

# [SmartPad4i]

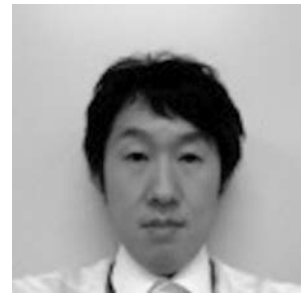
# スマートデバイスWebアプリケーション入門 —HTMLを使ったユーザーインターフェースの工夫

- はじめに
- JC/400、SmartPad4i における画面作成の基本
- HTML5 を利用した画面項目の工夫
- HTML5 を利用した画面表示の設定
- HTML5 を活用したグラフの実現
- さいごに



略歴 尾崎 浩司  
 1973年8月16日生  
 1996年 三重大学 工学部卒  
 1999年10月 株式会社ミガロ、入社  
 1999年10月 システム事業部配属  
 2013年4月 RAD 事業部配属

現在の仕事内容：  
 ミガロ、製品の素晴らしさをアピールするためのセミナーやイベントの企画・運営などを主に担当しています。



略歴 國元 祐二  
 1979年3月27日生  
 2002年3月 追手門学院大学 文学部アジア文化学科卒  
 2010年10月 株式会社ミガロ、入社  
 2010年10月 RAD 事業部配属

現在の仕事内容：  
 JC/400、SmartPad4i、Business4 Mobileの製品試験やサポート業務などを行っています。

## 1.はじめに

近年、スマートフォンやタブレットといったスマートデバイスを業務システムでも活用したいという要求が大きくなっている。

JC/400は、IBM iを活用したPC向けWebアプリケーション開発ツールだが、そのオプションであるSmartPad4iを使用するとスマートデバイス向けのWebアプリケーションも作成できるようになる。

PC向けの画面の場合、十分なサイズの画面や操作性の高いマウスやキーボードが使用できるが、スマートデバイスの場合、画面サイズに制約があり、入力操作などもソフトキーボードとなるため、表示や入力に制約が発生しやすい。

つまり、使いやすいスマートデバイス向けのWebアプリケーションを作成するには、ユーザーインターフェースの工夫が必要となるのである。

本稿では、SmartPad4iを使用した、スマートデバイス向けのWebアプリ

ケーションを作成する際に役立つスキルを紹介したい。

## 2.JC/400、Smart Pad4iにおける画面作成の基本

JC/400やSmartPad4iでアプリケーションを作成する場合、従来の5250アプリケーションで作成していたDSPF(画面ファイル)の代わりにHTMLを使用して画面を作成する。その際、入出力項目の要素にユニークなid名を設定するのがルールである。DSPFにおけるフィールドID同様、HTMLの要素にid名を設定することで、RPGあるいはCOBOLのプログラムから入出力項目にアクセスできるようになる。

作成したHTMLをDesignerで読み込み、入出力フィールドの型や長さを指定後、IBM iにRPGあるいはCOBOLを配布する。配布されたスケルトンのRPG/COBOLに業務ロジックを記述することで、簡単にWebアプリ

ケーションが作成できるのが特徴だ。【図1】

JC/400やSmartPad4iでは、型や文字長に応じたエラーチェックや入力制御が自動的に設定される。また、RPG/COBOL側のサブルーチンを呼び出すだけで、HTMLの属性を変更する機能も用意されているため、JavaScriptなどを使用しなくても、使いやすいWebアプリケーションが構築できるようになっている。

もちろん、標準機能で一通りのWebアプリケーションを構築できるのだが、さらにHTMLやJavaScriptなどを活用すれば、よりさまざまな機能やインターフェースを作成できる。

それでは本稿のテーマである、スマートデバイス向けのWebアプリケーション作成に役立つスキルとして、まずはHTML5を活用したSmartPad4iのインターフェースの拡張に焦点をあてて紹介していこう。

図1 JC/400, SmartPad4iWEBアプリケーション開発手法



図2 HTML5 Placeholder属性

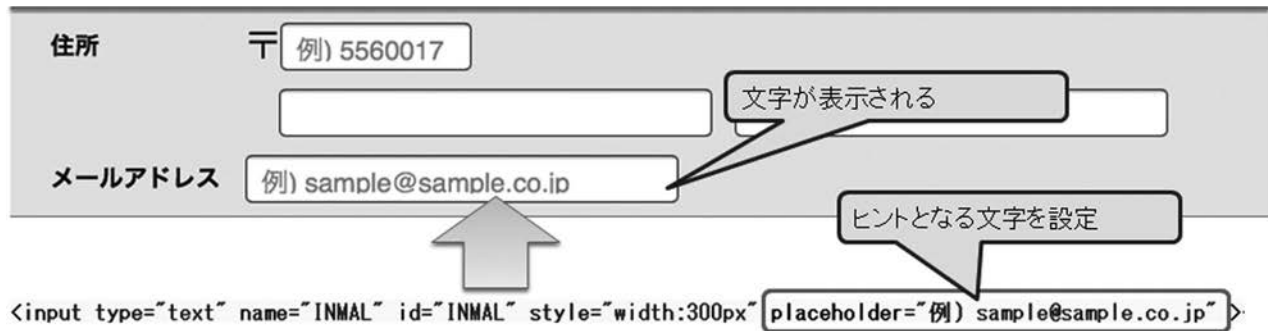
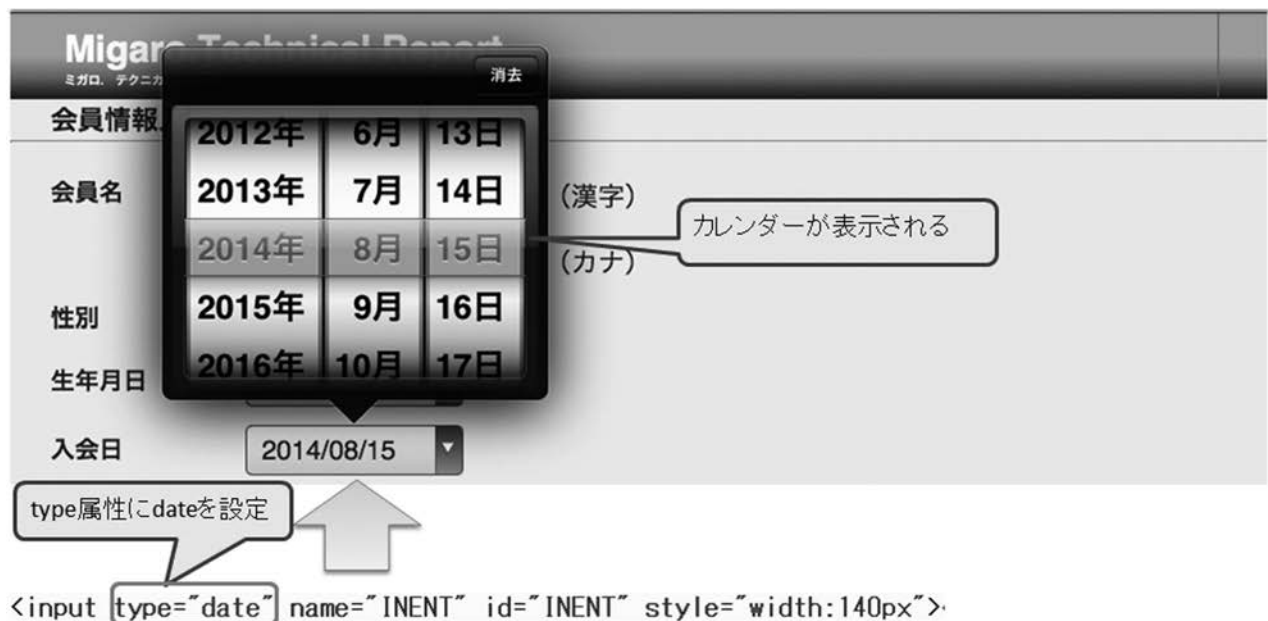


図3 HTML5 type属性date



## 3.HTML5を利用した画面項目の工夫

### HTML5とは?

HTML5 は、World Wide Web Consortium (W3C) で進められている HTML の 5 度目のメジャーバージョンアップで、2008 年頃に草案が発表され、現在正式勧告に向けて策定中の最新仕様である。ブラウザの種類ごとに対応状況は異なるが、さまざまな新しい機能が各ブラウザに実装されてきている。

従来の HTML では、flash や Java などのプラグインを利用しないと動画や音声、グラフィックなどを制御することは難しかったが、HTML5 では、新たに追加された video タグや、audio タグ、canvas タグなどを使用することで、プラグインなしでこれらを取り扱えるようになった。それ以外にも、クライアント上に大量データを保持する機能や、現在位置情報の取得、フォーム上の入力書式設定や妥当性チェックなどさまざまな機能が追加されている。

スマートデバイスに対応した SmartPad4i では、この HTML5 を利用することが可能である。特にスマートデバイスで利用される Safari や Chrome といったブラウザは HTML5 への対応が進んでいるため、HTML5 を活用することで、今までにないインターフェースや表現を容易に実現できるようになった。

まずは、SmartPad4i で比較的簡単に利用でき、効果的な HTML5 の機能について紹介したい。

### placeholder

Web アプリケーションで、薄いグレーの文字があらかじめ入っている入力欄を見たことがないだろうか。HTML5 で追加された placeholder 機能を利用すると、フォームの入力欄にあらかじめ入力のヒントとなる文字列を設定しておくことができるのである。例えば、メールアドレスの入力欄にあらかじめ、“例) sample@sample.co.jp” のような入力例を初期表示しておくことができる。

以前の HTML では、同様の機能を JavaScript で実装する必要があったが、placeholder を利用すると、input タグ

や textarea タグに placeholder 属性を追加するだけで利用できるようになる。

【図 2】

スマートフォンなどの限られた画面サイズでは、別途説明エリアを設けるのは難しいことが多いため、こういった工夫が効果的だろう。これは簡単に追加できるので、ぜひ入力項目に設定することをお勧めしたい。

### フォーム機能の拡張 type 属性

HTML5 では、フォーム機能が拡張されて、さまざまな type 属性が利用できるようになった。その中で今回は、type 属性の date について取り上げたい。

input タグの type 属性に date を設定すると、HTML5 に対応しているブラウザであれば、カレンダーなどの入力補助が表示されるようになる。例えば、iPhone や iPad で利用される Safari では、ドラム式のカレンダーが表示される。

【図 3】

このように type 属性を利用するだけで、リッチなインターフェースを利用できるので。しかも date を指定したものは日付値の入力ということが明確なため、日付値以外の値が入力できなくなるのも利点であろう。

画面自体はこれだけで利用できるようになるが、これを SmartPad4i の入出力項目として利用するには、ひと工夫必要だ。なぜならば、この date を指定した input 項目は、“2014-01-01” のように 10 桁の文字として値がセットされるからである。つまり date を利用する場合、SmartPad4i の Designer における System i Type (属性) は Atype に、System i length (桁数) は 10 に設定する必要がある。

IBM i 上のファイルでは、日付値を“20140101”のような数値 8 桁で保持することが多いだろう。そのため、例えば入力項目であれば、IBM i 側で日付文字列 10 桁を数値 8 桁に変換するようなロジックを作成すればよいのである。【ソース 1】は、RPG を使用した場合の記述例だ。【ソース 1】

逆に出力項目の場合には、8 桁の数値を“2014-01-01”のような書式に変換して出力すればよいこともご理解いただけるであろう。

### 入力妥当性チェックの追加

HTML5 では、入力欄の妥当性チェック機能として、未入力チェックの required 属性や、正規表現を使ったチェックが可能な pattern 属性が追加された。

SmartPad4i でも、これらの属性と CSS の invalid フィルターとを組み合わせることで、入力誤り通知機能を追加することができる。

必須入力欄に文字が未入力である場合や、形式に合わない文字列が入力されているなど、入力に不備がある場合に、入力欄の背景色を変更するには、以下の設定を行う。

まず、【ソース 2】のように HTML の head タグ内に style タグで CSS の定義を追加する。これは、項目値が不正 (invalid) となった場合の書式を設定しており、今回の場合は、background-color (背景色) の色コードを定義しているのである。【ソース 2】

あとは、HTML 内の各 input タグに必須項目であれば、required 属性を、書式をチェックしたい場合は、pattern 属性を追加するだけで実現できる。【図 4】

書式をチェックする pattern の場合、チェックしたい書式を正規表現で定義するのがポイントである。

正規表現とは、文字列の集合を 1 つの文字列で表現する方法で、例えば、郵便番号やメールアドレスなど特定の文字列パターンで表せるものをチェックするために利用することが多い。

例えば、郵便番号であれば、一般に 7 桁の数字で表現できる。数字 7 桁を定義する正規表現は、“^¥d {7}” のようになる。

これは、先頭文字 (^) から数字 (¥d) が 7 桁 ({7}) であることを表現しているのである。また、メールアドレスの場合は、例えば、“^[a-zA-Z0-9.!#\$%&'\*/=?^\_`{|}~-]+@[a-zA-Z0-9-]+(?:¥.[a-zA-Z0-9-]+)\*\$” のように表現するとよいだろう。

簡単な実装で、リアルタイムに入力の誤りを表現することができるので、ぜひ活用していただきたい。

## ソース1 type属性date RPG側で変換

```

0432.00 *-----
0433.00 * 文字型を日付に変換する
0434.00 *-----
0435.00 C          SBCRDY  BEGSR
0436.00 C*
0437.00 C          MOVEL*BLANK  WKYYMM
0438.00 C          MOVEL*BLANK  WKYYMD
0439.00 C*
0440.00 C          4          SUBSTWKGDAY:1  TYEAR  4          年 YYYY
0441.00 C          2          SUBSTWKGDAY:6  TMON   2          月  MM
0442.00 C          2          SUBSTWKGDAY:9  TDAY   2          日  DD
0443.00 C**
0444.00 C          TYEAR   CAT  TMON:0  WKYYMM
0445.00 C          WKYYMM  CAT  TDAY:0  WKYYMD
0446.00 C          MOVE   WKYYMD  WKIDAY          日付に変換
0447.00 C          ENDSR
  
```

10桁の文字を8桁の数値にする

## ソース2 CSS3 invalidフィルターの定義

HTMLファイル

```

<!DOCTYPE html>
<html>
<head>
<meta charset="Shift_JIS" />
<meta name="viewport" content="width=device-height" />
<meta name="format-detection" content="telephone=no" />
<link rel="stylesheet" href="DEMOCSS.css" type="text/css">
<title>MIGARO. Customer System</title>
<style>
input:invalid {
background-color: #ffcccc;
}
</style>
</head>
<body>
  
```

入力に誤りがある場合、背景色変更する定義

## 図4 HTML5 require、pattern属性による入力確認

require属性

未入力の場合は背景色が変わる

会員名  (漢字)

会員名  (漢字)

requiredを設定

```

<input type="text" name="INNMI" id="INNMI" required placeholder="例) 山田□太郎">
  
```

pattern属性

数値7桁でない場合は背景色が変わる

住所 〒

住所 〒

patternを設定

```

<input type="text" name="INAD1" id="INAD1" maxlength="7" pattern="^\d{7}$">
  
```



## 4.HTML5を利用した画面表示の設定

前節では、入出力項目となる input タグで活用可能な HTML5 を紹介したが、本節では画面全体の制御に関連する項目を紹介したい。

### viewport

SmartPad4i はスマートデバイスに対応した Web アプリケーションが作成できるツールである。スマートフォンや、タブレットはデバイスごとに画面サイズがそれぞれ異なるため、画面サイズに適切なインターフェースの作成が必要となる。

スマートデバイスに対応していない、Web サイトをスマートフォンで見たことがあるだろうか。

スマートデバイスで、PC 向けのサイトをそのまま表示すると、縮小してサイト全体を表示する。その結果、文字が小さくなってしまい、拡大しないとちゃんと文字が読めない状態になるだろう。【図 5 左】

しかし、最近の Web サイトでは、スマートフォン用画面が用意されており、スマートフォンでアクセスした際に利用しやすい表示になることが多い。【図 5 右】

この時、スマートデバイス向けに HTML を作成していることをブラウザに通知する方法が、viewport である。

スマートフォンやタブレット向けにインターフェースをデザインする場合には、head タグ内に meta 要素として viewport を指定すると、画面がデバイスのサイズに合わせて適切なサイズで表示してくれるのである。

例えば、【ソース 3】のように viewport を定義した場合、デバイスの横幅に合わせて描画をする指定となる。【ソース 3】

また、viewport には、画面の横/縦幅に合わせて描画する機能だけでなく、初期表示時の拡大率や、拡大縮小の操作の制御の可否なども設定することができる。【ソース 4】

例えば、初期表示される拡大率は initial-scale で設定可能である。100% サイズの場合を 1.0 とし、0~10 の範囲で数値が大きくなるに従って、初期表示される拡大率が大きくなるのである。同

様に minimum-scale は最小の拡大率、maximum-scale は最大の拡大率で、initial-scale と同様に 0~10 の範囲を指定可能だ。user-scalable はユーザーのピンチ操作（縮小や拡大）の可否を設定できる。

このように viewport の設定を適切に指定することで、ユーザーが利用しやすい画面を作成することができる。スマートフォンやタブレット向けのアプリケーション画面を作成する場合には、viewport の設定をお勧めしたい。

### レスポンシブルデザイン

viewport の項でも触れたが、PC とスマートフォンとは、画面サイズが異なるため、PC ブラウザ向け画面と、スマートフォン向け画面の 2 つを用意したい場合があるだろう。その場合、通常 2 種類の HTML を作成する必要があるが、レスポンシブルデザインという CSS の機能を利用した Web デザインの手法により、1 つの HTML を PC ブラウザとスマートフォンの両方に対応させることが可能になる。【図 6】

このレスポンシブルデザインは、SmartPad4i のアプリケーションでも有効で、1 つの HTML で PC ブラウザとスマートフォンの両方に対応した画面を表示するアプリケーションを作成することができるのだ。【図 7】

このレスポンシブルデザインで使用する CSS (カスケードリング・スタイルシート) とは、Web ページのスタイルを指定するための言語で、文書定義である HTML と組み合わせて利用するものだ。ここでは CSS の基本については触れないが、インターネットなどで検索すると説明があるので、ご存じない方は一度確認しておいてほしい。

レスポンシブルデザインは、HTML5 で追加された CSS のメディアクエリーという機能を利用することで作成できる。この機能を用いると、画面サイズごとにデザイン定義である CSS を切り替えて画面を表示できるのだ。

メディアクエリーの記述方法は簡単である。

例えば、@media 内に max-width : 640px と指定すると、表示横幅 640px 以下の場合のみ適用する CSS ができるのである。【ソース 5】

作成した CSS に名前を付けて保存しておき（ここでは“DEMOCSS\_mobile.css”として保存）、文書定義である HTML と同一階層へ配置の上、HTML の外部リンクで、通常サイズ用の外部スタイルシートを読み込んだあとに、640px 以下用の外部スタイルシートの読み込みを追加すればよい。【ソース 6】

このような定義を行うと、PC やタブレットを想定した 640px より大きいサイズをもつデバイスの場合、標準の CSS（この例では、“DEMOCSS.css”）のみが適用されるが、スマートフォンを想定した 640px 以下のデバイスの場合、標準 CSS に付加して、“DEMOCSS\_mobile.css”が適用されるのである。

SmartPad4i の場合には、1 つの HTML が 1 つの RPG プログラムと関連付けられるため、レスポンシブルデザインを利用することで、異なる画面サイズ用に類似のプログラムを複数作成しなくてもよいというメリットも出てくるであろう。そのため、PC 用とスマートフォン用の 2 つのデザインを用意したい場合には、レスポンシブルデザイン対応の HTML の作成をお勧めしたい。

## 5.HTML5を活用したグラフの実現

### canvas について

データを分析する際、最も有効なのはグラフの利用である。直感的に状況がわかるグラフを IBM i 上に登録されたデータから表示できると便利であろう。そこで今回は、HTML5 の canvas を使ったグラフ表示方法を紹介したい。

canvas は、HTML5 で追加された要素で、ブラウザ上に図形を描画するために利用するものである。

従来の HTML では、Flash や Java を別途利用しなければブラウザ上に図形を描画するのは困難であったが、HTML5 で追加された canvas により、JavaScript ベースで簡単に図形描画できるようになった。

もちろん、canvas を利用すれば、自由に図形描画ができるため、独自の実装でグラフを作成することも可能だが、それでは手間がかかるだろう。そこで今回は、canvas にグラフを描画する方法として、オープンソースで提供されている

図5 yahoo! トップページ



ソース3 Viewportの記述1

```

<!DOCTYPE html>
<html>
<head>
<meta charset="Shift JIS" />
<meta name="viewport" content="width=device-width" />
<meta name="format-detection" content="telephone=no" />
<link rel="stylesheet" href="DEMOCSS.css" type="text/css">
<title>MIGARO.Technical Report</title>

```

headタグ内にviewportを定義

ソース4 Viewportの記述2

```

<!DOCTYPE html>
<html>
<head>
<meta charset="Shift JIS" />
<meta name="viewport" content="width=device-width,initial-scale=1.0,
minimum-scale=1.0,maximum-scale=1.0,user-scalable=no">
<meta name="format-detection" content="telephone=no" />
<link rel="stylesheet" href="DEMOCSS.css" type="text/css">
<title>MIGARO.Technical Report</title>

```

headタグ内にviewportを定義  
拡大率固定、ユーザの拡大縮小操作をさせない



【Flotr2】ライブラリを利用したい。(公式サイト：<http://humblesoftware.com/flotr2/>)

Flotr2を利用すると、canvas上に、棒グラフや円グラフ、レーダーチャートグラフのような、さまざまなグラフを簡単に描画できる。

なお、グラフの描画には、外部 Web サービスである Google Chart API などを組み合わせた方法もあるが、今回の方法はインターネットに接続しなくても実行できることがメリットである。また、外部 Web サービスの場合、サービス提供者が API の仕様を変更する可能性もあるため、その際にはプログラムの変更対応が必要となるが、今回の方法では、ライブラリのスクリプトファイルをバージョンアップしない限り、同じプログラムを使い続けられることもメリットだ。

ここでは例として、会員の年代別割合を表示する円グラフの実現方法を説明する。データベースにある会員情報には、年齢が含まれるため、年齢から年代を取得して年代あたりの会員数を集計して、それをグラフで出力するのだ。【図 8】

## HTMLの作成

HTMLを利用したユーザーインターフェース部分を作成していこう。ここではグラフを埋め込むのに必要な箇所について説明したい。

### (A) Flotr2 の設定

まず、HTMLにFlotr2ライブラリを読み込む必要がある。<https://github.com/HumbleSoftware/Flotr2>からDownloadZIPボタンをクリックしてファイルをダウンロードする。【図 9】

次に、ダウンロードしたファイルを展開したデータ内にあるflotr2.min.jsを取得して、HTMLと同階層にjsフォルダを作成後、flotr2.min.jsを配置する。

配置後、HTMLのheadタグ内にFlotr2の外部JavaScriptファイルの参照を追加すればよい。【ソース 7】

### (B) 表示領域の設定

次に、HTMLにグラフを表示する領域を定義する。今回、グラフ本体を描画するdivタグには、id名にgraphViewと指定している。【ソース 8】

グラフ出力の実装例は後述するが、こ

のgraphViewに対してFlotr2のグラフ出力処理を実行するのである。また、id名がGDATAのtable要素には、会員の年代と会員数を出力する。つまり、SmartPad4iのサブファイルを利用して、RPG側から年代と会員数を出力するのだ。

## RPGの作成

### (A) 配列定義の設定

次にIBM i側の処理例を紹介しよう。今回はRPGでの実装例を紹介したい。RPG側の定義では、年代別のラベルと年代別の会員数の合計を保持する配列を定義する。【ソース 9】

今回の例では、コンパイル時配列を利用し、【ソース 10】のように10代から100代までのラベル名称を定義した。【ソース 10】

### (B) サブファイル出力の設定

【ソース 11】および【ソース 12】が会員ファイルを読み込んで、会員一覧を画面出力する際に、年代別の会員数をカウントして、その結果を年代リスト用のサブファイルに出力する例である。【ソース 11、ソース 12】

年代別のラベルと会員数を出力可能な仕組みができれば、RPG側の作業は完了だ。

## グラフの作成

### (A) initpage 関数の設定

次は、グラフの表示元データとなる、年代別のサブファイルをJavaScriptから読み込み、Flotr2のグラフ出力機能呼び出す処理を記述する。

ここで、initpage関数とcansubmit関数について説明しておこう。JC/400およびSmartPad4iでは、HTML内にinitpage関数を定義しておくと、RPG/COBOLプログラムが実行されて画面がブラウザに表示される時に、initpage関数に記述したJavaScriptが実行される。

また、cansubmit関数を定義しておくと、IBM iに値を送信(submit)する直前のタイミングにcansubmit関数が実行される。つまりページ表示時に、グラフを表示するには、initpage関数内でグラフ描画処理を記述することが必要だ。実装例が、【ソース 13】である。

## 【ソース 13】

### (B) 要素の取得とグラフ描画の設定

関数の先頭部分でSP4i.getElementByIdメソッドを利用しているのがわかると思うが、このSP4i.getElementByIdメソッドは、引数に指定したid名を元に実行時に要素を取得する方法である。この方法は、SmartPad4iにおいてJavaScript側で処理を実行する際に多用するので、ぜひ覚えておこう。

では、ソース 13の内容を確認していこう。まず初めに、①のように、SP4i.getElementByIdメソッドで年代別サブファイルのtable要素を取得する。

次に、②のように、サブファイルの情報をrowsプロパティとcellsプロパティを利用して配列に格納する。この際1行目は、列タイトルを表示しており、1行目を読み飛ばすためにforループの開始値は1と設定している点にご注意いただきたい。

グラフに必要なデータを配列に格納したら、あとはFlotr2を利用してグラフを出力すればよいのである。

出力には、Flotrオブジェクトのdrawメソッドを実行する。このdrawメソッドには、次の3つの引数がある。

1つ目には、図形を描画するdiv領域を指定する。今回はソース 8で定義したgraphViewのid名をもつ要素を使用する。このgraphViewはSmartPad4iのDesignerで定義した項目ではないため、③のようにdocument.getElementByIdで要素を取得する必要がある。

2つ目には、グラフの元になる配列データをセットする。今回は、②の部分で取得した配列を④のようにdgAttArr変数として渡している。

3つ目は、描画するグラフ表示を設定するオプション設定である。⑤のように、横軸のラベルや、縦軸のラベル、グラフの形状などを設定すればよい。

今回のように円グラフを表示する場合、横軸ラベルや、縦軸ラベルは不要のため引数にfalseを設定している。

以上でプログラムは完了である。完成したプログラムを実行すると、集計された年代別の人数をもとに円グラフがきれいに表示されるのである。

図6 レスポンシブルデザイン

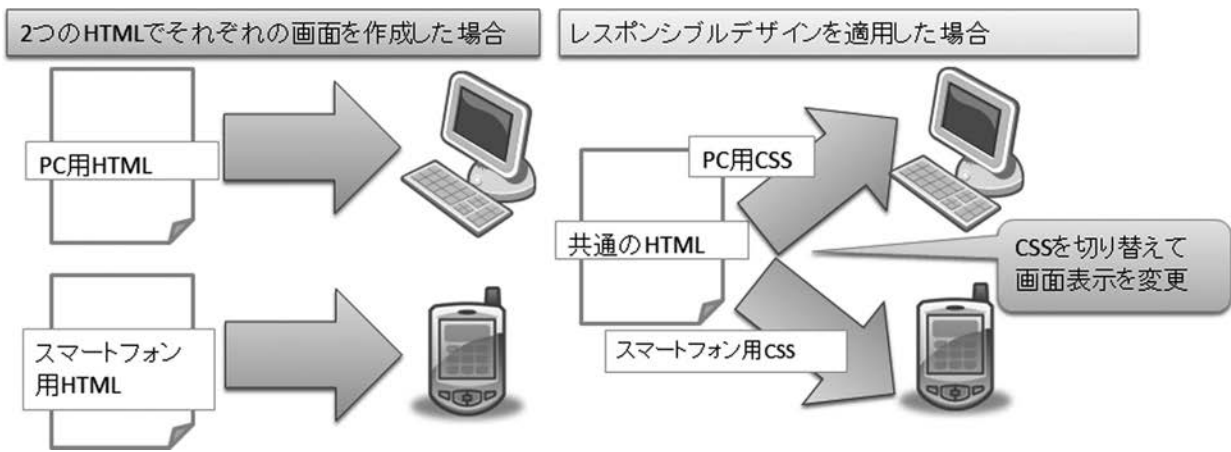


図7 レスポンシブルデザイン例

PC, iPad等で表示

No.	会員名(漢字)	会員名(カナ)	性別	年齢	入会日
00000001	細川 エリカ	ホソカワ エリカ	女性	20	2012/01/05
00000002	大原 結衣	オオハラ ユイ	女性	47	2012/02/12
00000003	藤澤 南朋	フジサワ ナオ	女性	18	2012/03/21
00000004	松田 恵麻	マツダ エマ	女性	52	2012/03/30
00000005	有村 理紗	アリムラ リサ	女性	69	2012/04/02
00000006	安藤 扶樹	アンドウ モトキ	男性	45	2012/04/04
00000007	若山 弘也	ワカヤマ ヒロナリ	男性	62	2012/05/02
00000008	菊田 竜也	キクダ リュウ	男性	22	2012/05/02
00000009	寺脇 育二	テラサキ ユウジ	男性	22	2012/05/02
00000010	高見 浩正	タカミ ヒロユキ	男性	57	2010/03/26
00000011	村井 莉央	ムライ リオ	女性	58	2010/06/12
00000012	宮本 雄大	ミヤモト ユウダイ	男性	22	2012/05/12

赤枠は、スマートフォン利用時に非表示に設定

スマートフォンで表示

スマートフォン表示時レイアウトを変更

No.	会員名(漢字)	性別	年齢
00000001	細川 エリカ	女性	20
00000002	大原 結衣	女性	47

ソース5 レスポンシブルデザイン用CSS

```
@media<
only screen and (max-width : 640px){ <
/* デバイスの横幅が640以下の場合に適用するCSS を以下に記述する*/<
<
/* タイトルのフォントサイズ変更 */<
.Header h1{<
font-size:20px;<
}<
/* 加名を表示しない */<
.name{<
display:none !important;<
}<
<
/* 区切り線を表示する */<
.mline{<
display:block;<
}<
/* 入力欄サイズを大きくする */<
.NNEN{<
width:360px !important;<
}<
}<
```

デバイスの横幅が640px以下の場合に適用するCSS



## Flotr2について

なお、今回は円グラフの例を紹介したが、Flotr2では設定により、さまざまなグラフを出力することが可能である。さらに静的なグラフだけでなく、アニメーションを持つグラフも作成できるなど、非常に高機能なグラフ出力ライブラリである。いろいろと応用してみたい。Flotr2の公式サイトにはさまざまな例が記載されているので、ぜひ参考にしていきたい。【図 10】

## 6.最後に

本稿では、スマートデバイスに対応する Web インターフェースの工夫について紹介した。SmartPad4iではHTML5が利用できるため、リッチなインターフェースを簡単に作成できることをおわかりいただけたかと思う。

さらに、SmartPad4iは、Flotr2のような外部のライブラリと連携しやすい点も魅力の1つである、便利なライブラリが多数あるので、ぜひいろいろな連携に挑戦していきたい。

**M**

## ソース6 レスポンシブルデザイン用CSS 外部読み込み

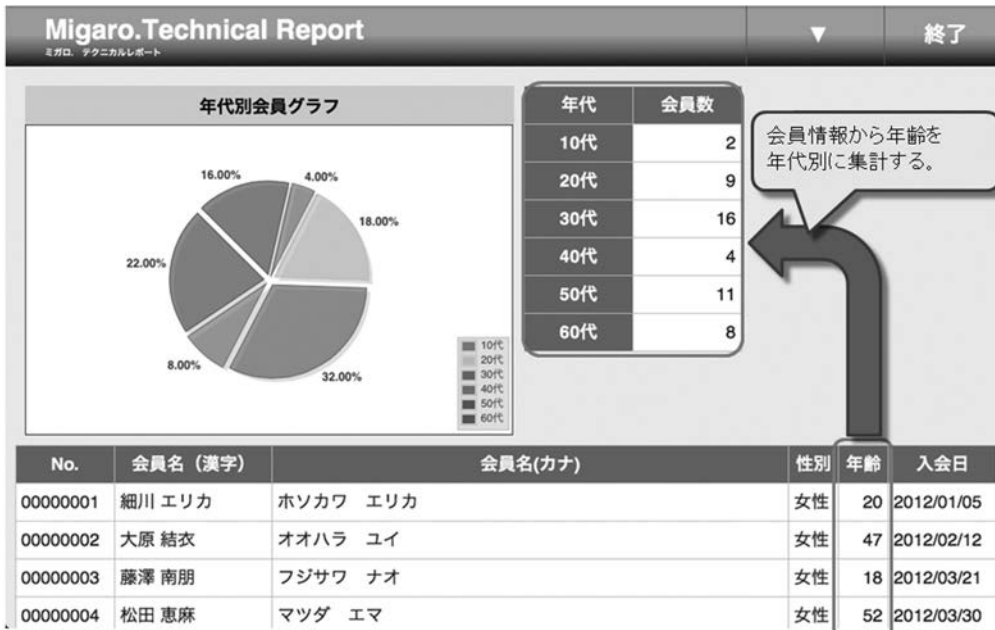
```

<!-- saved from url=(0014)about:internet -->
<!DOCTYPE html>
<html>
<head>
<meta charset="Shift_JIS" />
<meta name="viewport" content="width=device-height" />
<meta name="format-detection" content="telephone=no" />
<link rel="stylesheet" href="DEMOCSS.css" type="text/css">
<link rel="stylesheet" href="DEMOCSS mobile.css" type="text/css">
<title>MIGARO.Technical Report</title>

```

cssは最後に読み込まれるように追加

図8 グラフ完成画面



会員情報から年齢を年代別に集計する。

図9 flot2 ダウンロード

The screenshot shows the GitHub repository page for "HumbleSoftware / Flotr2". The repository has 2,123 stars and 407 forks. The file list includes:

- dev: Added spacing notes. 3 years ago
- examples: Add candle barchart to the displayed examples. 7 months ago
- js: Firefox mobile browser doesn't use window.event. So changed the code ... 2 months ago
- lib: Updated to jasmine-1.2.0 2 years ago
- make: Added examples for financial bar chart 7 months ago
- spec: Build and update reference. 5 months ago
- .gitignore: Add package.json for dependencies 5 months ago
- CONTRIBUTING.md: Added CONTRIBUTING.md 2 years ago
- LICENSE: 2012. 3 2 years ago
- Makefile: Add smooch dependencies to Makefile 5 months ago

On the right side, there are options to clone the repository or download it as a ZIP file. A callout box points to the "Download ZIP" button with the text "ダウンロード".

ダウンロード

### ソース7 flotr2.min.jsの定義

```

<!-- saved from url=(0014)about:internet -->
<!DOCTYPE html>
<html>
<head>
<meta charset="Shift_JIS" />
<meta name="viewport" content="width=device-height" />
<meta name="format-detection" content="telephone=no" />
<link rel="stylesheet" href="DEMOCSS.css" type="text/css">
<title>MIGARO.Technical Report</title>
<script type="text/javascript" src="js/flotr2.min.js"></script>

```

外部JavaScriptファイルの参照追加

### ソース8 グラフの表示域とグラフデータのサブファイルを定義

```

<!-- グラフの表示域 -->
<div id="gView" >
  <div id="graphTitle" class="graph-title"></div>
  <div id="graphView"></div>
</div>
<!-- グラフデータの表示域 -->
<div id="graphData" >
  <table id="GDATA" class="PAGE" border="1px" cellspacing="0" cellpadding="5px">
    <thead>
      <th style="width:100px;">年代</th><th style="width:100px;">会員数</th>
    </thead>
    <tbody>
      <tr>
        <th id="GD1"></th><td id="GD2" style="text-align:right"></td>
      </tr>
    </tbody>
  </table>
</div>
<!-- 会員情報の表示域 -->

```

グラフを表示する領域

グラフデータを表示する領域

### ソース9 RPG側 年代ラベルとデータの配列定義

```

0014.00 F010 * <YOURCODE>
0015.00 ---> * YOUR FILES
0016.00 F** <顧客ファイル>
0017.00 FCUSTFP IF E K DISK
0018.00 F010 * </YOURCODE>
0019.00 *
0020.00 E010 * <YOURCODE>
0021.00 ---> * YOUR ARRAYS
0022.00 E**<グラフラベル>
0023.00 E @GL1 1 10 30
0024.00 E**<グラフデータ>
0025.00 E @GD1 1 10 5
0026.00 E010 * </YOURCODE>

```

年代別のラベル

年代別の会員数を格納する配列



ソース10 RPG側 コンパイル時配列

```

0598.00  ** @GL1
0599.00  10代
0600.00  20代
0601.00  30代
0602.00  40代
0603.00  50代
0604.00  60代
0605.00  70代
0606.00  80代
0607.00  90代
0608.00  100代
0609.00  ** @GD1
0610.00  0
0611.00  0
0612.00  0
0613.00  0
0614.00  0
0615.00  0
0616.00  0
0617.00  0
0618.00  0
0619.00  0
    
```

ソース11 RPG側 ファイルの読み込み

```

0507.00  C**<会員ファイルの読み込み>
0508.00  C          SETOF          50
0509.00  C          Z-ADD*ZERO    IND    40
0510.00  C          CLEARSOOF02
0511.00  C          *LOVAL    SETLLCUSTFR
0512.00  C          DO *HIVAL
0513.00  C          READ CUSTFR          81
0514.00  C          *IN81    IFEQ *ON
0515.00  C          LEAVE
0516.00  C          ENDIF
0517.00  C*
0518.00  C          IND          ADD 1          IND
0519.00  C          OCUR SOOF03
0520.00  C          EXSR RDFILE
0521.00  C          ENDDU
0522.00  C          Z-ADD1          JCL103 40
0523.00  C          Z-ADDIND       JCL903 40
0524.00  C* <グラフ処理>
0525.00  C          EXSR GRAPHD
    
```

ファイルの読み込み

データを出力するためのサブルーチン

グラフデータを出力するためのサブルーチン

ソース12 RPG側 会員データの表示

```

0550.00 *-----
0551.00 * 顧客ファイル出力
0552.00 * 一覧に出力する項目を選択
0553.00 *-----
0554.00 C          RDFILE  BEGSR
0555.00 C**<一覧出力>
0556.00 C          MOVELCUSTNO  OLCNO      顧客番号
0557.00 C          MOVELCUNAME  OLCNM1   名前
0558.00 C          MOVELCUKANA  OLCNM2   カナ名
0559.00 C          CUSEX      IFEQ '0'
0560.00 C          MOVEL'男性'  OLSEX    性別
0561.00 C          ELSE
0562.00 C          MOVEL'女性'  OLSEX
0563.00 C          ENDIF
0564.00 C          Z-ADDCUBIRT  OLBIRT   生年月日
0565.00 C          Z-ADDCUENTD  OLENTD   入会日
0566.00 C**<一覧出力終了>
0567.00 C**<グラフ用データ取得>
0568.00 C          CUAGE      DIV 10      G      50      年代取得
0569.00 C          G          IFLE 10
0570.00 C          G          IFNE 0
0571.00 C          MOVE @GD1,G  TGD1    50
0572.00 C          ADD 1      TGD1
0573.00 C          MOVE TGD1   @GD1,G
0574.00 C          ENDIF
0575.00 C          ENDIF
0576.00 C*
0577.00 C**<グラフ用データ取得終了>
0578.00 C          ENDSR
0579.00 *-----
0580.00 * グラフデータ
0581.00 *
0582.00 *-----
0583.00 C          GRAPHD  BEGSR
0584.00 C          Z-ADD*ZERO  GDIND  40
0585.00 C          Z-ADD*ZERO  I      40
0586.00 C          1          DO 10      I
0587.00 C          @GD1,I    IFNE '0'
0588.00 C          ADD 1      GDIND
0589.00 C          GDIND     OCUR SOOF02
0590.00 C          MOVE@GL1,I  OGD1    P      年代
0591.00 C          MOVE@GD1,I  OGD2    P      会員数
0592.00 C          ENDIF
0593.00 C          ENDDO
0594.00 C          Z-ADD1      JCL102  40      FIRST LINE TO SEND
0595.00 C          Z-ADDGDIND  JCL902  40      LAST LINE TO SEND
0596.00 C*
0597.00 C          ENDSR

```

データを出力するためのサブルーチン

年代別の会員数を配列に格納

グラフデータを出力するためのサブルーチン

ソース13 initpage関数の定義

```
function initpage(){
  var gtable = SP4i.getElementById("GDATA"); //年代別のサブファイル取得
  document.getElementById("graphTitle").innerHTML= '年代別会員グラフ'; //タイトル
  var dgAttArr= new Array();
  for(var i=1; i<gtable.rows.length;i++){
    var slabel=gtable.rows[i].cells[0].innerHTML; //年代別ラベル
    var gdata=gtable.rows[i].cells[1].innerHTML; //年代別会員数
    dgAttArr.push({data:[0,parseInt(gdata)], label:slabel }); //配列に追加
  }
  //グラフ描画
  Flotr.draw(document.getElementById("graphView"), dgAttArr
  , {
    HtmlText: false,
    grid: {
      verticalLines: false, //canvas領域 縦グリッドの表示
      horizontalLines: false //canvas領域 横グリッドの表示
    },
    xaxis: {
      showLabels: false //横軸ラベル
    },
    yaxis: {
      showLabels: false //縦軸ラベル
    },
    pie: {
      show: true //円グラフ
    },
    mouse: {
      track:false //タッチイベントの取得
    },
    legend: {
      position: "se", //凡例位置
      backgroundColor: "#D2E8FF" //背景色
    }
  });
}
```

①要素の取得

②年代別サブファイルの情報を配列へ格納

③id名graphViewのdiv要素を設定

④グラフ描画データ配列

⑤グラフ描画オプション

図10 flot2 サンプルサイト

