

# プログラミング概論

<http://bit.ly/kosen01>

Week2@後期  
2016/09/27

# 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
- 演習
  - do while文の演習課題
  - これまでに学んだ繰り返し構文の復習

# 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

# while文の文法

```
while( 条件式 ) {  
    繰り返す処理  
}
```

条件式を満たす間、中括弧内の処理を繰り返す

# while文の条件判定のタイミング


- 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 }
```

# while文の条件判定のタイミング

- 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 }
```




もしも4行目が  
`int money = 1000000;`  
だったら？

# 条件判定のタイミング(1/2)

- 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 }
```



もしも4行目が  
`int money = 1000000;`  
だったら？



# 条件判定のタイミング(1/2)

- 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\n", month);
11    }
12 }
```

もしも4行目が  
`int money = 1000500;`  
だったら？

moneyは1000000を  
超えているので8-9行目は  
一度も実行されない  
(=なにも表示されずに終了)

# do while文

- while文は処理が一度も実行されないことがある
- **do while文**は処理が必ず一度は実行されるように拡張した繰り返し構文
  1. {}内の処理を実行した後に条件判定
  2. 条件を満たす間だけ処理を繰り返す

// do while文の文法

```
do {  
    繰り返す処理をここに記述  
} while(条件をここに記述);
```

# 例1) do while文

```
1 #include <stdio.h>
2
3 int main() {
4     int money = 1000500; // 貯まったお金
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 }
```

実行結果 :

1ヶ月目 : 1000500円

# 例1) 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/while/do whileの使い分け

## 1. for文

- 繰り返し回数があらかじめ決まっている場合

## 2. while文

- 繰り返し回数が決まっていない場合
- 無限に繰り返したい場合

## 3. do while文

- 繰り返す処理を最低1回は実行したい場合

どの繰り返し構文を使っても同じ処理を書くことはできるが、うまく使い分けると読みやすいプログラムになる

# 演習

- 演習0-2 : do while文の練習
- 演習3-5 : どの繰り返し構文を使っても良い  
(適切な構文を選ぶ)
- 放課後(16:00以降)も演習室を開けています  
ぜひ活用してください！  
(木曜日以外も誰か先生に言えば開けて  
もらえます)

# 演習0

- 前回の演習0（while文を用いた500円玉貯金）の4行目を以下に変更すると、なにも表示されずにプログラムが終了することを確認しなさい

4行目： `int money = 1000500;`

# 演習1

- 今週の講義資料の例1を入力し、do while文を使うと最低1回は{}内の処理が実行されることを確認しなさい



# 演習2-1

- 0が入力されるまで、int型の変数sumに値を足し続けるプログラムを作成しなさい  
(今回はwhile文で作成しましたが今回はdo while文を用いて記述しなさい)
- 値はキーボードから入力すること

実行例(下線はキーボードからの入力を示す))

\$ ./a.out

現在のsum: 0

5

現在のsum: 5

3

現在のsum: 8

9

現在のsum: 17

0

終了します。

## 演習3：2乗計算

- キーボードから整数値を入力すると、その数までの2乗値を一覧するプログラムを作成しなさい

実行例：

\$ ./a.out

整数を入力してください。

4

1 \* 1 = 1

2 \* 2 = 4

3 \* 3 = 9

4 \* 4 = 16

## 演習4:n乗計算

- 整数値nを入力すると、2のn乗を表示するプログラムを作成しなさい

実行例)

\$ ./a.out

整数を入力してください。

10

$2^{10} = 1024$

## 演習5：素数判定

- 1から1000までの整数のうち素数のみを表示するプログラムを作成しなさい。  
(なお、素数とは1とその数以外では割り切れない数です。1, 3, 5, 7, 11, 13, 17, 19などが素数です。)

# 次回

- 次回は繰り返し構文のまとめとして以下の内容を扱います
  - 繰り返し構文の入れ子  
(if文の入れ子と同じように繰り返しの中で繰り返し構文を使います)
  - 繰り返し構文を用いたやや複雑なプログラム