C言語講座
第2回

覚えるべき基礎事項
―char型、演算子、配列―

Character型

int型は整数を記憶するための変数だった。

じゃあ、文字は？ってことで紹介するのがchar型。

○構文

```
char c = 'a';
```

注意すべきなのは、「'」を使うことと、charは一文字しか表せないこと。

○プログラミング実践

入力された文字を表示する

```
#include <stdio.h>
void main(void)
{
    char c;
    printf("一文字入力してください:");
    scanf("%c", &c);
    printf("入力された文字:%c¥n", c);
}
```

算術演算子(+,-,*,/,%, +=,-=,*=,/=,%=)

前回、簡単に四則演算+剰余に関しては説明したと思う。

まずはそれを拡張というか、覚えておくべき表記について説明していく。

と言っても、見ればすぐに理解は出来るはずなので次ページの表を見てほしい。

算術演算子表

単項+	+a	あんまり意味は無い・・・
単項-	-a	aの符号を反転したもの。 10 ⇒ -10みたいに。
後置増分	a++	a = a + 1 と同じ。
後置減文	a--	a = a - 1 と同じ。
単純代入	a = b	aにbを代入
和	a += b	a = a + b
差	a –= b	a = a – b
積	a *= b	a = a * b
商	a /= b	a = a / b
余り	a %= b	a = a % b

論理演算子(!, &&, ||)

中学だか高校だかで習ったはずの、集合。
真(1)と偽(0)の二種類を組み合わせで表す。

x	y	!x(否定)	x&&y (論理積、かつ)	x||y (論理和、もしくは)
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

比較演算子(<,>,==, <=,>=,!=)

算術演算子と同じく、見ればだいたいの意味は分かるはず。

だが、一つだけ注意してほしいのが“==”

“=”だけだと代入になってしまうので、“等しい”という意味で使いたいときは必ず二つ重ねるように！！

比較演算子表

a < b	aがbよりも小さい
a > b	aがbよりも大きい
a == b	aとbは等しい
a <= b	aがb以下
a <= b	aがb以上
a != b	aとbが不等

分岐処理①(if)

英語での意味そのままに、ifは“もし～ならば”、elseは“それ以外”を意味します。

○構文

```
if (式1) { 処理1; } // もし、式1が非0ならば処理1を行い、
else if ( 式2 ) { 処理2; }   // 式1が0で、式2が非0ならば
                                処理2を行い、
else { 処理3; }          // どの処理も行われなければ処理3
                                を行う
```

まぁ、非0とか言われても意味不明だろうし、正直、これ作ってる人もそんな仕様忘れてました。

そこで使うのが論理演算子！

```
if (式1) { 処理1; }    // もし式1==trueなら処理1だけ行う
else if ( 式2 ) { 処理2 ; } //式2==trueなら処理2だけ行う
else { 処理3; }        // 全ての式==falseなら処理3を行う
```

...結局文章だけじゃ分かりにくいと思うので次ページのサンプル見てください。

○プログラミング実践

５で割り切れるかどうかを表示する

```
#include <stdio.h>
void main(void)
{
   int num;    printf(“整数を入力してください：”);
   scanf("%d", &num);
    if (num % 5 == 0) {
        printf("5で割りきれます¥n", num);
    } else {
        printf("5で割りきれません¥n", num);
    }
}
```

分岐処理②(switch)

switchは…日本語に直せませんでしたorz

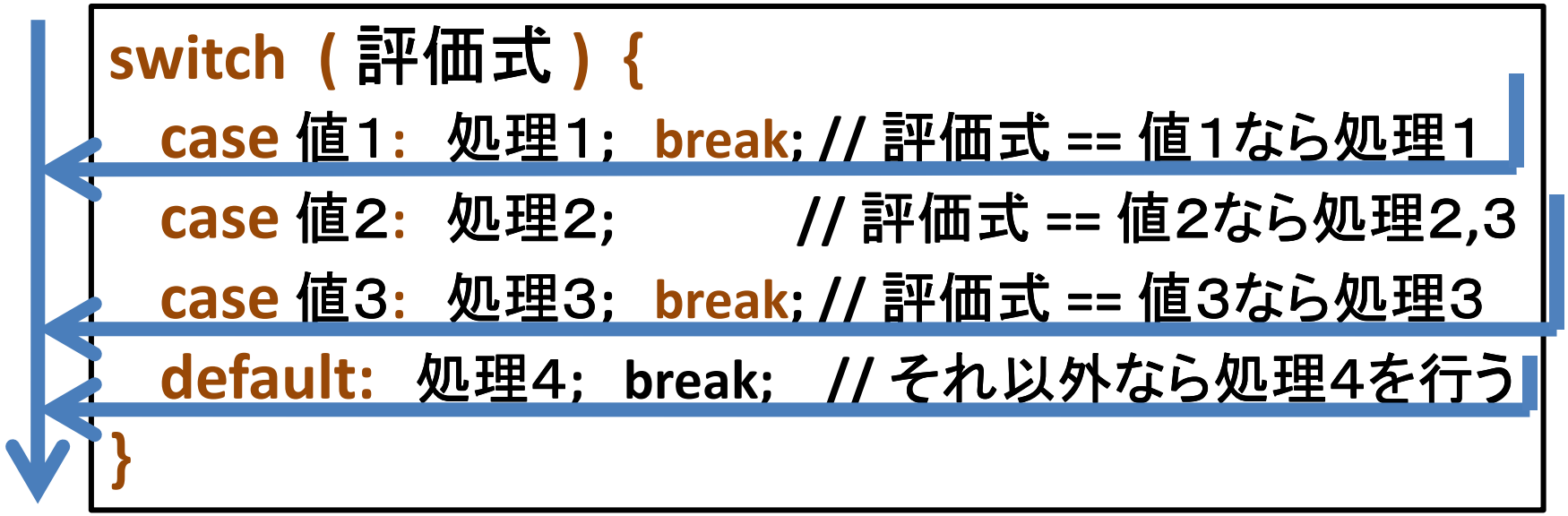
○構文

```
switch ( 評価式 ) {
   case 値1:  処理1;  break; // 評価式 == 値1なら処理1
   case 値2:  処理2;   break; // 評価式 == 値2なら処理2
   case 値3:  処理3;  break; // 評価式 == 値3なら処理3
   default:  処理4;  break;   // それ以外なら処理4を行う
}
```

“default”はif文におけるelseみたいなもの。

それよりも、switchで注意すべきなのは“**break**”

これが無いと意図しない処理まで行ってしまう。



○プログラミング実践

３の余りを表示

```
#include <stdio.h>
void main(void)
{
   int num;     printf("整数を入力してください：");
   scanf("%d", &num);
   switch (num % 3) {
      case 0: printf("3の倍数です¥n"); break;
      case 1: printf("1余りです¥n"); break;
      default: printf("2余り…だと思います¥n"); break;
   }
}
```

配列

学生番号や背番号など、同じ種類の“もの”が集まっている時、一つ一つに名前を付けるよりも番号を付けた方が分かりやすいことがある。

配列とは、指定した数の箱を作り、その箱に文字や数値を格納することでまとめて管理するもの。

○構文

```
データ型  配列(変数)名[要素数]
```

…はい、分かりにくい。

具体的には

```
int  a[3];          //「a」という変数を3個作る
```

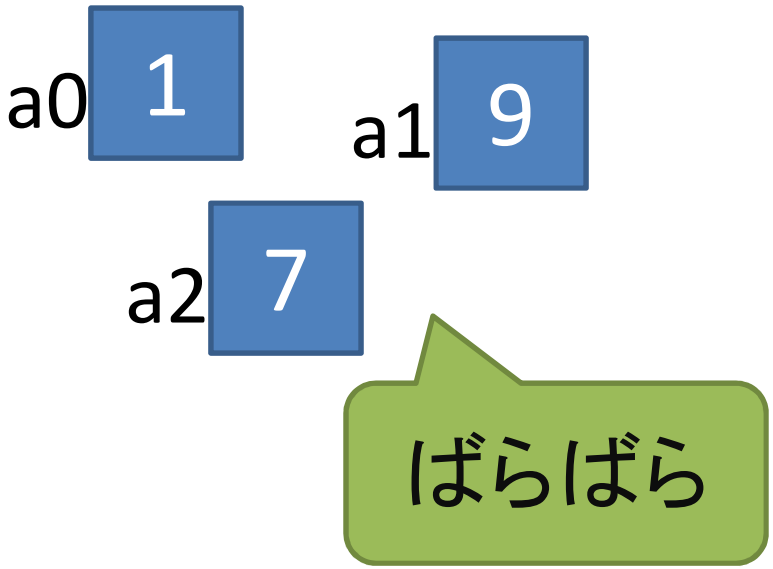
ちなみに“[]”を添字演算子と呼び、

index(要素番号)は必ず0から始まる。

イメージとしては、こんな感じ

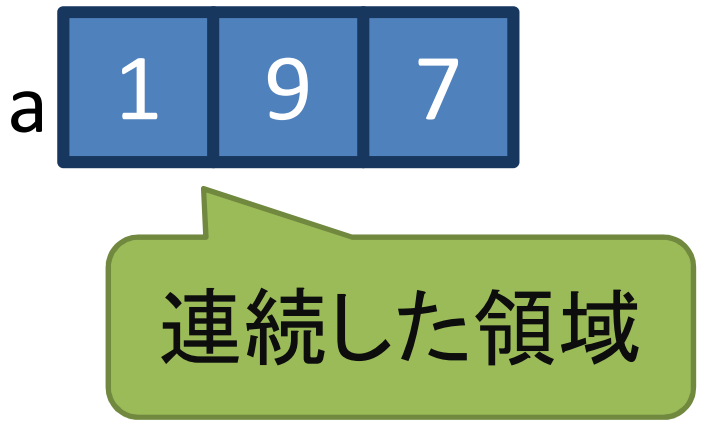
○個別の変数

```
int a0 = 1;
int a1 = 9;
int a2 = 7;
```



○配列

```
int a[3];
int a[0] = 1;
int a[1] = 9;
int a[2] = 7;
```



文字列

C言語の文字列操作は意外に面倒だったりするわけなんだが…

配列の概念があると、少し分かりやすい？

```
char  s[64];        //「s」という変数を64個作る
```

○プログラミング実践

入力された文字列を表示する

```
#include <stdio.h>
void main(void)
{
    char s[128];
    printf("文字列のinput:");
    scanf("%s", &s);
    printf("input文字列¥n%s¥n", s);
}
```

○演習

５つの要素を持つint型配列を用意し、その3番目を10に初期化して表示するプログラムを作成してください。

○解答

```
#include <stdio.h>
void main(void)
{
    int arrayInt[5];
    arrayInt[2] = 10;
    printf(“%d”, arrayInt[2]);
}
```