●免責

本書に記載された内容は，情報の提供だけを目的としています。したがって，本書を用いた運用は，必ずお客様自身の責任と判断によって行ってください。これらの情報の運用の結果について，技術評論社および著者はいかなる責任も負いません。

本書記載の情報は，2010年3月現在のものを掲載していますので，ご利用時には，変更されている場合もあります。

また，ソフトウェアに関する記述は，特に断わりのないかぎり，2010年3月現在でのバージョンをもとにしています。ソフトウェアはバージョンアップされる場合があり，本書での説明とは機能内容や画面図などが異なってしまうこともあり得ます。本書ご購入の前に，必ずバージョン番号をご確認ください。

以上の注意事項をご承諾いただいた上で，本書をご利用願います。これらの注意事項をお読みいただかずに，お問い合わせいただいても，技術評論社および著者は対処しかねます。あらかじめ，ご承知おきください。

●商標，登録商標について

・本書に登場する製品名などは，一般に各社の登録商標または商標です。なお，本文中に ™, ® などのマークは特に記載しておりません。

はじめに

現在私たちの生活を支えているWebアプリケーションシステムは，日々進化しており，一昔前のデスクトップアプリケーションと比べても遜色のないものになりつつあります。開発現場でもフレームワークや統合開発環境が進歩し，以前よりも少ない知識と労力で開発ができるようになってきました。

一方で，Webアプリケーション開発を支える技術は高度化・複雑化し，そのすべてを一度に理解することは困難になっています。そのため，開発現場の新人教育では，即戦力として使えるようにフレームワークの使い方など表面的な部分だけを教え込んで，新人を現場に投入せざるを得ない状況が続いています。しかし，根本の仕組みを理解していなければ，問題が発生したときにその原因を突き止めて解決することはできませんし，既存技術の問題点を捉えて新しい技術を生み出していくこともできません。

このような状況が続くと，日本のIT技術者の開発力がどんどん低下してしまうのではないかと筆者は危惧しています。日本のソフトウェア事情に目を向ければ，WindowsやLinux，Firefox，Apache，Tomcat，Oracleなど，現在のWebシステムを支えているOSやブラウザ，各種ミドルウェアなど，その多くは海外の開発者によって開発されたものが利用されています。その表面的な利用だけでは，これらの製品を超えるものを生み出していくことはできず，常にユーザの立場に甘んじることになってしまいます。より一歩先に進んで，日本の開発者が世界に向けて製品を生み出していくためには，新人のうちから技術の根本に触れ，その奥深さや面白さを知っていくことが重要なのではないでしょうか。

しかし学ぶべきことはあまりにも多く，膨大な時間をかけてさまざまな書籍や雑誌，ネットの記事を読まなくてはなりません。多くの人はそこで挫折してしまうでしょう。

本書は，何年先も通じるWebアプリケーションの基礎技術を，1冊で手軽に楽しく身につけられるようにすることを目標に執筆しました。本書で説明する内容は，現在では皆さんが使っているフレームワークなどに隠蔽され，通常は知らなくても開発に支障はないかもしれません。しかし，問題解決や既存技術を応用させて新たな技術を生み出していくためには，必ず理解しておかなければならないものです。

読者の皆様が本書を出発点としてさまざまな技術を身につけ，やがては第一線のエンジニアとして活躍していく。本書が，その一助になれば筆者にとってはこの上ない喜びです。

2010年3月　小森 裕介

Contents [目次]

はじめに …… ii

LESSON 0 プロローグ …… 1
- 市民権を得た「Web アプリケーション」 …… 2
- 「Web アプリケーション」開発の難しさ …… 2
- 「Web アプリケーション」の開発技術はどこで学べる？ …… 3
- なぜ，あなたは Web アプリケーション開発技術を学べないのか …… 3
- 本書の対象読者 …… 4
- 本書を読む上での想定知識 …… 4
- 最も効率よく「技術」を学ぶ方法 …… 5

LESSON 1 「Web アプリケーション」とは何か …… 7

1.1 デスクトップアプリケーション …… 8

1.2 Web アプリケーション …… 9

1.3 まとめ …… 12

LESSON 2 Web はどのように発展したか …… 13

2.1 WWW の誕生と普及 …… 14
- 世界中のコンピュータを結ぶインターネット …… 14
- インターネット普及の立役者・World-Wide Web と Mosaic …… 15
- WWW の誕生 …… 16
- 現代 Web ブラウザの祖先・NCSA Mosaic …… 18

2.2 Web を支える技術の発明 …… 20
- Web サーバと Web クライアント …… 21
- なぜ，クライアントとサーバに分けるのか …… 22

コラム▶「クライアント」と「サーバ」偉いのはどっち？ …… 22
- 「そのリソースはどこにある？」- URL …… 23
- HTTP …… 26

コラム▶ インターネットに公開された技術仕様・RFC …… 28

2.3 CGI の誕生 …… 29
- 動的なコンテンツへの要求 …… 29
- CGI の誕生 …… 30
- Web の爆発的な普及 …… 32

2.4 サーブレットの登場 …… 33
- CGI にまつわる問題点 …… 33
- Java ／サーブレットの誕生 …… 33
- Java でアプリケーションを開発することの利点 …… 34

コラム▶ 早すぎた技術，Java アプレット …… 36

2.5 JSP の誕生 …… 37
- サーブレットの問題点 …… 37
- 発想の逆転！ JSP の誕生 …… 39

2.6 Web アプリケーションフレームワークの時代 …… 40
- サーブレットや JSP の問題点 …… 40
- Web アプリケーションフレームワークの誕生 …… 41

2.7 まとめ …… 42

LESSON 3 HTTP を知る …… 43

3.1 HTTP の知識はなぜ必要か …… 44

コラム▶ ハードウェアさえも信じられない事態！？ …… 46

3.2 Web ブラウザと Web サーバの通信をのぞいてみよう …… 46
- 横取り丸と InetSpy のインストール …… 47
- HTTP 通信をのぞいてみよう …… 49
- HTTP リクエストをのぞく …… 50

コラム▶ URL と URI は何が違うのか？ …… 53
- HTTP レスポンスをのぞく …… 54
- HTTP では 1 回で 1 つのリソースを取得 …… 57
- ファイル名を省略した場合のリクエスト …… 58

3.3 情報はどうやってインターネットの大海原を越えるのか …… 59
- インターネット上の住所・IP アドレス …… 59
- IP アドレスを頼りに情報を届ける TCP/IP …… 60
- IP アドレスは誰が決めるのか …… 62

- グローバル IP アドレスとプライベート IP アドレス …… 62
コラム▶ IP アドレスと個人情報 …… 64
- ホスト名を IP アドレスに変換する DNS …… 64
- DNS はどのようにして実現されるのか …… 66
- ホスト内の宛先を決定するポート番号 …… 67

3.4 Web サーバへの要求をどのように伝えるか …… 70
- GET メソッドによるパラメータ渡し …… 70
- アプリケーション側でのパラメータの受け取り …… 73
- POST メソッドによるパラメータ渡し …… 75
- GET と POST どちらを使えばよい？ …… 78
- 日本語はどのようにして渡せばよいか …… 79

3.5 まとめ …… 82

LESSON 4 CGI から Web アプリケーションへ …… 85

4.1 宅配ピザ注文サイトを作ろう …… 86

4.2 画面構成を考える …… 87
コラム▶ 実際の Web システム開発の流れ …… 88

4.3 画面モックを作ろう …… 88

4.4 ログイン認証機能を作成する …… 91
- PHP で認証機能を作ろう …… 91
- 認証機能の動作を確認しよう …… 93
- リダイレクト動作の HTTP 通信を確認しよう …… 93
コラム▶ PHP はどのように実行されるのか～ CGI とモジュールの違い～ …… 95

4.5 ログイン状態をどのようにして記憶するのか …… 96
- ステートフルなプロトコルとステートレスなプロトコル …… 97
- ステートレスな HTTP 上で状態をどのように表現するか …… 100
- Cookie を利用して状態を保持する …… 101
- Cookie 利用の実際を確認する …… 104

4.6 安全に状態を保存するための技術 ーセッションー …… 108
- Cookie にまつわる問題点 …… 108
コラム▶ Cookie はどこに保存されている？ …… 109
- 銀行の窓口業務でセッションを理解しよう …… 109

- 口座開設業務の進行状況をどのように管理するか …… 112
- セッションで処理の進行状況を管理する …… 112
- セッションの状態をどこで保持するか …… 113
- HTTP におけるセッション ID の受け渡し方法 …… 115
- セッション ID 利用の実際を確認する …… 116
- セッション ID によるユーザの識別 …… 119

4.7 ピザ・ペントミノの完成 …… 121
コラム▶ Web サーバによる認証機能 ～ Basic 認証～ …… 123

4.8 まとめ …… 124

LESSON 5 Web アプリケーションの構成要素 …… 125
- なぜ Web アプリケーションの構成を理解しなければならないのか …… 126
コラム▶ コンピュータは「節」？ …… 128

5.1 Web サーバと Web クライアントの時代 …… 129
- WWW の黎明期 …… 129
- CGI の時代 …… 130
コラム▶ ソフトウェア？　プログラム？　アプリケーション？　サーバ？ …… 131

5.2 データベースサーバの登場 …… 133
- 大量の情報をどのようにして管理するか …… 133
- データベース管理システムの登場 …… 134
コラム▶ DB?　DBMS?　RDBMS? …… 135
- データベースに対する操作 …… 136
- データベースによる情報の管理 …… 137
コラム▶ データベース設計は IT システムの要 …… 138
- データベースから情報を抽出する …… 139
- 必要な情報を SQL でデータベースへ伝える …… 140
コラム▶ データベースに対する CRUD 操作と SQL 文の関係 …… 141
- データベースとクライアントの関係 …… 142
- データベースサーバの分離 …… 144
- Web アプリケーションとデータベースの通信 …… 147
コラム▶ 代表的なデータベース製品 …… 148

5.3 アプリケーションサーバの登場 …… 149
- Servlet や JSP はどこで動いているのか …… 149

- Servlet / JSP を動かすためのアプリケーションサーバ …… 150
- Web サーバとアプリケーションサーバの連携 …… 151
- Web サーバとアプリケーションサーバの分担 …… 152
- Web サーバとアプリケーションサーバ連携のメリット …… 154
- 複数の Tomcat への転送 …… 156
- Web サーバの機能を持ったアプリケーションサーバ …… 158

コラム▶ アプリケーションサーバの提供する機能 …… 160

5.4 Web システムの三層構成 …… 161

- 最小構成の Web システム …… 161
- 一般的な構成 …… 162
- Web システムの三層構成 …… 164

コラム▶ 現代の Web システムを支えるオープンソース …… 164

5.5 まとめ …… 165

LESSON 6 Web アプリケーションを効率よく開発するための仕組み …… 168

6.1 サーブレット／ JSP だけではいけないのか …… 168

- Web アプリケーション開発のスタンダード・Java …… 168
- サーブレットと JSP の連携 …… 169

6.2 サーブレット／ JSP で「ピザ・ペントミノ」のログイン処理を実現する …… 170

- JSP によるログイン画面の表示 …… 170
- サーブレットの呼び出し …… 172
- ログインサーブレットの処理 …… 173
- フォワードとリダイレクトの違い …… 174
- リクエストスコープにおける情報の受け渡し …… 176
- JSP におけるリクエストスコープからの情報の取り出し …… 178
- なぜリクエストスコープが必要なのか …… 179
- セッションスコープとリクエストスコープの違い …… 181

コラム▶ さまざまなセッションの実現方法 …… 182

6.3 Web アプリケーションのアーキテクチャ …… 183

- ロジックとデザインの分離 …… 183
- ソフトウェアの建築様式 …… 184

コラム▶ カスタムタグと JSTL …… 186

- 「ピザ・ペントミノ」の構造を俯瞰しよう …… 187
- MVC モデルによる Web アプリケーションのアーキテクチャ …… 189
- MVC モデルによる処理の流れ …… 192

6.4 フレームワークによるアーキテクチャの実現 …… 193

- フレームワークとは何か …… 193
- Struts による MVC モデルの実現 …… 195
- Struts による「ピザ・ペントミノ」のログイン処理 …… 198
- JSP からのログイン処理アクションの呼び出し …… 201

コラム▶ Java を部品化するための仕組み - Java Beans - …… 203

- ログイン処理アクションでのログインチェック処理 …… 204
- 商品一覧画面への遷移 …… 206

6.5 レイヤパターンによるデータアクセス層の分離 …… 208

- モデルをどのように実現するか …… 208
- JDBC によるデータベースからの情報の取得 …… 211
- レイヤパターンによるデータアクセス層の分離 …… 214
- DAO パターンによるデータアクセス層の実現 …… 217

6.6 O/R マッピングフレームワークによるデータアクセス層の実現 …… 219

- O/R マッピングフレームワークの必要性 …… 219
- RDB とオブジェクトのインピーダンス・ミスマッチ …… 220
- iBATIS による O/R マッピングの実際 …… 223
- Data Mapper と SQL マップファイルによる O/R マッピング処理 …… 224
- Dao Framework を利用した DAO の作成 …… 226

6.7 フレームワーク利用におけるメリットとデメリット …… 230

- フレームワーク利用のメリット …… 230
- フレームワーク利用のデメリット …… 231

6.8 まとめ …… 234

LESSON 7 セキュリティを確保するための仕組み …… 236

7.1 なぜセキュリティを確保しなければならないのか …… 236

- Web アプリケーションが守るべきセキュリティ …… 236

7.2 代表的な Web アプリケーションの攻撃手法とその対策 238
- SQL インジェクション 238
- クロスサイトスクリプティング（XSS） 241
- セッションハイジャック 244

コラム▶ SSL による通信路の暗号化 246
- クロスサイトリクエストフォージェリ 248

コラム▶ セキュリティの陰の立役者・ハッシュ関数 251
- 強制ブラウズ 253
- ディレクトリトラバーサル 255

コラム▶ より安全な認証方式～ Digest 認証～ 256

7.3 設計・実装ミスに起因する誤動作やセキュリティ問題を防ぐための対策 258
- 「戻る」ボタン対策 258
- ダブルサブミット対策 261
- hidden タグに重要な情報を持たせない 263
- デバッグ情報を出力させない 263
- グローバル変数に情報を持たせない 264

7.4 まとめ 266
- 謝辞 268

LESSON 8 おわりに 268

LESSON 9 付録 270

9.1 参考書籍・サイト 270
- Lesson 0 270
- Lesson 2 270
- Lesson 3 270
- Lesson 4 271
- Lesson 5 271
- Lesson 6 272
- Lesson 7 272

索引 274