Chapter 1

スマートフォンやタブレットに対応したサイト構築の考え方

1 BASIC スマートフォン&タブレットに対応した Web サイト制作の歴史

1-1 iPhone の登場

初期のスマートフォンは 1990 年代に登場し、PDA（Personal digital assistant）と電話の機能を足し合わせたようなデバイスが中心でした。しかし、2007 年に Apple 社から iPhone がリリースされたことによって状況が変わります。iPhone はスタイラスペンやキーボードなどの代わりに大きなタッチスクリーンを持ち、フルブラウザを搭載した強力なデバイスであったことから、確たるシェアを確立します。

1-2 iPhone に最適化した Web ページ制作の誕生

iPhone の普及により、デスクトップ用に作られた Web ページは iPhone では読みづらいという問題が出てくるようになります。そこで、iPhone の画面解像度 320 × 480 ピクセルに合わせて専用のページを作り、iPhone に最適化したデザインで閲覧してもらえるようにしようという動きが生まれます。

iPhone は初代のリリース以降、 1 年おきに iPhone 3G、iPhone 3GS、iPhone 4、iPhone 4S と上位機種がリリースされていますが、iPhone に最適化したデザインの Web ページであれば、iPhone で閲覧する限り問題が出ることはありません。iPhone に最適化した Web ページ制作のテクニックが現在まで大きく変わらないのはこのためです。

iPhoneの画面解像度に合わせたレイアウトの手引き
（Layout and Metrics on iPhone and iPod touch）
https://developer.apple.com/library/safari/#documentation/AppleApplications/Reference/SafariWebContent/UsingtheViewport/UsingtheViewport.html

1-3 Android の登場

しかし、2008 年に Android を搭載したデバイスが登場し、iPhone の独壇場だったスマートフォンの勢力図が徐々に変化していきます。

Android は Linux をベースとしたモバイル用の OS で、Google 社を中心とした Open Handset Alliance（OHA）によってリリースされています。Apple 社が製造している iPhone と異なり、Android を搭載したデバイスは数多くの企業が開発・製造しているため、多様なスペックのデバイスがリリースされるようになります。

Open Handset Alliance（OHA）
http://www.openhandsetalliance.com/

1-4 タブレットの登場

モバイル用のデバイスは、スマートフォンのみではありません。スマートフォンよりも大きなデバイスはタブレットと呼ばれ、2010 年には Apple 社が iPad をリリースし、後を追うように Android を搭載したタブレットも登場します。さらに、タブレットに最適化した Android 3.0 が開発され、2011 年には Android ベースの OS を搭載した Amazon の Kindle Fire といったデバイスも現れるなど、タブレットデバイスの多様化が進んでいきます。

1-5 多様なデバイスに対応するための Web ページ制作

現在、こうしたデバイスの多様化により、スマートフォン用の Web ページ制作は iPhone といった特定のデバイスのみをターゲットにしておけばよい時代ではなくなっています。iPhone、iPad はもちろん、Android を搭載したスマートフォンやタブレットへも対応することが必要です。そのためには、「レスポンシブ Web デザイン」と呼ばれるテクニックを利用することが、現時点では最も有効な手法であると考えられています。

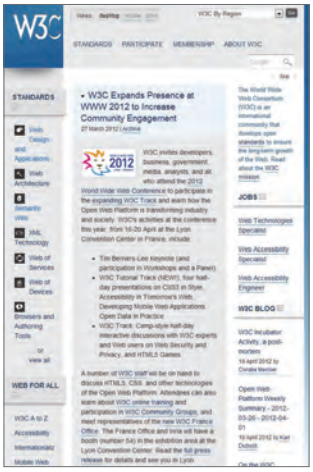
2 BASIC レスポンシブ Web デザイン

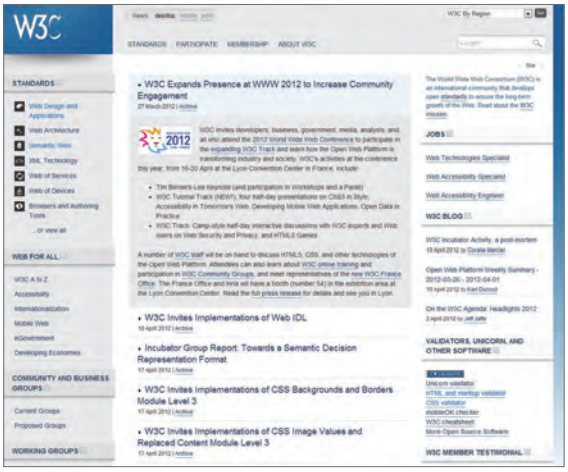
2-1 レスポンシブ Web デザイン

レスポンシブ Web デザインは、画面の横幅に合わせてデザインを変化させることにより、多様なデバイスに対応するという Web ページの制作手法です。

たとえば、HTML5 や CSS3 の標準規格を策定している W3C（World Wide Web Consortium）のサイトの場合、画面の横幅が大きいときは３段組みのレイアウトで表示されます。この３段組みはリキッドレイアウトになっており、画面の横幅を小さくしていくと、それに合わせて各段の横幅も変わっていくことがわかります。そして、最後には１段組みのレイアウトに変わり、横幅が小さな画面でもコンテンツの読みやすさが維持される仕組みとなっています。

この仕組みを利用すれば、新しいデバイスが出てきたり、想定外のデバイスで閲覧されたとしても、画面の横幅に合わせて Web ページが表示されますので、デバイスによって読みづらくなるといった問題を回避することができます。



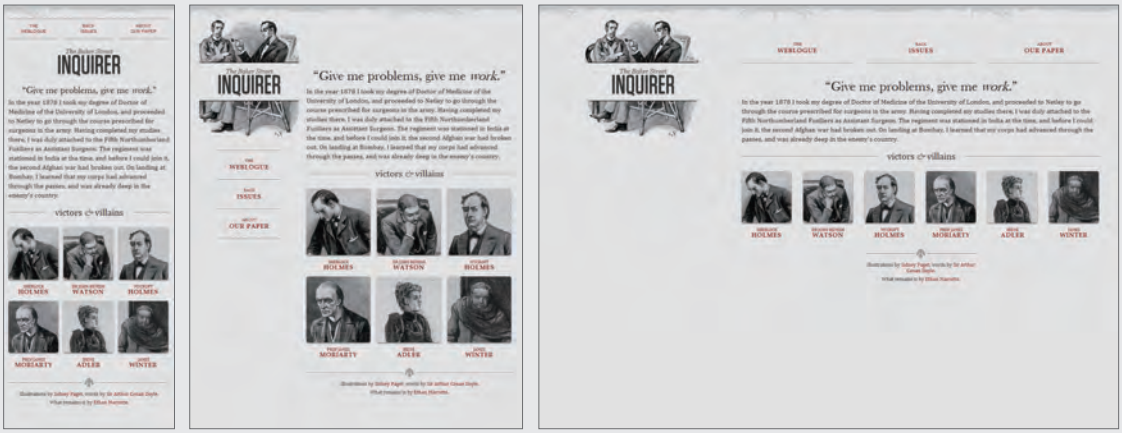


W3Cのトップページ（http://www.w3.org/）を画面の横幅を変えて表示したもの。
画面の横幅に合わせて段組みの数や各段の横幅が変わります。

NOTES レスポンシブ Web デザインの提唱

「レスポンシブWebデザイン」は、2010年にイーサン・マルコッテ氏が提唱した制作手法で、右の「A List Apart」の記事に基本的な考え方がまとめられています。また、この記事では画面の横幅に合わせて以下のように変化するサンプルが紹介されています。

"Responsive Web Design" by ETHAN MARCOTTE :
http://www.alistapart.com/articles/responsive-web-design/



「A List Apart」の記事で紹介されているサンプルを画面の横幅を変えて表示したもの。
画面の横幅に合わせて段組みの数や画像の配置・大きさなどが変化していることがわかります。

1

NOTES 単一ページでマルチデバイスに対応するメリット ～「One Web」と「One URL」

レスポンシブWebデザインは単一ページでマルチデバイスに対応するためのテクニックです。単一ページでマルチデバイスに対応することはW3Cにおいても推奨されており、「One Web」と呼ばれています。「One Web」は、Webが誕生した当初から提唱されている「Webページの情報はどのような閲覧環境からでも等しく取得可能であること」という基本理念に通じる考え方で、どのようなデバイスからページを閲覧した場合でも、同等の情報やサービスを得られるようにすることが求められています。

同様に、GoogleやBingといった検索エンジンでも、SEO（検索エンジン最適化）の観点から同一のコンテンツを単一のURLで管理する「One URL」という考え方を推奨しており、「レスポンシブWebデザイン」のテクニックに注目が集まっています。

Mobile Web Best Practices 1.0
http://www.w3.org/TR/mobile-bp/

「One Web」の考え方が提唱されたW3Cの勧告。

2-2 画面の横幅

では、画面の横幅とは何でしょうか？ ディスプレイサイズから導き出される横幅（通常ディスプレイサイズは対角線の長さを示しているため）や、画面解像度の横ピクセル数が考えられます。このうち、Web ページを制作するうえで扱いやすいのが、画面解像度の横ピクセル数です。

ところが、主要なデバイスのスペックを並べてみると、一筋縄ではいかないことが見えてきます。というのも、デバイスが進化するにつれて高解像度化が進み、同じディスプレイサイズでも解像度が異なるものが登場していたり、４インチクラスでありながら一世代前の 10 インチクラス並みの解像度を持つものが登場してきているためです。そのため、ディスプレイサイズと、解像度の両方を考慮した値が必要となります。それが、density に基づいたビューポートの横幅です。

	デバイス	ディスプレイサイズ（インチ）	デバイスの画面解像度（ピクセル）
2インチクラス	● PocketWiFi S	2.8	240×320
3インチクラス	● iPhone 初代	3.5	320×480
	● iPhone 3G	3.5	320×480
	● iPhone 3GS	3.5	320×480
	● iPhone 4	3.5	640×960
	● iPhone 4S	3.5	640×960
	● Xperia Ray	3.3	480×854
	● Motorola Defy	3.7	480×854
4インチクラス	● Xperia arc	4.2	480×854
	● Xperia acro HD	4.3	720×1280
	● Motorola PHOTON	4.3	540×960
	● HTC J	4.3	540×960
	● Galaxy Nexus	4.65	720×1280
5インチクラス	● Galaxy Note	5.3	800×1280
7インチクラス	● Galaxy Tab	7.0	600×1024
	● Kindle Fire	7.0	600×1024
8インチクラス	● Optimus Pad	8.9	768×1280
9インチクラス	● iPad 第1世代	9.7	768×1024
	● iPad 第2世代	9.7	768×1024
	● iPad 第3世代	9.7	1536×2048
10インチクラス	● Motorola Xoom	10.1	800×1280
	● Galaxy Tab 10.1	10.1	800×1280

● … iOS搭載デバイス
● … Android搭載デバイス

※ディスプレイサイズは画面の対角線の長さをインチで示したものです。
※製造元・販売元のカタログスペックの値を参照しています。
　値が明示されていない場合は GSMArena（http://www.gsmarena.com/）の情報を参照しています。

BASIC

3 density に基づいたビューポート

3-1 density とは

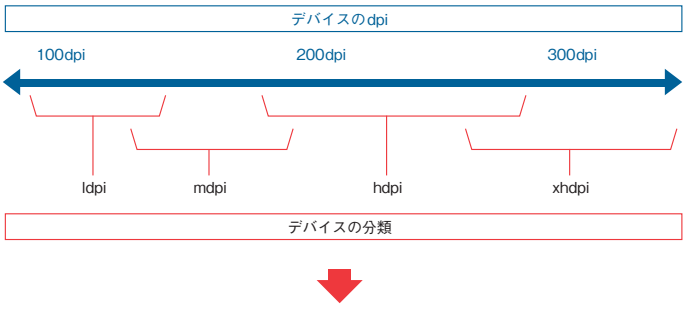
densityは、ディスプレイサイズや画面解像度が異なるデバイスにおいて、同じコンテンツが同じ大きさで見えるようにするために考えられたものです。

各種デバイスのdensityに基づいたビューポートに
160×160ピクセルの円を並べたページを表示したもの。

各種デバイスの画面解像度。

各デバイスでは dpi（または ppi）に応じて density の値が設定されています。dpi（dots per inch）または ppi（pixels per inch）はデバイスのディスプレイサイズと画面解像度から算出されるもので、その値に応じて ldpi ～ xhdpi の４つに分類され、density が設定されます。そして、density が「1」のデバイスを基準とし、コンテンツが同じ大きさで見えるように処理が行われる仕組みとなっています。

ldpi ～ xhdpi という分類は Android で使用されているものですが、iPhone や iPad でも同様の処理が行われ、デバイスごとに density が設定されます。たとえば、初代の iPhone は 163dpi なため density は「1」、iPhone 4S は 326dpi なため density は「2」となります。なお、この値は device pixel ratio とも呼ばれます。



dpiによるデバイスの分類	基準となるdpi	density (device pixel ratio)
ldpi (low)	120	0.75
mdpi (medium)	160	1
hdpi (high)	240	1.5
xhdpi (extra high)	320	2

3-2 ビューポートとは

ブラウザがデフォルトで設定するビューポート

ビューポートは、デバイスのスクリーンを何ピクセル×何ピクセルとして使うかを設定して作成されるものとなっています。iOS や Android に標準で搭載されたブラウザは、ビューポートの横幅をデフォルトで 800 ピクセルまたは 980 ピクセルに設定します。たとえば、iOS と Android 3.x/4.x では 980 ピクセル、Android 2.x では 800 ピクセルとなります。そのため、小さなディスプレイでは Web ページの文字が小さくなりすぎて読みづらいといった問題が発生します。

なお、デスクトップ環境ではブラウザ画面がビューポートとして扱われます。

NOTES

モバイル用のOperaやFirefox、Chromeといったブラウザは、iOSやAndroidのバージョンに関係なく、ビューポートの横幅をデフォルトで980ピクセルに設定します。

NOTES

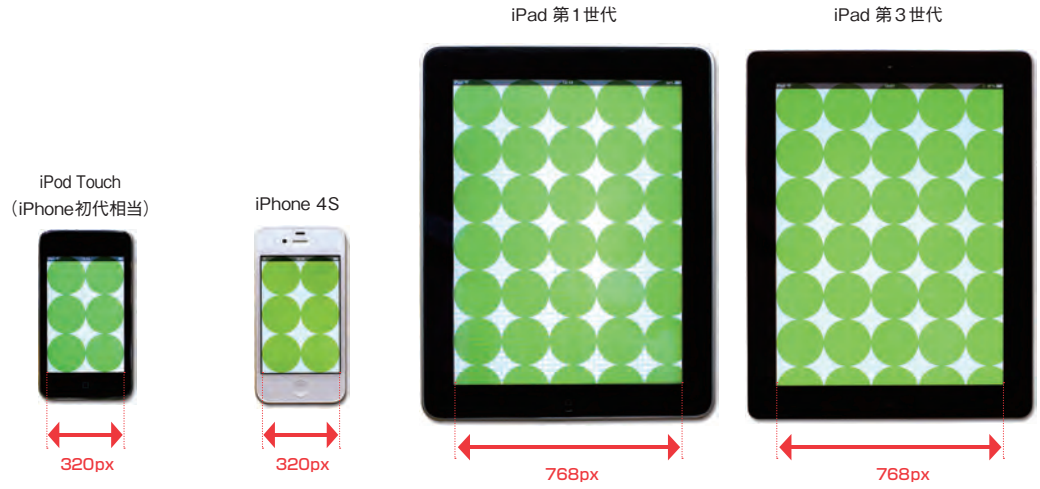
Windows Phone 7.5の標準ブラウザは、ビューポートの横幅をデフォルトで1024ピクセルに設定します。

デバイスの画面解像度に設定したビューポート

ブラウザが作成するビューポートの横幅は <meta name="viewport"> の指定で変更することができます。たとえば、デバイスの画面解像度に設定すると以下のような表示となります。この場合、デバイスのディスプレイサイズが同じでも、画面解像度によってコンテンツの表示範囲が大きく変わってくるという問題が発生します。

density に基づいたビューポート

ビューポートをデバイスの画面解像度と density に基づいた横幅に設定すると、異なるデバイスの間でもコンテンツが同じ大きさで見えるようにすることができます。レスポンシブ Web デザインでは、この横幅を利用します。



3-3 density に基づいたビューポートを利用する理由

density に基づいたビューポートを利用すると、異なるデバイスの間でもコンテンツが同じ大きさに見えるように表示されます。これは、density に基づいたビューポートの解像度が以下のように定義されるためです。

$$\text{densityに基づいたビューポートの解像度} = \frac{\text{デバイスの画面解像度}}{\text{デバイスに設定されたdensityの値}}$$

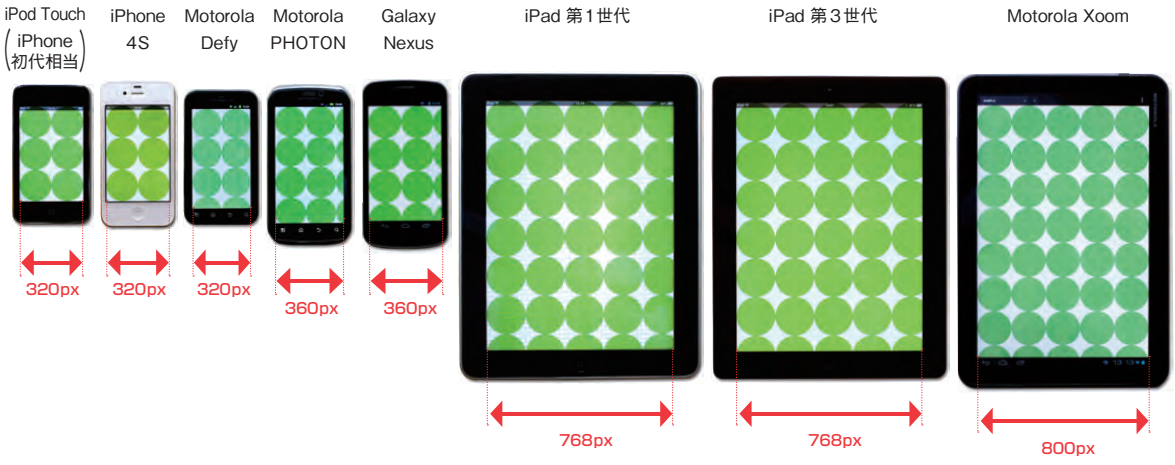
たとえば、iPhone 4S の場合、デバイスの画面解像度は 640 × 960 ピクセル、デバイスに設定された density の値は 2 となっていますので、density に基づいたビューポートの解像度は 320 × 480 ピクセルとなります。

同様にして導き出された主要なデバイスの density に基づいたビューポートの解像度は右ページのようになります。ディスプレイサイズとの関連性がなかったデバイスの画面解像度とは異なり、ディスプレイサイズに応じた値が並んでいます。

なお、デスクトップ環境ではブラウザ画面がビューポートとして扱われますが、これまでは density の概念がなく、density の値は「1」として扱われてきました。しかし、Retina MacBook Pro では density が「2」となっており、今後はデスクトップ環境においても density を考慮したページ制作が必要になりそうです。

	デバイス	ディスプレイサイズ（インチ）	デバイスの画面解像度（ピクセル）	dpi	density	densityに基づいたビューポートの解像度（ピクセル）
2インチクラス	● PocketWiFi S	2.8	240×320	143	0.75	320×427
3インチクラス	● iPhone 初代	3.5	320×480	163	1	320×480
	● iPhone 3G	3.5	320×480	163	1	320×480
	● iPhone 3GS	3.5	320×480	163	1	320×480
	● iPhone 4	3.5	640×960	326	2	320×480
	● iPhone 4S	3.5	640×960	326	2	320×480
	● Xperia Ray	3.3	480×854	297	1.5	320×569
	● Motorola Defy	3.7	480×854	240	1.5	320×569
4インチクラス	● Xperia arc	4.2	480×854	233	1.5	320×569
	● Xperia acro HD	4.3	720×1280	342	2	360×640
	● Motorola PHOTON	4.3	540×960	256	1.5	360×640
	● HTC J	4.3	540×960	256	1.5	360×640
	● Galaxy Nexus	4.65	720×1280	316	2	360×640
5インチクラス	● Galaxy Note	5.3	800×1280	285	2	400×640
7インチクラス	● Galaxy Tab	7	600×1024	171	1	600×1024
	● Kindle Fire	7	600×1024	169	1	600×1024
8インチクラス	● Optimus Pad	8.9	768×1280	167	1	768×1280
9インチクラス	● iPad 第1世代	9.7	768×1024	132	1	768×1024
	● iPad 第2世代	9.7	768×1024	132	1	768×1024
	● iPad 第3世代	9.7	1536×2048	264	2	768×1024
10インチクラス	● Motorola Xoom	10.1	800×1280	149	1	800×1280
	● Galaxy Tab 10.1	10.1	800×1280	149	1	800×1280

● … iOS搭載デバイス
● … Android搭載デバイス



NOTES Nexus 7

Nexus 7は、GoogleのAndroid独自ブランドであるNexusシリーズ初のタブレット型端末です。最新のAndroid 4.1（Jelly Bean）に対応した端末として注目されていますが、Webデザインという観点から見ても非常に興味深い製品です。というのも、これまでAndroidの端末は4種類のdensityに分類されていましたが、Nexus 7はAndroid3.2で新たに設定されていたtvdpiというカテゴリーに属する製品となっているためです。

tvdpiは、Google TVやタブレットなどをターゲットとして設定されたもので、213dpiと設定されており、Nexus 7にもこの値が設定されています。tvdpiの213dpiという数字から、ブラウザ上で扱われるdensityの値を計算すると、

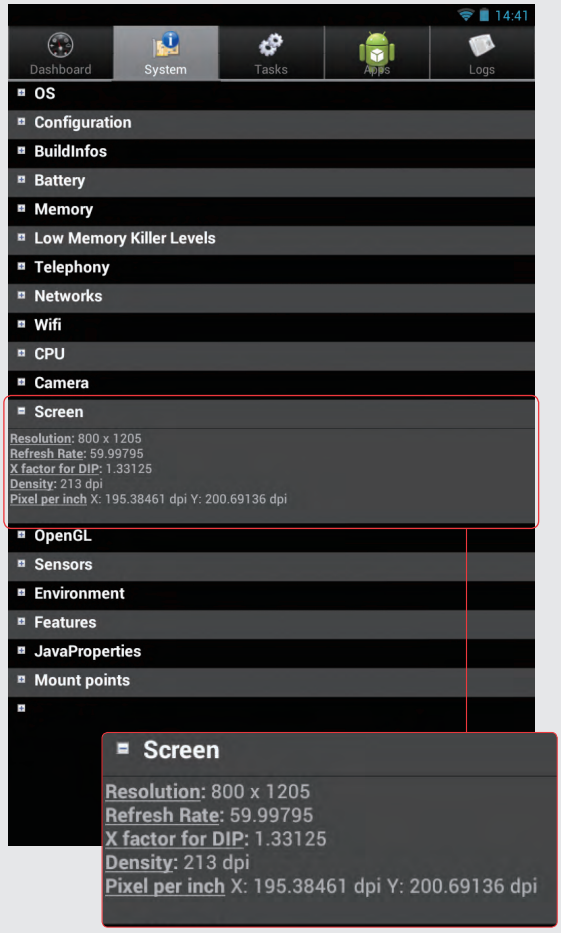
```
213 ÷ 160 = 1.33125
```

となり、mdpiとhdpiの間に位置することが確認できます。ただし、Androidのアプリケーションの開発などにおいてはsecondary densityといった位置づけとなっており、ユーザーインターフェースのために用意する画像データなどは、hdpi用に用意したデータを変換して利用するとされています。

dpiによるデバイスの分類	基準となるdpi	density (device pixel ratio)
ldpi (low)	120	0.75
mdpi (medium)	160	1
▶ tvdpi	213	1.33125
hdpi (high)	240	1.5
xhdpi (extra high)	320	2

ブラウザに関してはどういった扱いになっているのかを確認すると、 現時点では、Nexus 7の標準ブラウザであるChrome（バージョン18.0.1025123）はdensityの値として1.3250000476837158という数値を返してきます。また、Opera Mobileでは従来の分類に合わせてhdpiの1.5という数値を返してくるなど、アプリケーションレベルでの対応ができていないようで混乱しています。

これまで、tvdpiはその存在も含めてあまり意識されることはありませんでした。しかし、Nexus 7という、非常に注目度が高いデバイスとして登場してきたことで、今後はしっかりと意識していかなければならなさそうです。



Nexus 7におけるAndroid System InfoのSystem情報画面。

NOTES **Windows Phone**

現時点では、国内向けの端末がWindows Phone 7.5に対応した1機種（IS12T）しか存在していないこともあり、Webデザインの世界ではほとんど存在感の無いWindows Phoneですが、Windows 8に合わせて登場すると言われているWindows Phone 8では存在感が増してきそうです。そこで、Windows Phoneについて確認しておきます。

ブラウザ

OS標準のブラウザはもちろん、Internet Explorerです。Windows Phone 7.5ではIE9相当、Windows Phone 8では、IE10相当のブラウザとなっています。

解像度

Windows Phoneではハードウェア要件として、解像度は以下のものから選択することになっています。

	解像度
Windows Phone 7	800×480（15:9）
Windows Phone 8	800×480（15:9）
	1280×768（15:9）
	1280×720（16:9）

ただし、画面サイズに関する規定はなく、Windows Phone 7の場合、3.6インチや4.7インチの端末も存在しています。

また、画面サイズが自由ではあるものの、Androidにおけるdensityやi OSにおけるDevice Pixel Ratioに代わるものは、Windows Phone 7.5には見当たりません。そのため、現時点では、Windows Phone用の処理（P.116）が必要なわけですが、Windows Phone 8で何か変化があるのか、興味深いポイントでしょう。

Windows Phone Emulatorでの表示。

BASIC

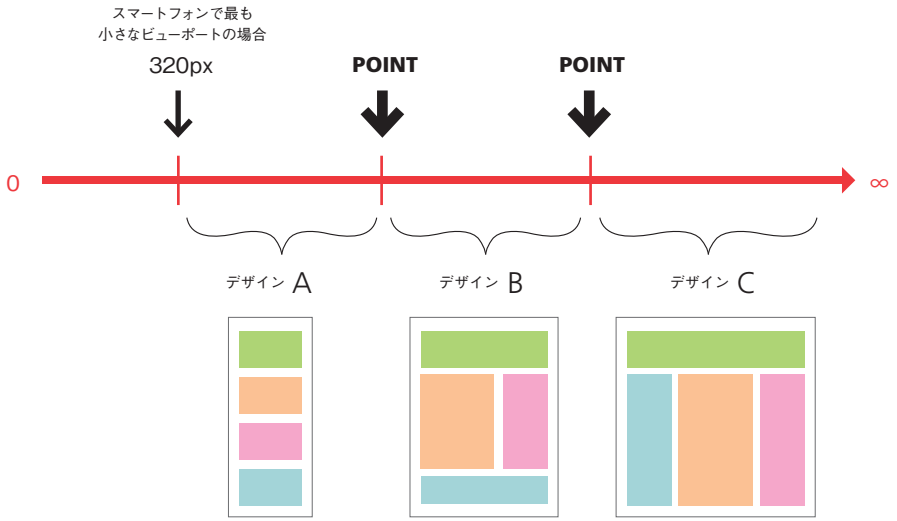
4 デザインを切り替えるための横幅の決定

レスポンシブ Web デザインの理想は、Web ページをあらゆる画面サイズにマッチする形で表示することです。しかし、画面サイズごとに個別にデザインを用意するのも、１つのデザインですべての画面サイズに対応するのも現実的には困難です。

そのため、画面サイズをいくつかの範囲に分け、範囲ごとに Web ページのデザインを切り替えることになります。そこで考えなければならないのが、切り替えるポイントとなる横幅です。また同時に、設定した横幅に対応できる柔軟なデザインを考えなければなりません。

切り替えのポイントとなる横幅については、表示するデバイスのディスプレイサイズを元に考えていくこともできますし、ページの構成要素を元に考えていくこともできます。

ページのデザインに関しては、リキッドレイアウトを元に可変ボックスと固定ボックスをいかに配置していくかがポイントとなります。詳しくは、Chapter 2 でサンプルを通して見ていきます。



Chapter 2

レスポンシブWebデザインの実践

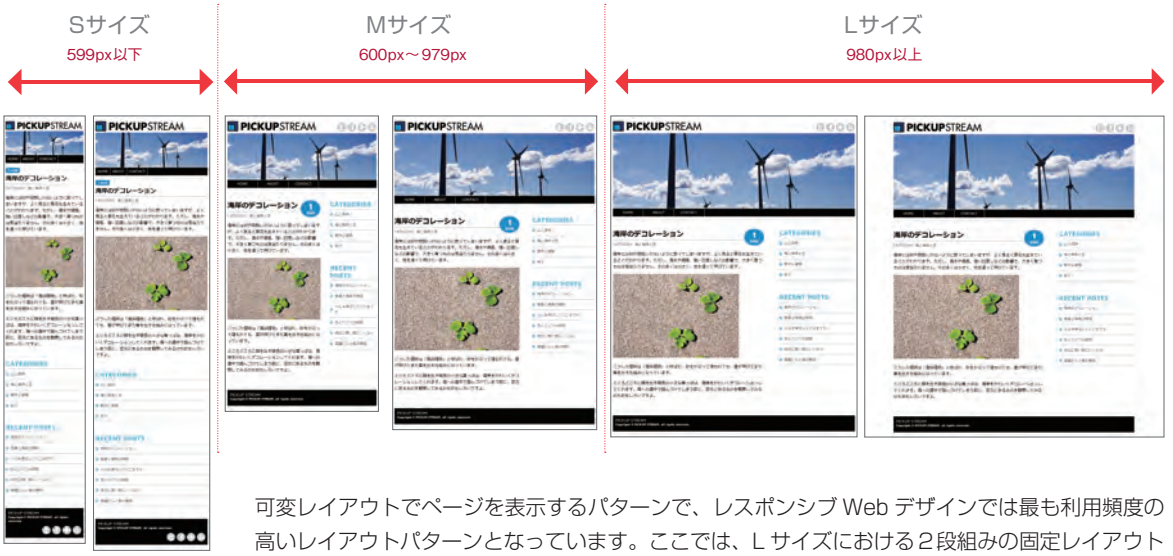
■レスポンシブWebデザインにおける主要レイアウトパターン
■レスポンシブWebデザインの基本設定
A 可変レイアウトによるレスポンシブWebデザイン
B 固定レイアウトによるレスポンシブWebデザイン
C 可変レイアウト＋固定レイアウトによるレスポンシブWebデザイン

SAMPLE

レスポンシブ Web デザインにおける主要レイアウトパターン

レスポンシブ Web デザインにおける主要なレイアウトパターンをサンプルを通して紹介していきます。ここでは、横幅が変化する「可変レイアウト」と、横幅を固定した「固定レイアウト」の組み合わせに応じて A ～ C の 3 パターンのサンプルを紹介します。いずれのサンプルにおいても画面サイズを S、M、L の 3 段階の範囲に分け、範囲ごとにページのデザインを切り替えるようにしています。ただし、L サイズの画面では常に「固定レイアウト」で表示を行い、デスクトップ環境の大きな画面で横幅が際限なく広がることがないようにしています。

A 可変レイアウトによるレスポンシブ Web デザイン



可変レイアウトでページを表示するパターンで、レスポンシブ Web デザインでは最も利用頻度の高いレイアウトパターンとなっています。ここでは、L サイズにおける 2 段組みの固定レイアウトをベースとし、M サイズでは可変レイアウトに変更した上で、ヘッダー、コンテンツ、サイドバー、フッターの各パーツの横幅が画面に合わせて変化するように設定します。同様に、S サイズの画面ではこれらを 1 段組みの可変レイアウトで表示します。また、ヘッダー画像や本文中の画像も画面に合わせて大きさが変わるようにしています。

B 固定レイアウトによるレスポンシブ Web デザイン

Sサイズ
639px以下

Mサイズ
640px～979px

Lサイズ
980px以上

各画面サイズで固定レイアウトを行うパターンです。ここでは画面サイズに応じて４段組み、２段組み、１段組みとレイアウトを変え、左右に挿入する余白によって多様な画面サイズに対応するように設定します。

C 可変レイアウト＋固定レイアウトによるレスポンシブ Web デザイン

Sサイズ
649px以下

Mサイズ
650px～979px

Lサイズ
980px以上

基本的にサンプル A と同じ可変レイアウトですが、固定レイアウトを混在させたパターンで、広告のように横幅を変えることができないパーツがある場合などに有効です。ここではサイドバー上の広告の大きさを変えずに表示するため、サイドバーの横幅を固定したレイアウトにしています。

SAMPLE

レスポンシブ Web デザインの基本設定

A ～ C の各サンプルで利用するレスポンシブ Web デザインの基本的な設定を行います。

1 ビューポートの横幅の指定

レスポンシブ Web デザインを行う上で重要なのは、デバイスの画面サイズとして扱われるビューポートの横幅の設定です。何の設定も行わないと、P.020 のようにデフォルトの横幅（800 または 980 ピクセル）に設定されてしまいます。ここでは、P.022 のようにビューポートの横幅を各デバイスの density に基づいた解像度に設定するため、以下のように <meta name="viewport"> の設定を追加し、content 属性の値を「width=device-width」と指定します。

なお、各サンプルは HTML5 で記述し、sample.html というファイル名で保存して作業していきます。

ビューポートの横幅を設定しなかったときのサンプルBの表示。

ビューポートの横幅をデバイスのdensityに基づいた解像度に設定したときのサンプルBの表示。

sample.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>SAMPLE</title>

<meta name="viewport" content="width=device-width">
</head>
<body>
 …
</body>
</html>
```

HTML5のDOCTYPE宣言を記述。

エンコードの種類をUTF-8に指定。

ページのタイトルを指定。

ビューポートの横幅を指定。

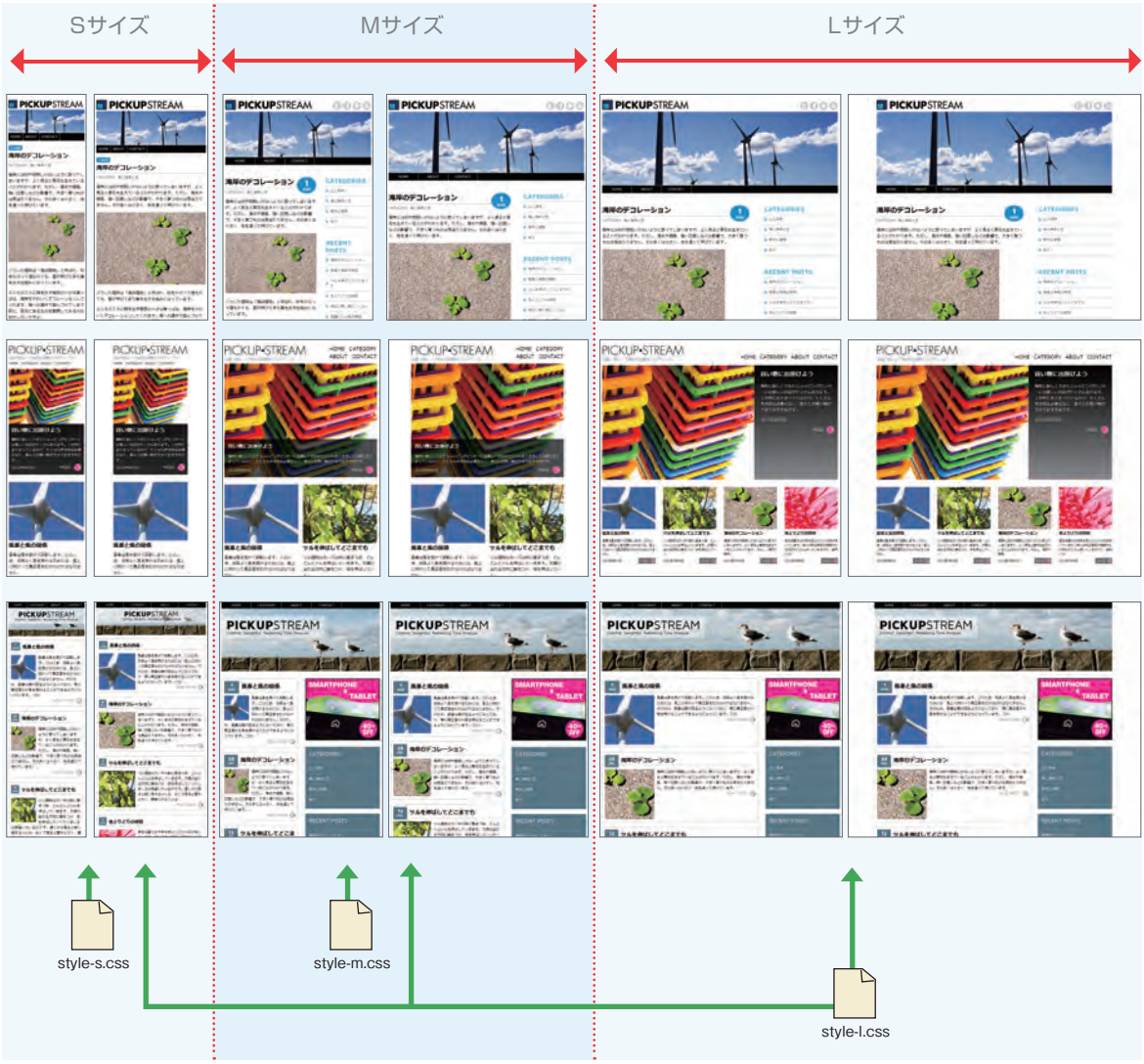
NOTES

<meta name="viewport" content="～">で指定できるパラメータについて詳しくはP.136を参照してください。

2 メディアクエリとスタイルシートの設定

次に、レスポンシブ Web デザインでは画面（ビューポート）の横幅をいくつかの範囲に分け、範囲ごとにデザインを切り替えます。デザインの切り替えには JavaScript などを利用する方法もありますが、ここでは最も手軽に利用することができる CSS3 のメディアクエリの機能を利用します。メディアクエリの機能を利用すると、画面の横幅に応じて適用するスタイルシートを指定することができます。

そこで、サンプル A ～ C では、S、M、L の各画面サイズで表示したときに適用する外部スタイルシートファイル style-s.css、style-m.css、style-l.css を用意します。ただし、ページのデザインは L サイズの画面をベースに設定していきますので、実際には、S サイズの画面では style-s.css と style-l.css を、M サイズの画面では style-m.css と style-l.css を、L サイズの画面では style-l.css を適用する形になります。



デザインのベースとなるスタイルシート

各画面のデザインのベースとなるスタイルシートは右のように <link> で指定します。sample.html に <link> を追加すると以下のようになります。ここではＬサイズ用の設定（style-l.css）を適用するようにしています。

```
<link rel="stylesheet" href="〜.css">
```

sample.html

```
<title>SAMPLE</title>

<meta name="viewport" content="width=device-width">

<link rel="stylesheet" href="style-l.css">

</head>
…略…
```

デザインのベースとなるスタイルシート（style-l.css）を指定。

特定の画面サイズで表示したときに適用するスタイルシート

2

特定の画面サイズで表示したときにのみ適用するスタイルシートを指定する場合、右のように <link> の media 属性で適用条件を指定します。

「メディアタイプ」ではスタイルシートを適用する閲覧環境の種類を指定します。スマートフォンやタブレット、デスクトップ PC のように、モニタスクリーンで表示を行う環境を適用対象とする場合、メディアタイプは「screen」と指定します。

さらに、「特性」では画面（ビューポート）の横幅やデバイスの向きといった条件を指定します。たとえば、「min-width:980px」と指定すると、画面の横幅が 980 ピクセル以上の環境を適用先とすることができます。なお、and でつなげることにより、複数の特性を指定することも可能です。

```
<link rel="stylesheet" href="〜.css"
media="only メディアタイプ and (特性)">
```

主なメディアタイプ	閲覧環境の種類
all	すべて
screen	モニタスクリーンで表示を行う環境
print	印刷環境

主な特性	閲覧環境の特性	値
width	画面（ビューポート）の横幅	数値
min-width	画面（ビューポート）の横幅の最小値	数値
max-width	画面（ビューポート）の横幅の最大値	数値
orientation	デバイスの向き（横／縦）	landscape portrait

NOTES

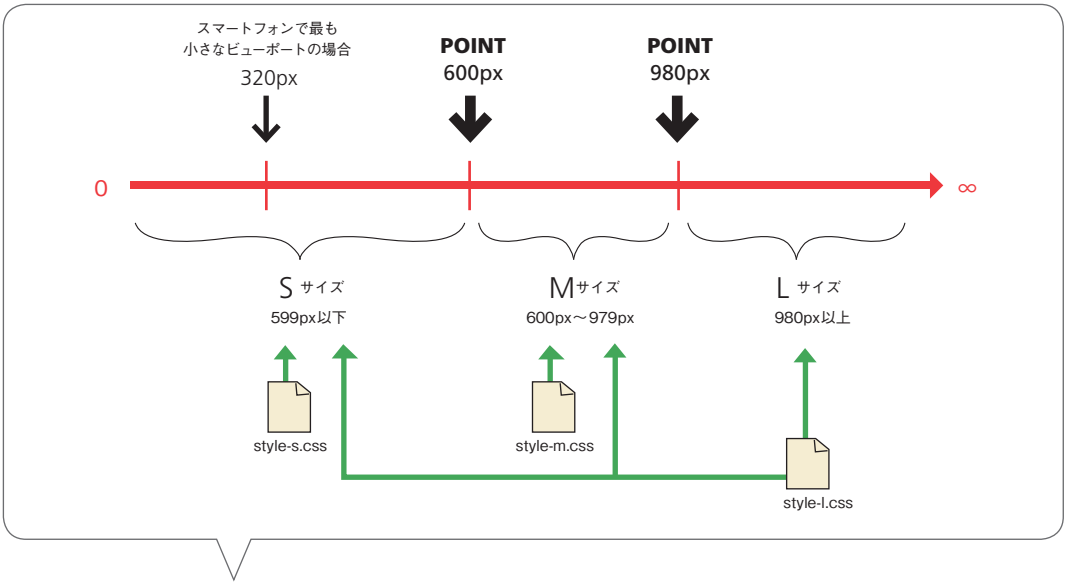
media属性による適用条件の指定が「メディアクエリ」と呼ばれる機能です。メディアタイプの指定はCSS2.1からサポートされていましたが、特性の指定はCSS3からサポートされた機能となっています。指定できる値などについて詳しくはP.097〜098を参照してください。

NOTES

デスクトップ環境では、ブラウザ画面の横幅がビューポートの横幅として扱われます。

たとえば、画面の横幅が 600 ピクセル以上～ 979 ピクセル以下のときに style-m.css を、599 ピクセル以下のときに style-s.css を適用したい場合、以下のように <link> の指定を追加します。なお、指定する横幅の範囲はサンプルごとに調節します。

以上で、レスポンシブ Web デザインの基本的な設定は完了です。次のステップからはサンプルごとの設定を行っていきます。



sample.html

```
<title>SAMPLE</title>

<meta name="viewport" content="width=device-width">

<link rel="stylesheet" href="style-l.css">

<link rel="stylesheet" href="style-m.css"
media="only screen
and (min-width:600px) and (max-width:979px)">

<link rel="stylesheet" href="style-s.css"
media="only screen and (max-width:599px)">

</head>
…略…
```

画面の横幅が600ピクセル以上～979ピクセル以下の場合に適用するスタイルシート（style-m.css）を指定。

画面の横幅が599ピクセル以下の場合に適用するスタイルシート（style-s.css）を指定。

NOTES

サンプルではスタイルシートの設定を個別のファイルにわけて管理していますが、1 ファイルで管理することもできます。その場合、@mediaを利用してメディアクエリの指定をCSSファイル内に記述します。@mediaについて詳しくはP.096を参照してください。