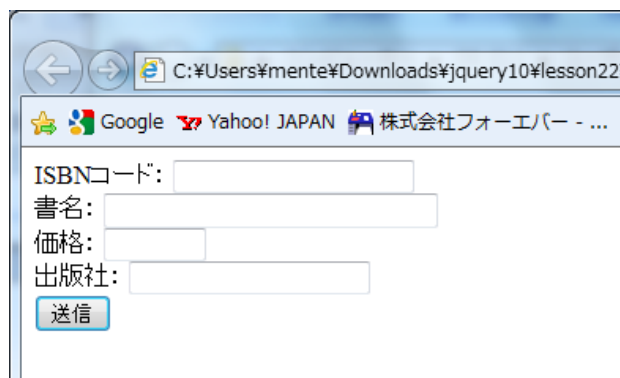


課題8：フォームの入力検証機能を実装しよう



The screenshot shows a web browser window with the address bar displaying a local file path: C:\Users\mente\Downloads\jquery10\lesson22. The browser's search bar contains "Google", "Yahoo! JAPAN", and "株式会社フォーエバー...". Below the search bar is a form with four input fields labeled "ISBNコード:", "書名:", "価格:", and "出版社:". A blue "送信" (Send) button is located below the "出版社:" field.

以下のような入力フォームがある。それぞれ以下のような検証が必要である

項目名	検証
ISBNコード	4-89311-648-7のようなISBNコードのフォーマット、必須入力
書名	30文字以内
価格	100から10000の範囲
出版社	翔泳社、技術評論社、秀和システム、マイコミのいずれかであること

以下のようなソースコードを組み、validation.js と組み合わせることで検証を行うことができる。

【validation.html】

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8" />
<link type="text/css" href="validation.css" rel="stylesheet" />
<title>イベント</title>
<script type="text/javascript" src="../jquery-1.6.4.min.js"></script>
<script type="text/javascript" src="validation.js" ></script>
</head>
<body>
<ul id="error_summary"></ul>

<form id="fm">
<div class="field">
  <label for="isbn">ISBNコード</label> :
  <input type="text" id="isbn" size="20" class="valid required regexp"
    data-pattern="[0-9]{1}-[0-9]{3,5}-[0-9]{3,5}-[0-9X]{1}" />
```

```
</div>
<div class="field">
  <label for="title">書名</label> :
  <input type="text" id="title" size="30" class="valid length"
    data-length="30" />
</div>
<div class="field">
  <label for="price">価格</label> :
  <input type="text" id="price" size="5" class="valid range"
    data-max="10000" data-min="100" />
</div>
<div class="field">
  <label for="publish">出版社</label> :
  <input type="text" id="publish" class="valid inarray"
    data-option="翔泳社 技術評論社 秀和システム マイコミ" />
</div>
<div class="field">
  <input type="submit" value="送信" />
</div>
</form>
</body>
</html>
```

ここに

```
data-pattern="[0-9]{1}-[0-9]{3,5}-[0-9]{3,5}-[0-9X]{1}"
```

の部分が ISBN コードのフォーマットを表しており、[0-9]が 0 から 9 までの文字、つまり数字で{1}が 1 文字、つづく-がかならず-が入る、続けて同様に[0-9]で{3,5}が 3 文字か 5 文字を表している。

これで x-xxx-xxxxx-x のフォーマットを指示している。

その他は required,data-pattern,data-length,data-max,data-min,data-option の各パラメータで指示した検証を validation.js で行うようにすればよい。

たとえば、ISBN は valid と required と data-pattern の class が指定してるが、valid が検証をすることを指示し required が必須を、data-pattern が regexp 関数を使ったパターンチェックを指示している。

ほかも同様に class で指示が書いてある。

【validation.js】

```
$(function () {  
  var msgs;  
  var setError = function (elem, msg) {  
    msgs.push('<li>' + msg + '</li>');  
    $(elem)  
    .addClass('error_field')  
    .after('<span class="error_mark">*</span>');  
  };  
  
  $('#fm').submit(function (e) {  
    msgs = [];  
    $('.error_mark').remove();  
    $('.valid', this)  
    .removeClass('error_field')  
    .filter('.required')  
    .each(function () {  
      if ($(this).val() === "") {  
        setError(this,  
          $(this).prev('label').text() + 'は必須入力です。');  
      }  
    })  
    .end()  
    .filter('.length')  
    .each(function () {  
      if ($(this).val().length > $(this).data('length')) {  
        setError(this,  
          $(this).prev('label').text() + 'は' +  
          $(this).data('length') + '文字以内で入力してください。');  
      }  
    })  
    .end()  
    .filter('.range')  
    .each(function () {  
      var v = parseFloat($(this).val());  
      if (v < $(this).data('min') || v > $(this).data('max')) {  
        setError(this,  
          $(this).prev('label').text() + 'は' +
```

```

        $(this).data('min') + '~' + $(this).data('max') +
        'の範囲で入力してください。');
    }
})
.end()
.filter('.inarray')
.each(function () {
    var opts = $(this).data('option').split(' ');
    if ($.inArray($(this).val(), opts) === -1) {
        setError(this,
            $(this).prev('label').text() + 'は' +
            opts.toString() + 'のいずれかで入力してください。');
    }
})
.end()
.filter('.regex')
.each(function () {
    var reg = new RegExp($(this).data('pattern'), 'gi');
    if (!reg.test($(this).val())) {
        setError(this,
            $(this).prev('label').text() + 'の形式が間違っています。');
    }
});

if (msgs.length === 0) {
    $('#error_summary').css('display', 'none');
} else {
    $('#error_summary')
        .css('display', 'block')
        .html(msgs.join(""));
    e.preventDefault();
}
});
});

```

jQuery の関数は 2 つあり、ひとつは setError でエラーを表示するもの。もう一つが長く、フォームの submit のタイミングで class に指示してある各検証パラメータをチェックしエラーがあればメッセージを追加していき、最後にエラーがあればその文字列を表示する。

【validation.css】

```
#error_summary {  
  color: #F00;  
  display: none;  
}
```

```
.error_field {  
  background-color: #F9C;  
}
```

```
.error_mark {  
  color: #F00;  
}
```

error_summary はフォーム先頭の一括エラー表示領域で、error_field が入力テキストボックスで、error_mark が入力テキストボックスの直後の領域です。

filter メソッドの次の行の each メソッドは、要素個々に対して順にアクセスする方法です。end()が各 filter の終了の宣言です。

まずはこれで、実行を確認してみてください。

Let's Try:教科書 184 ページあった検証プログラムを今回の方法で改造してみよう。むずかしい場合は課題 8 に「お名前」「郵便番号」「電話番号」「メールアドレス」を追加してみよう。Regex のような正規表現はむずかしいが Internet で調べて自分なりに設定してみよう。