レスポンシブウェブの新しいデザインの提案
Proposal of a new design for Responsive web

尚美学園大学　斎藤 忍
Shobi University　Shinobu SAITO

要旨：スマートフォン（モバイル）用とパソコン（デスクトップ）用の両方に最適化された一つの HTML ファイルにより各ページが構成されたレスポンシブウェブが多くのホームページに使われ始めている。
しかし現状のレスポンシブウェブには、①構成が複雑（個人で制作するためには高度な専門知識が必要となり、アウトソーシングするためには高額な制作費が必要）、②モバイル用での視読性が劣る、③モバイルとデスクトップでの共有化が不完全、等の問題があり、個人事業主のテレワーカーにとって「ホームページ」は仕事の受発注やスキルを告知する上で重要なツールであるにもかかわらずスマートフォン用ホームページそのものが普及はしていない。
これらの問題の解決を目指したレスポンシブウェブの新しいデザインを提案する。

Abstract: Responsive Web has begun to be used in a number of homepage that each page has been constructed by both optimized one HTML file in for the desktop and mobile.
But in Responsive web of status quo,
① configuration complexity (requires highly specialized knowledge in order to produce an individual, requires expensive production costs in order to outsourcing),
② vision readability for mobile　is inferior,
③ sharing of mobile and desktop are incomplete,　etc. of the problem,
"Homepage" for teleworkers of a sole proprietorship is not popular to be an important tool despite Homepage itself for the smart phone in order to notice the ordering and skills of work.
this paper proposes a new design of Responsive web that aims to solve above problems.

キーワード: レスポンシブウェブ、ホームページ、HTML、モバイル
Keyword: Responsive web, website, HTML, mobile

1. スマートフォン用ホームページの実情

2013 年後半から国内の多くのホームページでのアクセスログのモバイル用がデスクトップ用を上回ったと言われている。しかし多くのホームページは未だにモバイル用に対応していない。

2014年11月よりモバイルでのGoogleの検索結果ではモバイルに対応しているホームページには「スマホ対応」の文字が表示されるようになった。

「テレワーク」というキーワードでモバイルでの Google の検索をすると、上位 100 件のうち「スマホ対応」は 38 件（2015 年 2 月 21 日現在）で、その大半が大手企業、中央省庁、新聞社、Facebook などで、中小企業や個人事業主のホームページは含まれていなかった。

スマートフォン用ホームページへの対応の遅れは業態、業種を問わず中小企業や個人事業主に共通している問題であり、上記①～③がその大きな要因であることが推測できる。

2. 現在のレスポンシブウェブの問題点

2.1 構成が複雑

HTML はハイパーテキストを利用した相互間文書参照のフレームワークであり、プレーンテキストの文書を要素で括って意味付けすることで、文書の特定要素に URI（Uniform Resource Identifier）を用いた他文書へのリンクを記載しておけばユーザエージェントはそれを解釈して指定された他文書を表示させることが可能となるものである。プログラム言語の中でも比較的難易度が低く、HTML1.0（1990 年代初期）では HTML を活用することで一般の人が気軽に実用的なホームページを制作することができた。

しかしHTML5.0となりレスポンシブウェブが登場すると、HTML のコーディングは難解な css や JavaScript、PHP などとの連携により高度で複雑になり、ホームページの制作には専門的な知識とスキルが必要とされるようになり、一般の人が気軽に作ることはできなくなってきている。

また、近年では HTML を直接入力してホームページを作る人は少なくなり、CMS（content management system）やウェブ上でのホームページ作成サービスを利用する場合が一般的になっている。CMS によるレスポンシブウェブでは<head>から</head>までの間が 100 行を超えるものも一般的になっている。

2.2 モバイル用での視読性が劣る

レスポンシブウェブの多くはデスクトップ用のページを制作した後にモバイル用に変換するため、コンテンツはデスクトップ用をそのまま表示している。画面サイズを紙に例えるとデスクトップ用が A4 サイズ相当で、モバイル用は名刺サイズ相当ということになる。モバイル用でも画面をスクロールしていくことで A4 サイズ相当のテキストを表示することが可能であるが、同じ文字数のテキストをモバイル用でスクロールしながら読むことは、視覚的にも精神的にもストレスを感じ、視読性に劣るということになる。

A4 サイズに書かれたテキストを名刺サイズに納めるためには「テキストを短くするための推敲」「省略された単語の多用」「命令形の多用」などによりテキストの簡略化を行う必要がある。逆にモバイル用に簡略化されたテキストをそのままデスクトップ用で表示したとしても問題がない場合がほとんどである。

従って最初にモバイル用のテキストを作ってからデスクトップ用に展開することが望ましいが、現状のレスポンシブウェブの多くは、その作成作業をデスクトップ上で行うために、自然とデスクトップを優先した構成になっている。

2.3 モバイルとデスクトップでの共有化

現在のレスポンシブウェブの多くはモバイルとデスクトップでの共有化についてはまだ不完全であるといって良い。

レスポンシブウェブの多くは横長のデスクトップ用画面を縦長のモバイル用画面に強制的に表示させている。またデスクトップ用画面ではブラウザの横幅の可変に応じてボタンの位置をオーバーフローさせて配置を変化させて画面を有効に使っているものもある。

利用者側からはデスクトップ用とモバイル用が同じページを共有していることが理解されず、利用者がそれぞれ異なったデバイス（デスクトップ用とモバイル用）や異なった環境、異なった設定で見ていた場合にコミュニケーションに不具合を生じる（例えば 2 段目の一番右のボタンは状況によって異なる）こともある。

また電話でのコミュニケーションの場合、通話中にモバイル用のページを確認することができないため、デスクトップ用を見ながらの会話では通話内容が噛み合わない場合もある。

<図 1> デスクトップとモバイルの 400 字の見え方

<図 2> 今までのレスポンシブウェブでは同じページであってもデバイスや設定によってボタンの位置が異なる

3. 本提案の具体的内容

3.1 徹底したシンプルなコーディング

viewport は content="width=device-width とはせず content="width=600"と幅にピクセルの数値を入れて固定(ピクセル数の 600 は任意であるが、以下 600 で統一して説明する)。以降の img や table など全て width=600 とすることで解像度の数値の変化によるモバイルでの見え方を一定にする。またモバイルの機種によってコーディングの表示結果が若干異なるため nitial-scal と user-scalable は敢えて入れない（入れない場合の初期値は nitial-scale =1.0, user-scalable=yes となる)。

HTML での構造化タグは header、aside、article、footer 及び div class=right をデスクトップ用(p.css)では全て利用。モバイル用（m.css）では nav と article のみを利用し、他の構造化タグは全て display:none;として非表示とする。こうすることでリダイレクトは不要となり、リダイレクトのための htaccess または JavaScript 等での記述も不要となる（一般的にレスポンシブウェブではリダイレクトは使用しないと定義されているが最終的に使わざるを得ない場合も多い)。また全く同一の HTML ファイルでデスクトップ用とモバイル用が兼用でき、アクセス数もデスクトップとモバイルで振り分けられたりリダイレクトのためだけのアクセス数が加算されることもない。

デスクトップ用（p.css）では横方向の分割を aside を 35%、article を 600pix、.right を 30%とすることで article は 600pix 固定のままブラウザの幅に応じては左右のスペースの幅は割合が同じままで可変する。また aside、article、.right のそれぞれの高さを 100%にして、aside、.right は position を fixed とすることでブラウザの高さ方向のサイズに追従するかたちとした。

index.html が約 70 行、p.css が約 70 行、m.css が約 35 行である。また他のページの HTML ファイルもサンプルファイルではコーディングの行数が 100 行以内におさめている。

＜ファイル一式＞ http://con-sma.com/h/mss/smpl.zip

この量であれば決められた時間内にコーディングを 1 行ずつ解説しながら初心者（大学生などにも）に指導し、実際にサンプルホームページを制作することも可能である。

3.2 デバイスの違いによる差異を最小限とした

次項＜図 5＞でデスクトップの中央やや右寄り縦長の部分（nav と article）がそのままモバイルで反映され、この縦長の部分はブラウザの幅方向のサイズやデスクトップモニターの解像度によって可変することはない。前記の通り header、aside、article、footer、div class=right（中央やや右寄り縦長の部分以外）はモバイルでは非表示となる。

p.css

```
aside{
position:fixed;
left:0px;top:70px;
margin-top:0px;
margin-left:0px;
width:35%;
min-height:100%;
background:#ddd;
z-index:1;
}

nav{
position:absolute;
left:35%;top:70px;
margin-top:0px;
margin-left:0px;
width:600px;
height:auto;
background:#fff;
z-index:2;
}

article{
position:absolute;
left:35%;top:190px;
margin-top:0px;
margin-left:0px;
width:600px;
min-height:100%;
background:#fff;
z-index:2;
}

.right{
position:fixed;
right:0px;top:70px;
margin: 0; padding: 0;
width:30%;
min-height:100%;
background:#ddd;
z-index:0;
}
```

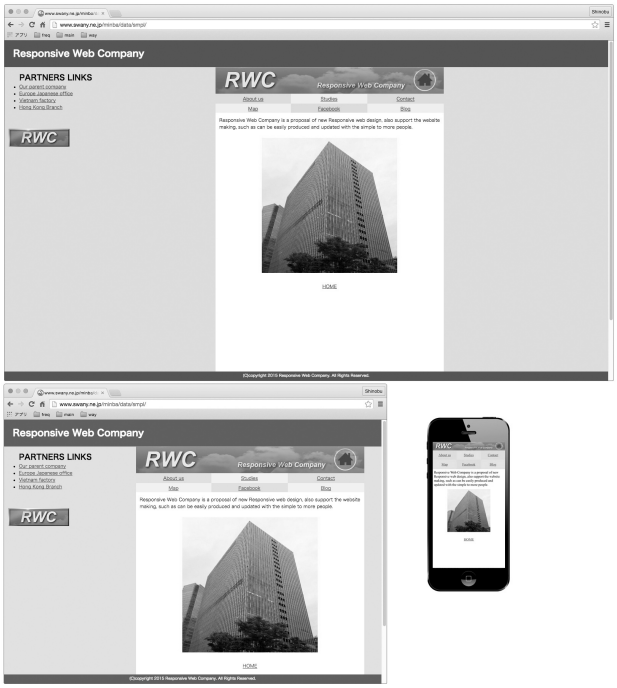
m.css

```
header{display:none;}
aside{display:none;}


nav{
left:0px;top:0px;
margin-top:0px;
margin-left:0px;
width:600px;
height:auto;
background:#ddd;
}

article{
left:0px;top:240px;
margin-top:0px;
margin-left:0px;
width:600px;height:100%;
}

td{
height:70px;
}


.right{display:none;}
footer{display:none;}
```

＜図 4＞ デスクトップ用（p.css）とモバイル用（m.css）

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=600">
  <link media="only screen and (max-device-width:1200px)" href="m.css" type="text/css" rel="stylesheet" />
  <link media="screen and (min-device-width:1201px)" href="p.css" type="text/css" rel="stylesheet" />
</head>
```

＜図 3＞ HTML ファイルの viewport と css の呼び出し部分



＜図 5＞ デスクトップ（ブラウザの幅が広い場合と狭い場合）とモバイルでの表示の比較

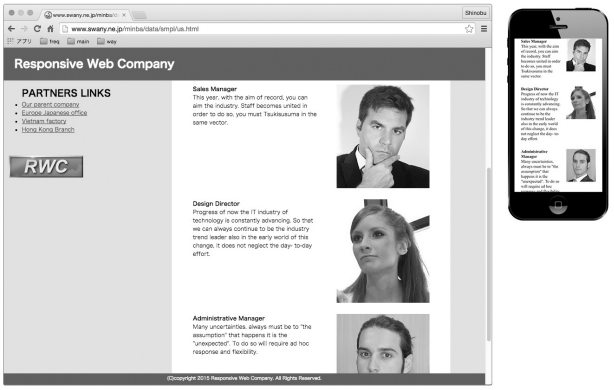
3.3 table による画像とテキストの配置

viewport を content="width=600"とし、ここに width=550（600 以下の数値で）の table を配置する。幅方向に対して 1/3 のサイズの画像を配置する場合にはその画像を width=200 と指示することで、常に横幅に対して 1/3 のサイズで画像が表示される。

これは画像の幅が 200pix ということではなく、viewport が幅方向を 600pix と仮定した場合の相対的なサイズであり、この相対関係はデバイスの種類が変わっても一定である。

従って横幅に対して 1/2 のサイズが必要であれば width=300、1/6 であれば width=100、というように常に content="width=との相対値で固定できる。

従来のモバイル用 HTML では横方向の複数の画像の配置、横方向でのテキストと画像の配置をすることが非常に難しかった。また HTML5 による構造化によって詳細な画像の配置を指定するためには座標の確認作業を含めて大きな労力を必要とされた。



＜図 7＞table 部分のデスクトップとモバイル

3.4 クリカッブルマップの活用

クリカッブルマップを利用することで 1 枚の画像の任意の部分にリンクの設定ができ、リンクのために画像を細かく分割する必要がなくなる。

特にモバイル用の限られた面積の中でのクリカッブルマップの活用は有効である。

ただしモバイル用の HTML ではデスクトップ用のクリカッブルが使用できないため、jquery を利用してクリカッブルマップに対応する必要がある。jquery には js（JavaScript）ファイルをダウンロードしてサーバー内に設置する方法があるが、本サンプルではコーディングを簡略化するために jquery は外部リンクとした。

HTML では viewport を content="width=600"とすることで設定した座標の数値が相対値としてデバイスの種類による可変に影響されることなく使用することが可能になる。ここで入力する座標値もあくまで 600pix を想定した相対値である。

＜図 8＞ クリカッブルマップ部分のデスクトップとモバイル

```
54  ␣␣↓
55  ␣␣<table cellpadding=0 cellspacing=20 width=550 align=center><tr><td valign=top>↓
56  ␣␣<b>Sales Manager</b><br>↓
57  ␣␣This year, with the aim of record, you can aim the industry. Staff becomes united in order to do so,
    Tsukisusuma in the same vector.<br><br>↓
58  ␣␣<td><td valign=top><img src ="sm.jpg" width =200></td></tr>↓
```

＜図 6＞HTML ファイルの table 部分

<図 11>YouTube の動画へのリンク部分

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=600">
  <link media="only screen and (max-device-width:1200px)" href="m.css" type="text/css" rel="stylesheet" />
  <link media="screen and (min-device-width:1201px)" href="p.css" type="text/css" rel="stylesheet" />
  <script type="text/javascript" src="js/jquery.rwdImageMaps.min.js"></script>
</head>
```

```
<article>
  <img src="stdy01.jpg" width="600" usemap="#stdy01">
  <map id="stdy01" name="stdy01">
  <area shape=rect coords="57,11,276,153" href="http://www.ball-park.jp/" target=_blank>
  <area shape=rect coords="324,11,543,153" href="http://www.swany.ne.jp/minba/u/" target=_blank>
  </map>
```

<図 9>jquery 呼び出し部分とクリカッブルマップ部分

4. 応用事例

4.1 製造業会社案内

埼玉県富士見市に工場を設立し、汎用モーターや発電機の修理と調整を行う。業務内容が専門的でありテキストによる説明だけではイメージを伝えることが難しいために動画を中心にしたコンテンツを展開した。動画はウェブサーバーのディスクスペース節約のため YouTube にアップしてリンクした。

<図 10>会社案内ホームページのデスクトップとモバイル

ここではYouTubeにリンクする際にはviewportをcontent="width=600"と指定（図 3）し、動画サイズを width="560"と設定することでモバイル用では画面に対して約 93%の幅で動画が表示される。またスマートフォンを横に傾けた場合には、傾けた画面の幅に対して約 93%の幅で動画が表示される。デスクトップ用では画面中央（nav と article 部分）に対して約 93%の幅で動画が表示される。

```
<iframe width="560" height="315"
  src="https://www.youtube.com/embed/6AypVp8T7as?rel=0"
  frameborder="0" allowfullscreen></iframe>
```

4.2 スポーツ用品ネットショップ

さいたま市にある野球ユニフォーム専門のネットショップ。一昨年前からモバイルのアクセスがデスクトップを上回り、モバイルからの注文も増えてきたため、モバイルファーストを意識したホームページの製作依頼があったため、今回のレスポンシブウェブデザインを採用した。

ネットショップであるため、モバイルでの視認性の高くする CSS ファイルやブラウザとサーバーとの間で情報をやりとりする CGI ファイルを用意する必要があり、レスポンシブウェブ化するためには基本構成が極めてシンプルにする必要があった。

<図 12>モバイル用ページの注文フォーム

このネットショップでは 2015 年 4 月現在のアクセス数はモバイル 55%に対しデスクトップ 45%程度。最終的に注文したページは不明（レスポンシブであるため同じファイルから送信されるため）である。ただし消費者からのメールによる質問や過去の経験から、デスクトップからの注文のほうが依然として多いことも想像される。

従ってモバイルファーストであってもデスクトップ用のページも軽視はできない。

<図 13>デスクトップ用ページの注文フォーム

本提案のデザインではデスクトップ用画面の中央部分が、モバイル用と基本的に同等であるため、どちらか一方の画面を見て、もう一方の画面を容易に想像できる。

これは消費者側がモバイル用ページを見ながら（または見た記憶で）色々と質問してきた場合に、ネットショップの担当者がデスクトップ用の画面を見ながら、その内容に対して即答できるというメリットがある。

5. 今後の課題

5.1 モバイルの割合の動向

スマートフォンの国内での出荷台数は 2014 年から微減傾向にある。

しかし 2015 年 5 月にアメリカと日本を含む世界 10 ヶ国で、“Google 検索はデスクトップよりもモバイルで行われるようになった”と Google が発表した。

今後デスクトップ対モバイルの割合はモバイルが優勢になっていくことは予想されるが、スマートフォンの出荷台数が頭打ちである以上は、今のままの割合で当面は推移する可能性もある。

その動向次第ではデスクトップとモバイルのホームページの役割がそれぞれ独立したものとなり、レスポンシブウェブという形態のホームページの必要性がなくなる可能性もある。

また反対にリーズナブルでコストパフォーマンスの高いレスポンシブウェブが要求が増える可能性もある。いずれにしても今後の動向をウォッチングしていく必要がある。

5.2 HTML の変化

HTML5 になり HTML4 で使われていたいくつかのタグが廃止になり、新たに推奨されるタグが追加された。またブラウザ毎のベンダープレフィックスタグや video や audio のファイル形式のブラウザ毎の対応などは頻繁に更新されている。

また HTML だけではなく jquery や各 OS やブラウザも頻繁にアップデートされている。従って一度完成したホームページはそれらのアップデートに全て対応していく必要があり、その情報に常に敏感でなければならない。

またアップデートへの対応が極力少なくて済むようなシンプルな構成のホームページのデザインを心がけるようにする必要もある。

本提案で動作確認した環境

Windows8.1　Internet Explorer11.0
Windows8.1　Google Chrome 42.0
Windows8.1　FireFox 36.0
Windows8.1　Opera 29.0
MacOS 10.10.3　Google Chrome 42.0
MacOS 10.10.3　Safari 8.0.6
MacOS 10.10.3　FireFox 37.0.2
MacOS 10.10.3　Opera 27.0
iOS 8.3　Safari 8.0
Andoroid 4.3　webkit

参考文献

定平 誠・斎藤 忍［共著］（電子書籍）
尚美学園大学 定平研究室［制作協力］（2014）
「初めての HTML5&CSS3 魅せる！Web デザイン」
オーム社［編集協力］
株式会社ファステップス［制作］

参考サイト

2014 年第 4 四半期および 2014 年 国内携帯電話・スマートフォン市場実績値を発表
http://www.idcjapan.co.jp/Press/Current/20150312Apr.html

Google、ついにモバイル検索数がデスクトップを超えたと発表
http://seo-scene.com/mobile-query-overtakes-desktop