

第 13 章

予定の詳細表示 (2)

『Ruby on Rails 5.0 初級②』の最終章です。まず、予定の詳細表示ページのビジュアルデザインを調整します。予定の一覧ページに詳細ページへのリンクを設置し、詳細ページには「戻る」ボタンを設定します。最後に、予定の開始日と終了日に曜日を表示できるようにします。

13.1 テキストの背景色と揃え位置の調整

前章に引き続き、予定の詳細表示ページのビジュアルデザインを整えていきましょう。属性の名前を表示する領域の背景色を設定します。また、PC ブラウザで表示した場合には、属性の名前が右揃えで表示されるようにします。

```
app/views/plan_items/show.html.erb
```

```
:
3 - <div class='col-12 col-md-4'>件名</div>
3 + <div class='col-12 col-md-4 bg-info text-white text-md-right'>件名</div>
:
7 - <div class='col-12 col-md-4'>説明</div>
7 + <div class='col-12 col-md-4 bg-info text-white text-md-right'>説明</div>
:
15 - <div class='col-12 col-md-4'>開始日時</div>
15 + <div class='col-12 col-md-4 bg-info text-white text-md-right'>
16 +   開始日時
17 + </div>
:
```

第 13 章 予定の詳細表示 (2)

```
23 - <div class='col-12 col-md-4'>終了日時</div>
23 + <div class='col-12 col-md-4 bg-info text-white text-md-right'>
24 +     終了日時
25 + </div>
    :
```

HTML クラス `bg-info` は背景色を水色 (#5bc0de) にします。info の部分を他の名前でも置き換えれば、さまざまな色を背景色として利用できます (表 13.1)。

表 13.1 背景色を設定するクラス

クラス名	背景色	16 進数表示
<code>bg-primary</code>	青	#0275d8
<code>bg-success</code>	緑	#5cb85c
<code>bg-info</code>	水色	#5bc0de
<code>bg-warning</code>	オレンジ色	#f0ad4e
<code>bg-danger</code>	赤	#d9534f

HTML クラス `text-white` は文字色を白色にします。

HTML クラス `text-md-right` は、ブラウザの横幅が Medium (768px) 以上のときにテキストを右揃えにします。`right` の部分を `left` で置き換えれば、左揃えになり、`center` で置き換えれば中央揃えになります。

`md` の部分を `sm`、`lg`、`xl` などで置き換えれば、さまざまなブレイクポイントに対してテキストの揃え位置を設定できます。すべての表示幅のブラウザでテキストの揃え位置を指定したければ `text-right`、`text-left`、`text-center` などのブレイクポイントと指定のない HTML クラスを使用します。

ブラウザを再読み込みすると、図 13.1 のようになります。

13.1 テキストの背景色と揃え位置の調整



図 13.1 背景色と揃え位置を調整 (PC 用レイアウト)

スマートフォン用の表示に切り替えると 図 13.2 のようになります。

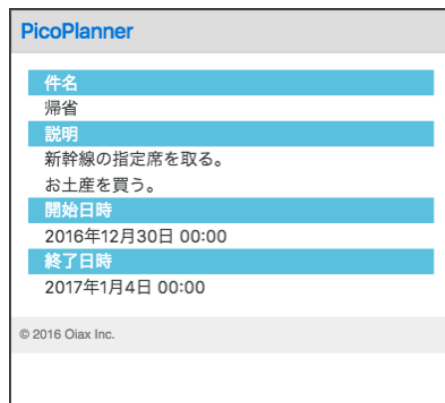


図 13.2 背景色と揃え位置を調整 (スマートフォン用レイアウト)

PC ブラウザでは属性の名前が右揃えで表示されるのに対し、スマートフォンではそのまま左揃えで表示されることを確認してください。

13.2 行間の調整

続いて、行と行の間を少し広げましょう。新規ファイル `plan_item.scss` を次の内容で作成してください。

```
app/assets/stylesheets/plan_item.scss (New)
1  div.plan-item {
2    div.row {
3      margin-bottom: 0.5rem;
4    }
5    div.row:last-child {
6      margin-bottom: 0;
7    }
8  }
```

よく似た名前の `plan_items.scss` がすでにありますので、注意してください。

HTML クラス `plan-item` が設定されたグリッドシステムの行 (`row`) に対して、下辺のマージンを `0.5rem` (通常は、`8px` に相当) に設定しています。ただし、最後の行に関しては下辺のマージンを `0` にします。

合わせてテンプレートを書き換えます。

```
app/views/plan_items/show.html.erb
1 - <div class='container-fluid'>
1 + <div class='container-fluid plan-item'>
:
```

ブラウザを再読み込みすると、図 13.3 のようになります。



図 13.3 行間を調整 (PC 用レイアウト)

スマートフォン用の表示に切り替えると 図 13.4 のようになります。

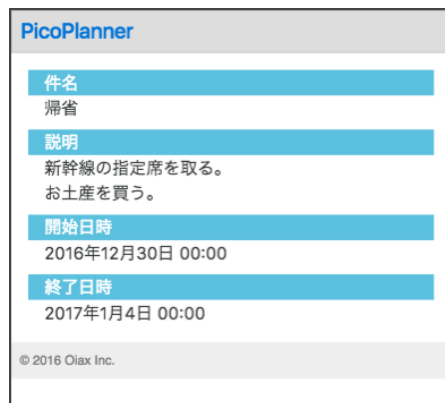


図 13.4 行間を調整 (スマートフォン用レイアウト)

13.3 予定リストにリンクを設置

予定の詳細表示ページがいちおう完成しましたので、今度は予定の一覧ページに詳細ページへのリンクを設置しましょう。

```
app/views/plan_items/index.html.erb
:
5 - <span class='plan-item-name'><%= item.name %></span>
5 + <span class='plan-item-name'><%= link_to item.name, item %></span>
:
```

ブラウザで動作確認をしましょう。図 13.5 のようになります。



図 13.5 予定リストにリンクを設置

「買い物」、「読書」、「帰省」というテキストがリンクになっており、この部分をクリックするとそれぞれの予定の詳細情報が表示されます。

さて、いま私たちが予定の一覧ページに設置したリンクでは、次のような書き方をしました。

```
link_to item.name, item
```

変数 `item` には `PlanItem` オブジェクトがセットされています。これまで、`link_to` メソッドの第 2 引数に文字列やシンボルを指定するケースはありましたが、モデルオブジェクトを指定する書き方は初登場です。

主キーの値が 3 である `PlanItem` オブジェクトが `link_to` メソッドの第 2 引数に指定されると、Rails は次のように読み替えます。

```
link_to item.name, '/plan_items/3'
```

この読み替えが行われるためには、`config/routes.rb` で `plan_item` という名前の経路が設定されている必要があります。そうでない場合、`plan_item_path` という未定義のメソッド (`undefined method`) が使用されたというエラーが発生します。

13.4 「戻る」ボタンの追加

今度は、予定の詳細ページから一覧ページに戻るためのリンク (ボタン) を設置しましょう。

```
app/views/plan_items/show.html.erb
```

```

1 <div class='container-fluid plan-item'>
2 +   <div class='row'>
3 +     <div class='col-12 hidden-md-up text-right'>
4 +       <%= link_to :plan_items do %>
5 +         <i class='fa fa-list fa-lg'></i>
6 +       <% end %>
7 +     </div>
8 +     <div class='col-md-12 hidden-sm-down text-right'>
9 +       <%= link_to :plan_items, class: 'btn btn-secondary btn-sm' do %>
10 +        <i class='fa fa-list'></i> 予定表へ戻る
11 +       <% end %>
12 +     </div>

```

第 13 章 予定の詳細表示 (2)

```
13 + </div>
14 <div class='row'>
15 <div class='col-12 col-md-4 bg-info text-white text-md-right'>件名</div>
   :
```

「戻る」ボタンを設置するためグリッドシステムに行 (row) を追加しました。そして、その内側に二つの div 要素を置いています。

ブラウザを再読込すると、図 13.6 のようになります。



図 13.6 「戻る」ボタンを追加 (PC 用レイアウト)

スマートフォン用の表示に切り替えると 図 13.7 のようになります。



図 13.7 「戻る」ボタンを追加（スマートフォン用レイアウト）

では、ソースコードを詳しく見ていきましょう。3 行目をご覧ください。

```
<div class='col-12 hidden-md-up text-right'>
```

HTML クラス `hidden-md-up` は初登場です。ブラウザの横幅がブレイクポイント `md` 以上のとき、この要素は見えなくなります。英語の “hidden” は「隠された」という意味の形容詞です。言い換えれば、この要素はブラウザの横幅がブレイクポイント `md` 未満のときしか見えない、ということになります。

次に、ソースコードの 4~6 行目をご覧ください。

```
<%= link_to :plan_items do %>  
  <i class='fa fa-list fa-lg'></i>  
<% end %>
```

Font Awesome のアイコンをボタンとして使用します。「list」という名前のアイコンを使うため `fa-list` という HTML クラスを指定しています。`fa-lg` はアイコンの大きさを「大 (large)」にするための HTML クラスです。これを指定するとアイコンのサイズが通常の 33% 増しになります。表 13.2 にアイコンのサイズを調整するための HTML クラスをまとめます。

ソースコードの 8 行目をご覧ください。

第 13 章 予定の詳細表示 (2)

表 13.2 アイコンのサイズを調整するための HTML クラス

HTML クラス	アイコンのサイズ
fa-lg	通常の 33% 増し
fa-2x	通常の 2 倍
fa-3x	通常の 3 倍
fa-4x	通常の 4 倍
fa-5x	通常の 5 倍

```
<div class='col-md-12 hidden-sm-down text-right'>
```

3 行目のコードと対になっています。この div 要素は、ブラウザの横幅がブレイクポイント md (768px) 未満のときに隠されます。

HTML クラス hidden-sm-down が「md (768px) 未満のときに隠される」という意味になるのは、少し不思議です。「sm (576px) 未満のときに隠される」の間違いではないのでしょうか。いいえ、そうではありません。

末尾が -down で終わる HTML クラスの場合、そのひとつ上位のブレイクポイントから 1px を引いた幅を上限として、ブラウザの幅がそれ以下のときに適用されることになっています。つまり、lg (992px) 未満の表示幅のブラウザで隠したければ、HTML クラス hidden-md-down を指定することになります。表 13.3 にクラス名と適用範囲をまとめてあります。普段は使わない xs というブレイクポイントが現れる点に注意してください。

表 13.3 要素を隠すための HTML クラス

HTML クラス	対象となるブラウザの表示幅
hidden-xs-down	sm (576px) 未満
hidden-sm-down	md (768px) 未満
hidden-md-down	lg (992px) 未満
hidden-lg-down	xl (1,200px) 未満
hidden-xl-down	すべて

ソースコードの 9~11 行目をご覧ください。

```
<%= link_to :plan_items, class: 'btn btn-secondary btn-sm' do %>
  <i class='fa fa-list'></i> 予定表へ戻る
<% end %>
```

HTML クラス `btn` は、指定された要素に対してボタンのように見えるスタイルを設定します。HTML クラス `btn-secondary` は背景色を白に設定します。`secondary` の代わりに、`primary`、`success`、`info`、`warning`、`danger` などを使用すれば背景色を変化させることができます (表 13.4)。

表 13.4 ボタンの背景色を設定するクラス

クラス名	背景色	16 進数表示
<code>btn-primary</code>	青	<code>#0275d8</code>
<code>btn-secondary</code>	白	<code>#ffffff</code>
<code>btn-success</code>	緑	<code>#5cb85c</code>
<code>btn-info</code>	水色	<code>#5bc0de</code>
<code>btn-warning</code>	オレンジ色	<code>#f0ad4e</code>
<code>btn-danger</code>	赤	<code>#d9534f</code>

HTML クラス `btn-sm` はボタンのサイズを「小 (small)」にします。逆にサイズを大きくしたいときは `btn-lg` を指定してください。

13.5 曜日の表示

ヘルパーメソッド `format_wday` の追加

本巻の締めくくりとして、予定のリストページと詳細ページに表示される日付に曜日を付けましょう。例えば、次のような形式になります。

```
10月8日 (土) 16:00 ~ 16:30
12月30日 (金) 00:00 ~ 2017年1月4日 (水) 00:00
```

まず、新規のヘルパーメソッド `format_wday` を作成してください。

```
app/helpers/application_helper.rb
```

```
19 s << item.ends_at.strftime('%Y年%-m月%-d日 %H:%M')
```

第13章 予定の詳細表示(2)

```
20     end
21   end
22 +
23 +   def format_wday(datetime)
24 +     %w(日 月 火 水 木 金 土)[datetime.wday]
25 +   end
26 end
```

メソッドの中身を次のように書き換えると分かりやすいかもしれません。

```
wday_names = %w(日 月 火 水 木 金 土)
wday_names[datetime.wday]
```

記号 %w については『初級①』の第 12 章で説明しました。文字列の配列を作るための % 記法です。次に例示する二つの式は等価です。

```
%w(日 月 火 水 木 金 土)
[ '日', '月', '火', '水', '木', '金', '土' ]
```

仮引数 `datetime` には日付時刻オブジェクトを指定します。日付時刻オブジェクトの `wday` メソッドは、曜日を 0 から 6 までの整数値で返します。0 が日曜日、1 が月曜日、6 が土曜日を表します。すなわち、ヘルパーメソッド `format_wday` は引数に与えられた日付時刻オブジェクトの曜日を「日」、「月」などの漢字 1 文字に変換して返します。

予定リストページでの曜日表示

では、これを利用して予定表の日付に曜日を付けましょう。まず、開始日時と終了日時それぞれの曜日を漢字 1 文字に変換し、変数 `w1` と `w2` にセットします。

```
app/helpers/application_helper.rb
:
4   def format_duration(item)
5 +     w1 = format_wday(item.starts_at)
6 +     w2 = format_wday(item.ends_at)
```

```
7 +
8   s = String.new('')
:
```

そして、これらを括弧付きで日付と時刻の間に挿入します。

```
app/helpers/application_helper.rb
:
9   if item.starts_at.year == Time.current.year
10 -   s << item.starts_at.strftime('%-m月%-d日 %H:%M')
10 +   s << item.starts_at.strftime("%-m月%-d日 (#{w1}) %H:%M")
11   else
12 -   s << item.starts_at.strftime('%Y年%-m月%-d日 %H:%M')
12 +   s << item.starts_at.strftime("%Y年%-m月%-d日 (#{w1}) %H:%M")
13   end
:
19  elsif item.ends_at.year == item.starts_at.year
20 -   s << item.ends_at.strftime('%-m月%-d日 %H:%M')
20 +   s << item.ends_at.strftime("%-m月%-d日 (#{w2}) %H:%M")
21  else
22 -   s << item.ends_at.strftime('%Y年%-m月%-d日 %H:%M')
22 +   s << item.ends_at.strftime("%Y年%-m月%-d日 (#{w2}) %H:%M")
23  end
:
```

ブラウザを再読込すると、図 13.8 のようになります。

第13章 予定の詳細表示(2)



図 13.8 予定リストページに曜日を表示

予定詳細ページでの曜日表示

詳細ページの日付表示も変更しましょう。新たにヘルパーメソッド `format_datetime` を次のように定義します。

```
app/helpers/application_helper.rb
22     s << item.ends_at.strftime("%Y年%-m月%-d日 (#{w2}) %H:%M")
23     end
24 end
25 +
26 + def format_datetime(datetime)
27 +   w = format_wday(datetime)
28 +   datetime.strftime("%Y年%-m月%-d日 (#{w}) %H:%M")
29 + end
30
31 def format_wday(datetime)
:
```

テンプレートを書き換えます。

```

app/views/plan_items/show.html.erb
:
31 - <%= @plan_item.starts_at.strftime('%Y年%-m月%-d日 %H:%M') %>
31 + <%= format_datetime @plan_item.starts_at %>
:
39 - <%= @plan_item.ends_at.strftime('%Y年%-m月%-d日 %H:%M') %>
39 + <%= format_datetime @plan_item.ends_at %>
:

```

ブラウザを再読込すると、図 13.9 のようになります。



図 13.9 予定詳細ページに曜日表示

以上で、本書『Ruby on Rails 5.0 初級②』はおしまいです。

次巻『Ruby on Rails 5.0 初級③』は 2016 年冬の刊行予定です。予定を追加、編集、削除する機能を PicoPlanner に追加していきます。