

プログラミング基礎 #03

変数と配列

担当：向井 智彦

前回のまとめ

- プログラムは1文ずつ(「;」区切り)実行される
- C/C++プログラムの本体はmain関数
- 「cout << データ」で画面表示
- 加減乗除「+」「-」「*」「/」, 剰余「%」
- 演算子には優先順位がある
- 数値には型(type)がある
- 演算を通じて型は自動変換される

本日の内容

- 目標：電卓代わりにC++プログラミングを利用
- 講義内容：データ値の記憶と利用
 - 変数・配列と型
 - 変数・配列の初期化, 代入
 - 変数や配列を用いた演算

@ wandbox.org

```
#include <iostream>
int main()
{
    int x = 0;
    x = 10 * 2;
    int y = x;
    std::cout << x << ", " << y << std::endl;
}
```

- 注意：
- ・全て半角文字で入力すること
 - ・大文字／小文字は区別されるので間違えないように
 - ・プログラム文の最後のセミコロン「;」を忘れずに

文法図解

```
#include <iostream>
```

```
int main()
```

型名

初期値

```
{  
  int x = 0;  
  x = 10 * 2;  
  int y = x;  
  std::cout << x << ", " << y << std::endl;  
}
```

変数名 (プログラマが指定)

変数の宣言

- ・指定された型の変数を, 変数名「x」で利用開始
- ・初期値0は省略可
(「int x;」でもOK)

文法図解

```
#include <iostream>
```

```
int main()
```

```
{
```

```
    int x = 0;
```

```
    x = 10 * 2;
```

```
    int y = x;
```

```
    std::cout << x << ", " << y << std::endl;
```

```
}
```

変数への**代入**

- ・左辺の変数の内容を
 右辺の値で置き換え
(変数xの内容が 0 から 20 になる)

文法図解

```
#include <iostream>
```

```
int main()
```

```
{
```

```
    int x = 0;
```

```
    x = 10 * 2;
```

```
    int y = x;
```

```
    std::cout << x << ", " << y << std::endl;
```

```
}
```

変数値の利用 & 変数の宣言

- ・左辺の変数を

右辺の変数の内容を用いて初期化
(変数yを, 変数xの値 20 を用いて初期化)

型 (type)

先週の復習

- 整数型 (... , -2, -1, 0, 1, 2, ...)
- 浮動小数点型 (-0.00009, 1043.1232, etc...)
- 文字型 ('a', 'A', ..., '0', ...)
- 文字列型 ("Hello", "world", etc...)
- ブール型 (true, false)
- ユーザー定義型 (クラス)
- ポインタ型

変数の型

- 整数型 `int, short, long`
- 浮動小数点型 `float, double`
- 文字型 `char`
- 文字列型 `std::string`（講義外）
- ブール型 `bool`
- ユーザー定義型（クラス）
- ポインタ型

変数の型 contd.

```
#include <iostream>
using namespace std;
int main()
{
    int    x = 0;
    double y = 10.2;
    char   z = 'a';
    cout << x << ", " << y << ", " << z << endl;
}
```

変数と演算

以下のプログラムの出力は？

```
int a = 0, b = 0; // 2つまとめて宣言 & 初期化  
a = 10;  
b = a;  
a = a + 20;  
cout << a << ", " << b << endl;
```

変数と演算

1. 変数 a と b が, 初期値 0 で作成

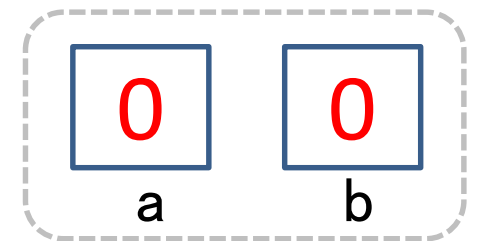
```
int a = 0, b = 0;
```

```
a = 10;
```

```
b = a;
```

```
a = a + 20;
```

```
cout << a << ", " << b << endl;
```



変数と演算

2. 変数 a に値 10 が代入される

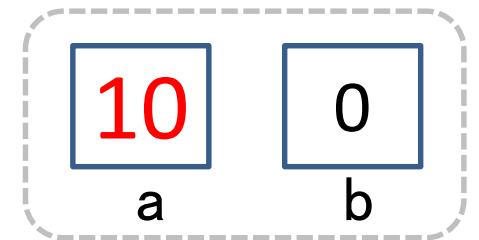
```
int a = 0, b = 0;
```

```
a = 10;
```

```
b = a;
```

```
a = a + 20;
```

```
cout << a << ", " << b << endl;
```



変数と演算

3. 1. 変数 a が保持する値 10 を読み出し

```
int a = 0, b = 0;
```

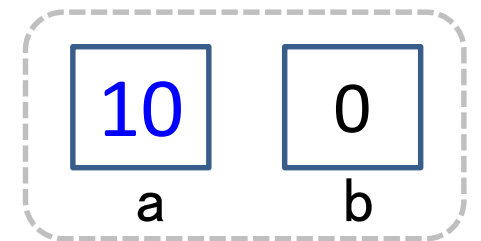
```
a = 10;
```

```
b = a,
```

10

```
a = a + 20;
```

```
cout << a << ", " << b << endl;
```



変数と演算

3. 2. 変数 b に変数 a が保持している値 10 が代入される

```
int a = 0, b = 0;
```

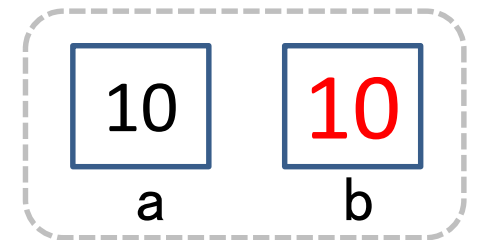
```
a = 10;
```

```
b = a;
```

10

```
a = a + 20;
```

```
cout << a << ", " << b << endl;
```



変数と演算

4. 1. 変数 a が保持している値 10 を読み出し

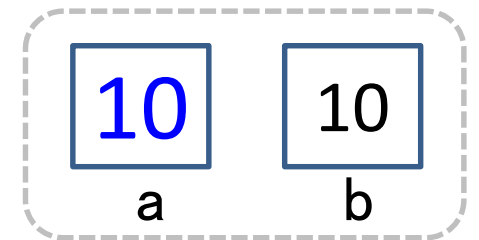
```
int a = 0, b = 0;
```

```
a = 10;
```

```
b = a;
```

```
a = a + 20;
```

```
cout << a << ", " << b << endl;
```



変数と演算

4. 2. 読み出した値 10 と 20 を加算

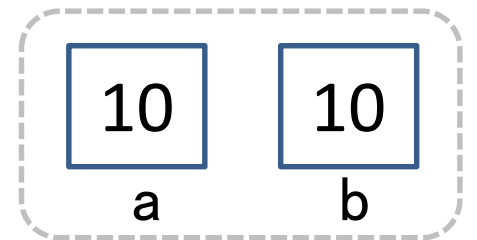
```
int a = 0, b = 0;
```

```
a = 10;
```

```
b = a;
```

```
a = a + 20;
```

```
cout << a << ", " << b << endl;
```



10

変数と演算

4. 3. 加算結果 30 を変数 a に代入

```
int a = 0, b = 0;
```

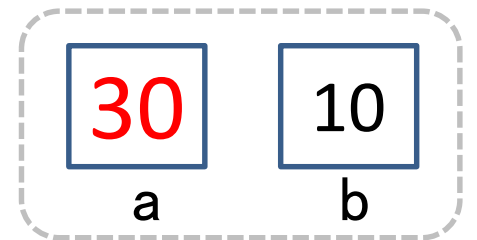
```
a = 10;
```

```
b = a;
```

```
a = a + 20;
```

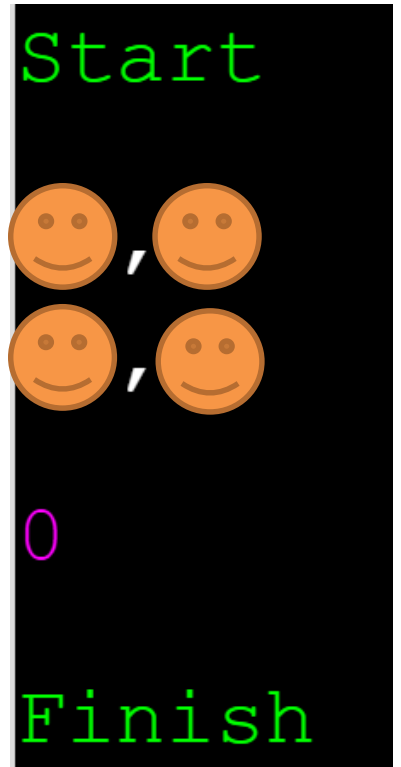
```
cout << a << ", " << b << endl;
```

30



クイズ： 出力結果は？

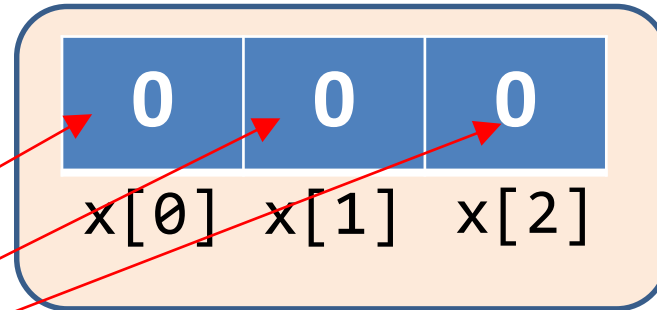
```
#include <iostream>
using namespace std;
int main()
{
    int x = 10, y = 20;
    cout << x << "," << y << endl;
    x = 30;
    x = y + x;
    y = x - 10;
    cout << x << "," << y << endl;
}
```



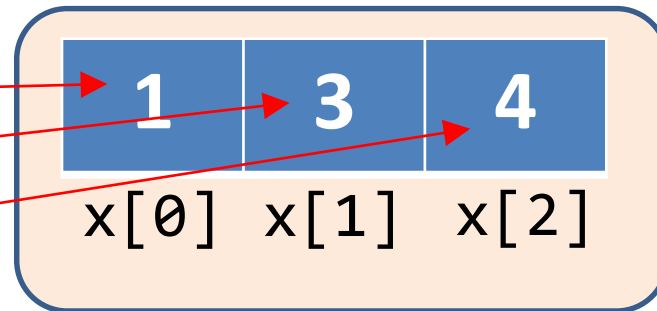
配列

- 複数の変数を番号付けて並べる

```
#include <iostream>
using namespace std;
int main()
{
```



代入



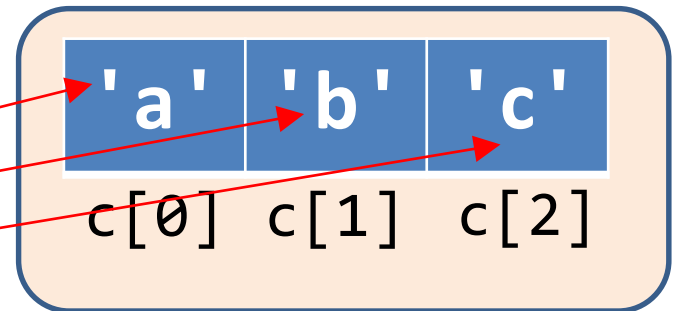
```
    int x[3] = {0, 0, 0};
    x[0] = 1;
    x[1] = 3;
    x[2] = 4;
    cout << x[0] << " " << x[1] << "," << x[2] << endl;
}
```

文字型配列

- 複数の変数を番号付きで並べる

```
#include <iostream>
using namespace std;
int main()
{
    char c[3] = {'a', 'b', 'c'};
    c[0] = c[1];
    c[1] = c[2];
    c[2] = 'd';
    cout << c[0] << c[1] << c[2] << endl;
}
```

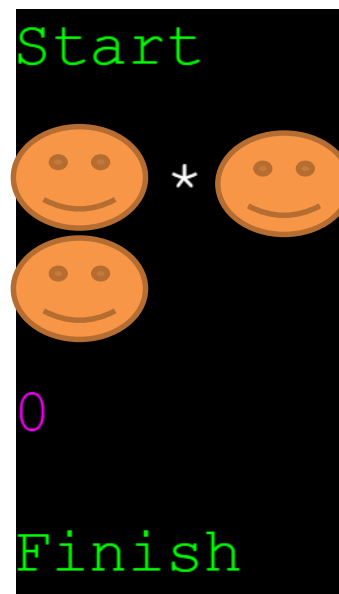
配列の初期値



出力結果は？

応用クイズ： 出力は何？

```
#include <iostream>
using namespace std;
int main() {
    int c[3] = {0, 0, 0};
    c[0] = 3.8;
    c[1] = 2.0;
    c[2] = c[0] * c[1];
    cout << c[0] << " * " << c[1] << endl;
    cout << c[2] << endl;
}
```



課題1.数値の入れ替え: 03-01.cpp

```
#include <iostream>
using namespace std;
int main()
{
    int x = 30, y = 20;
    cout << x << "," << y << endl;

    // TODO: 適切なコードを記述

    cout << x << "," << y << endl;
}
```

変数xとyの数値を
入れ替えて出力

Start

30,20

20,30

0

Finish

出力例

課題2. ベクトルの足し算: 03-02.cpp

```
#include <iostream>
using namespace std;
int main()
{
    double v0[2] = {11.0, 10.0};
    double v1[2] = {5.0, 7.0};
    double sum[2];
    // TODO: 適切なコードを記述

    cout << sum[0] << "," << sum[1] << endl;
}
```

ベクトル [10, 10] と
[5, 7] の加算結果を
出力する

Start
16, 17
0
Finish

出力例

課題3. 総和: 03-03.cpp

- 7個の要素を持つ浮動小数点型(double型)の配列に適当な値を設定し, それら7個の値の総和を求め, 画面(cout)に出力するプログラムを作成

```
#include <iostream>
using namespace std;
int main()
{
    // TODO: 適切なコードを記述
}
```

まとめ

- 変数：1つのデータを一時保存できる
 - 変数宣言 「int a;」, 代入(上書き) 「a = 10;」
 - 変数からの代入(読み出し) 「int b = a;」
- 配列：同じ型の複数の変数を添字付きで格納
 - 配列宣言 「int a[10];」
 - 配列の先頭番号は 0 「a[0] = 1;」...「a[9] = 10;」
- 変数を用いた演算
 - 「a = a + b + 10 * a;」のように代入演算子 = の左右両側に同じ変数が登場する際の挙動

来週の予告

- 条件分岐
- 論理演算とブール変数
- プログラムファイルとコンパイル