

プログラミング応用  
<http://bit.ly/kosen02>

Week4@後期  
2016/10/13

# 本日の内容

- 講義
  - 先週の内容の補足
- 演習(先週の続き)
  - ペアプログラミング
  - Gitを用いた共同開発

# 補足：別ファイルに関数を書く

- 関数を別ファイルに記述し呼び出す場合  
以下の3つのファイルが必要
  1. 関数を呼び出して使うプログラム  
(例えば、`test_kaijou.c`など)
  2. ヘッダーファイル(`kaijou.h`)
  3. 関数本体ファイル(`kaijou.c`)
- 別ファイルに分けた場合のコンパイル  
`$ cc test_kaijou.c kaijou.c`

# 1. 関数を呼び出して使うプログラム (test\_kaijou.c)

```
1 #include <stdio.h>
2 #include "kaijou.h"
3
4 int main(void) {
5     // 3の階乗が正しく返るかテスト
6     if (kaijou(3) == 6) {
7         printf("3の階乗：テスト成功\n");
8     } else {
9         printf("3の階乗：テスト失敗\n");
10    }
11    return 0;
12 }
```

#include “kaijou.h”のように  
ヘッダーファイルをインクルード

# 1. ヘッダーファイルの書き方 (kaijou.h)

```
1 #ifndef _INCLUDE_KAIJOU_H_  
2 #define _INCLUDE_KAIJOU_H_  
3 // 関数 kaijou のプロトタイプ宣言  
4 int kaijou(int);  
5 #endif // _INCLUDE_KAIJOU_H_
```

1, 2, 5行目はマクロ定義という記述で、  
\_INCLUDE\_(大文字のファイル名)\_H\_と書く

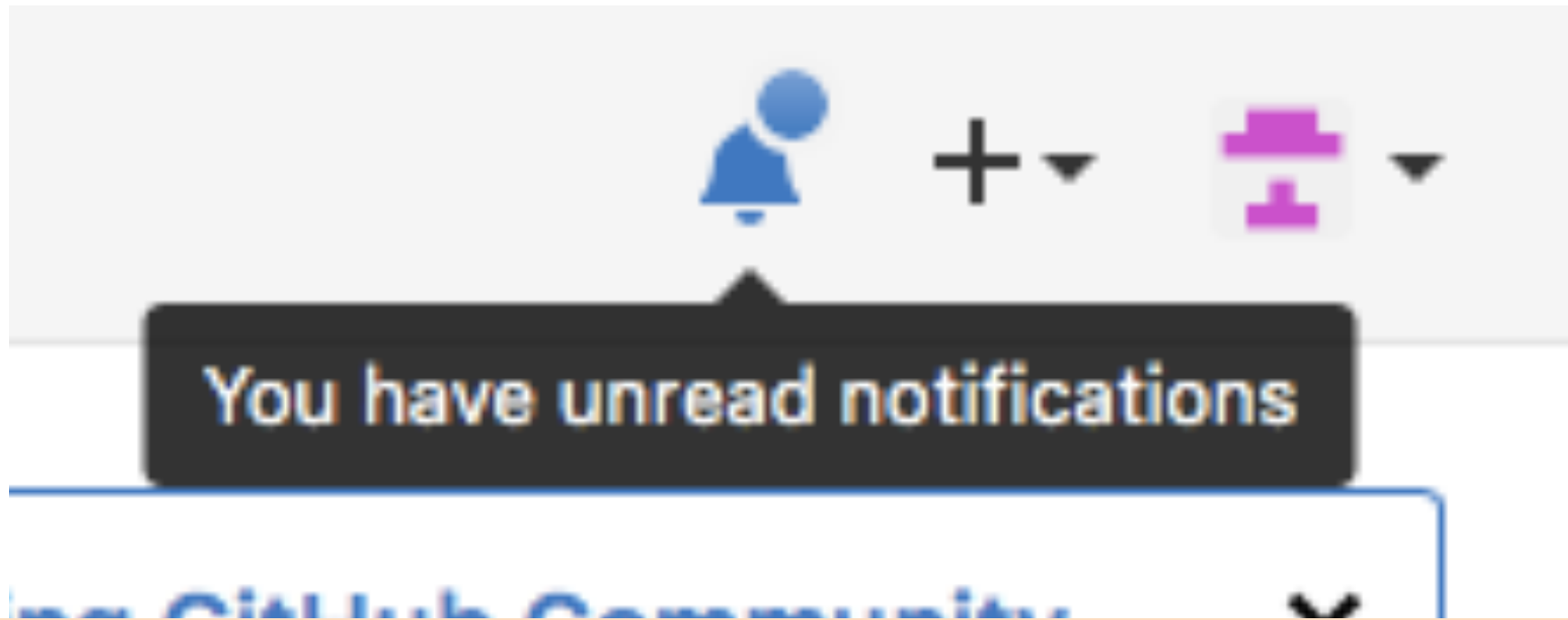
4行目に関数の形式宣言を記述

## 2. 関数本体ファイル (kaijou.c)

```
1 #include <stdio.h>
2 #include "kaijou.h"
3
4 int kaijou(n) {
5     int result = 1;
6     int i;
7     for(i=1; i <= n; i++) {
8         result *= i;
9     }
10    return 0;
11 }
```

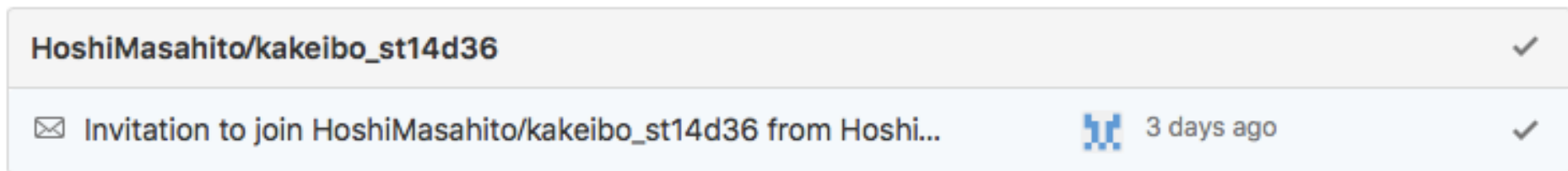
#include “kaijou.h”のようにヘッダーファイルをインクルード。関数の本体を記述する

# 演習の前に(1/3)



Collaborators(共同開発者)に新たに登録されたユーザに招待が届いているので、承認する必要がある。  
(Collaboratorsに追加されたユーザにだけ届く)  
GitHubにログインし右上の青いベルマークをクリック

# 演習の前に(2/3)



Invitation to join...の部分をクリックして開く



# 演習の前に(3/3)



**HoshiMasahito** invited you to collaborate

Accept invitation

Decline

Is this user sending spam or malicious content? You can  
[block @HoshiMasahito.](#)

Accept Invitationをクリックして開く

# 演習

- 先週の資料を見ながら進める
  - 演習1: 2人でプルリクとマージの練習
  - 演習2/演習3:
    - A: 合計値を出す関数をテスト駆動開発  
test\_sum.c/sum.h/sum.c
    - B: 平均値を出す関数をテスト駆動開発  
test\_average.c/average.h/average.c
  - 演習4: お互いコードレビューしマージ
  - 演習5: 家計簿にお互いの関数を統合
  - 演習6: 追加機能を話し合い実装

# 次回

- 共同開発を進めるための  
ソフトウェア設計技法を学習します
  - オブジェクト指向プログラミング
  - UML
  - デザインパターン
- 11/1は休講、次回までの内容が中間試験範囲です
- 演習室は放課後も使えるので活用してください！