

# プログラミング概論

<http://bit.ly/kosen01>

Week4@後期  
2016/10/19

# 本日の内容

- 講義
  - これまでに学習した繰り返し構文の復習
    - for文、while文, do while文
  - 繰り返し構文の入れ子
  - 代入演算子
- 演習
  - 繰り返し構文の練習

# 復習：for文によるn回繰り返し

- for文は繰り返し回数が決まっている場合に便利

```
1 #include <stdio.h>
2
3 int main() {
4     int i;
5     // for文による5回繰り返し
6     for(i=1; i<=5; i++) {
7         printf("こんにちは。 \n");
8     }
9     return 0;
10 }
```

nを繰り返したい回数に変える

# 復習：for文の文法(2/2)

初期化  
= はじめに一度だけ  
実行される処理

条件式  
= この条件を満たす間  
繰り返される

更新  
= 実行が終わる度に  
実行される  
(iの値を1増やす)

5 // for文による5回繰り返し  
6 for(i=1; i<=5; i++) {  
7 printf("こんにちは。 \n");  
8 }

繰り返し実行する文

# 復習: while文

- while文では、{}内に記述された処理を実行する前に条件が真であるか判定する
- 例)

```
1 #include <stdio.h>
2
3 int main() {
4     int money = 0; // 貯まったお金
5     int month = 0; // 現在何ヶ月目か
6     // while文の使用例
7     while(money <= 1000000) {
8         money = money + 500;
9         month++;
10        printf("%dヶ月目 : %d円\n", month, money);
11    }
12 }
```

# 復習: do while文

```
1 #include <stdio.h>
2
3 int main() {
4     int money = 1000500;
5     int month = 0; // 現在何ヶ月目か
6     // do while文の使用例
7     do {
8         month++;
9         printf("%dヶ月目 : %d円\n", month, money);
10        money = money + 500;
11    } while(money <= 1000000);
12    return 0;
13 }
```

すでに1000000を  
超えている

実行結果 :

1ヶ月目 : 1000500円

8-9行目が実行されてから  
条件が真か偽か判定される

# for文の入れ子

- 繰り返し構文(for, while, do while)の中に繰り返し構文を記述できる

```
1 #include <stdio.h>
2 int main() {
3     int i, j; // カウンタ変数を2つ宣言
4     for(i=1; i<=5; i++) {
5         printf("%d ", i); // iの値を表示
6         for(j=1; j<=i; j++) {
7             printf("*"); // *をi個表示
8         }
9         printf("\n");
10    }
11 }
```

# 例1) 入れ子による \* の表示

```
1 #include <stdio.h>
2 int main() {
3     int i, j; // カウンタ変数を2つ宣言
4     for(i=1; i<=5; i++) {
5         printf("%d ", i); // iの値を表示
6         for(j=1; j<=i; j++) {
7             printf("*"); // *をi個表示
8         }
9         printf("\n");
10    }
11 }
```

出力)  
\$ ./a.out  
1 \*  
2 \*\*  
3 \*\*\*  
4 \*\*\*\*  
5 \*\*\*\*\*



# 例1) 入れ子による \* の表示

```
1 #include <stdio.h>
2 int main() {
3     int i, j; // カウンタ変数を2つ宣言
4     for(i=1; i<=5; i++) {
5         printf("%d ", i); // iの値を表示
6         for(j=1; j<=i; j++) {
7             printf("*"); // *をi個表示
8         }
9         printf("\n");
10    }
11 }
```

1. i=1から外側のループ開始

\$ ./a.out

1 \*

2 \*\*

3 \*\*\*

4 \*\*\*\*

5 \*\*\*\*\*

# 例1) 入れ子による \* の表示

```
1 #include <stdio.h>
2 int main() {
3     int i, j; // カウンタ変数を2つ宣言
4     for(i=1; i<=5; i++) {
5         printf("%d ", i); // iの値を表示
6         for(j=1; j<=i; j++) {
7             printf("*"); // *をi個表示
8         }
9         printf("\n");
10    }
11 }
```

1. i=1から外側のループ開始

\$ ./a.out

1 \*

2 \*\*

3 \*\*\*

4 \*\*\*\*

5 \*\*\*\*\*

2. 内側のループ開始

# 例1) 入れ子による \* の表示

```
1 #include <stdio.h>
2 int main() {
3     int i, j; // カウンタ変数を2つ宣言
4     for(i=1; i<=5; i++) {
5         printf("%d ", i); // iの値を表示
6         for(j=1; j<=i; j++) {
7             printf("*"); // *をi個表示
8         }
9         printf("\n");
10    }
11 }
```

1. i=1から外側のループ開始

2. 内側のループ開始

\$ ./a.out

```
1 *
2 **
3 ***
4 ****
5 *****
```

3. j=1からiの値(=1)になるまで \* の表示を繰り返す。

→ 1回だけ \* を表示してこのfor文は終わる

# 例1) 入れ子による \* の表示

```
1 #include <stdio.h>
2 int main() {
3     int i, j; // カウンタ変数を2つ宣言
4     for(i=1; i<=5; i++) {
5         printf("%d ", i); // iの値を表示
6         for(j=1; j<=i; j++) {
7             printf("*"); // *をi個表示
8         }
9         printf("\n");
10    }
11 }
```

1. i=2にして2周目開始 (出力)

\$ ./a.out

1 \*

2 \*\*

3 \*\*\*

4 \*\*\*\*

5 \*\*\*\*\*

# 例1) 入れ子による \* の表示

```
1 #include <stdio.h>
2 int main() {
3     int i, j; // カウンタ変数を2つ宣言
4     for(i=1; i<=5; i++) {
5         printf("%d ", i); // iの値を表示
6         for(j=1; j<=i; j++) {
7             printf("*"); // *をi個表示
8         }
9         printf("\n");
10    }
11 }
```

1. i=2にして2周目開始 (出力)

\$ ./a.out

1 \*

2 \*\*

3 \*\*\*

4 \*\*\*\*

5 \*\*\*\*\*

2. 内側のループ開始

# 例1) 入れ子による \* の表示

```
1 #include <stdio.h>
2 int main() {
3     int i, j; // カウンタ変数を2つ宣言
4     for(i=1; i<=5; i++) {
5         printf("%d ", i); // iの値を表示
6         for(j=1; j<=i; j++) {
7             printf("*"); // *をi個表示
8         }
9         printf("\n");
10    }
11 }
```

1. i=2にして2周目開始 (出力)

2. 内側のループ開始

3. j=1からiの値(=2)になるまで \* の表示を繰り返す。  
→ 2回 \* を表示してこのfor文は終わる

\$ ./a.out

```
1 *
2 **
3 ***
4 ****
5 *****
```

# 代入演算子

sum = sum + 10;        sum += 10;

どちらの記法も同じ意味  
短く書けたほうが良いので+=がおすすめ

+=以外にも -=、 \*=、 /=なども使用可  
(他にも演算子があるので調べてみると良い)

# 演習

- for文の入れ子を使うプログラムの練習
- 演習を始める前に以下のコマンドで作業用のディレクトリを作ると良い

```
$ mkdir kouki4
```

```
$ cd kouki4
```



# 演習0

- 講義資料の例1を「enshu0.c」というファイルに記述しfor文の入れ子の動作を確認しなさい。

# 演習1

- 例1では「5 \*\*\*\*\*」を表示するとプログラムが終了します。  
「20 \*\*\*\*\*... (20個 \* が続く)」まで表示できるようにし、「enshu1.c」というファイル名で保存しなさい。

## 演習2：横向きのグラフ

```
1 *
2 **
3 ***
4 ****
5 *****
```

\*で書いた $y = x$ のグラフ(横向き, 縦軸 $x$ , 横軸 $y$ )

- 演習0/演習1のプログラムは $y = x$ のグラフを横向きにしたものと考えることが出来る
- \*を用いて $y = 2x^2 - 1$ のグラフを表示しなさい  
(ファイル名は「enshu2.c」)

# 演習2の出力例

\$ ./a.out

1

2 \*\*\*

3 \*\*\*\*

4 \*\*\*\*\*

5 \*\*\*\*

## 演習3-1) 九九表

- 以下のような九九表を表示するプログラムを作成しなさい(enshu3-1.c)

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

## 演習3-2) 九九表

- printf()関数では以下のように表示する数値の桁数を指定することが出来ます
  - printf(“%3d”, i\*j); // 3桁で表示
- 以下のように九九表を見やすい形にしてください(enshu3-3.c)

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

## 演習3-3: 見やすい九九表

- 以下の形式の九九表を表示しなさい  
(enshu3-3.c)

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

# 演習4

- 次ページのプログラムを+=、-=、\*=、/=を用いて書き変えなさい(enshu4.c)



# 演習4(もとのプログラム)

```
1 #include <stdio.h>
2
3 int main(void) {
4     int n = 2;
5     // nに5を足す
6     n = n + 5;
7     printf("nの値 : %d\n", n);
8     // nに2を掛ける
9     n = n * 2;
10    printf("nの値 : %d\n", n);
11    // nを2で割る
12    n = n / 2;
13    printf("nの値 : %d\n", n);
14    // nから1引く
15    n = n - 2;
16    printf("nの値 : %d\n", n);
17 }
```

# 次回

- 次回は関数という機能を使って複数の処理をまとめて呼び出す方法を学習します
- 11/1は休講
- 次回までが試験範囲です。