実際にコンピュータのメモリ上で配列はどう扱われるのか？

配布資料“array_ex3.c”を開いてみてください。



&

- &を配列や、変数の先頭に着けると、そのコンピュータのメモリ上での存在位置を知ることができます。
- `a[0]`, `b[0]`などに対応する番地が存在することが分かります。つまり配列とは、メモリ上に連続した領域を用意し、そのアドレスと、`a[0]`などの名前を関連させる仕組みなのです。

来週は？