

# 「LINEBOT を作ってみよう」テキスト

2019 年 2 月 23 日

株式会社シオラボ

## 1. 講座の目的

今回の講座は、学習の成果を社会課題の解決等につなげる「課題解決型」の ICT 講座です。「ICT を様々な事象に活用し、自ら率先して課題を解決できる人材」を育成することを目的としています。

(ア)この講座で意識してほしいこと

- 自由な意見、自由な発想
- みんなで力を合わせる
- 分からなくなったら止まって質問する
- 世の中を変えよう

(イ)あまり意識しなくてよいこと

- プログラミング言語「Ruby」を理解すること
- 他の人の進み具合

## 2. 開発するシステムの概要

今回は、商品名を入れると金額を返す「LINE ボット」を作ります。

### (ア)LINE ボットとは

コミュニケーションアプリ「LINE」で動くチャットボットサービスのことで、ユーザーのメッセージに応じて自動応答するプログラムの総称です。LINE で「友だち追加」して使います。

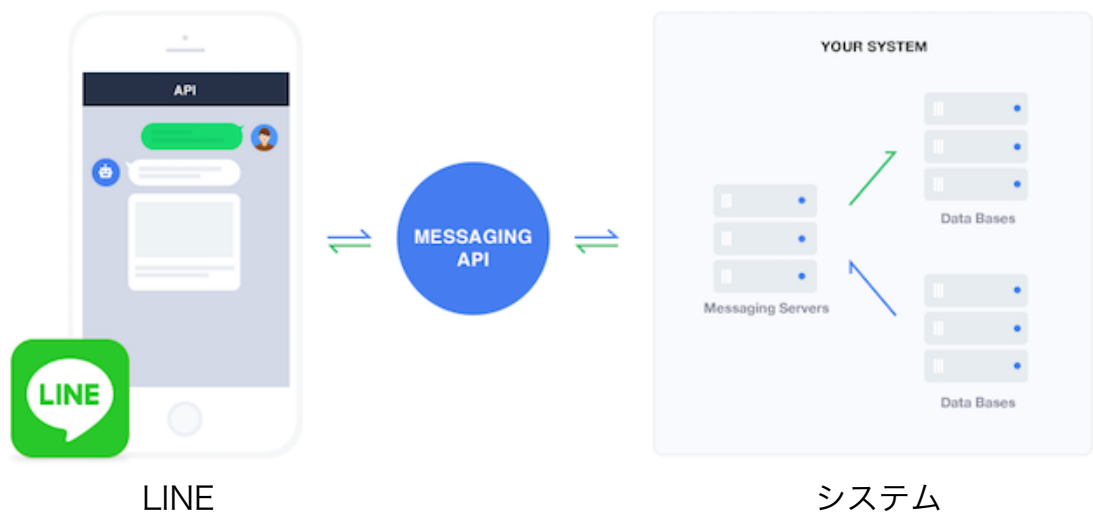
### (イ)LINE ボットの事例

#### 「ローソン公式アカウント」

天気予報、しりとり、新商品情報、近くのローソンを探す、将棋などができる。



### (ウ)仕組み



1. LINE でメッセージを入力して送信する。例えば「りんご」
2. システムでは、そのメッセージを受け取り、解釈して、適当な返事を作る。例えば「100 円」
3. 作成された返事を LINE に送る。
4. LINE では返事を受け取り、表示する。

今回の講座で作成するのは、2 の部分です。

### (エ)LINE ボット「シオラボット」

LINE ボットを用意したので、以下の QR コードを読み込んで、友だち追加してください。



追加できたら、「りんご」とメッセージを送ってみてください。

### 3. 設計

#### (ア)担当を決める

開発は分担しておこないます。以下のカテゴリをそれぞれに割り当てます。

- 野菜 (vegetable)
- 果物 (fruit)
- 魚 (fish)
- その他の食品 (foods)
- ドラッグストア (drugstore)

自分の担当が決まったら、自分が担当するカテゴリの英単語を、シオラボットにメッセージしてみてください。（魚担当なら「fish」を送る）日本語が返ってきましたか？

#### (イ)仕様を決める

担当が決まったら、どのようなメッセージを受け取ったら、どのように返信するかを決めていきます。

チラシがあるので、そこから情報を拾って、ブルーとピンクの付箋が対応するように書きましょう。最初は単語レベルでよいです。

LINE が送信するメッセージ：ブルーの付箋（例：りんご）

LINE に返信するメッセージ：ピンクの付箋（例：100 円）

書いたら、壁に貼ります。これを 1 人 10 セットぐらい作ります。

#### (ウ)仕様をレビューする

壁に貼られている付箋を見て、ブルーの付箋に重複がないか確認しましょう。ブルーの付箋に書かれている言葉が重複すると、どちらを返信すればよいか、コンピュータが判断できずに困ってしまうからです。

## 4. 開発

(ア) ブラウザ（Google Chrome）を起動します

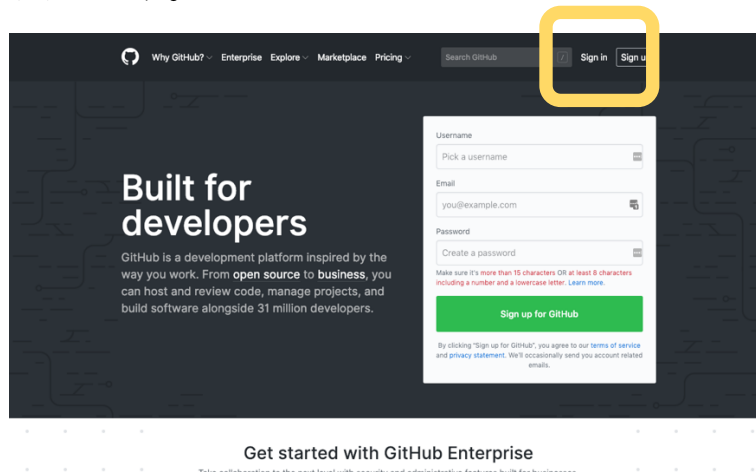
(イ) GitHub（ギットハブ）にログインします

今回の開発はブラウザのみでおこないます。また、アクセスするサイトは、ソフトウェア開発のプラットフォーム GitHub のみです。

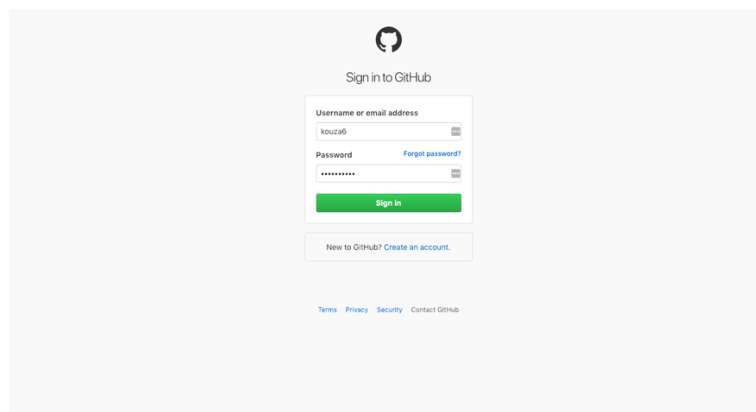
ブラウザのアドレスバーに、次の URL を入力してアクセスします。

<https://github.com/>

アクセスすると次のように表示されるので、右上の「Sign in」をクリックします。



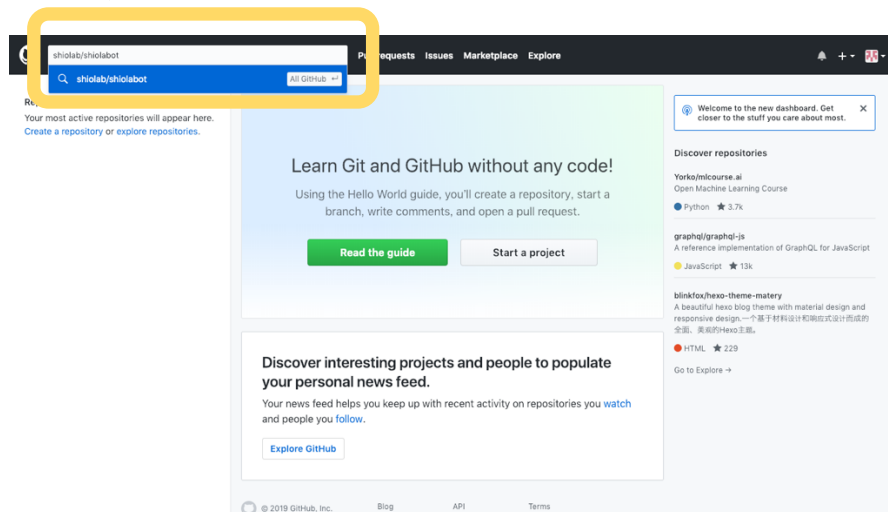
次のようなログイン画面になります。配布したアカウントとパスワードを入力して、「Sign in」をクリックしてください。



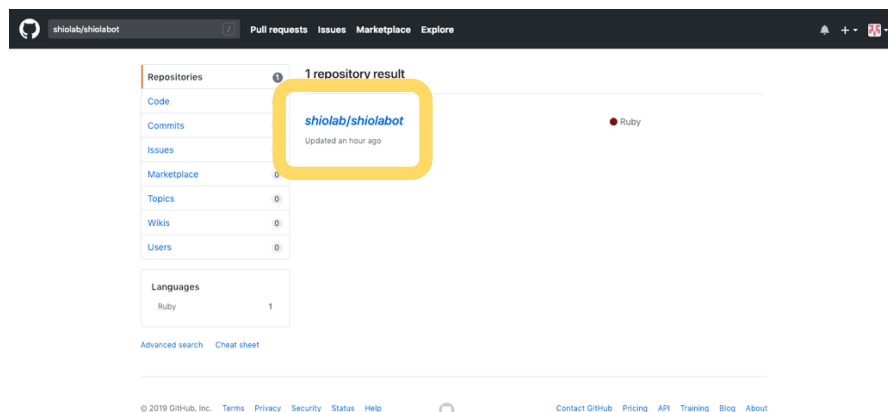
### (ウ) プログラムをフォークします

みんなで開発を進めるため、共通のベースとなるプログラムが必要です。それを自分の作業環境にコピーして使います。（フォークといいます）

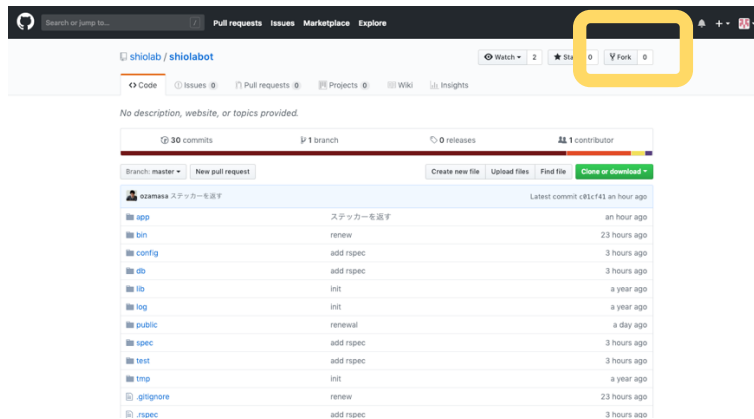
ログイン後の画面左上の入力欄に「shiolab/shiolabot」と入力してください。



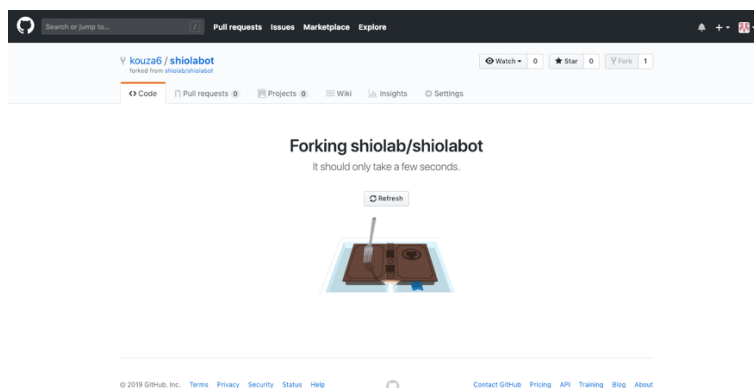
入力欄の下に、青字でハイライトされた「shiolab/shiolabot」が出てくるので、それをクリックします。すると次のような画面になります。



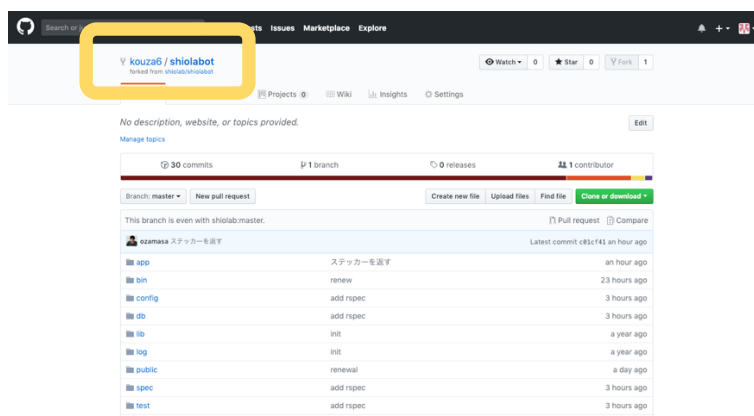
画面に表示されている「shiolab/shiolabot」をクリックします。すると次のような画面になります。



右上にある「Fork」というボタンを押します。すると、次のような画面になります。

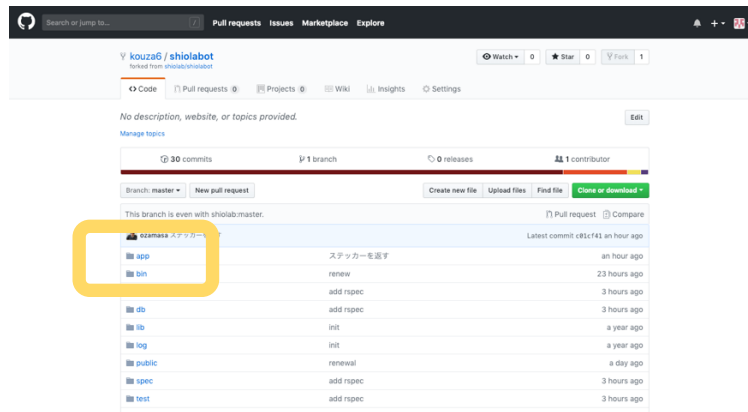


フォークが完了すると、自分のアカウントに開発環境がコピーされます。自分のアカウントが表示されていることを確認してください。

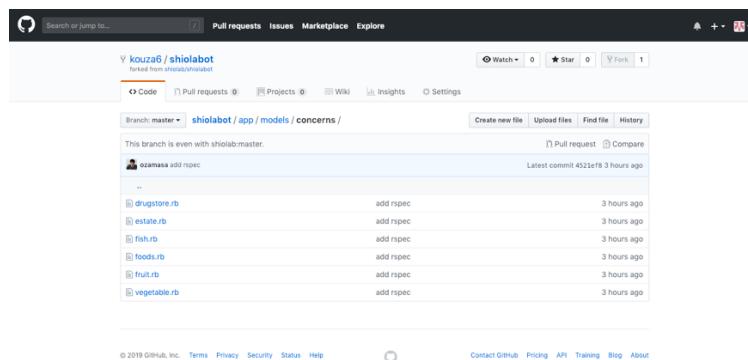




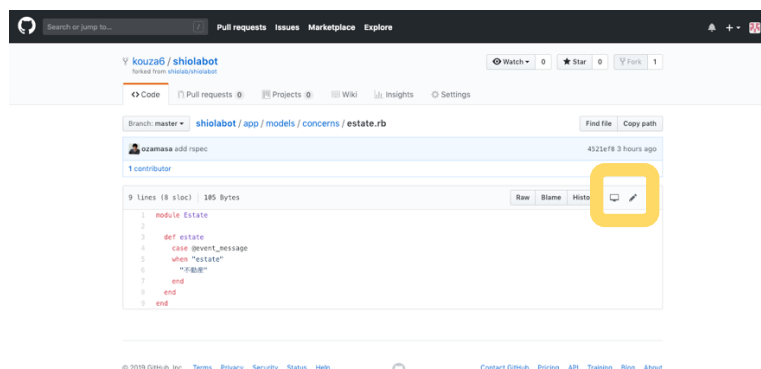
(エ) プログラムを書くファイルを表示します  
「app」 → 「models」 → 「concerns」  
の順にクリックしていきます。



移動できたら、次のような画面になっていることを確認してください。

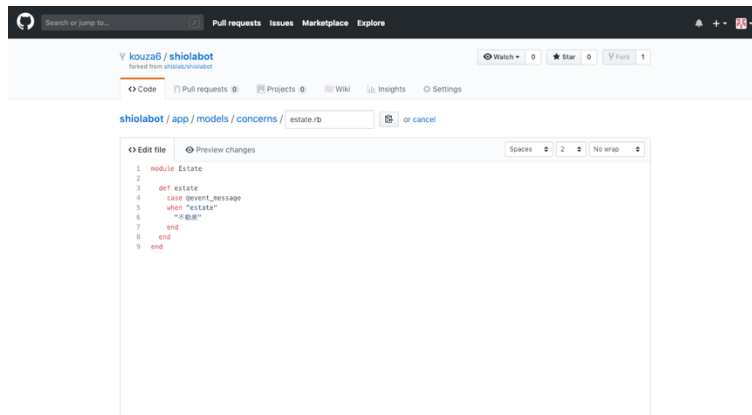


設計で分担したカテゴリを選択します。野菜担当なら  
「vegetable.rb」です。



(オ)いよいよプログラムを書きます

上記の画面で、右上に表示されている鉛筆マークを押して編集状態にします。



付箋を 1 セット（ブルーとピンク）持ってきて、以下のように入力します。

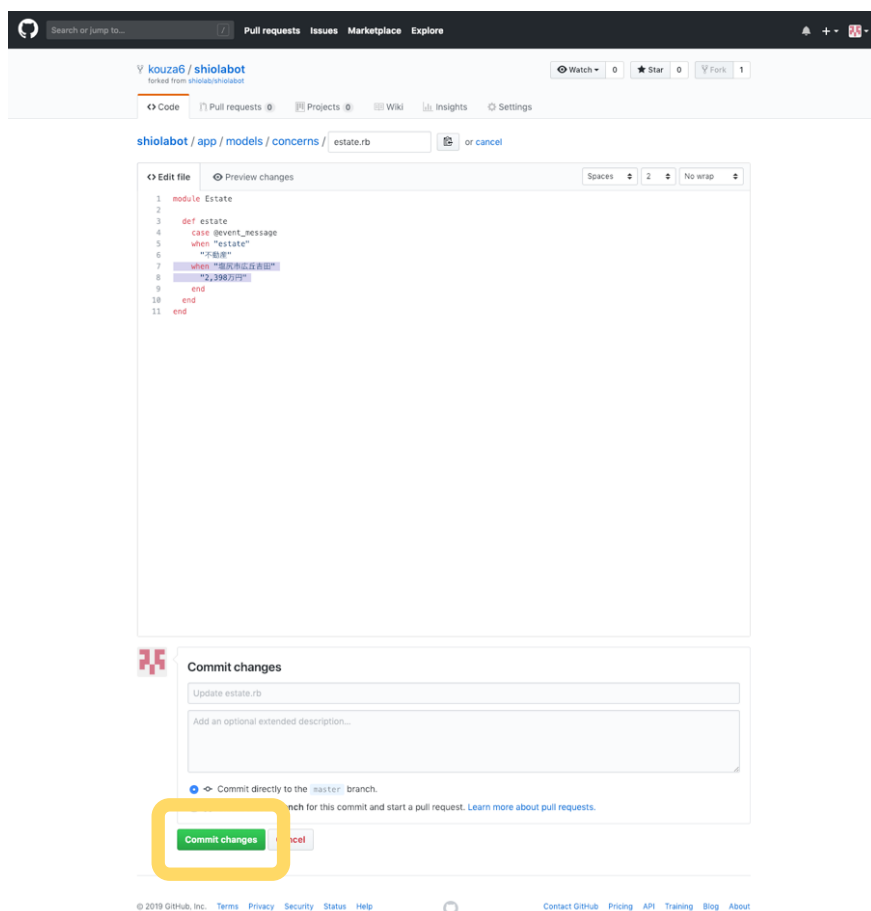
- 7 行目の先頭にフォーカスを移動して改行（空白行ができる）
- 空白行となった 7 行目にフォーカスを移動
- スペースを押して移動
- ブルーの付箋「みかん」、ピンクの付箋「50 円」の場合、以下のように入力

```
when "みかん"  
  "50 円"
```

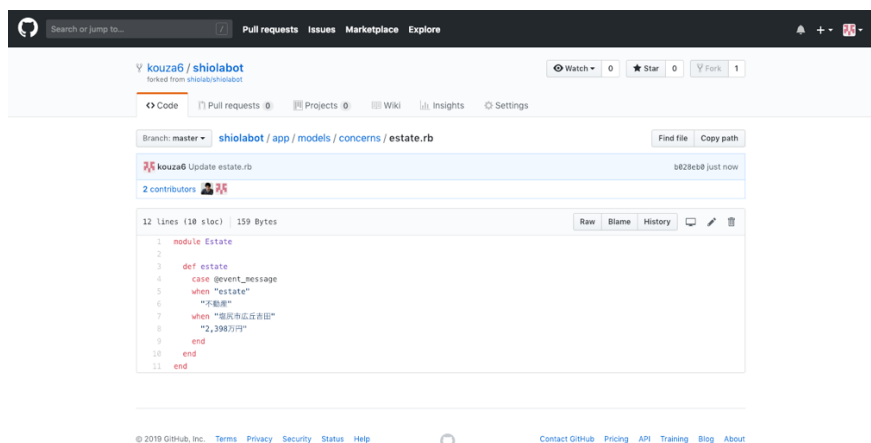
#### ※注意点

- この構文は、「”みかん”のとき（when）、”50 円”を返す」と読みましょう。
- プログラムは、基本的に半角英数字を使って書きます。一部を除いて全角で入力することはできません。
- その一部とは、ダブルクォーテーション (") で囲まれた箇所です。ダブルクォーテーションで囲まれたものを「文字列」と呼び、その中では全角を使えます。
- コンピュータは融通が利かないので、「みかん」と「ミカン」を、同じものと解釈できません。ここでは、付箋に書いた通りに記述しておきましょう。

次のようになります。ハイライトされているところが入力したプログラムです。



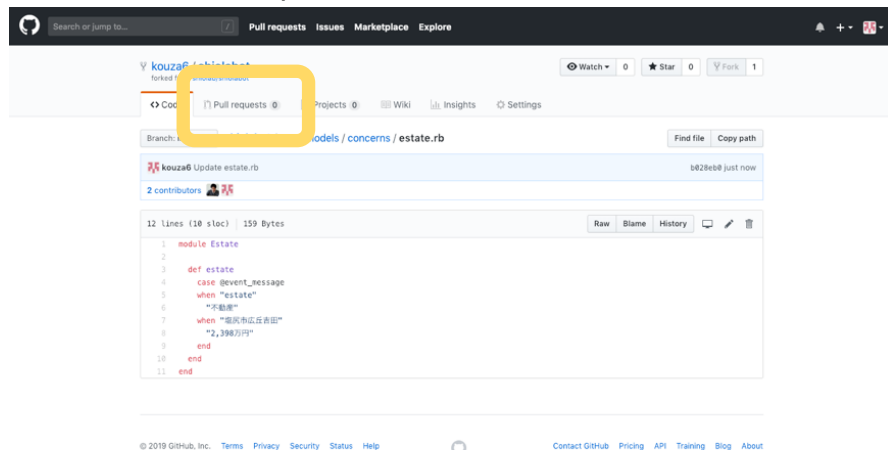
画面下にある「Commit changes」を押して、プログラムを保存します。保存すると、編集前の画面に戻ります。



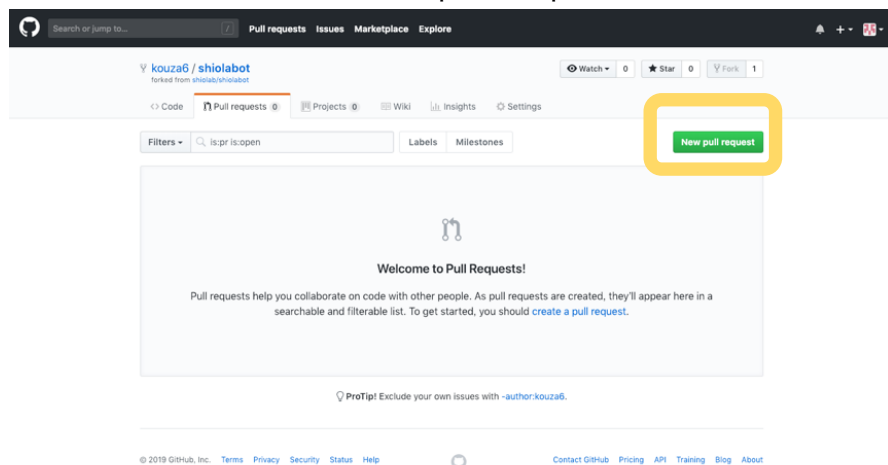
(カ) プログラムを反映します

プログラムを書いただけではシステムに反映されません。フォークしてきたベースのリポジトリに取り込むという作業が必要になります。

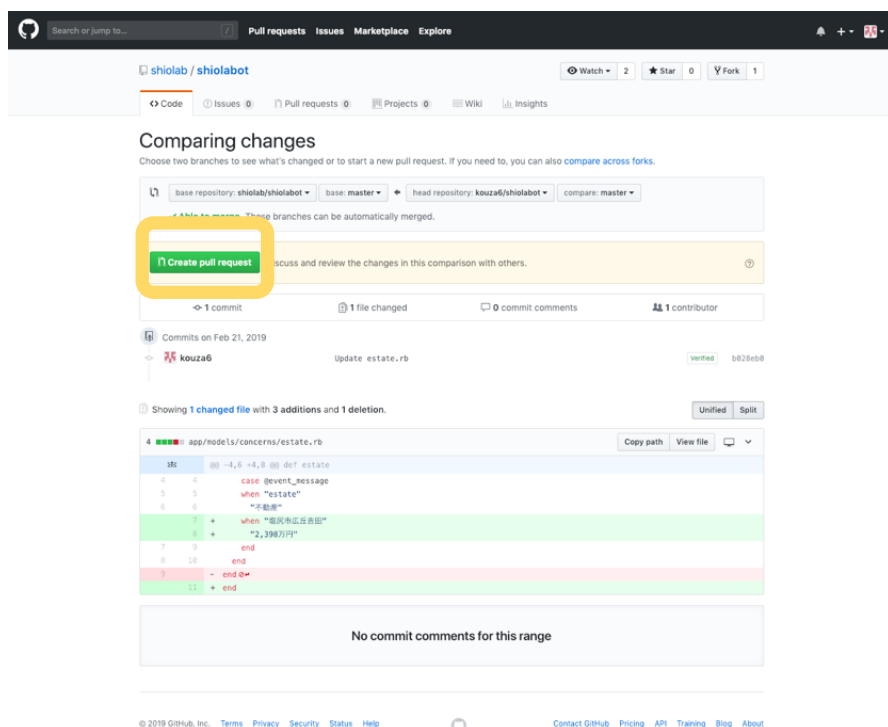
画面上の「Pull requests」と書かれているタブをクリックします。



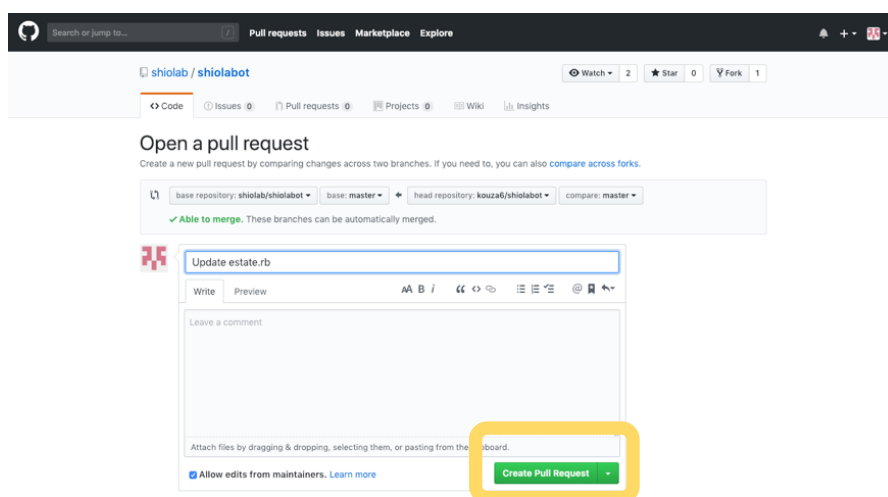
さらに、右上の緑色の「New pull request」をクリックします。



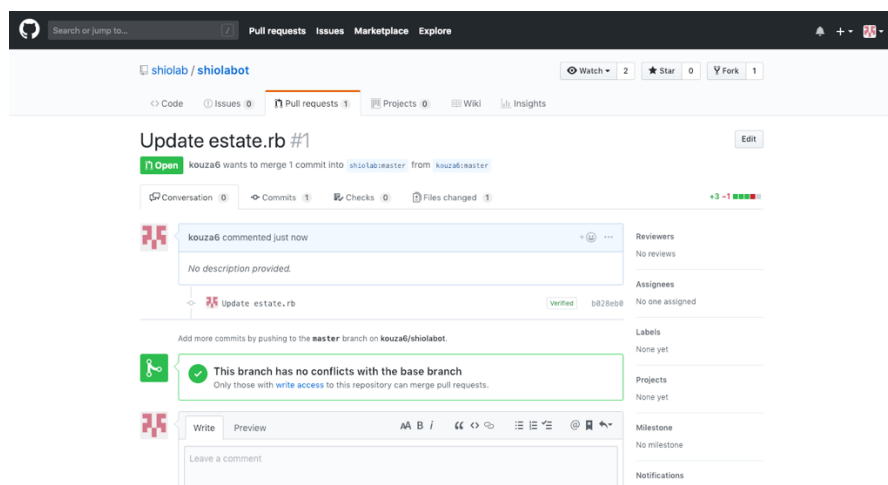
今回の修正箇所が緑色で表示されます。念のため確認して、左上の緑色の「Create pull request」をクリックします。



すると、次のような画面になるので、また、緑色の「Create pull request」をクリックします。



次のような画面になります。取り込みがおこなわれるまで、少し時間をください。



(キ) プログラムを修正し続けます  
(エ)～(カ)の作業を繰り返します。

(ク) プログラムが取り込まれたら確認します  
講師がプログラムを取り込んだら声を掛けるので、自分が作成したプログラムが正常に動作しているか確認します。  
LINE を取り出し、メッセージ送信してみます。先ほどの例では「みかん」と送信します。「50 円」と返事が返ってくれば成功です。

(ケ) 改善ポイントを見つけて改良しましょう  
こうしたい、というポイントを見つけて、プログラムを修正しましょう。  
例えば、

- 「みかん」と入力されても「ミカン」と入力されても、同じ返事をしたい。
- 「50 円」だけでなく、「愛媛産🍊50 円」と返事をしたい。

## 5. 今回使った技術や仕組み

今回の講座では、さまざまな技術や仕組みを使いましたが、時間の都合上、すべて説明しませんでした。なかには、意識せずに使ってもらったものもあります。

しかし、実際に開発する場合、技術や仕組みを知っておくと、どのような場面でどのように使えば便利なのか、自分のやりたいことは実現できるのか、など、解決の糸口を見つけることができます。興味があればぜひ調べてみてください。参考サイトも記載しておきます。Google で検索しても沢山の情報を得ることができます。

また、講座が終了したあと、講師に質問していただいても OK です。もう少し知りたいとか、ぜんぜん分からないとか、なんでもお聞きください。問い合わせ先を、テキストの最後に記載しておきます。

### (ア) プログラミング言語「Ruby (ルビー)」

今回使ったのは、「文字列」「case 文」くらいでした。

コンピュータに指示を出すために、プログラミング言語には様々な機能が実装されています。コンピュータとコミュニケーションするための便利な機能を、ぜひ勉強してみてください。

- Ruby 公式サイト

<https://www.ruby-lang.org/ja/>

### (イ) Web アプリケーションフレームワーク「Ruby on Rails (ルビーオンレイルズ)」

Web アプリケーションを構築するのに、Ruby の機能だけでは貧弱なので、Ruby の機能を補うライブラリ (gem=ジェムと読みます) を使いました。それが「Ruby on Rails」です。「Ruby on Rails」は、クックパッドなどのホームページでも使われています。

- Ruby on Rails 公式サイト

<https://rubyonrails.org/>

- Ruby on Rails リポジトリ

<https://github.com/rails>

(ウ)クラウドアプリケーションプラットフォーム「Heroku（ヘロク）」  
自分の作ったシステムやアプリケーションを誰かに見せたい、使ってもらいたい、全世界に公開したい、となったとき、昔は、高価なコンピュータを自分で用意する必要がありました。しかし、現在は「クラウド」という考え方によって、自分でコンピュータを用意せずとも、簡単にアプリケーションを公開することができるようになりました。今回は「Heroku」という仕組みを使っています。  
もし、自分でアプリケーションを作ったら、Heroku を使って公開し、全世界の人々に使ってもらいましょう。

- Heroku 公式サイト  
<https://jp.heroku.com/>

(エ)ソフトウェア開発プラットフォーム「GitHub（ギットハブ）」  
自分の作ったプログラムを置いておく場所です。今回の講座で頻繁に使いました。  
ソフトウェア開発は、みんなで力を合わせて開発を進めると、とてもよいものになります。GitHub には、みんなで開発を進める仕組みが充実していますし、その他のサービスや仕組みとも簡単に連携できます。今回も、裏側で Heroku と連携していました。

- GitHub 公式サイト  
<https://github.co.jp/>

(オ)LINE ボットを作るための「LINE Messaging API」  
LINE 上で提供するボットを作成するために、Messaging API という仕組みを使いました。LINE には「応答メッセージを送る」「プッシュメッセージを送る」「さまざまなタイプのメッセージを送る」といった機能があり、相手が人間でなくても返事ができることを体験しました。いま話題の AI がもっと進化すれば、もっと柔軟なやり取りができるようになるかもしれません。

- LINE Developers サイト  
<https://developers.line.biz/ja/docs/messaging-api/>



(カ)LINE ボットのためのアカウント「LINE@アカウント」

上記の「Messaging API」を利用する、つまり、LINE ボットを作るには、LINE ボット専用の LINE アカウント（LINE@アカウント）を開設する必要があります。

- LINE@ サイト

<https://at.line.me/jp/>

## 6. お問い合わせ

講座が終了したあと、質問などは下記までお気軽にお問い合わせください。

- 株式会社シオラボ  
住所：塩尻市大門八番町 1-2 塩尻インキュベーションプラザ 106  
電話：0263-52-3211  
メール：ozamasa@shiolab.com  
URL：<https://www.shiolab.com>
- CoderDojo 塩尻  
住所：塩尻市大門八番町 1-2 7 塩尻情報プラザ  
メール：coderdojo@shiojiri.fun  
URL：<https://coderdojo.shiojiri.fun>

※CoderDojo 塩尻は、毎月第 2 水曜日と第 4 水曜日の 17 時～18 時 30 分、塩尻情報プラザで開催しています。様子はホームページをご覧ください。