

時間割表示サイトの改良

遠藤 拓海†

1. 動機・目的

私は PC 部で Things という時間割表示サイトの運営をしており、時間割の編集をする際に扱いづらい点があった。先輩の負担を減らし、そして現在利用者が少ない Things をより多くの生徒に利用してもらうことで生徒の負担を軽減するために、より管理者にも使用者にも扱いやすくすることはできないかと思い改良を始めた。Things の管理者にとって需要があると考えた時間割編集の簡略化と、豊岡高校の全生徒を対象に Things に求めている機能についてのアンケートを取り（図 1）、比較的要望の多かったタイムスケジュールの表示の二つを目的として設定し研究を始めた。

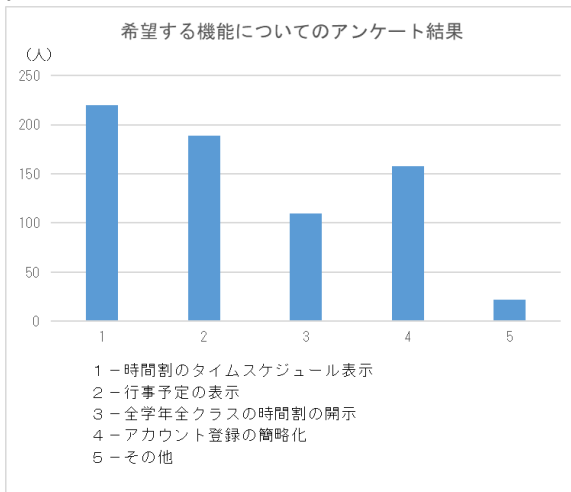


図 1

2. 方法・内容

今回のプログラムにはフロントエンド言語として JavaScript、HTML、サーバーサイド言語として PHP を使用してプログラムを組んだ。JavaScript で記述する箇所は jQuery を使用することでプログラムの記述量を減らし、プログラムの見易さを向上させた。入力フォームを作成する form 関数の部分では今まで input 要素で構成されていたものを select 要素で構成するものに変え、セレクトボックスを作成することにした。（図 2）これにより手入力の必要をなくすことで編集の簡略化につながると考えた。そして選択肢を時間に変えた物を作成することでタイムスケジュールの表示も可能になるため、同じ方法で利用者の需要も満たせるのではないかと考えた。データを送信する部分では ajax 通信を用いることによって非同期通信を可能にし、サーバーの応答を待ち続け、動作が停止してしまうことがないようにした。（図 3）そして通信の秘匿性を上げるため通信形式を POST 形式にした。

```
<div class="form-group">
  <label for="job2">subject</label>
  <select id="job2" name="job2" class="form-control">

    <optgroup label="国語">

      <option value="">現代文</option>
      <option value="">古典</option>
    </optgroup>

    <optgroup label="数学">

      <option value="">数I</option>
      <option value="">数II</option>
      <option value="">数III</option>
      <option value="">数A</option>
      <option value="">数B</option>
    </optgroup>

    <optgroup label="英語">

      <option value="">英</option>
      <option value="">表英</option>
    </optgroup>

  </select>
</div>
</form>
```

図 2

```
$(function(){
  // Ajax button click
  $('#ajax').on('click',function(){
    $.ajax({
      url:'http://localhost/Untitled-3.php',
      type:'POST',
      data:{

        'job':$('#job').val(),
        'job2':$('#job2').val(),
      }
    })
    // Ajaxリクエストが成功した時
    .done( (data) => {
      $('#result').html(data);
      console.log(data);
    })
    // Ajaxリクエストが失敗した時
    .fail( (data) => {
      $('#result').html(data);
      console.log(data);
    })
  });
});
</script>
```

図 3

3. 結果

入力フォームを改良したプログラムを記述し、タイムスケジュールの表示をすることもできた。しかし、今回作成したプログラムを Things に実装できておらず、実際に動かすことが出来なかった。

4. 考察

入力フォームは改善され、手入力をする必要はなくなったが、一限一限教科を選択して入力するため、これだけでは管理者の負担を軽減するには不十分な物であると考えられる。

5. 結論

今回プログラムを組むことは出来たがそれを実装することは出来ておらず、さらに今回の改良だけではあまり管理者の負担が軽減できないと考えられるため、今回は改良といえるほどの結果にはならなかったといえる。さらに改良していくには、機能を実装してそれが管理者や利用者にとって十分なものであるか確認する必要があると考えられる。

6. 今後の課題と展望

入力フォームを改善するプログラムは記述できたが、これだけでは実装できたとしてもあまり管理者の負担を改善することにはつながらないと考えられる。最初は PHP に送られたデータをデータベースに送信することで、曜日、時間割、タイムスケジュールの三つが対応した時間割のテンプレートを作成し、入力を減らすことができると考えていたが、今回その記述が出来なかったためその部分を完成させ、Things に実装するのが今後の課題であると考えられる。そして今回利用者にとって必要な機能を調べるためにアンケートを取ったが、Things を知っている人、利用している人がとても少なくそれらの人が実際に利用することで新たに必要な機能が増えることも考えられるため、より多くの人に Things を利用してもらい、その意見を聞くことも必要と考えられる。

7. 参考文献

- ①たにぐちまこと
これから Web をはじめる人の HTML&CSS、JavaScript
のきほんのきほん マイナビ出版 2017
- ②PHP: 言語リファレンス - Manual
<https://www.php.net/manual/ja/langref.php>