

プログラミングII

<http://bit.ly/Prog3i>

ソースコードの管理 (1)

～GitHubの準備～

前期 第2週

2019/4/16

GitHubとは

- ▶ プログラムコードをバージョン管理するためのシステム
- ▶ GitHub Classroomは、授業で使うプログラムコード（課題）を配付したり、各自が作成した全てのプログラムを管理してくれる

今回は

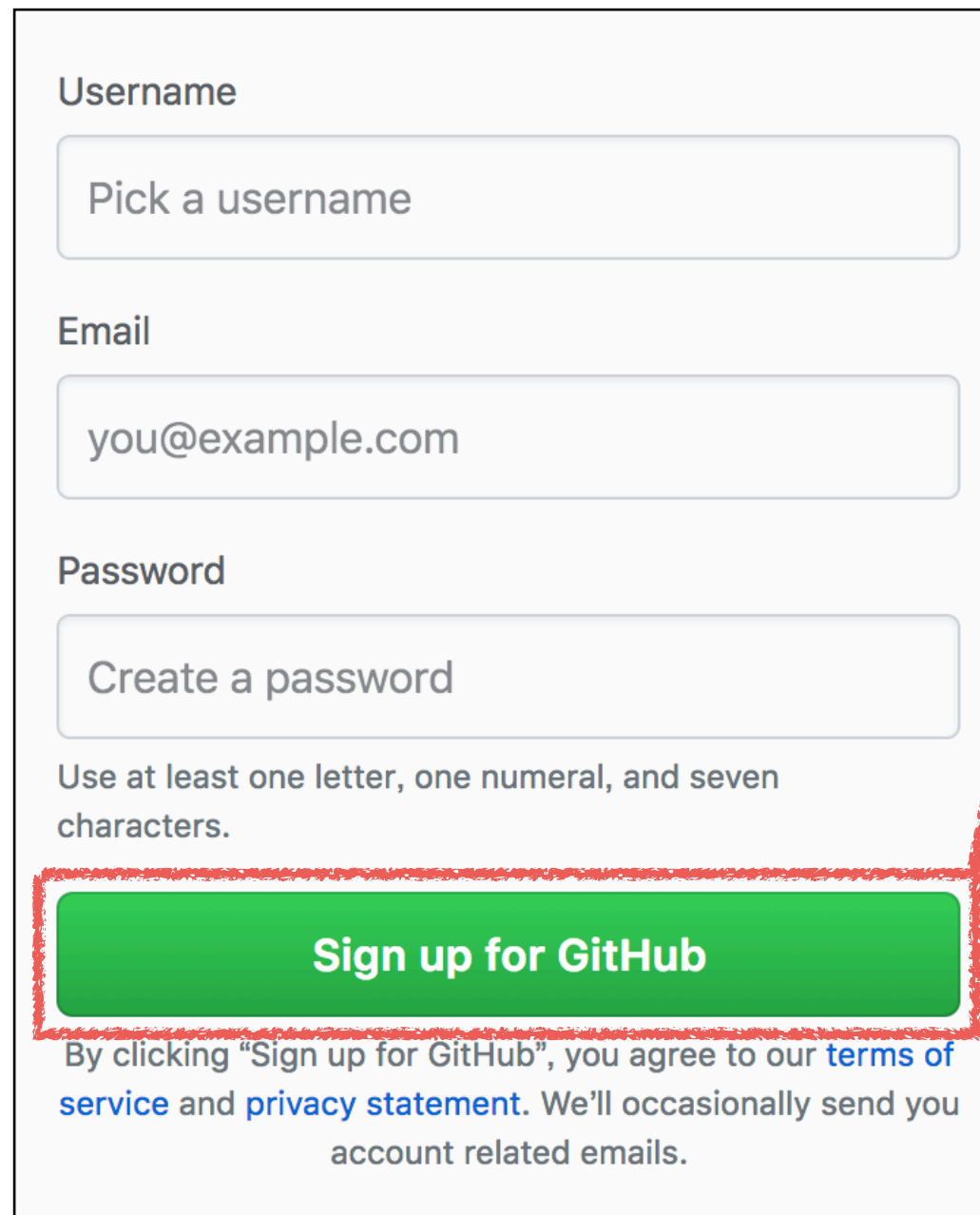
▶ **GitHubを使うための準備**

▶ **GitHubを使った基本的な操作方法**

GitHubへの会員登録

以下のWebサイトにアクセスしてアカウントを作成する

`https://github.com/`



The image shows a screenshot of the GitHub sign-up form. It includes three input fields: 'Username' with the placeholder 'Pick a username', 'Email' with the placeholder 'you@example.com', and 'Password' with the placeholder 'Create a password'. Below the password field is a note: 'Use at least one letter, one numeral, and seven characters.' At the bottom of the form is a green button labeled 'Sign up for GitHub'. A red dashed box highlights the button, and a red arrow points to it from the right with the Japanese text 'クリック' (Click).

Username

Pick a username

Email

you@example.com

Password

Create a password

Use at least one letter, one numeral, and seven characters.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

クリック

GitHubへの会員登録

Create your personal account

Username

This will be your username. You can add the name of your organization later.

Email address

We'll occasionally send updates about your account to this inbox. We'll never share your email address with anyone.

Password

Use at least one lowercase letter, one numeral, and seven characters.

By clicking "Create an account" below, you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

Create an account

← st17d??

← 本校のGmail

← パスワード

← クリック

GitHubへの会員登録

以下のページが表示されたら、そのままにして、
メールアドレス確認の処理を進める

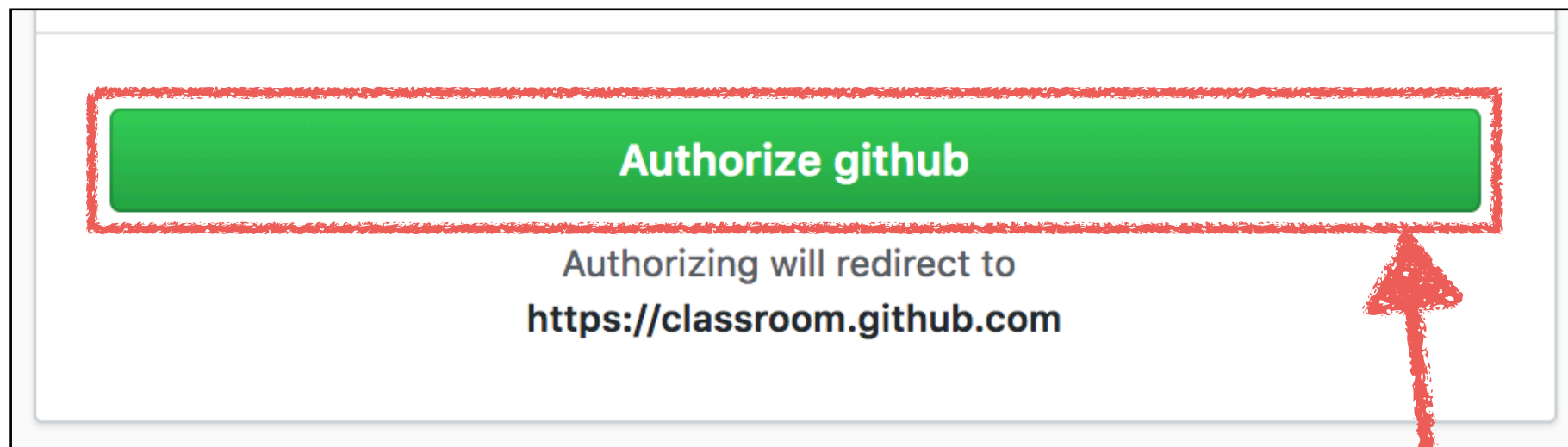


Please verify your email address

Classroomへの参加

以下のWebサイトにアクセスしてClassroomに参加する

<https://classroom.github.com/a/roS267Nr>



クリック

Classroomへの参加

課題名が表示される

Accept the assignment

Accepting this assignment will give you access to the

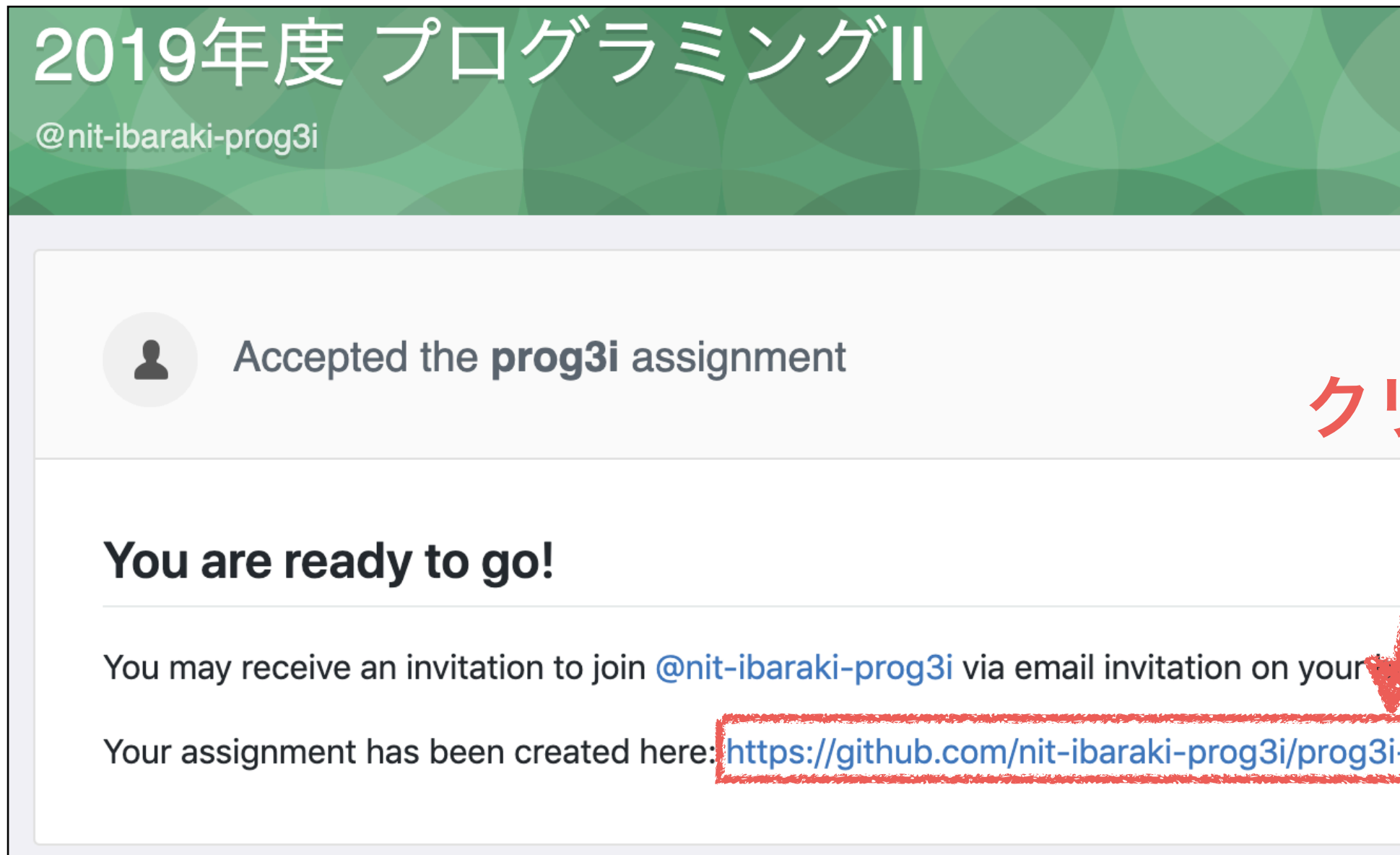
クリック

Accept this assignment

「prog3i」という課題の受け取り完了

課題提出のための設定

「prog3i」という課題の画面を表示する



The screenshot shows a GitHub notification interface. At the top, a green banner reads '2019年度 プログラミングII' with the handle '@nit-ibaraki-prog3i' below it. The main notification area has a light gray background and contains the following text: 'Accepted the prog3i assignment' next to a user icon. Below this, it says 'You are ready to go!'. Further down, it states 'You may receive an invitation to join @nit-ibaraki-prog3i via email invitation on your'. At the bottom, it says 'Your assignment has been created here:' followed by a URL 'https://github.com/nit-ibaraki-prog3i/prog3i-'. The URL is highlighted with a red rectangular box. To the right of the notification, the Japanese word 'クリック' (Click) is written in red, with a red arrow pointing down to the URL box.

2019年度 プログラミングII

@nit-ibaraki-prog3i

Accepted the **prog3i** assignment

You are ready to go!

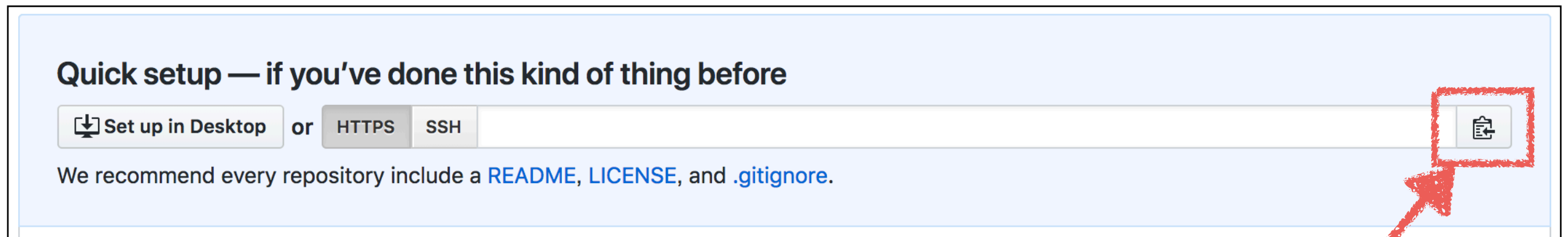
You may receive an invitation to join @nit-ibaraki-prog3i via email invitation on your

Your assignment has been created here: <https://github.com/nit-ibaraki-prog3i/prog3i->

クリック

課題提出のための設定

URLをコピーする



クリック

課題提出のための設定

端末内で以下のコマンドを入力して、**プロキシ**の設定をする
(校内ネットワークからGitHubにアクセスするために必要)

```
$ git config --global http.proxy http://po.cc.ibaraki-ct.ac.jp:3128/  
$ git config --global https.proxy http://po.cc.ibaraki-ct.ac.jp:3128/
```

作業用ディレクトリをGitHubからダウンロードする

```
$ git clone (さっきコピーしたURLをここに貼り付ける)
```

作業用ディレクトリが作られたことを確認する

```
$ ls  
(「prog3i-ユーザ名」というディレクトリが作られていれば完了)
```

課題提出のための設定

以下を入力して自分の名前とメールアドレスを設定しておく

```
$ git config --global user.name "自分の名前（ローマ字）"  
$ git config --global user.email "自分の学校用Gmailアドレス"
```

【練習2-1】

以下の流れで、最初の課題を提出してみましょう。

1. 端末を起動して、以下のコマンドを実行

```
$ cd prog3i-(ユーザ名)
```

```
$ gedit README.md
```

2. テキストエディタ内で、自分の名前を書いて保存

3. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “はじめての課題提出”
```

```
$ git push origin master
```

4. 以下の自分の課題用リポジトリを開いて、README.mdがアップロードされていることを確認する


```
https://github.com/nit-ibaraki-prog3i/prog3i-\(ユーザ名\)
```

今回の課題は…

- ▶ 自分で作ったプログラムファイルをGitHubにアップロードする流れを把握する（ソースコードのバージョン管理については次回以降）
- ▶ ついでに、関数について復習する
（以降に、D科2年生プログラミングでの資料を載せます。説明文中のページは、やさしいCのページ番号になります。）

【Point 1】 関数の名前と引数などを先にプロトタイプ宣言しておく
(p. 261)

```
1: #include <stdio.h>
2:
3: /* 関数のプロトタイプ宣言 */
4: int max(int a, int b);
5:
```



【Point 2】 戻り値の型 max(仮引数1, 仮引数2, ...)
の書式で書く (p. 218)

```
6: /* 大きい方の整数を求める関数 */
7: int max(int a, int b) {
8:   int result = 0;
9:
10:  /* 比較した結果をresultへ代入する */
11:  if (a < b) {
12:    result = b;
13:  } else {
14:    result = a;
15:  }
16:
17:  /* 比較結果を返す */
18:  return result;
19: }
```

【Point 3】

- **return**で呼び出し元に値を返す
- 変数resultの型が、関数の**戻り値の型**と一致する必要がある
- 何も値を戻さない関数の場合、戻り値の型は**void**となり、**return**を省略できる (p. 236)

【Point 4】 仮引数aに5、仮引数bに10が代入される
(p. 226~p.227)

```
20:
21: int main(void)
22: {
23:     int x, y, z;
24:
25:     /* 実引数に値を直接書く場合 */
26:     z = max(5, 10);
27:     printf("max: %d\n", z);
28:
```

【Point 5】 関数実行後は、returnの後にくる値に置き換わる（この場合はresult代入された10に置き換わる）

【Point 6】 関数max()の戻り値がzに代入される (p. 239)

```
27: printf("max: %d\n", z);
```

```
28:
```

```
29: /* 実引数に変数を使う場合 */
```

```
30: printf("x > ");
```

```
31: scanf("%d", &x);
```

```
32: printf("y > ");
```

```
33: scanf("%d", &y);
```

```
34: z = max(x, y);
```

```
35: printf("max: %d\n", z);
```

```
36:
```

```
37: /* 戻り値を代入せずにそのまま使う場合 */
```

```
38: printf("max: %d\n", max(20, 50));
```

```
39: printf("max: %d\n", max(x, y));
```

```
40:
```

```
41: return 0;
```

```
42: }
```

【Point 7】 実引数には変数も使える

【Point 8】 戻り値は変数に代入することなく、呼び出しの記述をそのまま使うこともできる

【課題の準備】

以下の流れで、課題のプログラムを作るためのフォルダを準備しましょう。

1. 端末を起動して、以下のコマンドを実行して前期第2週のフォルダを作る
\$ cd prog3i-(**ユーザ名**)
\$ mkdir week102
\$ cd week102
2. テキストエディタで、「kadai1.c」というファイルを新規に開く

【課題2-1】

関数max()を参考にして、引数で指定した2つの整数のうち小さい方を戻す関数min()のプログラムを、ファイル「kadai1.c」に作成してください。

[この関数のプロトタイプ宣言]

```
int min(int a, int b);  
/* max()の処理を参考に作ってみる */
```

[main()での処理]

```
printf("min: %d\n", min(5, 10));  
printf("min: %d\n", min(20, 10));
```

[実行例]

```
min: 5  
min: 10
```

【課題のアップロード】

以下の流れで、作ったCプログラムをGitHubにアップロードして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題2-1提出”
```

```
$ git push origin master
```

2. 自分の課題用リポジトリを開いて、week102がアップロードされているのを確認する
(以下の場所は、github.comを開いてログインしてもリンクが現れる)

[https://github.com/nit-ibaraki-prog3i/prog3i-\(ユーザー名\)](https://github.com/nit-ibaraki-prog3i/prog3i-(ユーザー名))

【課題2-2】

3つの引数の値に対する平均値を求める関数
`average()`のプログラムを、ファイル「`kadai2.c`」
に作成してください。

[この関数のプロトタイプ宣言]

```
double average(double a, double b, double c);  
/* 3つの引数の平均値を求め、この関数の戻り値とする */
```

[`main()`での処理]

```
printf("average: %lf\n", average(30, 20, 40));  
printf("average: %lf\n", average(15, 8, 21));
```

[実行例]

```
average: 30.000000  
average: 14.666667
```

【課題のアップロード】

以下の流れで、作ったCプログラムをGitHubにアップロードして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m "課題2-2提出"
```

```
$ git push origin master
```

2. 自分の課題用リポジトリを開いて、week102がアップロードされているのを確認する
(以下の場所は、github.comを開いてログインしてもリンクが現れる)

[https://github.com/nit-ibaraki-prog3i/prog3i-\(ユーザー名\)](https://github.com/nit-ibaraki-prog3i/prog3i-(ユーザー名))

【課題2-3】

2つの整数を指定した演算方法で計算する関数
op_num()のプログラムを、ファイル「kadai3.c」
に作成してください。

[この関数のプロトタイプ宣言]

```
int op_num(int a, int b, char op);
```

```
/* aとbを、opで指定した文字に応じて次のように計算し、*/
```

```
/* その結果を戻り値とする */
```

```
/*     • opが '+' の場合、 加算する */
```

```
/*     • opが '-' の場合、 減算する */
```

```
/*     • opが '*' の場合、 乗算する */
```

```
/*     • opが '/' の場合、 除算する（整数値として計算してよい） */
```


【課題2-3】

[main()での処理]

```
printf("op_num: %d\n", op_num(8, 3, '+'));  
printf("op_num: %d\n", op_num(6, 9, '*'));  
printf("op_num: %d\n", op_num(8, 3, '/'));  
printf("op_num: %d\n", op_num(2, 7, '-'));
```

[実行例]

```
op_num: 11  
op_num: 54  
op_num: 2  
op_num: -5
```

【課題のアップロード】

以下の流れで、作ったCプログラムをGitHubにアップロードして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題2-3提出”
```

```
$ git push origin master
```

2. 自分の課題用リポジトリを開いて、week102がアップロードされているのを確認する
(以下の場所は、github.comを開いてログインしてもリンクが現れる)

[https://github.com/nit-ibaraki-prog3i/prog3i-\(ユーザー名\)](https://github.com/nit-ibaraki-prog3i/prog3i-(ユーザー名))

まだ余裕のある人は…【課題2-4】

2つの整数a, bの最大公約数を求める関数gcd()のプログラムを、ファイル「kadai4.c」に作成してください。

最大公約数は、次のようなユークリッドの互除法を利用すると求めることができます。

- (1) aからbを割った余りtを求める
- (2) aにbを代入する
- (3) bにtを代入する
- (4) 上記の処理を、bが0になるまで繰り返す (bは必ずいつか0なる)
- (5) 繰り返し終了時、つまりbが0となった時のaの値が最大公約数

(参考Webサイト)

<http://math-arithmetic.blogspot.jp/2011/01/blog-post.html>

【課題2-4】

[この関数のプロトタイプ宣言]

```
int gcd(int a, int b);
```

```
/* 繰り返し処理によって、前述のユークリッドの互除法を作る */
```

[main()での処理]

```
printf("gcd: %d\n", gcd(18, 12));  
printf("gcd: %d\n", gcd(221, 323));
```

[実行例]

```
gcd: 6  
gcd: 17
```

【課題のアップロード】

以下の流れで、作ったCプログラムをGitHubにアップロードして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題2-4提出”
```

```
$ git push origin master
```

2. 自分の課題用リポジトリを開いて、week102がアップロードされているのを確認する
(以下の場所は、github.comを開いてログインしてもリンクが現れる)

[https://github.com/nit-ibaraki-prog3i/prog3i-\(ユーザー名\)](https://github.com/nit-ibaraki-prog3i/prog3i-(ユーザー名))