

第6章 2重ループ&配列

2重ループと配列をやります

今回の内容

- 2重ループ

- 配列

- コメント

- define

以下のプログラムを実行してみよう

```
int main()
{
    int x, y;

    for (y = 0; y < 3; y++)
    {
        for (x = 0; x < 5; x++)
        {
            printf("★");
        }
        printf("\n");
    }

    return 0;
}
```



2重ループ

- 「ループ」とは、繰り返しのこと（while文とfor文）
- 「2重ループ」とは、ループを2重にしたもの

```
int i, j;
for (i = 0; i < 繰り返す回数; i++)
{
    for (j = 0; j < 繰り返す回数; j++)
    {
    }
}
```

九九の表のy行目を表示

- 「%2d」は、2ケタで表示するという意味
 - 1ケタならスペースが入って2文字になる
 - %2d は表を作るときによく使う
- これをy=1から9まで繰り返すと、九九の表ができそう

2 4 6 8 10 12 14 16 18

```
int main()
{
    int x, y;

    y = 2;

    for (x = 1; x <= 9; x++)
    {
        printf("%2d ", x * y);
    }
    printf("\n");

    return 0;
}
```

練習問題1

- 掛け算の九九の表を表示するプログラムを作りなさい

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

練習問題1 解答例

```
int main()
{
    int x, y;

    for (y = 1; y <= 9; y++)
    {
        for (x = 1; x <= 9; x++)
        {
            printf("%2d ", x * y);
        }
        printf("\n");
    }

    return 0;
}
```

以下のプログラムを実行してみよう

```
int main()
{
    int hp[3];

    hp[0] = 100;
    hp[1] = 80;
    hp[2] = 200;
    printf("%d %d %d\n", hp[0], hp[1], hp[2]);

    return 0;
}
```

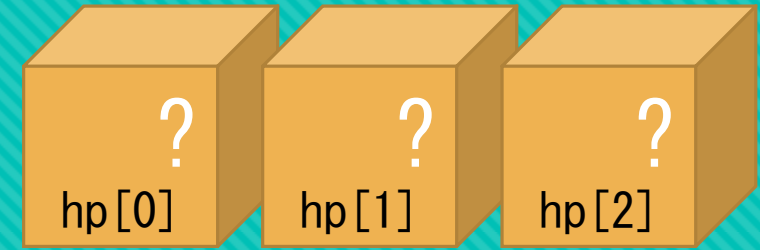
100 80 200

配列

配列の宣言

```
int 名前[大きさ];
```

```
int hp[3];
```

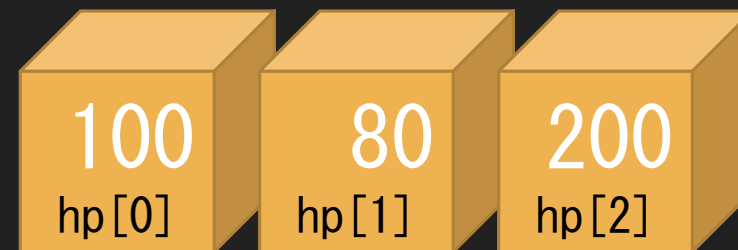


- 配列とは、変数（箱）がいくつか並んでいるもの
- 配列の宣言を1回行うだけで、たくさんの変数（箱）ができる
- 「int hp[3];」と宣言すると、hp[0], hp[1], hp[2] という名前の3つの箱ができる
- hp[3]という箱はありません（0番から始まるため）

配列の初期化

```
int hp[3] = { 100, 80, 200 };
```

```
int hp[] = { 100, 80, 200 };
```



- 配列の宣言のときに初期化を行うことで、最初から箱の中身を指定することができる
- イコールと中カッコ「={ }」を使い、コンマ「,」で区切る
- 初期化をする場合、配列の大きさは省略してもよい

初期化と代入

初期化

```
int x = 5;
```

```
int hp[3] = { 100, 80, 200 };
```

代入

```
x = 5;
```

```
hp = { 100, 80, 200 };
```

```
hp[0] = 100;  
hp[1] = 80;  
hp[2] = 200;
```

- 初期化は変数を作るときに行える、代入はいつでも行える
- 配列は初期化ができるが、代入はできない（代入したいなら、変数1つ1つに代入するしかない）

練習問題2

- 入力された3つの整数を表示するプログラムを作りなさい

```
1P > 100
```

```
2P > 200
```

```
3P > 300
```

```
1P:100 2P:200 3P:300
```

練習問題2 解答例

```
int main()
{
    int i;
    int hp[3];

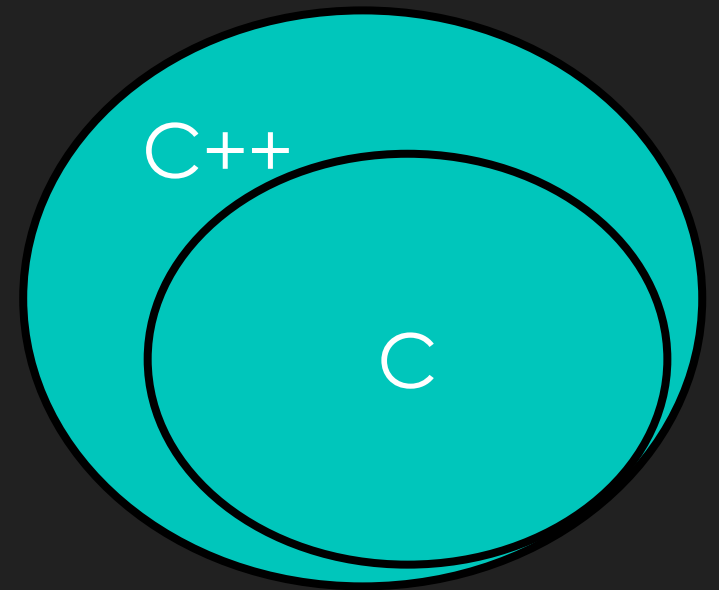
    for (i = 0; i < 3; i++)
    {
        printf("%dP > ", i + 1);
        scanf("%d", &hp[i]);
    }

    for (i = 0; i < 3; i++)
        printf("%dP:%d  ", i + 1, hp[i]);
    printf("¥n");

    return 0;
}
```

CとC++

- 実は今使っている Visual Studio は、C++ という言語を使う
- CでできることはC++でもできるので、今までCが使えた
- C++でできてCでできないことがある



以下のプログラムを実行してみよう

```
int main()
{
    // ハローワールド
    printf("Hello, world!¥n");
    printf("%d¥n", 1 /* + 2 */ );

    return 0;
}
```

```
Hello, world!
1
```

コメント

```
/* ここはコメント */
```

```
// 行の最後までコメント
```

- コメントは、C言語として実行されない（無視される）
- 「/*」から「*/」までの部分、「//」から行の最後まで部分、がコメントになる
- 例えば、このプログラムが何をするのかをメモする、プログラムの一部をコメントにして一時的に消す、などなど、いろいろと役に立つかもしれない
- ちなみに、Cのコメントは「/* */」のみで「//」は使えない（はずだが、最近のCは「//」が使える）

define

#define 名前 値

- defineは定数
- 同じ数字を何度も使うときはdefineがおすすめ
- defineの値を変えると全部変わる

1P:100 2P:200 3P:300

```
#define NUM 3

int main()
{
    int i;
    int hp[NUM];

    for (i = 0; i < NUM; i++)
        hp[i] = 100;

    for (i = 0; i < NUM; i++)
        printf("%dP:%d ", i + 1, hp[i]);
    printf("¥n");

    return 0;
}
```

終わり

お疲れ様でした