

文芸的ウェブプログラミング

脇田建

平成 22 年 8 月 30 日

概要

簡単なウェブアプリケーションを文芸的プログラミングの手法を用いて例示することによって、文芸的プログラミングの特徴を学ぶ実験。文芸的プログラミングのツールとしては NuWeb を用いた。

目次

1	HTML ファイルの概要	2
2	アプリケーション	2
3	アプリケーションの装い	5
A	ファイル一覧	6
B	モジュール一覧	6

1 HTML ファイルの概要

本例題はウェブアプリケーションなので、HTML ファイルを通して必要となる、スタイルシートやプログラムのファイルが読み込まれる。HTML ファイルの大まかな構造は以下の通りである。

まずこの HTML は HTML5 で記述されている。HTML5 では、大幅に簡素化された記述が許されており、html タグ、head タグ、body タグなどを省略することができる。先頭の DOCTYPE 宣言で、このファイルが HTML であることを宣言し、次の行でこのファイルが UTF-8 エンコーディングされていることを宣言している。この HTML ファイルには、CSS ファイル群と JavaScript ファイル群を読み込むことのみが宣言されている、HTML の本体にあたる部分は本稿へのリンクのみである。その他の HTML に相当する部分は、すべて JavaScript で動的に生成される。

```
"hello.html" 2a ≡
  <!DOCTYPE html>
  <meta charset="utf-8">
  { CSS section 5b }
  { JavaScript section 2b }
  <ul>
  <li><a href="hello.pdf">文芸的「Hello!」</a></li>
  <li><a href="hello.w" charset="utf-8">アプリケーションとドキュメントの元になっ
  た文芸的ななにか</a></li>
  ◇
```

2 アプリケーション

ウェブアプリケーションの装飾を司る CSS の宣言については後述することとし、まずはアプリケーションを構成する JavaScript の方から解説する。

本アプリケーションでは、基本ライブラリとして jQuery を用いている。jQuery は、jQuery Core パッケージを中心とした拡張可能なパッケージシステムで、準標準的な UI パッケージのほかユーザグループから提供されたさまざまなパッケージ群を利用することができる。以下が、本アプリケーションで利用している JavaScript のファイルである。

jquery-...js jQuery Core

jquery-ui-...js jQuery で準標準的に用いられているユーザインタフェイス関連のパッケージ

jQuery.radmenu.min.js 放射的メニューを扱うためのパッケージ

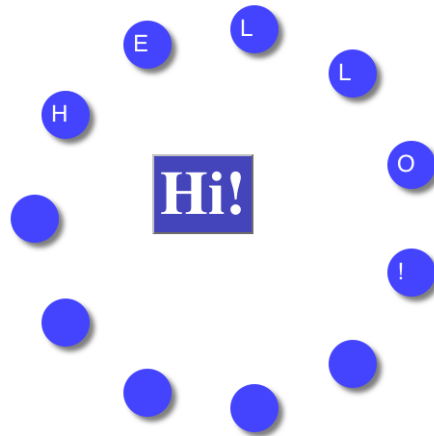
hello.js 本アプリケーションの本体

```
{ JavaScript section 2b } ≡
  <script type="text/javascript" src="/lib/jquery-1.4.2.min.js"></script>
  <script type="text/javascript" src="/lib/jquery-ui-1.8.4.custom.min.js"></script>
  <script type="text/javascript" src="js/jquery.radmenu.min.js"></script>
  <script type="text/javascript" src="hello.js"></script>
  ◇
```

Macro referenced in 2a.



(a) “Hi!” ボタン



(b) “Hi!” ボタンをクリックすると放射状メニューが表示される

図 1: 本アプリケーションで使われるボタンとメニュー

以下では、アプリケーションの本体となる JavaScript のプログラムを解説する。

先頭の 3 行は、ウェブブラウザのウィンドウとボディに対応した DOM オブジェクトに対応する jQuery オブジェクトを取得し、それぞれ `$window`、`$body` と名付けている。jQuery オブジェクトは DOM オブジェクトに対応したラッパーであり、これを利用することで、DOM 木の操作が簡略になる。本例題では、jQuery オブジェクトを表す変数名を `$` で始めることで普通の DOM オブジェクトに対するラッパーであることを明記している。

jQuery オブジェクトが提供する主な機能として、属性を設定するための `attr` メソッド、スタイルを設定するための `css` メソッド、テキスト要素を定義するための `text` メソッドなどがある。ここでは、`$body` の大きさをウィンドウの大きさと同じに設定している。

これらの基本的な設定ののちに、“Hi!” ボタン図 1(a) とそのボタンをクリックしたときに表示する放射状のメニュー図 1(b) の設定を行っている。ボタンの作成方法は jQuery で典型的な方式に乗っ取っている。まず、簡単な HTML 記述からタグを作成し、`text` メソッドでテキストを埋め込み、`click` メソッドでボタンをクリックしたときの振る舞いを与え、最後に `$body` 要素に追加したのちに、画面中央に配置している。`position` メソッドは引数の部分をそのまま英文として読みくたせば、意味合いが理解できるだろう。ボタンをクリックした時には、`click` メソッドで定義したように後述のメニューを開く。

```
"hello.js" 4a ≡
$(function () {
  var $window = $(window);
  var $body = $(document.body)
    .css({ width: window.innerWidth, height: window.innerHeight });
  $('<div></div>').position();
  <Menu definition 4b, ... >
  $('<button id="hello_button"></button>')
    .text('Hi!')
    .click(function () { $hello_menu.radmenu('show'); })
    .appendTo($body)
    .position({ my: 'center center', at: 'center center', of: $body });
});
◇
```

つぎに先ほど省略したメニューの作成方法について説明する。このメニューは前述のボタンをクリックした時に表示されるものである。このメニューは単なるジョークで Hello!の一文字ずつを選択項目名とするダイヤル式のメニューである。メニューは以下で説明するように、HTML のリスト構造として定義してある。これにダイヤル式メニューの装飾を施すのが radmenu パッケージである。

まず、メニュー項目の定義方法から説明する。最初に、メニュー全体を表す div 要素を作成し、hello_menu と名付けている。つぎにこのメニューに登録すべき各メニュー項目に対応する jQuery オブジェクトを定義したものが Menu_item である。以下ではこのオブジェクトを雛形として、そのなかの div 要素のテキスト要素にメニュー項目名を与えることによってメニュー項目の構造を作成している。

実際のメニュー項目の作成とそのメニューへの追加は、関数的プログラミングによって実現している。まずメニューに現れる文字からなる文字列を一文字ずつに分解し、それぞれを雛形の複製のテキスト項目として登録し、最期にメニューに追加している。

```
<Menu definition 4b> ≡
var $hello_menu = $('<div><ul class="list"></ul></div>');
var $Menu_item = $('<li class="item"><div></div></li>');
'HELLO!'.split('')
  .map(function (c) {
    var $mi = $Menu_item.clone();
    $mi.children().first().text(c);
    $mi.appendTo($('.list', $hello_menu));
  });
◇
```

Macro defined by 4b, 5a.
Macro referenced in 4a.

このように作成されたメニューに対して、ボタンを作成した時と同じような方法で HTML 文書に追加し、場書を文書の中央に設定し、最期にきれを放射状メニューとして設定している。

〈Menu definition 5a〉≡

```
$hello_menu
  .appendTo($body)
  .radmenu({ radius: 200, angleOffset: Math.PI,
    selectEvent: 'click',
    onSelect: (function () { $hello_menu.radmenu('hide'); }) })
  .position({ my: 'center center', at: 'center center', of: $body });
```

◇

Macro defined by 4b, 5a.
Macro referenced in 4a.

3 アプリケーションの装い

アプリケーションの見かけは概ね CSS を用いて整えた。HTML のヘッダでは、jQuery 標準の CSS と、このアプリケーションのために私が作成した hello.css を読み込んでいる。

〈CSS section 5b〉≡

```
<link rel="stylesheet" type="text/css" href="css/js-ipad/jquery-ui-1.8.4.custom.css">
<link rel="stylesheet" type="text/css" href="hello.css">
```

◇

Macro referenced in 2a.

hello.css の先頭では、body タグに対して、余白がないことを設定している。これは、ブラウザの表示領域全体を有効に利用するための設定である。

"hello.css" 5c ≡

```
body {
  margin: 0; padding: 0;
  font-family: Helvetica;
  /*position: relative;*/
}
```

◇

File defined by 5cd, 6.

以下は画面中央に表示される“Hi!” ボタンのスタイルの設定である。画面中央に配置したボタンは、青地 (#44b) に白い大きな太文字で描画される。

"hello.css" 5d ≡

```
#hello_button {
  color: white; background: #44b;
  font-family: serif;
  font-size: 48pt;
  font-weight: bold;
}
```

◇

File defined by 5cd, 6.

最後に、放射状メニューのメニュー項目の設定である。ここでは radmenu 開発者が提供しているサンプルの設定をそのまま利用している。

"hello.css" 6 ≡

```
.radial_div_item {  
  position: absolute;  
  font-size: 18pt;  
  text-align: center;  
  background-color: #44f;  
  color: white;  
  width: 30px;  
  height: 30px;  
  padding: 10px;  
  -webkit-border-radius:30px;  
  -webkit-box-shadow: 5px 5px 5px gray;  
}
```

File defined by 5cd, 6.

A ファイル一覧

"hello.css" Defined by 5cd, 6.

"hello.html" Defined by 2a.

"hello.js" Defined by 4a.

B モジュール一覧

〈 CSS section 5b 〉 Referenced in 2a.

〈 JavaScript section 2b 〉 Referenced in 2a.

〈 Menu definition 4b, 5a 〉 Referenced in 4a.