

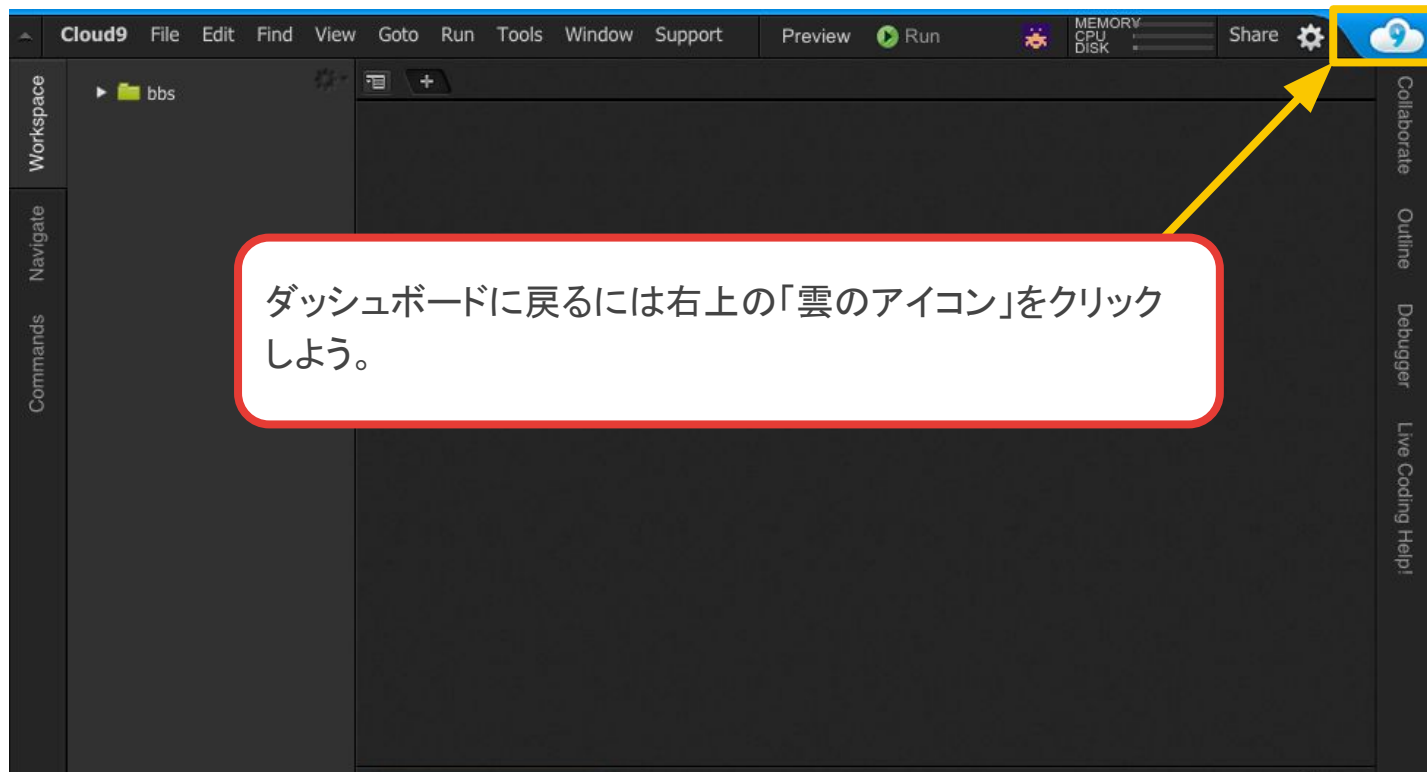
# 家計簿アプリ



オリジナルアプリ制作

Webサービス開発コース

# 開発の準備をしよう①



## 開発の準備をしよう②

### Create a new workspace

Workspace name

account\_book

1

account\_bookと入力

Hosted workspace

Clone workspace

Remote SSH workspace



**Private**

This is a workspace for your eyes only



**Public**

This will create a workspace for everybody to see

Clone from Git or Mercurial URL (optional)

e.g. ajaxorg/ace or git@github.com:ajaxorg/ace.git

Choose a template



HTML5



Node.js



Meteor



PHP, Apache &...



Python



Ruby



Wordpress



Rails Tutorial



Blank

CS50

Harvard Unive...

2

Customを選択

3

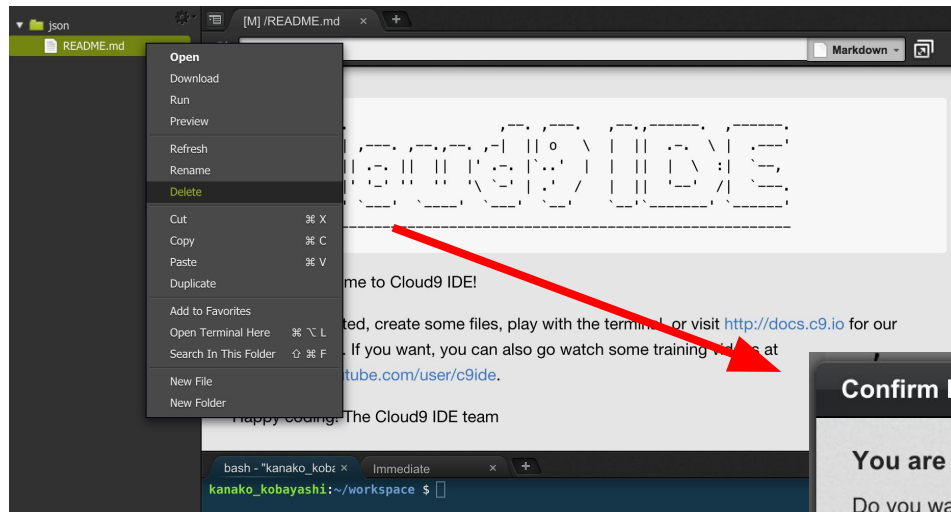
CREATEをクリック！

Create workspace

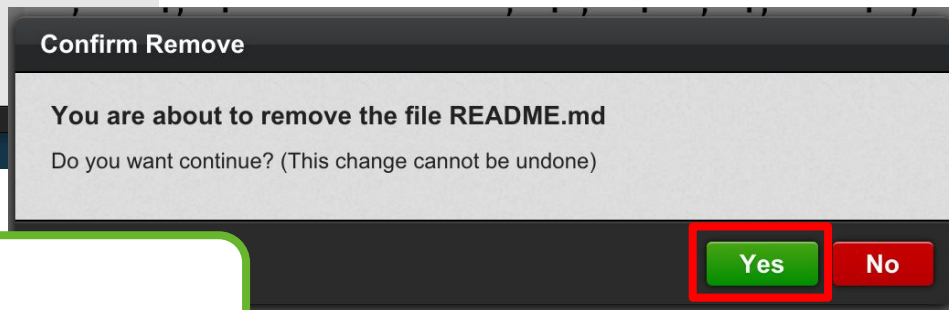
# プロジェクトを空にしよう！

プロジェクトが作成されます。

次のページで、素材ファイルを取り込むので、README.mdを削除しておこう。

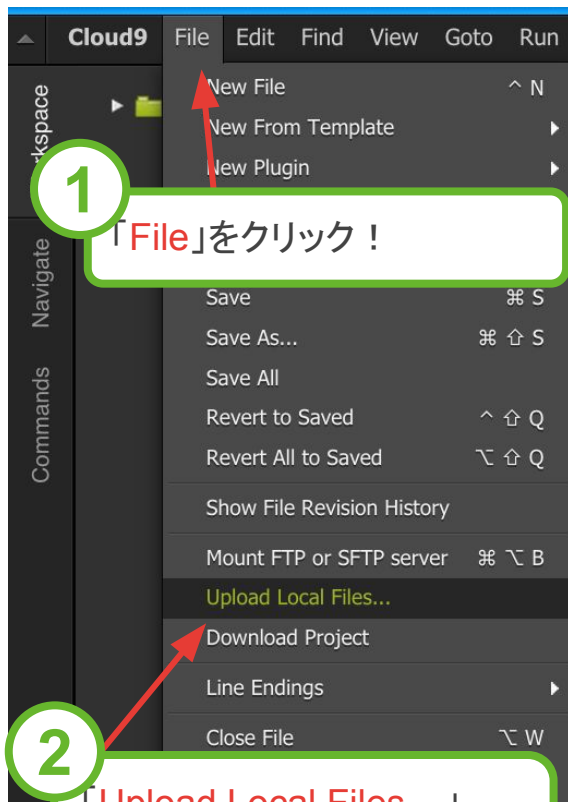


「README.md」を選択して、Controlを押しながらクリック！  
「Delete」をクリックして削除しよう。



既にかかれている「README.md」は、  
タブの「×」マークをクリックして閉じましょう。

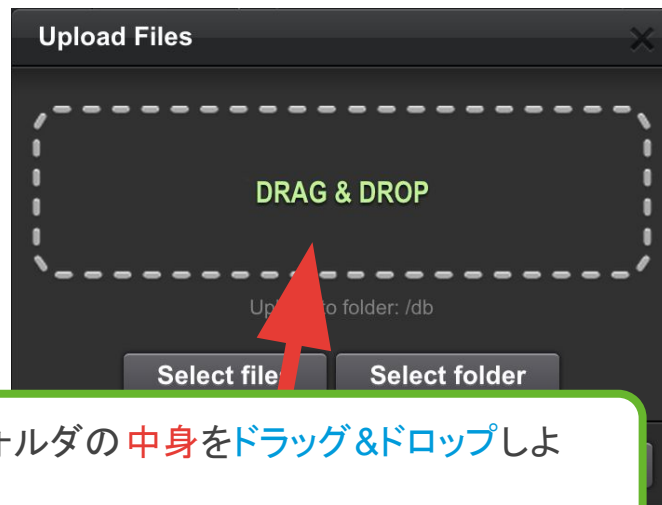
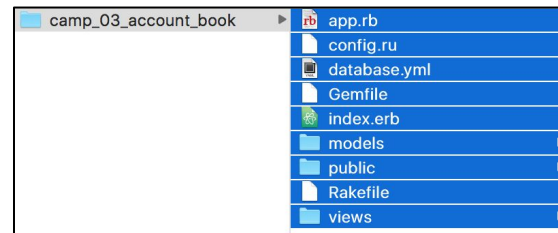
# フォルダをアップロードしよう！



「File」をクリック！

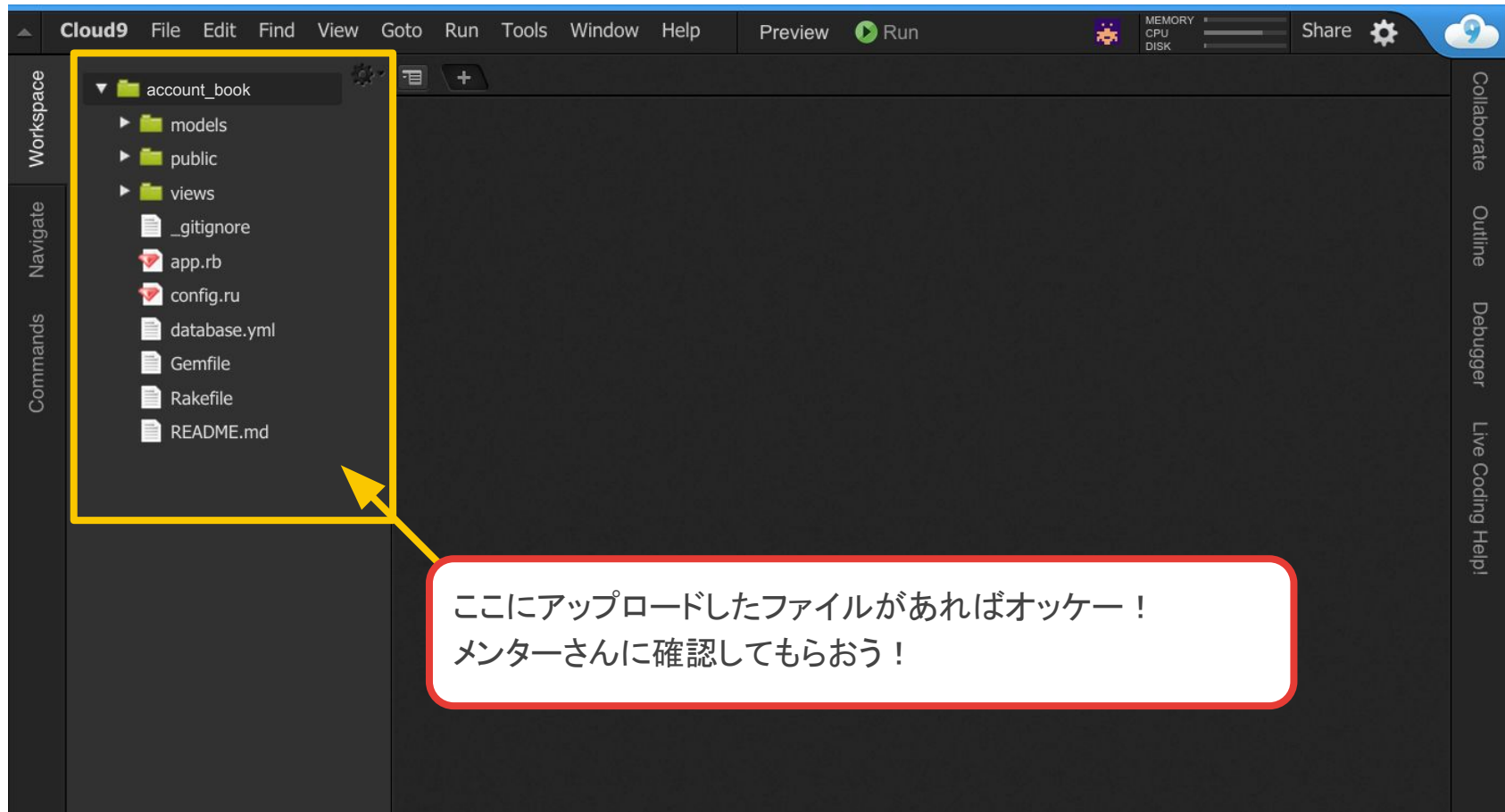
「Upload Local Files...」  
をクリック！

フォルダの **中身** はこんな風に  
選択しよう！



フォルダの **中身** をドラッグ&ドロップし  
よう！

これで準備完了です！



# 必要なプラグインをインストールしよう！

ターミナルで次の2つの **コマンド** を入力して **Enter** を押そう！

```
username@count:~/workspace $ gem install bundler
```

Enter

```
username@count:~/workspace $ bundle --without production
```

Enter



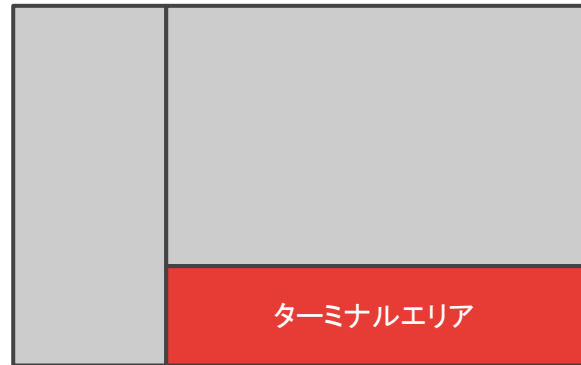
```
Fetching gem metadata from https://rubygems.org/.....
Resolving dependencies...
Installing rake 10.4.2
```

```
~~~~~
```

```
Installing sqlite3 1.3.10
Using bundler 1.7.6
Your bundle is complete!
Gems in the group production were not installed.
Use `bundle show [gemname]` to see where a bundled gem is
installed.
```

```
username@count:~/workspace $
```

cloud9のワークスペースの簡易地図



これで完了！



どのような機能が必要なんだろう？

- ・商品名と金額が登録できる
- ・現在の合計金額が表示される

今回はこの2つの機能をつけていくよ～！



# まずはデータベースの準備をしよう！

①商品名と金額が登録できる

②現在の合計金額が表示される

この2つの機能をつけるために必要なデータを考えよう！

①は最低限、**商品名(title)**と**金額(price)**が必要だから、このカラムを持つテーブルを準備しよう！

	title(string型)	price(integer型)

↑データベースのイメージ  
(ここに1列ずつデータが追加されていくよ)

まず、titleは文字列なので **string(文字列)**、priceは数字な

ので **integer(整数)** という型にするよ！

(整数:… -2, -1, 0, 1, 2 … といった符号付きの1刻みの数字！)

次のページからデータベースを作るためのファイルを作るよ！

# マイグレーションファイルを生成しよう！

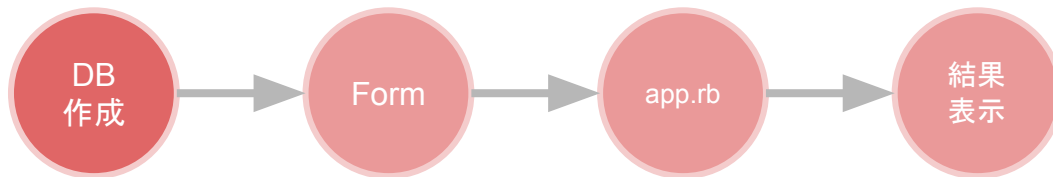
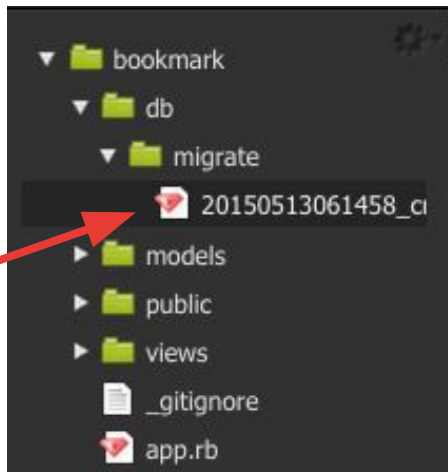
データベースをつくるのに必要なマイグレーションファイルを作っていくよ！

下のようにコマンドをターミナルに入力してEnterを押そう！

```
username@account_book::~~/workspace $ rake db:create_migration  
NAME=create_items  
db/migrate/20150515081154_create_items.rb
```

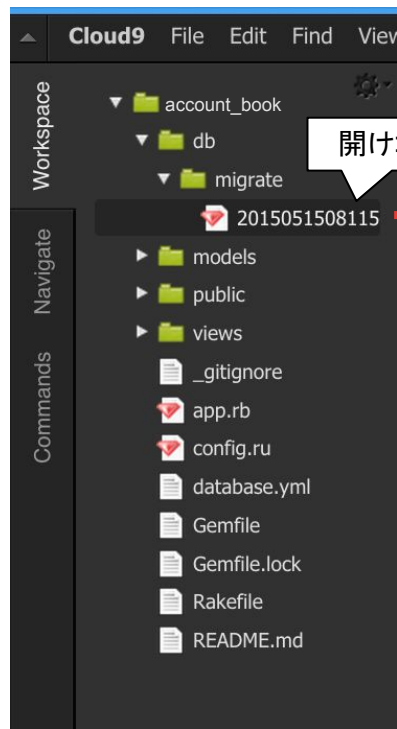
Enter

このファイルが「db/migrate」の中に生成されたよ！



# マイグレーションファイルの中身を書いていこう！

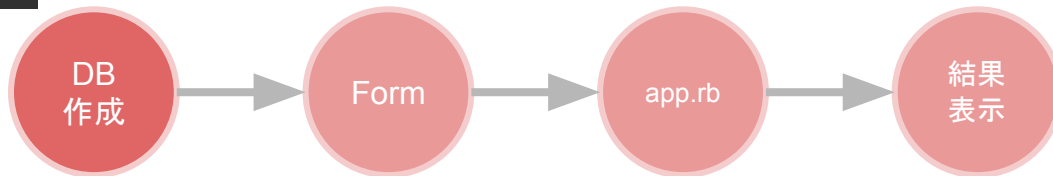
書き終わったら「command + S」または「ctrl + S」で保存しよう！



db/migrate/(日付)\_create\_items.rb

```
1 class CreateItems < ActiveRecord::Migration
2   def change
3     create_table :items do |t|
4       t.string :title
5       t.integer :price
6       t.timestamps null: false
7     end
8   end
9 end
```

- ① テーブルの作成: `create_table :テーブル名(複数形)`
- ② カラムの作成: `t.型 :カラム名`
- ③ タイムスタンプ(作成日時・更新日時を自動的に入れてくれる)  
`t.timestamps null: false`



# マイグレーションファイルを反映させよう！

ターミナルで次のコマンドを入力してEnterを押そう！

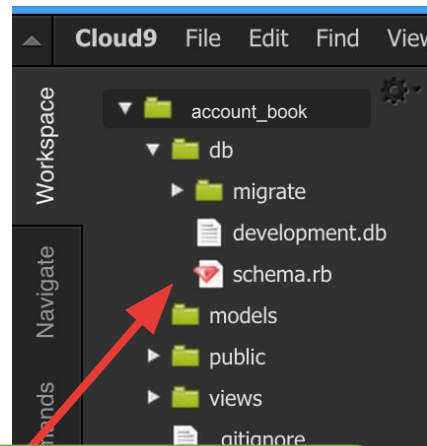
```
username@account_book::~~/workspace $ rake db:migrate
```

Enter

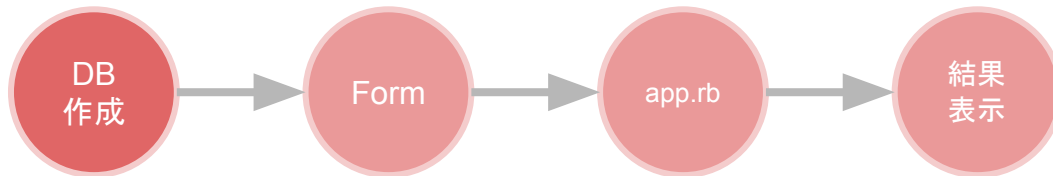


db/schema.rb

```
1 ActiveRecord::Schema.define(version: 20150515081154) do
2
3   create_table "items", force: :cascade do |t|
4     t.string "title"
5     t.integer "price"
6     t.datetime "created_at", null: false
7     t.datetime "updated_at", null: false
8   end
9
10 end
```



「**schema.rb**」というファイルが  
「db/」の中にあると、データベース  
のできあがり！



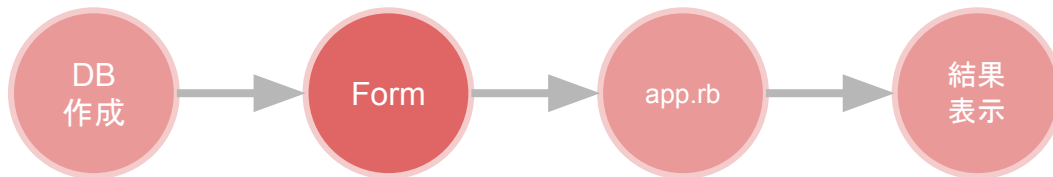
# フォーム(投稿画面)をつくろう

書き終わったら「command + S」または「ctrl + S」で保存しよう！

viewフォルダの中にある **index.erb** を下のようを書いてみよう！

views/index.erb

```
9 <body>
10
11 <form action="/create" method="post">
12   <label for="title">商品名</label>
13   <input type="text" name="title" id="title">
14   <label for="price">金額</label>
15   <input type="text" name="price" id="price">
16   <input type="submit" value="追加">
17 </form>
18
19 </body>
20 </html>
```



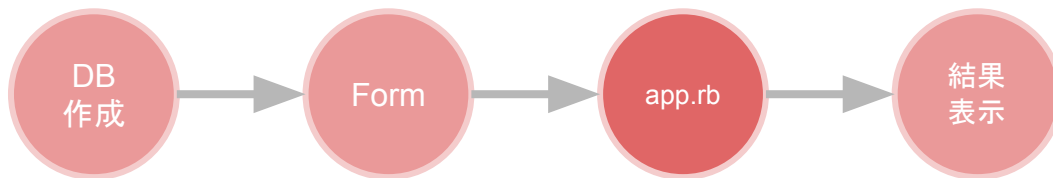
# テンプレートを読み込んでみよう！

書き終わったら「command + S」または「ctrl + S」で保存しよう！

app.rb を開いて、下のコードを追加しよう！

app.rb

```
1 require 'bundler/setup'
2 Bundler.require
3 require 'sinatra/reloader' if development?
4
5 require './models/item.rb'
6
7 get '/' do
8   erb :index
9 end
```

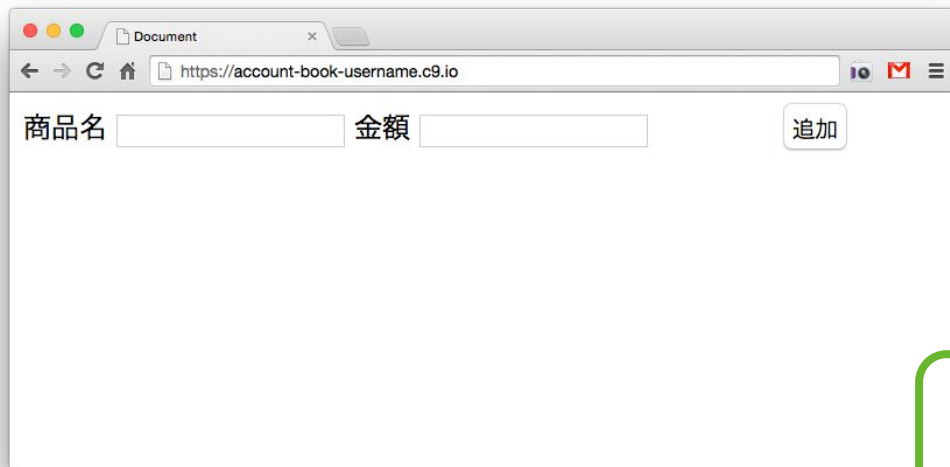


# サーバを起動して確認しよう！

ターミナルで次のコマンドを入力してEnterを押そう！

```
username@account_book:~/workspace $ ruby app.rb -p $PORT -o $IP
```

Enter



A screenshot of a web browser window. The address bar shows the URL `https://account-book-username.c9.io`. The page content includes two input fields: the first is labeled '商品名' (Item Name) and the second is labeled '金額' (Amount). To the right of these fields is a button labeled '追加' (Add).

こんな風に表示されたかな？  
商品と金額を追加をしてみよう！

# フォームから飛んでくるデータをデータベースに保存しよう！

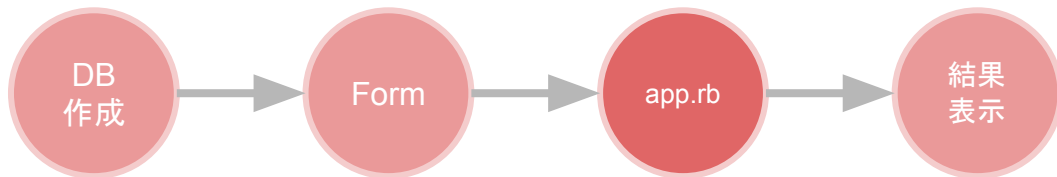
app.rb を開いて、下のコードを追加しよう！

書き終わったら「command + S」または「ctrl + S」で保存しよう！

app.rb

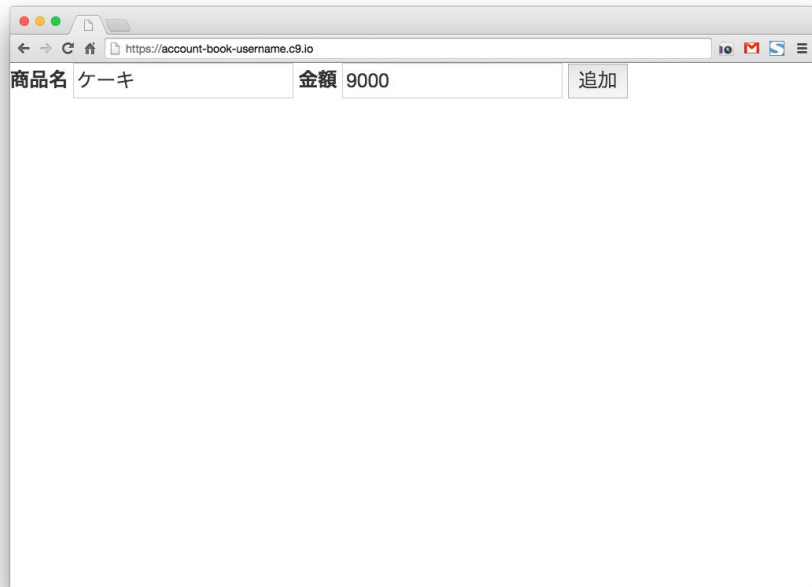
```
7 get '/' do
8   erb :index
9 end
10
11 post '/create' do
12   Item.create({
13     title: params[:title],
14     price: params[:price]
15   })
16
17   redirect '/'
18 end
```

`redirect '/'` を使うと別ページに飛ばすことができるよ！コードを読んでもみると、`Item` にデータベースを作成した後に、トップページにリンクを飛ばしているよ！

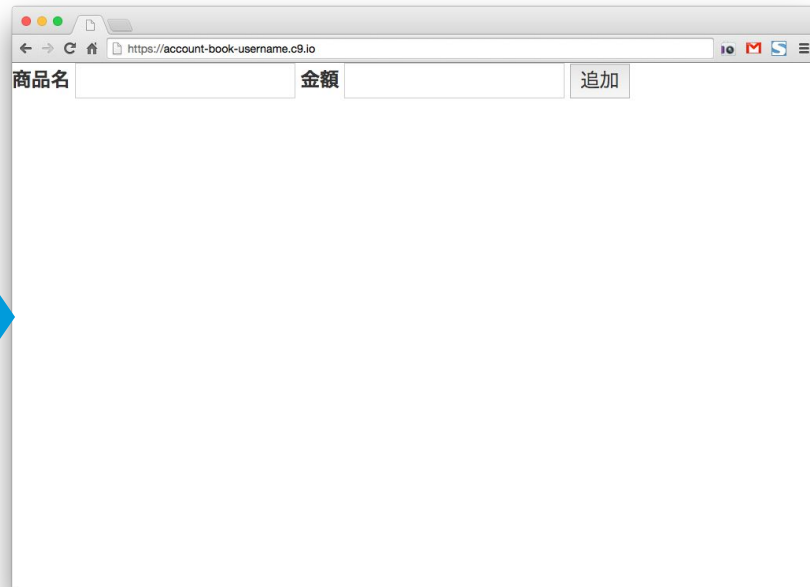




もう一度、フォームに何か投稿して、確認してみよう！



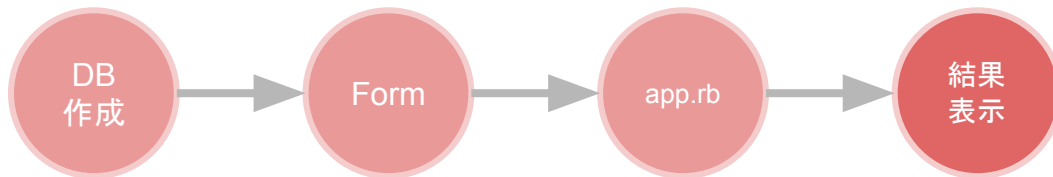
A screenshot of a web browser window with the address bar showing `https://account-book-username.c9.io`. The page contains a form with two input fields: '商品名' (Product Name) containing 'ケーキ' (Cake) and '金額' (Amount) containing '9000'. To the right of these fields is a button labeled '追加' (Add).



A screenshot of a web browser window with the address bar showing `https://account-book-username.c9.io`. The page contains an empty form with labels '商品名' (Product Name) and '金額' (Amount), and a button labeled '追加' (Add).

商品名と金額を追加すると...

何も表示されない！



# データベースの中身を確認してみよう！

①「control + C」でサーバ停止

②「sqlite3 db/development.db」を入力

③「select \* from items;」を入力

```
^C== Sinatra has ended his set (crowd applauds)
[2015-05-13 14:29:35] INFO going to shutdown ...
[2015-05-13 14:29:35] INFO WEBrick::HTTPServer#start done.
```

control + C

```
username@bookmark:~/workspace $ sqlite3 db/development.db
```

```
SQLite version 3.8.2 2013-12-06 14:53:30
```

```
Enter ".help" for instructions
```

```
Enter SQL statements terminated with a ";"
```

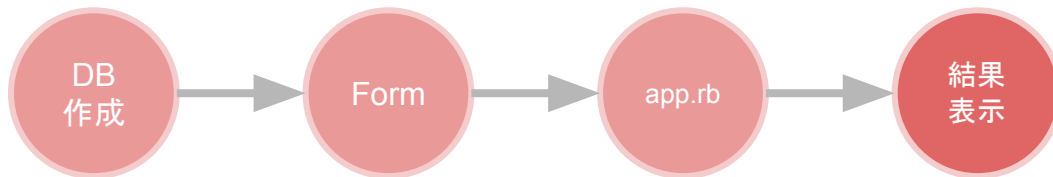
```
sqlite> select * from items;
```

```
1|ケーキ|9000|2015-05-15 08:48:03.532844|2015-05-15
08:48:03.53284
```

Enter

Enter

フォームに入力したものが表示されているかな？  
メンターに確認してもらおう！



# 「商品名」と「金額」を表示しよう！

書き終わったら「command + S」または「ctrl + S」で保存しよう！

`Item.all` でデータベースに保存されている  
`Item` を呼び出して、`@items` に格納するよ！  
(`"="`は代入という意味)

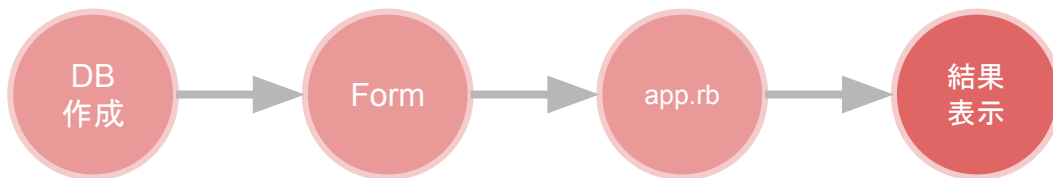
1. `@items.each` ですべての `Item` を  
一個一個取り出して `item` という変数に  
入れているよ！
2. `item.title` で商品名を、
3. `item.price` で金額を、  
それぞれとりだしてるよ

app.rb

```
7 get '/' do
8   @items = Item.all
9   erb :index
10 end
```

views/index.erb

```
10 <form action="/create" method="post">
11   ....
16 </form>
17
18 <div>
19   <% @items.each do |item| %>
20     <div> 2.
21       <p><%= item.title %></p>
22       <p><%= item.price %>円</p>
23     </div> 3.
24   <% end %>
25 </div>
```



# サーバを起動して表示を確認しよう！

`sqlite> .exit` と入力してsqliteを終了させよう。

そして、`ruby app.rb -p $PORT -o $IP` と打ってEnterを押してサーバを起動しよう！

```
sqlite> .exit
username@account_book:~/workspace $ ruby app.rb -p $PORT -o $IP
[2015-05-13 14:46:03] INFO WEBrick 1.3.1
[2015-05-13 14:46:03] INFO ruby 2.1.5 (2014-11-13) [x86_64-linux]
== Sinatra/1.4.5 has taken the stage on 8080 for development with backup from WEBrick
[2015-05-13 14:46:03] INFO WEBrick::HTTPServer#start: pid=20520 port=8080
```

サーバを起動したら、ページを確認してみよう！

データベースの中身が表示されたかな？

きちんと表示されるか確認しよう！



商品名

ケーキ

金額

9000円

追加

「商品名」と「金額」が出てきたかな？  
メンターに確認してもらおう！

# 合計金額を表示しよう！

書き終わったら「command + S」または「ctrl + S」で保存しよう！

app.rb

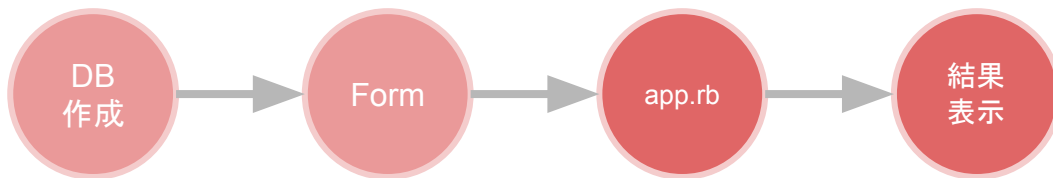
```
7 get '/' do
8   @items = Item.all
9   @total = Item.sum(:price)
10  erb :index
11 end
```

views/index.erb

```
18 <div>
19   <% @items.each do |item| %>
20     <div>
21       <p><%= item.title %></p>
22       <p><%= item.price %>円</p>
23     </div>
24   <% end %>
25 </div>
26 <div>
27   <p>合計:<%= @total %>円</p>
28 </div>
```

`Item.sum(:price)` で `Item` の  
`price` の合計を返してくれるよ！

`app.rb` から飛んでくる `@total` を  
`index.erb` で表示させるよ！



# サーバを起動して表示を確認しよう！

①「商品名」と「金額」を入力

②金額は負の数でもおっけー！

③追加をクリックすると合計金額が計算されて表示される！

商品名	金額	追加
ケーキ		
9000円		
合計：9000円		

商品名	金額	追加
チーズケーキ	10000	

商品名	金額	追加
ケーキ		
9000円		
チーズケーキ		
10000円		
クーポン		
-2000円		
合計：17000円		

# 完成！

この教科書から他の機能を作る  
教科書に続くよ！



The screenshot shows a web browser window with the URL `https://account-book-username.c9.io`. The application interface includes a table with two columns: '商品名' (Item Name) and '金額' (Amount). There is an '追加' (Add) button next to the table. The table contains the following entries:

商品名	金額
ケーキ	9000円
チーズケーキ	10000円
クーポン	-2000円
ガス代	2600円
電気代	10000円
合計	29600円

カテゴリー機能

編集・削除機能

グラフを表示する

家計簿アプリの基礎部分ができあがったよ！  
ここからは好きな機能を選んでつくってみよう！



# Tips - データベースを作る流れ

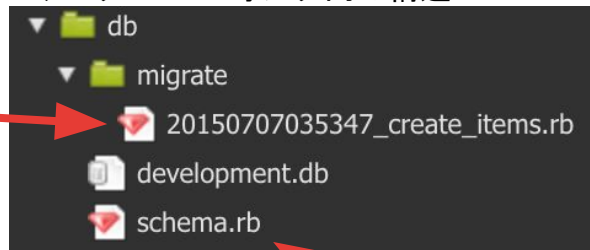
カウントアプリではもともとデータベースが作成されている素材を使いましたが、ブックマークアプリと家計簿アプリではデータベースは自分で作成しました。  
データベースが作成されるまでの流れを追ってみましょう！

## ① マイグレーションファイルを作成する

```
$ rake db:create_migration NAME=create_items
```

このファイルが  
作られる

### データベースフォルダ内の構造



## ② ①で作ったマイグレーションファイルを編集する

db/migrate/20150515081154\_create\_items.rb

```
1 class CreateBookmarks < ActiveRecord::Migration
2   def change
3     create_table :items do |t|
4       t.string :title
5       t.integer :price
6       t.timestamps null: false
7     end
8   end
9 end
```

↑カラムの名前と型を指定します

schema.rbが  
作られる

## ③ マイグレーションファイルを データベースに反映させる

```
$ rake db:migrate
```

## Tips - schemaって？

`$ rake db:migrate` コマンドが通ると、自動的に `schema.rb` が作成されます。このファイルは、マイグレーションファイルに書かれた内容を反映させたデータベースの内容が自動的に書かれるので、自分で書き換えるということはしません。マイグレーションが上手く通ったかの確認に使いましょう！

### マイグレーションファイル

db/migrate/20150515081154\_create\_items.rb

```
1 class CreateBookmarks < ActiveRecord::Migration
2   def change
3     create_table :items do |t|
4       t.string :title
5       t.integer :price
6       t.timestamps null: false
7     end
8   end
9 end
```



### schema

db/schema.rb

```
1 ActiveRecord::Schema.define(version: 20150515081154) do
2   create_table "items", force: :cascade do |t|
3     t.string "title"
4     t.integer "price"
5     t.datetime "created_at", null: false
6     t.datetime "updated_at", null: false
7   end
8 end
```