





# プログラミング概論

<http://bit.ly/kosen01>

Week9@後期  
2016/11/24

# 全体像-プログラミング概論-

1.  C言語プログラムの実行方法  
(コンパイルcc、実行a.out)
2.  変数  
(int, float, char)
3.  条件分岐  
(if文、switch文、条件式、論理演算)
4.  繰り返し  
(for文、while文、do while文)
5. 関数
6. 配列

# 本日の内容

- 講義
  - 復習: 関数とは？
  - 引数と戻り値
  - グローバル変数とローカル変数
- 演習
  - 引数と戻り値を使ったプログラム

# 本日の内容

- 講義
  - 復習: 関数とは?
  - 引数と戻り値
  - グローバル変数とローカル変数
- 演習
  - 引数と戻り値を使ったプログラム

# (復習)関数とは？

- 関数(function) :  
よく使う処理をまとめ、簡潔な記述で  
呼び出す機能
- 関数について学ぶべき内容
  1. 関数の宣言と呼び出し(前回)
  2. 関数の戻り値と引数 (今回)
  3. 関数の再帰呼び出し(次回)

# 関数の動作

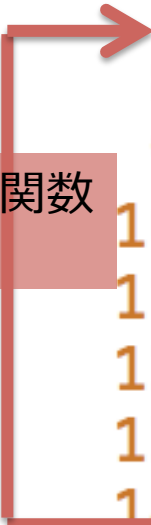
```
1 #include <stdio.h>
2
3 // 1. 形式宣言
4 void average();
5
6 // 2. 処理記述
7 void average() {
8     // 1と2と3の平均値を表示
9     printf("%d", (1+2+3) / 3);
10 }
11
12 int main() {
13     // 3. main()関数からの呼び出し
14     average();
15     return 0;
16 }
```

1. まずはmain()から実行開始

# 関数の動作

```
1 #include <stdio.h>
2
3 // 1. 形式宣言
4 void average();
5
6 // 2. 処理記述
7 void average() {
8     // 1と2と3の平均値を表示
9     printf("%d", (1+2+3) / 3);
10 }
11
12 int main() {
13     // 3. main()関数からの呼び出し
14     average();
15     return 0;
16 }
```

average()関数  
を実行



1. まずはmain()から実行開始

# 関数の動作

```
1 #include <stdio.h>
2
3 // 1. 形式宣言
4 void average();
5
6 // 2. 処理記述
7 void average() {
8     // 1と2と3の平均値を表示
9     printf("%d", (1+2+3) / 3);
10 }
11
12 int main() {
13     // 3. main()関数からの呼び出し
14     average();
15     return 0;
16 }
```

2. average()関数  
を実行

1. まずはmain()から実行開始

3. main()に戻る



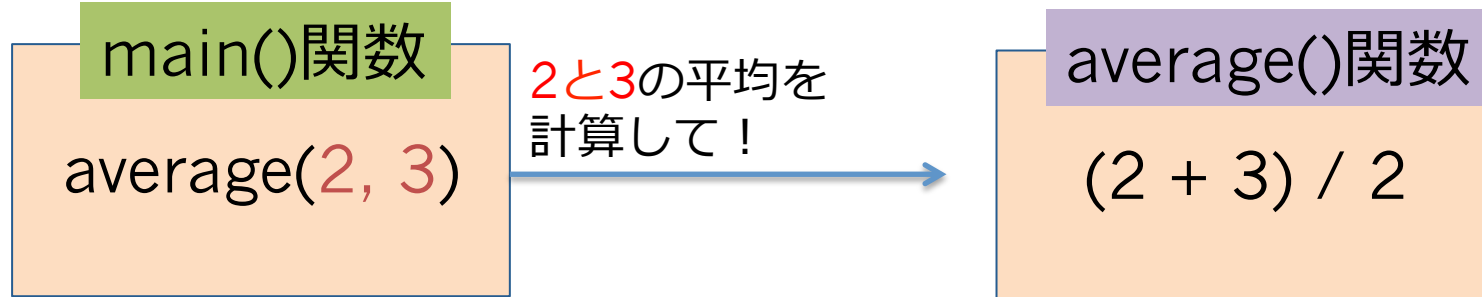
# 本日の内容

- 講義
  - 復習: 関数とは？
  - 引数と戻り値
  - グローバル変数とローカル変数
- 演習
  - 引数と戻り値を使ったプログラム

# 引数と戻り値

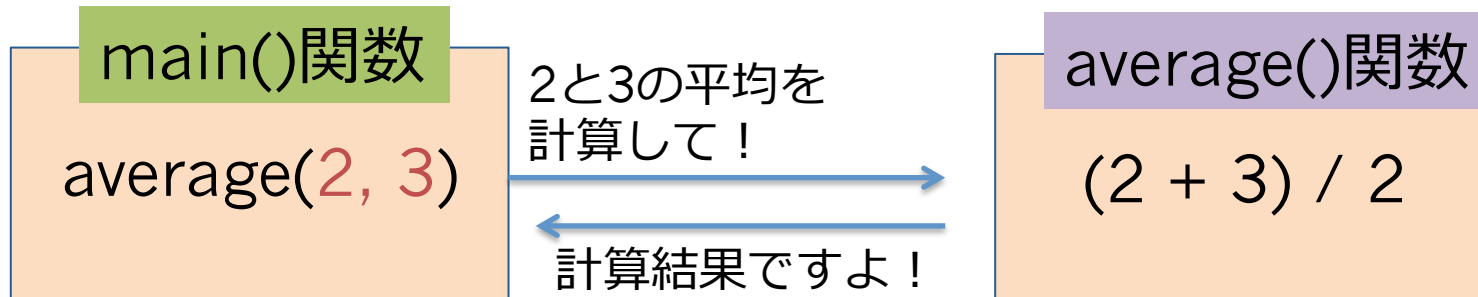
- 引数

- 関数に値を送る仕組み



- 戻り値

- 関数から値を受け取る仕組み



# 例1) 引数の利用法

```
1 #include <stdio.h>
2
3 // 1. 引数を使う場合の形式宣言
4 void average(int a, int b);
5
6 // 2. 引数を使う関数の本体
7 void average(int a, int b) {
8     result = (a + b) / 2;
9     printf("%d", result);
10 }
11
12 // 3. 引数を使う関数をmain()関数から呼び出す
13 int main() {
14     average(2, 3);
15     return 0;
16 }
```

# 例1) 引数の利用法

```
1 #include <stdio.h>
2
3 // 1. 引数を使う場合の形式宣言
4 void average(int a, int b);
5
6 // 2. 引数を使う関数の本体
7 void average(int a, int b) {
8     result = (a + b) / 2;
9     printf("%d", result);
10 }
11
12 // 3. 引数を使う関数をmain()関数から呼び出す
13 int main() {
14     average(2, 3);
15     return 0;
16 }
```

1. ()内に変数宣言  
を書く  
(変数を引数と呼ぶ)

2. 引数を使って  
処理を書く

3. 呼び出すときは  
具体的な数値等を記述

# 引数を使った場合の処理の流れ

```
1 #include <stdio.h>
2
3 // 1. 引数を使う場合の形式宣言
4 void average(int a, int b);
5
6 // 2. 引数を使う関数の本体
7 void average(int a, int b) {
8     result = (a + b) / 2;
9     printf("%d", result);
10 }
11
12 // 3. 引数を使う関数をmain()関数から呼び出す
13 int main() {
14     average(2, 3);
15     return 0;
16 }
```

← 1. main()関数から呼び出される

# 引数を使った場合の処理の流れ

```
1 #include <stdio.h>
2
3 // 1. 引数を使う場合の形式宣言
4 void average(int a, int b);
5
6 // 2. 引数を使う関数の本体
7 void average(int a, int b) {
8     result = (a + b) / 2;
9     printf("%d", result);
10 }
11
12 // 3. 引数を使う関数をmain()関数から呼び出す
13 int main() {
14     average(2, 3);
15     return 0;
16 }
```

2. 変数a, bに2, 3がそれぞれ代入

1. main()関数から呼び出される

# 引数を使った場合の処理の流れ

```
1 #include <stdio.h>
2
3 // 1. 引数を使う場合の形式宣言
4 void average(int a, int b);
5
6 // 2. 引数を使う関数の本体
7 void average(int a, int b) {
8     result = (a + b) / 2;
9     printf("%d", result);
10 }
11
12 // 3. 引数を使う関数をmain()関数から呼び出す
13 int main() {
14     average(2, 3);
15     return 0;
16 }
```

2. 変数a, bに2, 3がそれぞれ代入

3. 処理を実行してmain()に戻る

1. main()関数から呼び出される

## 例2)戻り値の利用法

```
1 #include <stdio.h>
2 // 1. 戻り値を使う場合の形式宣言
3 int average();
4 // 2. 戻り値を使う関数の本体
5 int average() {
6     int result;
7     result = (2 + 3) / 2;
8     return result;
9 }
10 // 3. 戻り値を使う関数をmain()関数から呼び出す
11 int main() {
12     printf("%d", average(2, 3));
13     return 0;
14 }
```



## 例2) 戻り値の利用法

```
1 #include <stdio.h>
2 // 1. 戻り値を使う場合の形式宣言
3 int average();
4 // 2. 戻り値を使う関数本体
5 int average() {
6     int result;
7     result = (2 + 3) / 2;
8     return result;
9 }
10 // 3. 戻り値を使う関数をmain()関数から呼び出す
11 int main() {
12     printf("%d", average(2, 3));
13     return 0;
14 }
```

1. 形式宣言と関数本体の1行目を  
main()関数に渡したい値の型にする  
(整数を渡すならint、小数ならfloat)

2. return (渡したい値)と書く

# 戻り値を使った場合の処理の流れ

```
1 #include <stdio.h>
2 // 1. 戻り値を使う場合の形式宣言
3 int average();
4 // 2. 戻り値を使う関数の本体
5 int average() {
6     int result;
7     result = (2 + 3) / 2;
8     return result;
9 }
10 // 3. 戻り値を使う関数をmain()関数から呼び出す
11 int main() {
12     printf("%d", average(2, 3));
13     return 0;
14 }
```

1. average()関数を呼び出し

# 戻り値を使った場合の処理の流れ

```
1 #include <stdio.h>
2 // 1. 戻り値を使う場合の形式宣言
3 int average();
4 // 2. 戻り値を使う関数の本体
5 int average() {
6     int result;
7     result = (2 + 3) / 2;
8     return result;
9 }
10 // 3. 戻り値を使う関数をmain()関数から呼び出す
11 int main() {
12     printf("%d", average(2, 3));
13     return 0;
14 }
```

2. 処理を実行

1. average()関数を呼び出し

# 戻り値を使った場合の処理の流れ

```
1 #include <stdio.h>
2 // 1. 戻り値を使う場合の形式宣言
3 int average();
4 // 2. 戻り値を使う関数の本体
5 int average() {
6     int result;
7     result = (2 + 3) / 2;
8     return result;
9 }
10 // 3. 戻り値を使う関数をmain()関数から呼び出す
11 int main() {
12     printf("%d", average(2, 3));
13     return 0;
14 }
```

2. 処理を実行

1. average()関数を呼び出し

## 例3) 引数/戻り値の両方を使う

```
1 #include <stdio.h>
2 // 1. 戻り値/引数を使う場合の形式宣言
3 int average(int a, int b);
4 // 2. 戻り値/引数を使う関数の本体
5 int average(int a, int b) {
6     int result;
7     result = (a + b) / 2;
8     return result;
9 }
10 // 3. 戻り値/引数を使う関数をmain()関数から呼び出す
11 int main() {
12     printf("%d", average(2, 3));
13     return 0;
14 }
```

# 本日の内容

- 講義
  - 復習: 関数とは？
  - 引数と戻り値
  - グローバル変数とローカル変数
- 演習
  - 引数と戻り値を使ったプログラム

# グローバル変数とローカル変数

- ローカル変数
  - 関数内で宣言された変数のこと
  - 関数内でしか使うことが出来ない
  - これまでの授業で使っていた変数はコレ
- グローバル変数
  - 関数外で宣言された変数のこと
  - どの関数からでも使うことが出来る

例を見ていきましょう

## 例) ローカル変数

```
1 #include <stdio.h>
2 // 1. 戻り値/引数を使う場合の形式宣言
3 int average(int a, int b);
4 // 2. 戻り値/引数を使う関数の本体
5 int average(int a, int b) {
6     int result;
7     result = (a + b) / 2;
8     return result;
9 }
10 // 3. 戻り値/引数を使う関数をmain()関数から呼び出す
11 int main() {
12     printf("%d", average(2, 3));
13     return 0;
14 }
```

main()関数内で変数a, bを呼び出したり、  
値を代入することはできない



## 例4) グローバル変数

```
1 #include <stdio.h>
2 // 1. グローバル変数の定義
3 int price; // 値段
4 int weight; // 重さ
5 // 2. 関数の形式宣言
6 void buy(int amount); // 給油
7 // 3. 関数の本体
8 void buy(int amount) {
9     price = amount * 1000000; // 数量×100万円
10    weight = amount * 1000; // 数量×1000kg
11 }
12 // 4. main()関数からの呼び出し
13 int main() {
14     // グローバル変数を初期化
15     price = 0;
16     weight = 0;
17     printf("値段 : %d円 重さ : %dキログラム\n", price, weight);
18     // 車を買う関数buy()呼び出し
19     buy(5); // 車を5台買う
20     printf("値段 : %d円 重さ : %dキログラム\n", price, weight);
21     return 0;
22 }
```

## 例4) グローバル変数

```
1 #include <stdio.h>
2 // 1. グローバル変数の定義
3 int price; // 値段
4 int weight; // 重さ
5 // 2. 関数の形式宣言
6 void buy(int amount); // 給油
7 // 3. 関数の本体
8 void buy(int amount) {
9     price = amount * 1000000; // 数量×100万円
10    weight = amount * 1000; // 数量×1000kg
11 }
12 // 4. main()関数からの呼び出し
13 int main() {
14     // グローバル変数を初期化
15     price = 0;
16     weight = 0;
17     printf("値段 : %d円 重さ : %dキログラム\n", price, weight);
18     // 車を買う関数buy()呼び出し
19     buy(5); // 車を5台買う
20     printf("値段 : %d円 重さ : %dキログラム\n", price, weight);
21     return 0;
22 }
```

グローバル変数は  
main()関数からも  
buy()関数からも使える

# 変数のスコープ

変数が見える範囲のことをスコープと呼ぶ

- ローカル変数のスコープは関数内
- グローバル変数のスコープはプログラム全体

# 演習0

- 例1, 例2, 例3、例4のプログラムを入力し動作を確認しなさい。  
ファイル名は  
week9\_rei1.c,  
week9\_rei2.c,  
week9\_rei3.c  
week9\_rei4.cとすること

# 演習1-1

半径 $r$ を引数として受け取り、面積、直径、円周を返す3つの関数を作成します。

1. 空のC言語ファイル「circle.c」を作成しなさい
2. 以下の2つの関数の形式宣言を行いなさい  
int menseki(int r); // 面積を返す  
int tyokkei(int r); // 直径を返す  
int ensyu(int r); // 円周を返す

## 演習1-2

- 関数 `menseki()`、`tyokkei()`、`ensyu()` がそれぞれ半径  $r$  の円の面積 ( $r * r * 3.14$ )、直径 ( $2 * r$ )、円周 ( $2 * 3.14 * r$ ) を返すように関数を記述しなさい。
- `main()` 関数を以下のように記述し正しく動くことを確認しなさい。

```
int main() {  
    printf("面積 : %d\n", menseki(3));  
    printf("直径 : %d\n", tyokkei(3));  
    printf("円周 : %d\n", ensyu(3));  
    return 0;  
}
```

# 演習1の実行例

- 演習1が正しく実装できると以下のように出力されます。

```
$ cc circle.c
```

```
$ ./a.out
```

```
面積：28
```

```
直径：6
```

```
円周：18
```

# 演習2

- 以下の処理を行うcalc()関数を作成しなさい
  - 3つの整数を引数として受け取る
  - 引数の合計値をグローバル変数sumに代入
  - 引数の平均値をグローバル変数averageに代入
  - 引数の最大値をグローバル変数maxに代入
- main()関数から以下のようにcalc()関数を呼び出し、sum/average/maxをprintf()で表示

```
calc(4, 5, 6);
printf("合計： %d\n", sum);
printf("平均： %d\n", average);
printf("最大値： %d\n", max);
```



# 演習2の実行例

\$ ./a.out

合計：15

平均：5

最大値：6

# 次回

- 関数から関数を呼び出す
  - これまではmain()関数から自作関数を呼び出していました。次回は自作関数から自作関数を呼び出して使います。
- 再帰呼び出し
  - 関数の中で自分自身を再度呼び出すことを「再帰」といいます。
  - 「再帰」を使うとどのようなことができるかみていきます。