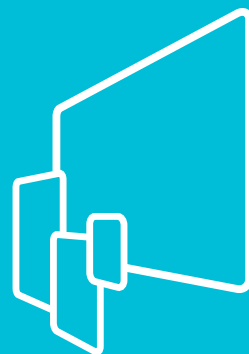


レスポンス Webデザイン

制作の実践的ワークフローとテクニック

渡辺 竜



Responsive Web Design

Practical Development Workflow and Techniques

Ryo Watanabe

➡ サンプルファイルについて

本書で解説しているサンプルのデータおよび第 6 章で作成するサンプルサイトのデータは、本書の連動ウェブサイト (<http://rwd-book.info>) よりダウンロードいただけます。

・ソースコードについて

本書のサンプルファイルに使用されている HTML、CSS、JavaScript などのソースコードは、オープンソースソフトウェアまたはそれらのコードを改変して作成されたものです。ご使用の際は、必ずオリジナルのライセンス・著作権をご確認いただいたうえ、ご自身の責任と判断によってご利用ください。

・イメージファイルについて

本書のサンプルファイルに含まれているイメージファイルは、クリエイティブ・コモンズでライセンスされたものを使用しています。使用される場合は、各イメージに記載されたクレジットのオリジナルのライセンスをご確認のうえご利用ください。

なお、サンプルファイルの使用により発生した、いかなる損害についても、著者およびソシム株式会社は一切の責任を負いかねますのでご了承ください。

本書の連動ウェブサイト
<http://rwd-book.info>

免責事項

● 本書記載の情報は、特別な明記がない場合 2012 年 11 月現在の情報を掲載しています。ご利用時には、変更されている場合もあります。また、紹介しているソフトウェアやフレームワークなどは、バージョンアップされる場合があり、本書の説明とは異なる場合があります。

● 本書の出版にあたっては、できる限り調査と検証をして正確な記述に努めましたが、環境の違いなどによって記述とは異なる動作となる可能性があります。本書はすべての正しい動作を保証するものではありません。また、本書の内容に基づく運用結果について、著者およびソシム株式会社は一切の責任を負いかねますのでご了承ください。

● 本書に記載されている会社名、商品名、製品名などは一般に関係各社の商標または登録商標です。本文中では、TM、®などのマークは省略しています。

はじめに

レスポンス Web デザインは、これからのウェブ制作を支える重要な制作手法の一つです。基本理念を理解して活用すれば、ユーザにも制作者にも多くのメリットをもたらす可能性があります。

本書の目的は、レスポンス Web デザインの理念、基本的な制作手法、必要とされるワークフローの考え方から制作の実践を通して、レスポンス Web デザインの可能性を感じてもらうことです。筆者がレスポンス Web デザインでサイトを制作し 1 年ほど運営してきた経験から、共有できることを整理し、基礎から具体的なノウハウまでを丁寧に解説しました。

レスポンス Web デザインには、まだ解決すべき課題が残っています。また、これが唯一あるべき制作の方法だとは思いません。しかし、この手法が持つ可能性、また、それがもたらす制作に対する考え方の変化が、これからのウェブ制作に多大な影響を及ぼすことは間違いありません。ウェブ関連の技術だけでなく、手法自体が着実に進歩しています。多くの課題は遅かれ早かれ解決されるでしょう。

モバイル・インターネット環境が整備されスマホやタブレットなどのスマートデバイスが急速に普及したことで、その中身を構成するウェブが進化を求められています。いまウェブは変化の時代であり、その変化ゆえに、これからのウェブはいつそう面白くなると感じています。そして、ウェブを楽しむ要因と言えるのが、レスポンス Web デザインのような、「Future Friendly」で柔軟な考え方を取り入れた制作手法であり、「モバイル」という可能性だと感じています。

レスポンス Web デザインについては賛否両論あるようですが、まずはそれが持つ可能性を読み解いてから判断しても良いのではないのでしょうか？ 本書が、その際のお役に立てば幸いです。

本書は以下の 3 部で構成されています。

第 1 部 基本編では昨今のウェブにまつわる潮流をおさらいして、なぜ、いまレスポンス Web デザインのような手法が必要とされているのか、そして、レスポンス Web デザインの真の意味や考え方をみていきます。

第 2 部 準備編では、制作を進めるにあたり知っておくべき考え方やワークフローを学んで実践に備えます。

第 3 部 実践編では、より具体的な制作手法を学び、架空のプロジェクトをベースにプロトタイプを構築して、レスポンス Web デザインでの制作を体験します。

では、さっそくはじめましょう。

2012 年 12 月
渡辺 竜（わたなべ りょう）

CONTENTS

第 1 部 基本編

第 1 章

なぜ、いまレスポンス Web デザインなのか？ …11

1.1. レスポンス Web デザインとは ……12

1.1.1. 1 分でわかるレスポンス Web デザインの歴史…12

1.1.2. レスポンス Web デザインを支える 3 つの手法…12

1.1.3. レスポンス Web デザインの例…13 / 1.1.4. 大学やブランド企業でのレスポンス Web デザイン…14

1.2. モバイルの波 ……16

1.2.1. インターネットの閲覧がモバイルメインになる日も近い…16

1.2.2. フィーチャーフォンとスマートフォンの大きな違い…16

1.2.3. スマホで見るデスクトップ向けウェブサイト…17 / 1.2.4. スマホ（モバイル）対応の必要性…18

1.3. 未知の端末の可能性と「未来にやさしい」ウェブ ……19

1.3.1. ウェブを閲覧するのに使うのは PC だけではない…19 / 1.3.2. 未知の端末にも対応する必要性…20

1.4. コンテンツを起点に考える「コンテンツ・ファースト」 ……22

1.4.1. ユーザはウェブサイトを使いたいように使う…22

1.4.2. コンテンツが「データ」としてさまざまな形で消費される時代…23

1.4.3. コンテンツを起点に考えるウェブ制作とコンテンツ戦略…24

1.5. モバイルを起点に考える「モバイル・ファースト」 ……26

1.5.1. ユーザのネット環境の変化…26 / 1.5.2. モバイル・ファーストとは？…27

1.5.3. モバイル・ファーストのメリット…29

1.6. コンテキストをより強く意識したコンテンツ制作 ……31

1.6.1. コンテンツ・ファーストとコンテキスト…31 / 1.6.2. 「コンテキスト」に、より密接なモバイル端末…32

1.6.3. いまモバイルから得られる「コンテキスト」情報…32 / 1.6.4. なぜ、いま「コンテキスト」なのか？…33

1.7. 複数の端末をシームレスに使い分ける時代 ……34

1.7.1. 同じウェブサイトを複数の端末で閲覧する時代…34 / 1.7.2. ビジネスが生き残るための必須条件…35

1.8. 複数サイト構築で増え続ける運営コスト ……37

1.8.1. 複数端末対応で増え続ける制作・運営コスト…37 / 1.8.2. ウェブサイトの多言語化…37

1.9. そもそもウェブは1つしかない「One Web」	39		
1.9.1. 「One Web」とは	39 / 1.9.2. 「One Web」とアクセシビリティ	40 / 1.9.3. 「One Web」の可能性	41
1.9.4. 「One Web」の理想と現実	42		

第2章

レスポンス Web デザインの基礎知識	43
---------------------	----

2.1.	「レスポンス Web デザイン」の真の意味	44	
2.1.1.	命名によって加速的に普及する技術	44 / 2.1.2. レスポンス Web デザインの根底となる考え方	44
2.1.3.	ウェブの哲学 - 「A Dao of Web Design」	45 / 2.1.4. 周囲に呼応する建築「Responsive Architecture」	45
2.1.5.	レスポンス Web デザインの魅力	46	
2.2.	フレキシブル・グリッド	48	
2.2.1.	グリッド・レイアウトとは	48 / 2.2.2. フレキシブル・グリッドとは	49
2.2.3.	グリッド・システムとは	49 / 2.2.4. 可変グリッドシステム	51
2.3.	フレキシブル・イメージ & メディア	53	
2.3.1.	フレキシブル・イメージ & メディアとは	53 / 2.3.2. フレキシブル・イメージの課題	54
2.3.3.	いまある解決策	57	
2.4.	メディアクエリ	59	
2.4.1.	メディアクエリとは	59 / 2.4.2. メディアクエリで指定できる主なメディア特性	60
2.4.3.	ブラウザの対応状況と非対応ブラウザ向けの対策	61	
2.5.	レスポンス Web デザインのいま	62	
2.5.1.	Google がモバイル対応にレスポンス Web デザインを推奨	62	
2.5.2.	その他のレスポンス Web デザイン関連の動き	63 / 2.5.3. レスポンス Web デザイン用のツールの充実	66
2.6.	ビジネス視点から見たレスポンス Web デザイン	68	
2.6.1.	どんなプロジェクトにレスポンス Web デザインは向いているのか	68	
2.6.2.	ビジネスゴールとレスポンス Web デザイン	69 / 2.6.3. 機会損失を最小限に抑えるという考え方	70
2.6.4.	SEO の視点から見たレスポンス Web デザイン	71	
2.6.5.	広告キャンペーンから考えるレスポンス Web デザイン	71 / 2.6.6. 制作費と運営費の比較	72

第2部 準備編

第3章

レスポンスな制作に必要なこと73

3.1. レスポンス Web デザインに必要なワークフロー74

- 3.1.1. いままでとは違ったワークフローが必要な理由...74 / 3.1.2. 現代のウェブに必要な「デザイン」...75
- 3.1.3. レスポンス Web デザインでの制作の違い...75 / 3.1.4. 基本的なワークフロー型の種類...76
- 3.1.5. アジャイル型がレスポンス Web デザインに向いている?...77

3.2. レスポンス Web デザインに必要な制作体制78

- 3.2.1. レスポンス Web デザインに望ましいチーム構成...78
- 3.2.2. レスポンス Web デザインで制作する際の注意点...79

3.3. アジャイルな制作方法81

- 3.3.1. 「アジャイル (agile)」とは?...81 / 3.3.2. アジャイル型がレスポンス Web デザインに向いている理由...81
- 3.3.3. アジャイル開発の6つの原則とレスポンス Web デザイン...82
- 3.3.4. プロジェクトの意思決定者を巻き込んだ制作...84

3.4. 自分の領域を超えた制作ができるチーム86

- 3.4.1. 垂直統合型のスキルセットが必要な理由...86 / 3.4.2. デザイナーが CSS で可能なことを学ぶ...87
- 3.4.3. コーダーがデザイン・コミュニケーションを学ぶ...88
- 3.4.4. ライター・編集者がレスポンス Web デザインを学ぶ...88

3.5. HTML プロトタイプでの検証90

- 3.5.1. HTML プロトタイプを活用するメリット・デメリット...90 / 3.5.2. プロトタイプを取り入れたワークフロー...92
- 3.5.3. プロトタイプで検証する内容...92 / 3.5.4. プロトタイプの検証のポイント...93

3.6. ウェブの柔軟性を許容する制作95

- 3.6.1. 完璧を求めない、目的に反するもの以外は許容する制作...95 / 3.6.2. ブラウザ間の表示の違いを許容する...95
- 3.6.3. ブラウザの違いを顧客に理解してもらう...96 / 3.6.4. ウェブの柔軟性を最大限に活かせるように工夫する...97
- 3.6.5. スタイルシートはページのデザインを提案するもの...98

3.7. 目的重視の意思決定と現場チームへの権限移譲99

- 3.7.1. 柔軟かつスピーディな意思決定が不可欠な理由...99 / 3.7.2. 先送りを恐れない柔軟な社内体制...100
- 3.7.3. 優先順位を決める際の指針...101

3.8. コンテンツのモジュール化とパターンライブラリ	102
3.8.1. コンテンツのモジュール化とパターンライブラリ...	102
3.8.2. モジュールを組み合わせた「レイアウト・テンプレート」...	103 / 3.8.3. パターンライブラリの例...104
3.9. サイトのパフォーマンス最適化	107
3.9.1. レスポンシブ Web デザインを支える「第 4 の手法」...	107 / 3.9.2. どこから始めれば良い?...108
3.9.3. もっとも効果がある最適化は?...109	

第 4 章

レスポンシブなワークフロー.....111

4.1. レスポンシブなワークフローの概要	112
4.1.1. レスポンシブなワークフローが必要な理由...	112 / 4.1.2. ワークフローや手法を確立する意味...112
4.1.3. レスポンシブなワークフローの 1 つの可能性...	113 / 4.1.4. ワークフローをうまく使うコツ...114
4.1.5. 制作チームに必要なスキルセット...	115 / 4.1.6. 各チームメンバーの役割...115
4.2. ブラウザ・端末対応の仕様を決める	117
4.2.1. ブラウザ対応を決める際の注意点...	117 / 4.2.2. ブラウザ対応を顧客にどう説明するべきか...118
4.2.3. ブラウザ検証の優先順位...	119 / 4.2.4. 古いバージョンの IE 対応...120 / 4.2.5. スマホのブラウザ対応...120
4.3. ブレイク・ポイントとレファレンス・ポイントの仕様を決める ...	122
4.3.1. ブレイク・ポイントとは...	122 / 4.3.2. 2 種類のブレイク・ポイント...123
4.3.3. モバイル・ファースト、デスクトップ・ファースト?...124	
4.4. サイト・フローチャートの作成	128
4.4.1. フローチャートを作る意味...	128 / 4.4.2. レスポンシブなフローチャートの注意点...128
4.4.3. フローチャートとナビゲーション...	129
4.5. ナビゲーション・パターンの洗い出し	131
4.5.1. ナビゲーション・パターンの例...	131
4.6. ナビゲーション・プロトタイプの実成	135
4.6.1. ナビゲーション・プロトタイプとは...	135 / 4.6.2. ナビゲーション・プロトタイプで検証すること...136
4.6.3. レスポンシブなナビゲーションの注意点と課題...	136
4.7. レスポンシブなコンテンツづくり	138
4.7.1. 最も重要なのはサイトの中身...	138 / 4.7.2. コンテンツ・ファースト...139
4.7.3. 「伝えたいこと」と「伝えてほしいこと」をマッチングする仕事...	139
4.7.4. レスポンシブなコンテンツ制作のために...	140 / 4.7.5. レスポンシブなコンテンツづくりのヒント...141

CONTENTS

4.8. パターンライブラリの作成	143
4.8.1. パターンライブラリとは…143 / 4.8.2. パターンライブラリが必要な理由…143	
4.8.3. パターンライブラリを作るメリット…144 / 4.8.4. 標準的なパターンを整理する…144	
4.8.5. サイト特有のパターン要素を一覧に追加する…145 / 4.8.6. パターンライブラリのプロトタイプ…147	
4.8.7. フレームワーク化について…147	
4.9. レイアウト・テンプレートとサイト全体のプロトタイプの作成 …	148
4.9.1. レイアウト・テンプレートとは？…148 / 4.9.2. レイアウト・テンプレートで検証すること…149	
4.9.3. レイアウト・テンプレート作成の際の注意点…150 / 4.9.4. サイト全体のプロトタイプの作成…150	
4.10. デザイン	152
4.10.1. デザインとは…152 / 4.10.2. デザインには意図や意思がある…153	
4.10.3. レスポンシブ Web デザインだと魅力的なデザインはできない？…154 / 4.10.4. デザインする際の注意点…155	
4.11. ブラウザチェック・実機チェック	158
4.11.1. テストの頻度とタイミング…158 / 4.11.2. 3つのテスト方法…158	
4.11.3. テストするブラウザ・実機の見分け方…160 / 4.11.4. ブラウザとモバイル端末の選択の例…162	
4.11.5. テストの効率化とプランニング…163 / 4.11.6. テストプロセスの例…164	
4.11.7. テスト環境の整備とツール集…165	
4.12. アクセス解析と結果検証	170
4.12.1. アクセス解析の考え方…170 / 4.12.2. レスポンシブ Web デザインでのアクセス解析…170	
4.12.3. Google アナリティクスで行うレスポンシブなアクセス解析…171	
4.12.4. レスポンシブ Web デザインでのコンバージョンの最適化…173	
4.12.5. どの程度の結果を妥当とするべきか…174 / 4.12.6. 思ったような成果が得られない原因…175	

第 3 部 実践編

第 5 章

レスポンシブ Web デザインの基礎の実践

5.1. メディアクエリを使ってみる	178
5.1.1. メディアクエリの仕様…178 / 5.1.2. メディアクエリの設置方法…181	
5.1.3. メディアクエリの指定方法の考え方…183 / 5.1.4. モバイル・ファーストなメディアクエリの指定方法…183	
5.1.5. デスクトップ・ファーストなメディアクエリの指定方法…184	
5.1.6. 画面サイズを限定するメディアクエリの指定方法…185 / 5.1.7. サポートのないブラウザへの対応…186	

5.2. Viewport メタタグの設定と詳細 188

5.2.1. Viewport メタタグを指定してスマホ表示に...188 / 5.2.2. デスクトップとスマホでの Viewport の定義...189

5.2.3. スマホの Viewport に関わる 2 つの領域...189 / 5.2.4. Viewport をコントロールする...190

5.2.5. Viewport メタタグの仕様...191 / 5.2.6. iOS 5.1 以前の Safari のバグ...194

5.2.7. 現状のベスト・プラクティス...195

5.3. リキッド・レイアウト..... 196

5.3.1. リキッドレイアウトとは?...196 / 5.3.2. 画像やビデオをリキッド対応にする...197

5.3.3. リキッドにレイアウトする際の注意点...199

5.4. グリッド・システムを構築する..... 207

5.4.1. グリッドで考えるレイアウト...207 / 5.4.2. レスポンシブなグリッド・システムの基本...208

5.4.3. 制作現場で使えるグリッド・システムの条件...209 / 5.4.4. 基本的なグリッド・システムを構築する...212

5.4.5. カラムの表示順を変える PUSH と PULL...220 / 5.4.6. モバイル表示の対応...223

5.5. サイトのパフォーマンス（表示速度）最適化の実践..... 226

5.5.1. 最適化の基本...226 / 5.5.2. いまから始める 7 つのパフォーマンス最適化...227

5.5.3. HTTP リクエストを減らす...230 / 5.5.4. CodeKit で CSS の管理を効率化...232

5.5.5. CSS を上に、JavaScript を下に設置する...235 / 5.5.6. 画像の最適化...236

5.5.7. JPEG/JPG の特徴と向いている画像のタイプ...237 / 5.5.8. PNG と GIF の特徴と向いている画像のタイプ...239

5.5.9. CSS/JavaScript の圧縮を行う...241 / 5.5.10. サーバサイドの設定はファイルの圧縮だけでも必ずやる...243

5.6. CSS スプライトの実装方法 244

5.6.1. CSS スプライトの基本...244 / 5.6.2. CSS スプライトのソースコードの全貌...245

5.6.3. CSS スプライトのソースコードの説明...248 / 5.6.4. アイコン表示に を選んだ理由...252

5.6.5. スプライト画像作成の際の注意点...253

5.7. IE6 と IE7 対応のヒント..... 256

5.7.1. IE6 と 7 の対応の選択肢...256 / 5.7.2. IE 対応しない場合の対処方法...257 / 5.7.3. IE 対応の準備...258

5.7.4. メディアクエリへの対応...259 / 5.7.5. HTML5 への対応...260 / 5.7.6. CSS3 への対応...261

第 6 章

レスポンシブ Web デザインでサイトを構築する... 263

6.1. サンプルサイトの定義..... 264

6.1.1. サンプルプロジェクトの概要...264 / 6.1.2. サンプルサイトの仕様...265

6.1.3. サンプルサイト作成のフロー...267 / 6.1.4. フローチャート...267

6.1.5. ワイヤフレームをスケッチする...268 / 6.1.6. 各ページのスケッチの紹介...268

CONTENTS

6.2. ベーステンプレートの準備271

6.2.1. ベーステンプレートに使うもの...271 / 6.2.2. HTML5 Boilerplate について...271

6.2.3. ベーステンプレートの準備...275 / 6.2.4. HTML5 Boilerplate のカスタマイズ...277

6.3. ナビゲーション・プロトタイプの実成284

6.3.1. ナビゲーション・プロトタイプのコーディング...284 / 6.3.2. ナビゲーションのスタイリング...287

6.3.3. ナビゲーション用 JavaScript の設置...288 / 6.3.4. ブラウザ・テストとレイアウトの調整...288

6.4. レイアウト・テンプレートのプロトタイプの実成292

6.4.1. プロトタイプ作成の目的...292 / 6.4.2. プロトタイプの作成...293

6.4.3. スタイルの追加とレイアウトの調整...294 / 6.4.4. タブレット表示の調整...295 / 6.4.5. スマホ表示の調整...297

6.4.6. プロトタイプの完成とブラウザ・テスト...299 / 6.4.7. その他のページのプロトタイプ完成イメージ...299

6.5. フレキシブルなスライドショーと画像301

6.5.1. スライドショー...301 / 6.5.2. スライドショーのファイルの準備...302 / 6.5.3. HTML の編集...303

6.5.4. responsiveslides.css の編集...306 / 6.5.5. その他のレスポンシブなスライドショー・プラグイン...309

6.5.6. レスポンシブ・イメージの実装...310 / 6.5.7. Picturefill の設置...311

索引...316

column

2012 年 4 ~ 9 月でスマホの出荷台数は全体の 69.4% に...16

コンテンツの重要性を訴える試み...25

どちらがファーストでもない、どっちもファースト...28

Luke Wroblewski 氏の「Mobile First」関連資料...30

日本のレスポンシブ・アーキテクチャ...47

未来の仕様「レスポンシブ・イメージ (仮)」...58

Google が推奨するレスポンシブ Web デザイン...62

デザイナーは「素材」の制限を知らないほうが良い?...87

ブラウザ上でデザインする「Designing in the Browser」とは...91

制作と運営の効率化で顧客との関係も良好に...106

ウェブ制作も「Back to the Future」...110

各工程の詳細について参考になる書籍——「ユーザー中心ウェブサイト戦略」...114

Internet Explorer 10...121

メディアクエリに指定する幅の単位— px それとも em?...127

カテゴリ分けとタイトル付けの上手なやり方...129

ナビゲーションの考え方が変わる?...137

コンテンツの重要性を理解するために...138

レスポンシブ Web サイトでも「PC サイト」ボタンが必要?...176

Apple の開発者向けウェブサイト「Safari HTML Reference」...195

レスポンシブなグリッド・システムの例...209

CodeKit と LESS 言語の参考サイト...235

画像最適化ツールやオンラインサービス...241

iPad のスケッチ・アプリ「Notesshelf」...270

ウェブサイトのナビゲーションが変わる?...291

第 1 部 基本編

第 1 章 なぜ、いまレスポンス Web デザインなのか？

2012 年 8 月の時点で国内のスマホ普及率は 23% 以上。5 月には携帯契約台数の過半数はスマホになっています。すでに日本でもユーザの「モバイル」環境は劇的に変わっています。企業や制作の現場でも、いまではタブレットやスマホなどを含めた「モバイル対応」を急ピッチで進めていることと思います。そんな中、「モバイル対応」の一つの手法として確立されつつあるのが、「レスポンス Web デザイン」です。

第 1 章では、レスポンス Web デザインやそれを取り巻くウェブの状況を足早におさらいしていきます。レスポンス Web デザインが必要になった理由やその考え方、それに付随するいくつかのウェブの潮流を理解して、具体的に実装する際の助けとなる「概念」を把握していただければと思います。

1.1. レスポンシブWebデザインとは

レスポンシブ Web デザインはユーザが閲覧する端末の画面サイズに合わせて、ページのデザインやレイアウトを自在に操るウェブ制作の手法です。この手法を使うとスマホやタブレットだけでなく、大画面テレビで閲覧するような場合でも、ユーザが使いやすいようにページを柔軟に順応させることが可能です。ここではレスポンシブ Web デザインの簡単な説明と、その手法を用いて作られたウェブサイトがどんなサイトなのかを紹介します。

1.1.1. 1分でわかるレスポンシブWebデザインの歴史

レスポンシブ Web デザインは、アメリカの Ethan Marcotte 氏が考案、命名したウェブ制作の手法です。「Web Standards」を広めた Jeffrey Zeldman 氏が創設したウェブ制作者向けの「A List Apart^{※1}」というウェブサイトで、2010年5月25日に「Responsive Web Design」という題名の記事でその詳細が紹介されました。この記事はアメリカのウェブ業界に衝撃を与え、その後、さまざまな議論がなされ賛否両論を乗り越え、今では次世代ウェブ制作の1つの手法として確立されています。いまアメリカでは、この手法を使って制作されたウェブサイトが続々と登場しています。日本でも2011年後半から、この手法を取り入れたウェブサイトが登場し始めています。

※1 A List Apart

常にウェブ制作の最先端を行くウェブ制作のベストプラクティスを紹介するアメリカのウェブサイト。

➡ <http://www.alistapart.com>

1.1.2. レスポンシブWebデザインを支える3つの手法

レスポンシブ Web デザインを支える3つの手法について簡単に紹介します（詳細は第2章で紹介します）。レスポンシブ Web デザインは、以下の3つを組み合わせることで成り立っています。

フレキシブル・グリッドをベースとしたレイアウト

レイアウト幅が可変する段組レイアウト。

フレキシブル・イメージ/メディア

フレキシブル・グリッド内でサイズが可変するイメージやビデオなどのメディア。

メディアクエリ

画面サイズを判別してスタイルを指定できる CSS3 の機能。

1.1.3. レスポンシブWebデザインの例

レスポンシブ Web デザインで作られたサイトでは、以下のようにデバイスのサイズにレイアウトが順応して表示されます（図 1.1.3a～c）。



↑ 図 1.1.3a テンプル大学のウェブサイト、iPhone で見た場合

⇒ <http://www.tuj.ac.jp>



↑ 図 1.1.3b 同サイトを iPad の縦位置（ポートレート）で見た場合



← 図 1.1.3c 同サイトを 1024 x 768px のデスクトップで見た場合

ぜひ、<http://www.tuj.ac.jp> にアクセスして、ブラウザのウィンドウ・サイズを変更して、ページのレイアウトが変わっていく様を見てみてください。

1.1.4. 大学やブランド企業でのレスポンス Web デザイン

以下のような大学、企業でレスポンス Web デザインを使ったウェブサイトの制作を採用しています。それぞれの画面サイズにおけるレイアウトの違いを見てみましょう。

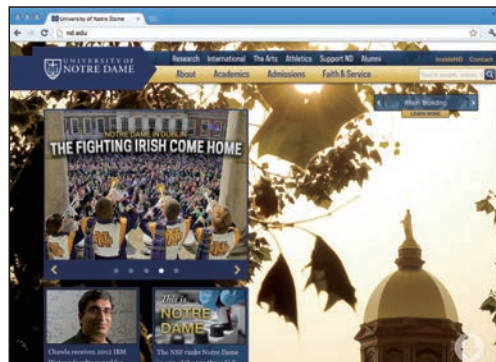
・米ノートルダム大学 <http://nd.edu/>



↑ iPhone での表示

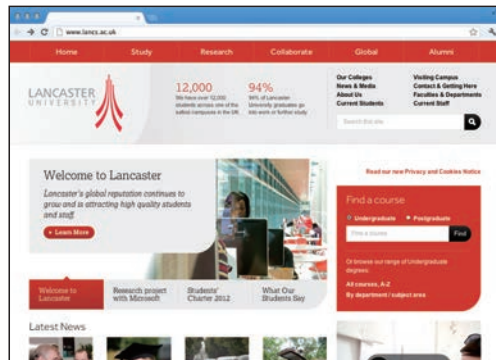
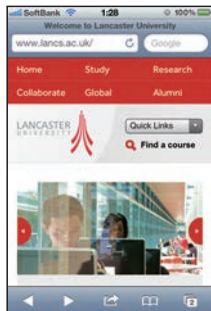


↑ iPad での表示

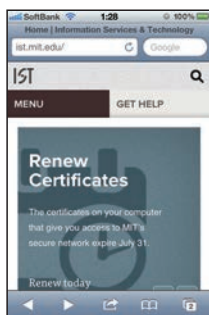


↑ デスクトップ (1024 x 768px) での表示

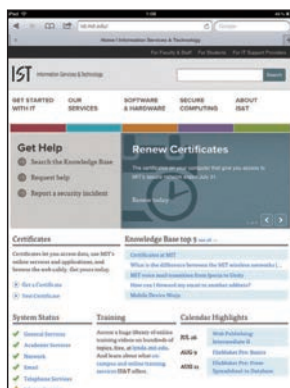
・英ランカスター大学 <http://www.lancs.ac.uk/>



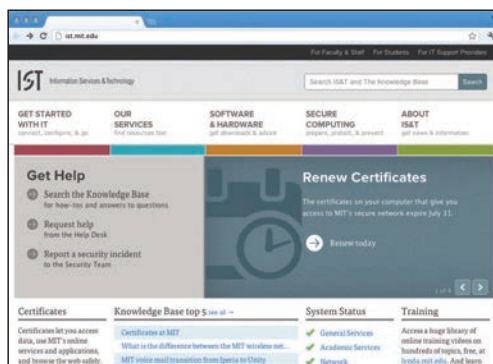
・米 MIT の情報技術サービスサイト <http://ist.mit.edu/>



↑ iPhone での表示

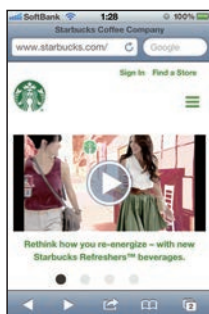


↑ iPad での表示

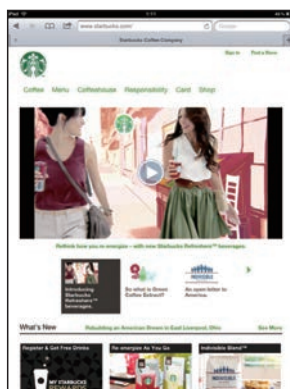


↑ デスクトップ (1024 x 768px) での表示

・Starbucks <http://www.starbucks.com>



↑ iPhone での表示



↑ iPad での表示



↑ デスクトップ (1024 x 768px) での表示

どうでしょうか？レスポンシブ Web デザインがどんなものか、ざっくりとでもご理解いただけたでしょうか？次項からはレスポンシブ Web デザインに関連するウェブの潮流を紹介します。なぜレスポンシブ Web デザインのような手法が広まり確立されつつあるのか紐解いていきましょう。

1.2. モバイルの波

2012年10月現在、携帯各社のラインナップはほとんどがスマートフォンになっています。2010年のiPadの登場からタブレット端末市場も急成長し、ユーザはスマホやiPadのようなタブレット端末からも、いつでも、どこからでもウェブサイトアクセスできるようになっています。そんななか、家や職場といった特定の場所からアクセスすることを想定して作られたデスクトップ向けウェブサイトが、ユーザのネット環境の変化にともない進化を求められています。

1.2.1. インターネットの閲覧がモバイルメインになる日も近い

2012年10月現在、筆者のネットアクセス環境も完全にモバイルに移行しつつあります。職場ではデスクトップPCからネットにアクセスしますが、家ではiPhoneやiPadでネットにアクセスすることが圧倒的に増えました。机に腰を据えて使う必要があるノートPCを開く機会が圧倒的に減っています。たとえば、MMD研究所（モバイルマーケティングデータ研究所）が行ったオンライン調査によると、スマホ所有者の77.3%がスマホを所有してからネットを使う頻度が高くなったと回答しているそうです^{※1}。

2012年4～9月でスマホの出荷台数は全体の69.4%に

2012年11月にMM総研が発表した調査結果によると、2012年度上期（4月～9月）の国内携帯電話端末の出荷台数は2,049万台で、そのうちスマートフォンは1,422万台の69.4%を占めています。

☞ <http://www.m2ri.jp/newsreleases/main.php?id=010120121101500>

※1 「スマートフォン所有者の接触頻度に関する実態調査」

☞ http://mmd.up-date.ne.jp/news/detail.php?news_id=1060

1.2.2. フィーチャーフォンとスマートフォンの大きな違い

ウェブの閲覧においてフィーチャーフォンとスマートフォンの最大の違いはブラウザにあります。スマートフォンにはデスクトップ版と機能的に同等のブラウザが搭載されています。これにより、スマートフォンユーザはデスクトップ向けのウェブサイトが普通に閲覧できるようになりました。ちょっとしたことのように思えるかもしれませんが、これは大きな違いです。フィーチャーフォンではモバイル対応していないウェブサイトが閲覧できなかったために、ユーザは携帯でウェブサイトを見ても使えないという意識が働いて

しまうからです。

フィーチャーフォンでのウェブは使えない

携帯版のウェブサイトは機能が限定されたものが多く、ユーザの要求が満たされないことが多くありました。結局、欲しい情報はPC サイトに行かないと探せない → 携帯ではあきらめる。フィーチャーフォンにも「フルブラウザ」が搭載されていますが、表示が遅くレイアウトが崩れてしまうウェブサイトが多く、快適に使えるものではありません。

スマホでウェブ閲覧が「普通」の行為に

スマートフォンでウェブサイトが閲覧できるようになってなにか起こったのでしょうか？

それは、モバイル端末でウェブサイトにアクセスすることが「普通」の行為になったのです。デスクトップ向けのウェブサイトスマホで見るのは画面サイズの制約などで困難なところがありますが、それでも、必要な情報を必要なときに取得する手段が生まれたのです。「携帯」が情報を探せる、ユーザの要求を満たせる端末になったのです。

1.2.3. スマホで見るデスクトップ向けウェブサイト

デスクトップ版のサイトをスマホで閲覧したことがある方なら体験したことがあると思いますが、使い勝手はあまり良くありません。ズームイン、ズームアウトを繰り返し、ようやく欲しい情報にたどり着いたと思ったら、リンクが小さくて間違ったリンクをタップしてしまったり、バックボタンを一回押したつもりが2つ前のページに飛んでしまったり。フラッシュを使ったウェブサイトは表示されないし、地下鉄の駅でサイトを見ていたら画像が重くてなかなかロードしなくて、電車が発車してしまってページ全体が非表示になってしまった……。フィーチャーフォンと比べれば、デスクトップ向けに作られたウェブサイトが見られるようになっただけでもましかもしれません。しかし、スマホでのデスクトップ向けウェブサイトの閲覧は、ユーザにとって普通に満足できるものではないのが現状です。



← 図 1.2.3a iPhone で見たデスクトップ版 Yahoo.com。
デスクトップの画面を縮小しただけなので、このままでは
利用しづらい

1.2.4. スマホ(モバイル)対応の必要性

スマホやiPadなどのタブレット端末が普及し、ユーザがそれらの端末でネットを日常的に利用するようになった今、それらモバイル端末にも「最適化」されたウェブサイトが必要になりました。それも早急にです。

そして、ウェブ担当者は以下のような悩みを持ち始めたのです：

- ・ スマホサイトを作るべきか？
- ・ スマホサイトとは別にタブレット向けのサイトは必要か？
- ・ それともアプリを作るべきなのか？
- ・ 使用中のCMSでスマホに対応できるのか？
- ・ ガラケーサイトはどうすればいい？

そんななか、レスポンシブ Web デザインが選択肢の一つとして浮上ってきたのです。

1.3. 未知の端末の可能性と 「未来にやさしい」ウェブ

ウェブサイトは、現在すでにさまざまなサイズや仕様を持つ端末で閲覧されています。スマートフォンやタブレットだけでもさまざまなサイズがあり、WiFi、3G、LTE などのネット接続環境があります。また、TV やゲーム機にもウェブブラウジングの環境が整いつつあります。たとえば PSP や Nintendo 3DS には NetFront というブラウザが搭載されています。2012 年 6 月には Internet Explorer 10 ベースのブラウザが Xbox 360 に搭載されることも発表されました。新しい端末がものすごいスピードで市場に投入される昨今、このような未知の端末にウェブサイトに対応させる方法はあるのでしょうか？

1.3.1. ウェブを閲覧するのに使うのはPCだけではない

つい最近までは「ウェブ」といえば PC で見るものでした。しかし、いまではデスクトップ PC、ノート PC、タブレット、スマホ、ゲーム機など、さまざまな端末からウェブを閲覧する環境が整ってきています。すでに Android を搭載したカーナビも存在しますし、近い将来、あらゆる電子機器にブラウザが搭載されるかもしれません（図 1.3.1a、図 1.3.1b、図 1.3.1c）。



← 図 1.3.1a 『これが「ウェブ」ではない』
By bradfrostweb.com @brad_frost



← 図 1.3.1b 『これがいまの「ウェブ」』
By bradfrostweb.com @brad_frost



← 図 1.3.1c 『そして、これが未来の「ウェブ」』
By bradfrostweb.com @brad_frost

1.3.2. 未知の端末にも対応する必要性

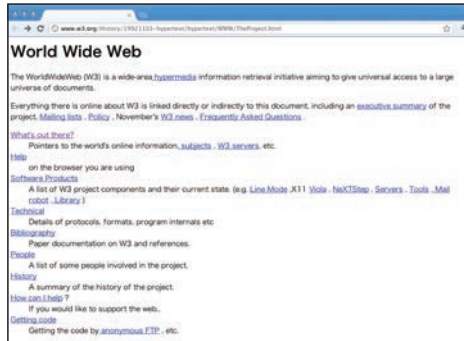
現在のウェブが置かれる状況を見ても、たとえば、米国 OpenSignalMaps の調査結果によると、195 カ国で Android 端末の機種数は 3,997 種類もあったそうです^{※1}。ユーザがどのようなスクリーンサイズの端末で、どのような OS やブラウザでウェブサイトアクセスするかを想定するのは非常に難しくなっています。そのような想定不可能な環境に、可能な限り対応するように制作するにはどうすれば良いのでしょうか？

※1 Android デバイスの断片化を示す統計（英語）
🔗 <http://opensignal.com/reports/fragmentation.php>

重要になるのは以下の 2 つです：

1. 標準に準拠したウェブ制作をする

まず、標準に準拠したウェブ制作をすることで、標準に準拠したブラウザでの閲覧はある程度担保できます。世界初のウェブページが、現在のモダンブラウザでも問題なく閲覧できるのは、標準に準拠したコーディングがされているからです（図 1.3.2a）。



← 図 1.3.2a 世界初のウェブページ
 ⇒ <http://www.w3.org/History/19921103-hypertext/hypertext/WWW/TheProject.html>

2. 「未来にやさしい」ウェブ制作

2つ目は、ユーザのスクリーンサイズとネット接続環境を例に考えてみましょう。日本では現在、固定幅でウェブサイトデザインするのが主流ですが、それでは将来登場する未知のスクリーンサイズに対応できません。現に多くのウェブサイトはスマートフォンで閲覧すると、表示が小さくなりすぎて使い勝手は断然悪くなっています。また、ブロードバンドが普及して、Flash や画像を多用したウェブサイトが増えましたが、それらのウェブサイトも 3G 回線のスマートフォンで見ると、Flash が見られなかったりダウンロードに時間がかかりすぎてしまう場合があります。

たとえばウェブサイトのレイアウトを可変にして、遅い回線を想定した制作が一般的だったとしたら、今のインターネットの状況は大きく違っていただいでしょう。この「モバイルの波」は、環境に依存し過ぎない柔軟な考え方で行う、標準に準拠したウェブサイト制作を見直す良い機会なのかもしれません。このような制作の考え方を、「モバイル・ファースト」を提唱している Luke Wroblewski 氏をはじめとする、主にアメリカのウェブ制作者が「Future Friendly (未来にやさしい)」と呼んでいます (図 1.3.2b)。



← 図 1.3.2b
 ⇒ <http://futurefriendly.ly>

1.4. コンテンツを起点に考える 「コンテンツ・ファースト」

モバイルの波が訪れ、その波にどのように対応するべきか技術や手法が議論されています。しかし、モバイルだからとかデスクトップだからではなく、「ユーザの役に立つコンテンツ」を提供することを起点にウェブサイトを作成するべきだと提唱しているのが「コンテンツ・ファースト」という考え方です。たとえば、携帯サイトだからコンテンツを制限するのではなく、まずは「ユーザに最適なコンテンツ」は何なのかを考えてみることで、ユーザにとってより有益なウェブサイトになります。そして、ユーザに有益なウェブサイトにすることで、サイトオーナーはビジネス・ゴールの達成に近づけます。いま、ユーザに求められるコンテンツを、ユーザが消費しやすい形で提供することが重要になってきています。

1.4.1. ユーザはウェブサイトを使いたいように使う

たとえば Instapaper や Readability といったツールを使うと、ウェブサイトの余計なものを取り除いた「コンテンツ」のみを表示して、好きなデバイスで読むことができます。

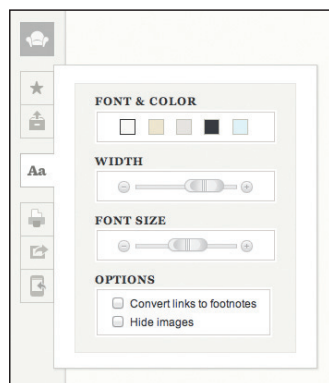


↑ 図 1.4.1a 記事を読むときには必要がないナビゲーションや広告が多い



↑ 図 1.4.1b Readability というサービスを使うと、読みたい記事の部分だけ抽出してくれる

➡ <http://www.readability.com>

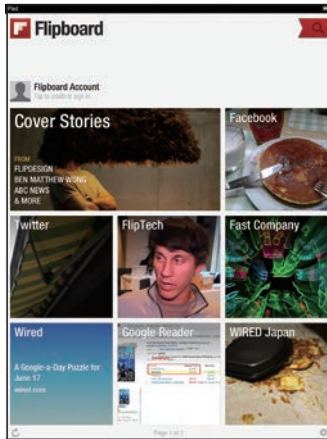


↑ 図 1.4.1c 画面左上に表示されるメニュー。記事の背景色やフォントサイズを変更できる機能もある

ユーザが求めているのは、コンテンツを好きな時に、好きなデバイスで、好きな方法で読むことです。Readabilityのようなツールを使わなくても、そういったユーザの要求に応えるウェブサイトを提供することが、最終的にはベストなアプローチなのではないでしょうか？ なぜなら、ユーザが要求する「コンテンツ」を起点に考えた制作を進めることは、ユーザにとってのメリットであり、ユーザにとってのメリットは直接ビジネスのメリットに繋がるからです。

1.4.2. コンテンツが「データ」としてさまざまな形で消費される時代

また、Flipboard というアプリを使うと、Facebook や Twitter、RSS をはじめ、さまざまなソースから集めてきた自分好みのコンテンツを、雑誌のような見た目で好きなように閲覧することができます。



← 図 1.4.2a 「友達から共有された情報を基に、あなただけの雑誌を作成できる」Flipboard (iPad 版)
<http://flipboard.com/>

ネットという情報の海から自分が必要な情報を探し、それらを効率的に消費する方法が充実してきています。筆者は Google Reader を使って複数のソースからの情報を、iPhone や iPad、職場の Windows PC や家の Mac で、好きなときに、好きな場所で、好きな形で閲覧しています。また、Facebook は友人の情報を、Twitter はフォローしたユーザの情報を収集して提供してくれます。

▶ ポータブルなコンテンツ提供

コンテンツ（情報）は「データ」としていろいろな形で共有され消費される時代が来ています。そして、昨今のソーシャルウェブの時代、この「共有されること」が大変重要になってきています。大切

なのはコンテンツをある特定の形に埋め込みコントロールすることではなく、より柔軟に、よりポータブルにユーザに提供することです。そもそもウェブサイトは情報の提供者とユーザとのコミュニケーションの場です。サイトオーナーが提供したいコンテンツとユーザが求めるコンテンツをうまくマッチングさせ、ユーザが好きなときに好きな方法でそれらを消費できる環境を提供することが、ビジネスの成功に繋がるのです。

1.4.3. コンテンツを起点に考えるウェブ制作とコンテンツ戦略

ユーザが求めるコンテンツを提供するには、最低限、以下の3つを知る（または想定できる）必要があります。

1. 提供する相手は誰か
2. 提供するコンテンツの目的はなにか
3. ユーザは、いつ、どこで、どのようにコンテンツを消費するのか

これら3つを考え、調査し、想定することで、ユーザに喜ばれるコンテンツが提供できるはずです。そして、これら3つを考慮した制作をすれば、安直に携帯サイトはデスクトップ版のコンテンツを制限したものにするといった判断はしないはずです。

▶ コンテンツは箱を埋めるものではない

コンテンツが重要だということは誰もが理解していることだと思います。しかし、SEO、マーケティング、UI、UX、デザイン、CMS、HTML5、CSS3、JavaScript など、ウェブ制作に必要なさまざまなタスクに埋もれて、コンテンツが二の次、または「箱を埋めるもの」になっていないでしょうか？ ナビゲーションや広告、企業の一人よがりなコンテンツで埋め尽くされ、ユーザが求めるものを提供できなくなったウェブサイトはユーザに見捨てられてしまいます。モバイルの波の第2波、第3波として、「コンテンツ・ファースト - コンテンツ起点の制作」の波も業界に押し寄せています。

コンテンツの重要性を訴える試み



← 図 1.4.3a

米ウェブ業界のスーパースター Zeldman 氏が彼のブログで、ナビゲーションをなくし、大きなフォントサイズ（デスクトップで見ると大きすぎ？）を使ってデザインをリニューアルをしたことで話題を呼びました。彼は「デザイナーがコンテンツにフォーカスしなければ、ユーザーが Readability や Instapaper のようなサービスを使ってそうするだけだ」と英雑誌「.net」の「Jeffrey Zeldman on his web Design Manifesto 2012」というインタビュー記事で語っています。

🔗 <http://www.zeldman.com/2012/05/18/web-design-manifesto-2012/>