

# iPhone 用 Web サイトの作成について

0848066 中川真衣

## 目次

1.はじめに	1
(1) 動機	1
(2) iPhone とは	2
(3) 対象者について	2
(4) 類似物について	3
2.制作物について	3
(1) 最適化とは	3
(2) iPhone 向け Web サイト作成	6
(3) 基本的な HTML について	10
(4) 基本的な CSS について	12
(5) iPhone 風ナビゲーションページ	14
(6) アクセスの振り分け	15
(7) 検証について	16
(8) iPad について	18
3.最後に	22

はじめに

### (1) 動機

iPhone について調べそれについての論文を作成しようと考えたのは、まず私が所属する福田ゼミの基本テーマが「人の役に立つものをつくる」ということに始まる。大谷大学に入学した当初、私自身に明確な学習目的はなく、人文情報学科という学科を選択したのも、理系は苦手だが Web 関連の勉強がしてみたいと考えていたからである。高校時からパソコンや Web 関連に興味はあったものの、これといった専門知識などは全く持っておらず、ましてやそれらを使い人の役に立つものをつくるということは考えたこともなかった。

そのため 2 回生に上がり、後期のゼミ選択時には、自分が本当に学んでいきたい、やりたいことはなんだろうかと深く考えることになった。そして、それぞれのゼミの先生にお話を聴くうちに、福田先生のこのテーマが印象深く自分の中に残った。やはり何かをするのなら、自分の自己満足ではなく、人のためになることを身につけていきたいと強く感じたためである。

そして、福田ゼミで学んだことを活かせる、人の役に立てるテーマとは何だろうかと考えた結果、「iPhone 用 Web サイトについて」というテーマを思いつき、これについて論文を作成することにした。iPhone は現在携帯電話に代わる次世代モバイルデバイスとして、着々と使用率を上げており、身近なものとなっている。そしてその iPhone を用いて、出先で Web サイトなどを利用することは多々あり、そんな世間一般的なものとなってきた iPhone 用の Web サイトについてどのような仕組みで成り立っているのか、PC で閲覧する時と中身はどう違っているのか等疑問があったため、今回のテーマを設定した。

## (2) iPhone とは

スマートフォン自体は世界各国の通信会社から 2000 年頃より登場していたが、iPhone は 2008 年 7 月 11 日に Apple 社が発売した「iPhone 3G」が普及のきっかけとなり、登場後日本においてシェアが大きく伸びたモバイルデバイスである。一般的な携帯電話とは違い、電話やメール機能以外の機能も数多く備えている。また現在では iOS を搭載した iPhone だけでなく、Android OS を搭載した AQUOS PHONE や GALAXY など国産スマートフォンもシェアを大きく伸ばしてきている。

iPhone だけでなく、どのスマートフォンにも共通していえる最も大きな特徴は、Web サイトをデスクトップ用のパソコンブラウザのように閲覧ができるという点である。従来はパソコンでしかできなかった作業も外出先で行うこともでき、出先で簡単に調べ物をおこなったり、サイトを閲覧することもできるため、ユーザーの多くはこれらの利点を使用目的としている。

## (3) 対象者について

本論文は、人文情報学科でこれから Web サイトを作成し公開しようとしている人、またこれまでにサイトを作成したことがあり、今後 Web サイトをスマートフォン用に最適化させていこうと考えている人に向けて作成している。また、HTML や CSS についての基本的な知識があるということなども対象の条件としている。そのため、デスクトップ用の Web ページについての説明は、iPhone 用のタグやソースとの比較時以外は基本省くこととする。

#### (4) 類似物について

これまでのゼミ生の制作物の中で、Web サイトを作成するための HTML や CSS についてまとめた教材や論文は存在したが、iPhone 用の Web サイトを作成することについてまとめた論文は、確認した中では制作物内にはない。これは近年になってからスマートフォンが世の中に定着し始めたためだと思われる。

しかし、ネットや書店などで調べてみると、iPhone についての書籍やサイトは既に数多く存在する。今回は iPhone について初心者の人にもわかるよう基本的なことを説明すると同時に、それら既存のものとの差別化を図るため、これらのサイト・書籍を比較し、最も理解しやすいものになるように作成を図った。

## 2. 制作物について

### (1) 最適化とは

Web サイトを閲覧するため iPhone に搭載されている Web ブラウザは、Apple 社が開発したデスクトップ用ブラウザ Safari のモバイル版 Mobile Safari というものである。これはデスクトップ用に作成されたほとんどの Web サイトを閲覧する機能があるが、iPhone とデスクトップ用パソコンでは、デバイスと画面の大きさ、そしてモバイル通信網による回線スピードが異なるため、デスクトップ向けに作成されたサイトは快適に閲覧することができない場合が出てくる。iPhone 用に最適化したサイトを公開しているメジャーな例としては、Yahoo!Japan、Google などの検索エンジン、ポータルサイトの livedoor、ショッピングサイトの Amazon や楽天、ソーシャルネットワーキングサービスの mixi などのサイトが挙げられる。例えばもし、これらのサイトが

iPhone 用に最適化されずそのままの状態で開催されていた場合、閲覧時に必要としている情報が見づらく表示され、ユーザーはサイトから離脱してしまう可能性が発生する。すると、本来そのサイトで発生するはずだった成約や売り上げの減少を招きかねない。それらの事態を防ぐためにも、事前に iPhone 用に Web サイトを最適化することには、大変意義がある。これは個人用サイトにも共通していえることである。

iPhone は Webkit という HTML レンダリングエンジンを使用しており、これは最新の Web 技術への対応が早いため、新しい HTML5 や CSS3 に対応している。そのため特別な理由がない限り、iPhone 向けの Web ページを作成する場合は HTML5 を使うのが一般的と言える。CSS も同様である。最新の技術を用いて Web サイトを作成することができるため、HTML の記述をより簡単に記述することも可能であり、サイトの作成者にとっても知識を身につける勉強になるという点でも利点はある。

そして上記した Mobile Safari にはいくつか制限事項がある。

- ・ファイルのアップロード/ダウンロードができない
- ・「マウスオーバー」の表現ができない
- ・Flash などのプラグインに対応していない
- ・フレーム表現ができない
- ・日本語フォントは 1 つのみ

iPhone にはパソコンのようにファイル管理の機能がない。例えば、Web ページ内の「写真アルバム」に保存をすること、PDF ファイルを Mobile Safari で開くこと、専用アプリに保存することは可能だが、そ

れ以外のほとんどのファイルはダウンロードを禁止されている。

「マウスオーバー」の表現ができないというのは、iPhone は画面をタッチすることにより操作する方法を取っているため、マウスのように「移動」と「押す」を分けることができず、2 つまとめて「タッチ」という動作になるためである。そのため、CSS の `:hover` や、JavaScript の `onmouseover` は使用できない。JavaScript を工夫すれば近い表現はできるものの、ボタンの上に指をのせると色が変わったりするような処理は難しいといえる。

そして iPhone にできないこととして最も有名であるのが、Flash に対応していないことである。アプリを使用すれば再生可能のものもあるが、基本的に再生できないのが一般的である。しかし現在では韓国の SAMSUNG が開発し、日本では NTT ドコモが販売している GALAXY S などが Flash 対応しており、今後スマートフォンとしては対応機器が増えるかもしれない。

`iframe` 要素を使ったデスクトップ向けサイトは少なくない。しかし Mobile Safari ではこのフレームを仕様通りに表示できない。HTML5 の規格では `frame`、`frameset`、`noframes` 要素が廃止対象となっており、`iframe` 要素のみが利用可能となっているが、使い勝手が悪いものとなるため iPhone で使うことはやめた方がいいといえる。

日本語フォントは 1 つのみというのは、Mobile Safari の使用上の制限ではないものの、日本語フォントはヒラギノ角ゴシックしか付属しておらず、明朝体などは使えないということである。日本語フォントは変更できないが、英文フォントは充実しているので使い分けすることが可能だ。

以上が、iPhone に最適化したサイトを作るためにも忘れてはいけな

い制限事項である。

## (2) iPhone 向け Web サイト作成

ここでは iPhone 向け Web サイト作成の基本的な方法と最低限の知識をまとめる。iPhone はパソコンと違い、画面サイズが小さく、モバイル回線を使用しているためテキストや画像のダウンロードにも時間がかかることなどを念頭に置きサイト設計を考える必要がある。

実際に私もこの論文内の条件を踏まえサンプルサイトを作成している。なお作成したサンプルサイトについては卒業制作をみてほしい。

### (i) ページサイズは横 320px×縦 480px

デスクトップでは、横幅を 740～960px 前後に想定して設計するのが一般的であるが、iPhone では横 320px×縦 480px または横 480px×縦 320px と設定する。通常は縦表示時の横幅 320px が基本である。ここで大事なのがファーストビューであり、アクセス時最初に表示されるページには最も伝えたい情報を盛り込まなければならない。Mobile Safari には上下に URL の表示領域などがあるため、一度に表示される領域は 356px となる。少しでも表示領域を増やすには、HTML の最後に script 要素を記述、そこから JavaScript を読み込むことにより、URL の表示領域が自動で隠れるようにすることができる。

### 【HTML】

```
<script type="text/javascript" src="js/iphone.js"></script>  
</body>  
</html>
```



### 【js.ファイル】

```
window.addEventListener('load',  
  function(){  
    setTimeout(function(){  
      scrollTo(0,1);  
    },100);  
  }, false);
```

以上の処理は必須というわけではないが、より良いサイト作成のためには記述した方がいいといえる。

#### (ii) 画像サイズも横幅 320px 以内

サイトの横幅が 320px なので、画像ももちろんこの幅に納める。また Mobile Safari は読みこめるファイルサイズに制限があるため注意が必要である。画像はそのファイルの「重さ（ファイルサイズ）」だけでなく「大きさ（ピクセル数）」が制限の対象になっていることに注意が必要である。

ファイルの種類	制限
GIF,PNG,TIFF	3 メガピクセルまで
JPEG	32 メガピクセルまで
HTML,CSS,JavaScript など	10 メガバイトまで

上記の他にも、

- ・文字サイズは 14px~16px、行間は 1.4~1.6
- ・ページの左右には 10px の余白を作る
- ・ナビゲーションリンクの横幅は 44px 前後

など、実際の画面で見た際に見やすい設定にしておく。

次にセマンティックス関連の要素についてである。セマンティックなマークアップを行うと、作成者が文書の構造を理解しやすくなるだけでなく、コンピュータにとっても理解しやすいデータとなるため、サーチエンジンなどでより正確にデータを取り出せるようになる。

例えば、`section` 要素は、見出しと本文で構成される 1 つのまとまりをマークアップするために使うものであり、見出しがないただの文章のまとまりに対しては利用するべきではない。そういう場合は `div` 要素を使用する。

```
<section>
```

```
  <h1>セクションの見出し</h1>
```

```
  <p>本文</p>
```

```
</section>
```

`section` 要素以外の代表的なものは、`article`、`nav`、`header`、`footer`、`figure`、`small` などがあり、HTML の各要素をわかりやすくするためにセマンティックス要素を使用することを忘れないようにしたい。

iPhone に最適化したデザインにするためには、`Viewport` の設定も忘れてはいけない。`Viewport` とは、表示領域を指定するためのプロパティで、パソコン用ブラウザではウィンドウのサイズによって表示領域が決まるものだが、iPhone は特に指定しない限り表示領域の横幅を

980px 分を画面に収まるように画面サイズに合わせ縮小している。ピンチイン/ピンチアウトを使えば閲覧は可能なものの、より快適にするためにも Viewport の設定が必要である。

Viewport は meta 要素を使用し、下記のように定義する。

```
<meta name="viewport" content="値" />
```

値 = 内容	意味	デフォルトの値
width	Viewport の横幅	980px
height	Viewport の縦幅	幅に応じて自動
initial-scale	ページ読み込み時の拡大率	1
user-scalable	ユーザーによる拡大操作の許可	yes
minimum-scale	拡大率の最小	0.25
maximum-scale	拡大率の最大	1.6

Viewport を指定せずに Mobile Safari のデフォルト値をそのまま使用した場合は、次のように指定したのと同じ状態になる。

```
<meta name="viewport" content="width=980px,initial-scale=1.0,
user-scalable=yes,minimum-scale=0.25,maximum-scale=1.6" />
```

ここでは、値を device-width と入力する。そうすることにより画面サイズが 320px 以外のデバイスでも最適化し表示することができるた

めである。

### (3) 基本的な HTML について

基本的なセマンティックス要素はデスクトップとほとんど変わらない。

- ・ サイトのロゴやページの見出しを `header` で囲む
- ・ 見出しがある節を `section` で囲む
- ・ 写真や挿絵などを `figure` で囲む
- ・ パンくずリストを `nav` で囲む
- ・ コピーライトなどを `footer` で囲む

これらを用いて、ページの各部位の意味を明確にしておくことを念頭において作業を進める。HTML ファイルの先頭で、`Viewport` や `CSS` の設定を行う。ここではその `head` 要素の定義について説明する。

① `<!DOCTYPE html>`

② `<html lang="ja">`

`<head>`

③ `<meta charset="utf-8"/>`

④ `<title>タイトル</title>`

⑤ `<meta name="viewport" content="width=device-width"/>`

⑥ `<meta name="format-detection" content="telephone=no"/>`

⑦ `<link rel="apple-touch-icon" href="アイコンのファイル名.png"/>`

⑧ `<link rel="stylesheet" href="css のファイル名.css" media="screen"/>`

</head>

#### ① DOCTYPE 宣言

どういう仕様の HTML ドキュメントであるかを宣言。単に「DOCTYPE html」と記述すると HTML5 ドキュメントとして処理される。

#### ② 言語の宣言

ドキュメントが何の言語で書かれているかの宣言。日本語であれば「lang="ja"」と記入。

#### ③ 文字コード

内容はサイト環境やドキュメントの文字コードに合わせるが、特に指定や制約がない場合 UTF-8 とするのが一般的。

#### ④ タイトル

デスクトップとは違い、表示できる文字数が少ないため注意が必要。

#### ⑤ Viewport の設定

上記で説明したとおり、値を「device-width」と設定。

#### ⑥ 電話番号認識の無効化

Mobile Safari には、- (ハイフン) で繋がれた文字や連続した数字の文字列を見つけると、電話番号などの連絡先と判定し自動的にリンクを生成する機能がある。しかし、ハイフンを用いた文字列も電話番号と判定されてしまう場合もあるため、ここではあえて「"telephone=no"」としている。

#### ⑦ ホーム画面用のアイコンの設定

Mobile Safari では、iPhone のホーム画面にブックマークアイコンを置くことができる。これを Web クリップと呼ぶ。アイコンに使用する

画像は、横 114px×縦 114px 以上の PNG 形式で作成する。アイコンの角丸と光沢は iPhone の方で自動的に付けてくれるため、必要ない。

またこの記述をせずにアイコンを指定する場合は、画像のファイル名を `apple-touch-icon.png` にしておき、サイトの一番上の階層に配置する。後は自動的にアイコンが生成される。

#### ⑧ CSS の読み込み

ファイル名やデバイスの種類を指定して CSS を読み込む。ここで注意するのが、`media` 属性である。`media` 属性で指定できる値には「`handheld`」というモバイルデバイス向けの設定値があるが、iPhone などのスマートフォンではパソコン用ブラウザと同じ「`screen`」を指定する。

#### (4) 基本的な CSS について

次に CSS の説明をする。デスクトップ向けページでは、Web ブラウザごとのスタイル設定の違いを吸収するために、最初にすべての要素のデフォルトスタイルをリセットするのが一般的である。iPhone の場合は Mobile Safari のみを考慮すればいいため、本来はリセットする必要はない。しかし、`margin` や `padding` などのレイアウトに関するスタイルは、リセットしておいた方がスタイリングしやすいため、CSS ファイルの先頭でリセットをしておく。

Mobile Safari には、文字サイズを自動的に調整する機能も付いている。それを制御するためのプロパティが「`-webkit-text-size-adjust-`」であり、デフォルト値は「`auto`」だが、CSS でサイズを調整する場合「`none`」を指定し、無効化する必要がある。

iPhone では HTML5 を使用しているため、HTML5 の一部の新要素

にも気を配らなければならない。HTML5 以前からある要素に対しては、それぞれブラウザ側でデフォルトの `display` プロパティの値がセットされている。例えば、`h1` 要素は「`display:block`」がデフォルトである。しかし新要素の場合、ブラウザによってはセットされていない場合があるため、こちらで設定しておく必要がある。header 要素のようなものでも、ブロックレベルとして設定されていない場合があるため、ブロックレベルに該当する要素は、「`display:block`」と指定することが望ましい。

画像、リスト、表などに関しても設定のリセットが必要である。`img` 要素はボーダーをなしに、縦方向の配置を下揃えに設定し、リストに使用する `li` 要素はリストスタイルをなしにする。`table` 要素はボーダー間隔を 0 にし、内容がないセルも表示する設定にする。

iPhone では CSS3 を使用するため、ベンダープレフィックスというものが必要となる。ベンダープレフィックスとは、正式に仕様が定められていないがブラウザが試験的に先行実装しているプロパティや、ブラウザが独自実装しているプロパティを表す。Mobile Safari のベンダープレフィックスは「`-webkit-`」であるが、現状一部のプロパティを除き、ベンダープレフィックスの記述が必須となっている。そのため、今後ベンダープレフィックスが正式に勧告された時のことも考え、ベンダープレフィックスなしの記述とありの記述を併用する。

```
-webkit-border-radius: 8px;           //ベンダープレフィックスあり  
border-radius:8px;                    //ベンダープレフィックスなし
```

#### (5) iPhone 風ナビゲーションページ

iPhone にはアプリの画面に似せたナビゲーションが多く存在している。それらは一体どうやって表現しているのか、ここで説明していく。

iPhone のホーム画面のように、グリッド状にアイコンを並べたナビゲーションはユーザーにとって感覚的にわかりやすいナビゲーションである。例としてはアイコンの角丸と影を CSS でつけ、縦横 60px の正方形アイコンを並べるといったようなことである。これはナビゲーション全体を nav 要素でマークアップし、個々のアイコンになる部分は ul、li 要素を使って記述する。また alt 属性には詳しい内容を記述する。HTML5 では img 要素の alt 属性に関する以下の詳細な設定が設けられている。

イメージの内容	alt 属性に記述すべき内容
図表などイメージがコンテンツにとって重要である場合	alt 属性を設け、前後の文脈に自然な形で、対象イメージの詳細な表現を記述
アイコン画像のように装飾や補足のためのイメージである場合	alt 属性を設けるが、中身を空にする

またリスト型ナビゲーションも iPhone アプリではよく見られる。単なるテキストリンクよりもタップしやすく、また項目を縦に追加していただくので作成者にも後々使いやすいナビゲーションである。

そしてヘッダー型のナビゲーションもその 1 つであり、タイトル表示とナビゲーションの機能を兼ね備えている。これを設置するとページ内でユーザーが迷いにくい、移動のしやすいサイトになる。しかしこれ



はヘッダーに表示領域を確保しなければならず、あまり文字数の多いものは収まらないため注意が必要だ。

iPhone の基本的なページはこれら 3 つの要素でアプリのように見せかけることができる。

#### (6) アクセスの振り分け

iPhone に最適化したサイトを作成したら、iPhone ユーザーをサイトへ誘導しなければならない。ここでは主に 3 つの方法を紹介する。

##### (i) .htaccess による振り分け

まず「.htaccess」という Web サーバの動作を制御するファイルを使って iPhone 向けサイトに転送させる方法がある。利用している Web サーバで、mod\_rewrite という URL を書き換える機能が有効でなければ使えないため、まずこれを確認する。次にテキストエディタを用い、下記の内容を「.htaccess」というファイルに書き込む。

```
<IfModule mod_rewrite.c>
RewriteEngine on
RewriteCond %{HTTP_USER_AGENT} iPhone
RewriteRule ^$ http://www.example.com/iphone/ [R]
</IfModule>
```

##### (ii) JavaScript による振り分け

デスクトップ向けトップページ(index.html)にかき込むと、デスクトップ向けのサイトの URL にアクセスしたとき自動転送させることができるようになる。

```
<head>
<script>
if (navigator.userAgent.indexOf('iPhone') != -1) {
    location.href=http://www.example.com/iphone/;
}
</script>
</head.>
```

### (iii) PHP による振り分け

これは PHP を使用しているサイトの場合使用できる。以下の記述を PHP ファイルに書き込むのだが、どの PHP ファイルに書き込むかはサイトの構成によって異なるため、PHP に詳しい人などと相談して加えることを勧める。

```
<? Php
    if (strops($_SERVER['HTTP_USER_AGENT'], 'iPhone') !== false){
        header("location: サイトの URL");
    }
?>
```

### (7) 検証について

次に上記で作成した Web ページが正しく表示されることを確認しなければならない。ここではその検証方法について紹介する。

#### (i) Mac の場合

Mac で検証する場合、iPhone シミュレータ (iOS シミュレータ) を利用する方法と、OSX のパーソナル Web 共有を起動し、iPhone 実機で確認する方法がある。本テキストでは、前者の方法で検証する方法を紹介する。

iPhone シミュレータとは、iPhone アプリを開発するための iOS SDK に含まれている、動作検証用のソフトウェアであり、ダウンロードには、iPhone Developer Program への無料登録が必要である。登録後、iOS SDK のダウンロードを行い、インストールする。インストール後に、iPhone シミュレータを開くには、「Developer」→「Platforms」→「iPhoneSimulator.platform」→「Developer」→「Applications」と、深い階層のフォルダまでいかなければならない。これを覚えておくのもいいが、Spotlight を使用し、「iPhone Simulator」で検索にかけることをお奨めする。

iPhone シミュレータでは、マウスが指のかわりになる。ピンチイン/アウトを行うときは Option キーを押したままドラッグすることで、同じように動作するということも覚えておくといい。

Web ページを検証するときは、「Safari」を起動し目的の Web ページの URL を入力する。ハードディスク内に保存された Web ページを見る場合は、「http://」の代わりに「file://」をつけ、その後に「/Users」から始まるファイルパスを入力すると表示される。このシミュレータは i「ハードウェア」メニューの「デバイス」を切り替えれば、iPad も検証可能である。

#### (ii) Windows の場合

iPhone シミュレータは、Windows 版のものは提供されていないため、上記の方法は使用できず、パソコンだけで検証することはできない。し

かし Windows 版 Safari を使えば、だいたいの見た目を確認することはできる。

また iPhone の実機があれば、Web サーバを導入することで検証が可能になる。これにはフリーソフトの XAMPP for Windows を利用することを推奨する。これをインストールすると、スタートメニューなどに XAMPP Control Panel が登録され、そこから Apache を起動する。特に設定を変更せずインストールした場合、XAMPP は C ドライブ直下の「xampp」フォルダにインストールされ、そのフォルダ内の「htdocs」フォルダの中に確認したいファイルをコピーする。そして iPhone で「http://パソコンのローカル IP アドレス/」にアクセスすることで検証可能となる。

#### (8) iPad について

iPad は 2010 年 5 月より発売され、大谷大学でも新しい試みとして 2011 年度から iPad を導入した。ここでは iPad と iPhone について比較する。

- ・ 画面サイズの違い
- ・ 画像ファイルの制限の違い
- ・ Web クリップのサイズの違い
- ・ ビデオの再生方法の違い

これらが主な iPad と iPhone の違いである。

画面サイズは見てわかるとおり、iPad の方が大きく作られており、解像度が違う。また iPhone で Viewport がデフォルトで横幅 980px に

設定されているように、iPad もデフォルトでは横幅 980px にレンダリングされている。しかし実際表示すると、物理的ディスプレイの大きさの違いにより、iPhone では少々見づらいサイトも、iPad では 9.7 インチのディスプレイのためそれほど見づらくはない。

JPEG 画像は両方とも 32 メガピクセルまで対応しているが、GIF、PNG、TIFF 画像の場合、iPhone4 では 3 メガピクセルまで対応しているのに対し、iPad では 2 メガピクセルまでとなっている。

iPad でも Web サイトのブックマークを、Web クリップとしておくことができ、アイコンの指定方法も iPhone と同じだが、アイコンサイズだけが多少違う。

iPhone では、ビデオを `iframe` や `embed`、HTML5 の `video` 要素などでマークアップし Web ページに埋め込むことが可能だが、再生するときは専用のプレーヤーが起動して再生される。対する iPad は、Web ページに埋め込まれた状態のまま再生が可能であり、デスクトップ向けサイトと同じように再生することが可能である。しかし Flash 動画は原則的に再生できないことに注意が必要である。

以上のことを踏まえると iPad は最適化の必要がほとんどないように感じられる。しかしながら、やはり快適な Web 閲覧を可能にするために iPad にも最適化は必要であるといえる。以下に iPad への最適化の注意点を端的に述べておく。

- ・ iPad に合わせた Viewport を設定する
- ・ Flash の代替要素を用意する
- ・ 画像サイズの制限に気をつける
- ・ スクロールバーは表示されない

- ・ `position:fixed` に頼らない
- ・ マウスオーバーの表現ができない

iPad での Viewport の設定は 2 通りあり、1 つめの方法としては「Viewport の width に何も指定しない」という方法である。これはつまりデフォルトの Viewport サイズのままにするということで、サイトによってはデフォルトでも問題ないサイズになる。2 つめは「Viewport の width にサイトの横幅+ $\alpha$ を指定する」方法である。これはデフォルトの 980px 以外のサイズを指定することで、例えば width=1024px という Viewport に設定した場合、ページの横幅が 982px のサイトであった場合は左右に余白ができた状態で表示される。逆に 930px など小さいサイズに指定すると、少し拡大されたような形で表示される。

一時期より Flash は使用されることが減ったものの、現在でも使用するところは多く使用されている。Flash が使用されることにより起こる問題は、Flash を主要メニューに使用した場合 Flash 非対応の環境で閲覧すると他のページに移動できないということである。これを避けるため、Flash コンテンツを埋め込んだ箇所に代替の要素をいれておき、Flash 非対応環境では代替要素が表示されるようにする必要がある。

画像サイズは上記に記したとおり、サイズ制限に気をつければ表示されないなどの問題はクリアできる。

iPhone も iPad もスクロールバーという概念がない。これは普段操作しているときに問題にならないが、iframe 要素や overflow プロパティを使った要素の大きさを固定したスクロールを行う場合、問題となってくる。例えば、デスクトップサイトのトップページで「新着情報」や「News」などのコンテンツがミニボックス内でスクロールできるよう

になっている場合がある。このスクロールは、iPhone、iPad ともに実質的に使用不可能である。「実質的に」というのは、指二本でスワイプすることでこのミニボックスのスクロールは可能なものの、この操作方をすべてのユーザーが知っているわけではないからである。つまり機能としては問題無いが、快適な閲覧のためにこれに関しても場合によって最適化が必要といえる。

最後に、スクロールバーと似たような問題だが、`position:fixed` の問題がある。例えばスクロールさせているが一部の要素が移動せず、画面に表示され続ける Web ページが存在する場合がある。これは `position:fixed` という CSS プロパティと値が使われているためである。こういったサイトを iPad で閲覧すると、`position:fixed` を適用しても要素が固定されず、スクロールごと動いてしまう。またフッターなどに固定している場合、フッターがコンテンツに被ってしまうことなどがある。これを回避するための方法として、iPad 閲覧時には次の CSS が適用されるようにする。

#### 【iPad 用 CSS】

```
.footer {  
    position: fixed;  
    width:100%;  
    bottom: 0;  
}
```

ここに記述される `position:static` により、iPad での閲覧時のみ `.footer` が普通にスクロールすることになる。

マウスオーバーに関しては iPhone の時に触れたとおり、iPad も同じ

く使用できない。これに代替するものとしては、JavaScript のタッチ系イベントである。

- `ontouchstart` (`onmousedown` に相当)
- `ontouchmove` (`onmousemove` に相当)
- `ontouchend` (`onmouseup` に相当)

これらを使用することにより、表現を再現することが可能である。しかし必ずしも表現を再現する必要はないので、ケースバイケースでイベントや表現は変えた方がいいだろう。

### 3. 最後に

これまでサイトを作成する際、iPhone や iPad などについて考えて作成したことはなかった。HTML や CSS などの勉強はこれまで学科の授業などでも行っていたため、一通りの基本的知識はあった。しかし、モバイルデバイスに関しては今まで興味を持ったことがほとんどなく、iPhone にしても iPad にしても自分から勉強したことはほぼなかった。昨年大谷大学で iPad2 が配布されたものの、私たちは 4 回生だったため授業カリキュラムはなく、宝の持ち腐れとなっていた生徒も多かったと思われる。私自身 iPhone・iPad についての知識は今回卒業論文のテーマにするまで全く持ち合わせておらず、また最適化についても暗中模索状態であった。

今後こういったモバイルデバイスは普及率を伸ばすだろうと考えられる。現在は携帯とスマートフォンの二台持ちという人もいつかはスマートフォン一台のみになるかもしれない。また今携帯だという人も、昔ポ



ケベルが携帯に取って代わったように、いつかすべてスマートフォンになる時代がくるかもしれない。私たちサイト作成者は、そんなもしもに備え、ユーザーが常によりよい環境で快適に **Web** を利用できることを念頭に置くべきである。

私がこの論文内で紹介した最適化は本当に基本的な一部分であり、他のやり方ももちろん多種多様に存在している。今この瞬間も世界中でネット利用者達が新たな技術を生み出しているかもしれない。今後も **HTML** や **CSS** の規格が改定されるにつれまた変わってくることも多く出てくると考えられる。しかし冒頭で記したとおり、これから **Web** サイトを作成し公開しようとしている人、またこれまでにサイトを作成したことがあり、今後 **Web** サイトをスマートフォン用に最適化させていこうと考えている人たちには、現在のことだけでなく今後どのように **Web** は展開していくかを考え情報を収集しながらサイトを作成して欲しいと考える。そして、そのきっかけに少しでもこの論文がなれば幸いである。

## 文献表

谷拓樹 著

『HTML5+CSS3 で作る魅せる iPhone サイト』

『Yahoo!デベロッパーネットワーク』

<http://techblog.yahoo.co.jp/html5/html5/>

『HTML5 についてのおさらい』

<http://w3g.jp/blog/studies/html5report>

『iPad Files』

<http://arigato-ipod.com/ipad-iphone-ipod-touch.html>

『to-R』

<http://blog.webcreativepark.net/2010/08/13-205903.html>

『iphone wiki』

<http://www.iphoone.com/pukiwiki/index.php?FrontPage>

『iPhone 対応 Web サイト構築入門』

<http://sourceforge.jp/magazine/08/10/30/1051230>

『iPhone iPad 向けサイトを制作するためのいろいろ』

<http://synapse28.com/iphone-ipad> 向けサイトを制作するためのいろいろ /