

## 4-5-基. Web アプリケーションに関する知識

### 1. 科目の概要

スクリプト言語や Java 言語を利用して、Ruby on Rails やその他 Web フレームワークを活用して、マークアップ言語およびスクリプト言語を用いた Web サービスを提供するアプリケーションを開発する際に必要とされるスキルを学習する。基本編では Web に関する基礎的な事項を中心に学ぶ。

### 2. 習得ポイント

本科目の学習により習得することが期待されるポイントは以下の通り。

習得ポイント	説明	シラバスの対応コマ
4-5-基-1. Webブラウザ, Webサーバ, CGI	Webアプリケーションの構成要素として、サーバとクライアントであるWebサーバとWebブラウザのOSSによる代表的な実装を紹介し、それらの動作の概略を説明する。また動的なサービス生成の基礎であるCGIについても触れる。	1
4-5-基-2. HTTP/HTTPS	Webアプリケーションにおいて、サーバとクライアント間で通信するために利用されるプロトコルのHTTPを紹介する。HTTPを暗号化し、セキュリティ機能を備えたHTTPSについても解説する。	1
4-5-基-3. HTML/CSS	そもそもWebコンテンツはどのような技術から構成されているのかを説明し、Webの最も基本的な要素であるHTMLについて解説する。さらに、CSSとは何か、なぜCSSを使うのかということも説明する。	2
4-5-基-4. DOM, JavaScript	動きのあるWebサイトを作る技術であるJavaScriptとDHTMLについて解説する。また、文書構造やデザインの構成を指定するために用いるDOM (Document Object Model)の意義を説明し、実際に動的な変化を可能とするページの作り方を示す。	3
4-5-基-5. XML/XSL	構造的なデータを記述するための汎用的な言語であるXML (eXtensible Markup Language)の概念を説明し、XMLの構造や定義、XMLにより記述されたデータの表示をXSLによりデザインする方法を解説する。	2
4-5-基-6. 3層アプリケーション、デザインとロジックの分離	開発効率を高めるために必須の技法であるデザインとロジックの分離について説明する。さらに一般的なWebアプリケーションの構成である3層アプリケーションとは何か、その概要を解説する。プログラムをモデル(Model)、ビュー(View)、コントローラ(Controller)に分けて設計することで、システム構成の抽象化を高めて説明する際に利用されるMVCモデルについて解説する。	7
4-5-基-7. スクリプト言語	Webアプリケーション開発でしばしば利用されるOSSのスクリプト言語を紹介し、その特徴やWebアプリケーション開発特有の使い方、表現方法などを説明する。	5
4-5-基-8. 開発フレームワーク	Webアプリケーションは通常、スクラッチから開発されることは稀であり、数多くの開発フレームワークが利用される。ここでは開発フレームワークとは何か、開発フレームワークの種類やOSSによる代表的な実装を紹介する。	7,8
4-5-基-9. DBインタフェース	アプリケーションからDBの操作を行う際に、SQLによるアクセスを隠蔽し生産性を高めるORマッパーなど、データベース操作のインタフェースについて説明する。	10
4-5-基-10. セッション管理	セッション管理の必要性について触れ、クッキーを用いたセッション管理やセッション管理を乱す要素(戻るボタン)の回避方法など、Webアプリケーション特有のセッション管理方法について解説する。	6

#### 【学習ガイダンスの使い方】

- 「習得ポイント」により、当該科目で習得することが期待される概念・知識の全体像を把握する。
- 「シラバス」、「IT 知識体系との対応関係」、「OSS モデルカリキュラム固有知識」をもとに、必要に応じて、従来の IT 教育プログラム等との相違を把握した上で、具体的な講義計画を考案する。
- 習得ポイント毎の「学習の要点」と「解説」を参考にして、講義で使用する教材等を準備する。

### 3. IT 知識体系との対応関係

「4-5-基. Web アプリケーションに関する知識」と IT 知識体系との対応関係は以下の通り。

科目名	1	2	3	4	5	6	7	8	9	10
4-5-基. Webアプリケーション開発に関する知識	Webアーキテクチャの概要	Webの表現技術 (HTMLとCSS)	Webの表現技術 (DOMとJavaScript)	Web標準およびWebアプリケーションの基礎演習	アプリケーションの記述言語	クッキーとセッション	Webアプリケーションの仕組み	PHPを用いたWebアプリケーション開発の概要	PHPの文法・演算子、フロー制御、関数と配列	ファイルとデータベースへのアクセス

11	12	13
PHPの文字列操作と正規表現	Webフレームワークとオブジェクト指向	再利用可能なWebコンポーネント

#### <IT 知識体系上の関連部分>

分野	科目名	1	2	3	4	5	6	7	8	9	10	11	12	13	
総論的基礎事項と情報セキュリティ	1	IT-IAS 情報保護と情報セキュリティ	IT-IAS2 情報セキュリティの仕組み (対策)	IT-IAS3 運用上の問題	IT-IAS4 ホリゾン	IT-IAS5 攻撃	IT-IAS6 情報セキュリティ分野	IT-IAS7 フェレシジック (情報状態)	IT-IAS8 情報のセキュリティ	IT-IAS9 情報セキュリティ	IT-IAS10 脅威分析モデル	IT-IAS11 脆弱性			
	2	IT-SP 社会的な観点とプロフェッショナルとしての課題	IT-SP1 プロフェッショナルとしてのコミュニケーション	IT-SP2 コンピュータの歴史	IT-SP3 コンピュータを取り巻く社会環境	IT-SP4 チームワーク	IT-SP5 知的財産権	IT-SP6 コンピュータの法的問題	IT-SP7 組織の中のIT	IT-SP8 プロフェッショナルとしての倫理的な問題と責任	IT-SP9 プライバシーと個人の自由				
応用技術	3	IT-IW 情報管理	IT-IW1 情報管理の概念と基礎	IT-IW2 データベース関係の連携	IT-IW3 データアーキテクチャ	IT-IW4 データモデリングとデータベース設計	IT-IW5 データと情報の管理	IT-IW6 データベースの応用分野							
	4	IT-WS Webシステムとその技術	IT-WS1 Web技術	IT-WS2 情報アーキテクチャ	IT-WS3 デジタルメディア	IT-WS4 Web開発	IT-WS5 脆弱性	IT-WS6 ソーシャルソフトウェア							
ソフトウェアの方法と技術	5	IT-PP プログラミング基礎	IT-PP1 基本プログラミング	IT-PP2 フロントエンドの基本的構成要素	IT-PP3 オブジェクト指向プログラミング	IT-PP4 アルゴリズムと問題解決	IT-PP5 イベント駆動プログラミング	IT-PP6 再帰							
	6	IT-PT 技術者としてのプログラミング	IT-PT1 システム開発	IT-PT2 データ取り扱いと交換	IT-PT3 統合的コーディング	IT-PT4 スクリプトプログラミング	IT-PT5 ソフトウェアセキュリティの実現	IT-PT6 種々の問題	IT-PT7 プログラミング言語の概要						
	7	OE-SWE ソフトウェア工学	OE-SWE0 歴史と概要	OE-SWE1 ソフトウェアプロセス	OE-SWE2 ソフトウェアの要求と仕様	OE-SWE3 ソフトウェアの設計	OE-SWE4 ソフトウェアのテストと検証	OE-SWE5 ソフトウェアの保守	OE-SWE6 ソフトウェア開発・保守ツールと環境	OE-SWE7 ソフトウェアプロジェクト管理	OE-SWE8 言語翻訳	OE-SWE9 ソフトウェアのフォールトレランス	OE-SWE10 ソフトウェアの構成管理	OE-SWE11 ソフトウェアの標準化	
	8	IT-SIA システムインテグレーションとアーキテクチャ	IT-SIA1 要求仕様	IT-SIA2 調達/手配	IT-SIA3 インテグレーション	IT-SIA4 プロジェクト管理	IT-SIA5 テストと品質保証	IT-SIA6 組織の特性	IT-SIA7 アーキテクチャ						
	9	IT-NET ネットワーク	IT-NET1 ネットワークの基礎	IT-NET2 ルーティングとスイッチング	IT-NET3 物理層	IT-NET4 セキュリティ	IT-NET5 アプリケーション分野	IT-NET6 ネットワーク管理							
	10	OE-NWK テレコミュニケーション	OE-NWK0 歴史と概要	OE-NWK1 通信ネットワークのアーキテクチャ	OE-NWK2 通信ネットワークのプロトコル	OE-NWK3 LANとWAN	OE-NWK4 クラウドサービスとセキュリティ	OE-NWK5 データのセキュリティと整合性	OE-NWK6 ワイヤレスコンピュータネットワーク	OE-NWK7 データ通信	OE-NWK8 組み込み機器向けネットワーク	OE-NWK9 通信技術とネットワーク概要	OE-NWK10 性能評価	OE-NWK11 ネットワーク管理	OE-NWK12 信頼と伸張
	11	IT-PT プラットフォーム技術	IT-PT1 オペレーティングシステム	IT-PT2 アーキテクチャと機構	IT-PT3 コンピュータインフラストラクチャ	IT-PT4 デバイスとソフトウェア	IT-PT5 ファームウェア	IT-PT6 ハードウェア							
	12	OE-OPS オペレーティングシステム	OE-OPS0 歴史と概要	OE-OPS1 実行性	OE-OPS2 スケジューリングと優先度	OE-OPS3 メモリ管理	OE-OPS4 セキュリティと保護	OE-OPS5 ファイル管理	OE-OPS6 リアルタイムOS	OE-OPS7 OSの構成	OE-OPS8 設計と保護	OE-OPS9 デバイスマネジメント	OE-OPS10 システム性能評価		
ソフトウェアエンジニアリング	13	OE-CAO コンピュータのアーキテクチャとドメイン	OE-CAO0 歴史と概要	OE-CAO1 コンピュータアーキテクチャの基礎	OE-CAO2 メモリシステムの構成とアーキテクチャ	OE-CAO3 インタフェースと通信	OE-CAO4 デバイスサブリシステム	OE-CAO5 GPUアーキテクチャ	OE-CAO6 性能・コスト評価	OE-CAO7 分散・並列処理	OE-CAO8 コンピュータによる計算	OE-CAO9 性能向上			
	14	IT-ITF IT基礎	IT-ITF1 ITの一般的なテーマ	IT-ITF2 組織の問題	IT-ITF3 ITの歴史	IT-ITF4 IT分野 (学科) とそれに関連する分野 (学科)	IT-ITF5 応用領域	IT-ITF6 IT分野における数学と統計学の活用							
複製領域にまたがるもの	15	OE-ESY 組み込みシステム	OE-ESY0 歴史と概要	OE-ESY1 低電力コンピュータ設計	OE-ESY2 高信頼性システムの設計	OE-ESY3 組み込み用アーキテクチャ	OE-ESY4 開発環境	OE-ESY5 ライフサイクル	OE-ESY6 要件分析	OE-ESY7 仕様定義	OE-ESY8 構造設計	OE-ESY9 テスト	OE-ESY10 プロジェクト管理	OE-ESY11 並行設計 (ハードウェア、ソフトウェア)	OE-ESY12 実装 (ハードウェア、ソフトウェア)
			OE-ESY13 リアルタイムシステム設計	OE-ESY14 組み込みマイコンコントローラ	OE-ESY15 組み込みプログラム	OE-ESY16 設計手法	OE-ESY17 ツールによるサポート	OE-ESY18 ネットワーク型組み込みシステム	OE-ESY19 インテグレーション	OE-ESY20 センサ技術	OE-ESY21 デバイスドライバ	OE-ESY22 メンテナンス	OE-ESY23 専門システム	OE-ESY24 信頼性とフォールトレラランス	

#### 4. OSS モデルカリキュラム固有の知識

OSS モデルカリキュラム固有の知識として、OSS を活用した Web アプリケーション開発環境がある。また、代表的な OSS のスクリプト言語である PHP も学習対象である。

科目名	第1回	第2回	第3回	第4回	第5回	第6回	第7回	第8回	第9回	第10回	第11回	第12回	第13回
4-5-基 Webアプリケーション開発に関する知識	(1) Webアプリケーションとは  (2) WWWの歴史  (3) WWWの仕事の仕組み  (4) Webアプリケーションのセキュリティ	(1) Webコンテンツの構成  (2) HTMLとXHTML  (3) XHTMLの記述法  (4) CSS  (5) その他の留意点	(1) 動きのあるWebサイト  (2) WebページへのJavaScriptの埋め込み  (3) DHTMLとDOM  (4) Webサイトの制作環境	(1) Web標準とは  (2) Webサイト構築演習	(1) プログラム言語とスクリプト言語  (2) Webアプリケーションに利用されることの多い言語	(1) セッション管理の必要性  (2) Webアプリケーションにおける問題点  (3) クッキーの利用  (4) クッキーを用いたセッション管理	(1) 3層アプリケーション  (2) 各種のモデル	(1) PHPの概要  (2) Webアプリケーションでの利用  (3) PHPフレームワーク	(1) PHPの基礎的な文法  (2) PHPのフロー制御  (3) 関数の利用  (4) 拡張モジュールの利用  (5) 配列の操作	(1) ファイルの読み書き  (2) デレクトリへのアクセス  (3) ファイルシステムのセキュリティ  (4) データベース関数の基本  (5) 各種フレームワークを利用したデータベース	(1) 文字列の操作  (2) 正規表現  (3) 日本語文字列の取扱い	(1) PHPによるオブジェクト指向の基礎  (2) 正規表現	(1) 再利用可能性とは  (2) PHPによる再利用可能なコンポーネント

(網掛け部分は IT 知識体系で学習できる知識を示し、それ以外は OSS モデルカリキュラム固有の知識を示している)

スキル区分	OSS モデルカリキュラムの科目	レベル
セキュリティ分野	4-5 Web アプリケーションに関する知識	基本
習得ポイント	4-5-基-1. Web ブラウザ, Web サーバ, CGI	
対応する コースウェア	第1回 Web アーキテクチャの概要	

## 4-5-基-1. Web ブラウザ, Web サーバ, CGI

Web アプリケーションの構成要素として、サーバとクライアントである Web サーバと Web ブラウザの OSS による代表的な実装を紹介し、それらの動作の概略を説明する。また動的なサービス生成の基礎である CGI についても触れる。

### 【学習の要点】

- \* Web システムは、Web サーバと Web ブラウザで構成される。
- \* Web ブラウザと Web サーバは HTTP プロトコルで通信を行い、Web ブラウザは URL に含まれるホスト名で示される Web サーバと HTTP 通信する。
- \* HTTP レスポンスの種類は、テキスト形式の HTML や、バイナリ形式の画像ファイルなどがあり、Web ブラウザによってユーザの端末に表示される。
- \* Web サーバは静的な HTML ファイルを返すだけでなく、CGI という仕組みを利用して動的にコンテンツを生成することができる。

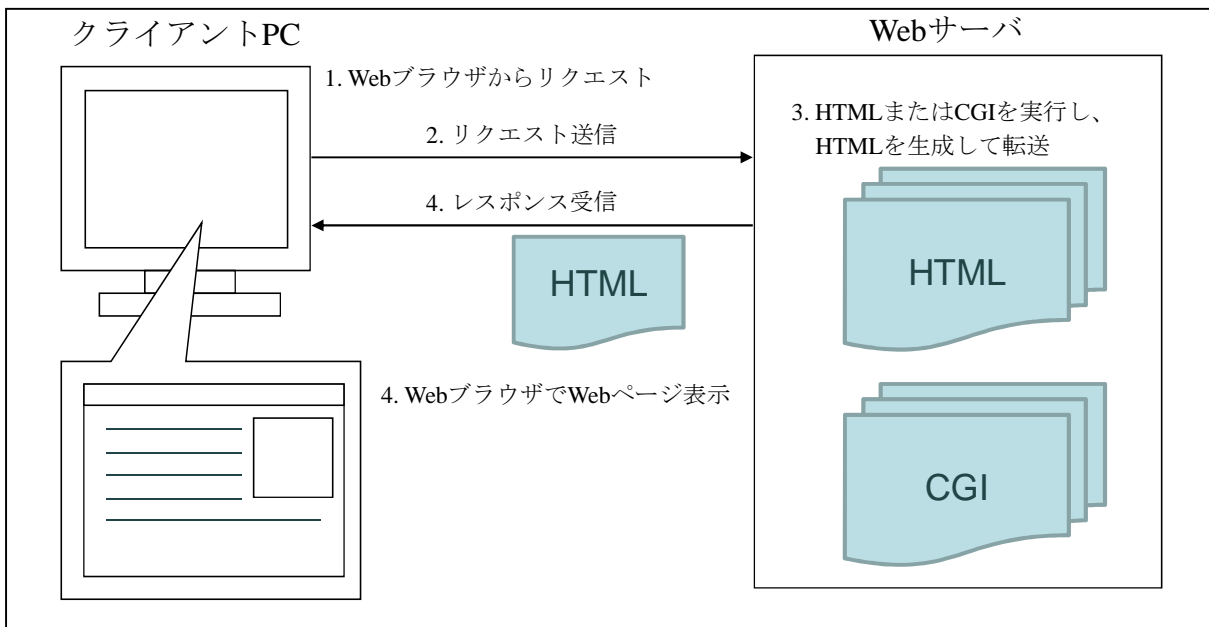


図 4-5-基-1. Web システムの仕組み

## 【解説】

### 1) Web システムの仕組み

- \* Web システムは、システムのサービスを提供する Web サーバとそのサービスを受けるクライアント PC で構成される。
- \* Web サーバは、Web ページのリソースとなる HTML ファイルや画像ファイルなどを保管し、インターネットを経由してクライアント PC からのリクエスト(要求)にレスポンス(応答)し、該当のリソースデータを送信する。
- \* クライアント PC は、Web ブラウザを使用して Web サーバとデータの送受信を行う。
- \* Web サーバには、Web ブラウザから Web サーバのホスト名、ポート番号、リソースの保管場所を表す URL を指定してアクセスする。

### 2) HTTP 通信

- \* Web ブラウザと Web サーバは、インターネットで広く使用されている HTTP プロトコルと TCP/IP プロトコルを使用して通信を行う。
- \* Web ブラウザは、HTTP プロトコルを使用して、URL に含むホスト名の Web サーバに HTTP リクエストを送信し、Web サーバからの HTTP レスポンスを受信する。
- \* TCP/IP プロトコルは、HTTP プロトコルよりも通信の基盤に関わるパケット通信、フロー制御や通信エラーの訂正などを行う。

### 3) HTTP レスポンスの種類

- \* HTTP レスポンスのデータの種類は、Web ページを構成する HTML データや画像データ、音声データなどがある。
- \* Web ブラウザは、受信した HTML データや画像データを解釈および実行し、Web ページとして画面表示する。

### 4) 代表的な Web サーバと Web ブラウザ

- \* 代表的な Web サーバには、Apache HTTP Server、lighttpd、nginx などがある。
- \* 代表的な Web ブラウザには、Mozilla FireFox、Google Chrome などがある。
- \* OSS 以外では、Web サーバの Microsoft Internet Information Services(IIS)、Web ブラウザの Internet Explorer、Opera、Safari が有名である。

### 5) 動的コンテンツの生成

- \* HTML で静的な Web ページを Web ブラウザに表示するだけでなく、動的に Web ページを生成するためには、CGI(Common Gateway Interface)などの技術を活用する必要がある。
- \* CGI は、Web サーバに保管されるプログラムで、Web ブラウザからのリクエストに対し、記述されているプログラムの処理を行い、HTML データを生成してレスポンスする。

スキル区分	OSS モデルカリキュラムの科目	レベル
セキュリティ分野	4-5 Web アプリケーションに関する知識	基本
習得ポイント	4-5-基-2. HTTP/HTTPS	
対応する コースウェア	第1回 Web アーキテクチャの概要	

## 4-5-基-2. HTTP/HTTPS

Web アプリケーションにおいて、サーバとクライアント間で通信するために利用されるプロトコルの HTTP を紹介する。HTTP を暗号化し、セキュリティ機能を備えた HTTPS についても解説する。

### 【学習の要点】

- \* HTTP リクエストについて、HTTP メソッド (GET、POST) や、HTTP/1.0 および HTTP/1.1 について説明する。
- \* HTTP レスポンスのヘッダ (ステータスコード、コンテンツタイプ、etc) や、コンテンツ本体について説明する。
- \* HTTP での通信は平文で行われるため、個人情報や電子決済など暗号化によって盗聴やなりすましを防ぐためには HTTPS が使われる。

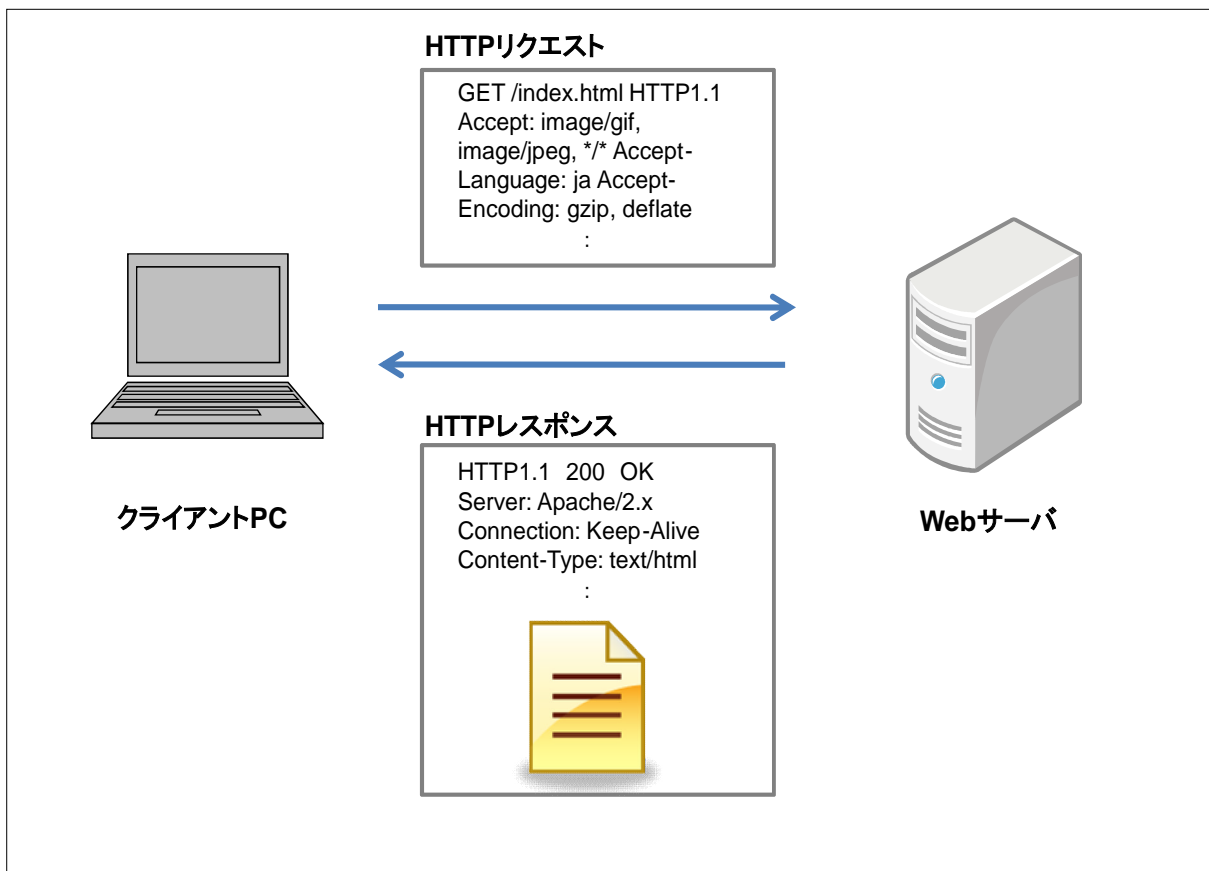


図 4-5-基-2. HTTP リクエストと HTTP レスポンス

## 【解説】

### 1) HTTP リクエスト

- \* HTTP リクエストはクライアントからサーバに送られる情報であり、リクエスト行、ヘッダ部分(メッセージヘッダ)、ボディ部分(メッセージボディ)からなる。
- \* リクエスト行は「メソッド名 対象リソースの URI HTTP の版数」形式で記述する。
- \* 現在多く用いられている HTTP メソッドには、GET メソッド、POST メソッドがある。GET メソッドはサーバの情報を得るために使用し、POST メソッドはサーバに情報を送るために使用する。その他、HEAD、PUT、DELETE などのメソッドが存在する。
- \* HTTP の版数には HTTP/0.9~HTTP/1.1 が存在し、現在多く利用されている版数は複数データを転送するためのキープアライブ機能やプロキシなどの使用も想定された HTTP/1.1 である。
- \* ヘッダ部分にはクライアントの受け入れ可能コンテンツタイプやリクエストの出所などのヘッダ情報を格納する。
- \* ボディ部分には POST などの場合にサーバに送る情報を格納する。

### 2) HTTP レスポンス

- \* HTTP レスポンスはサーバからクライアントに送られる情報であり、ステータス行、ヘッダ部分(メッセージヘッダ)、ボディ部分(メッセージボディ)からなる。
- \* ステータス行にはリクエストに対する処理結果を表す 3 桁の数字(ステータスコード)が格納され、100-199 が情報提供、200-299 が成功、300-399 がリダイレクト、400-499 がクライアントエラー、500-599 がサーバエラーを表す。
- \* ヘッダ部分にはサーバから送られる情報のタイプを表すコンテンツタイプ、コンテンツサイズ、更新日時などのヘッダ情報が格納される。
- \* ボディ部分には、クライアントがリクエストの際に指定した URL で示されるコンテンツ情報が格納される。

### 3) HTTPS 通信

- \* HTTPS は HTTP による通信について安全性を向上させる目的で幅広く利用されており、通信プロトコルには、HTTP over SSL/TLS が用いられている。
- \* HTTP over SSL/TLS はメッセージを平文のまま送受信する HTTP とは違い、SSL (Secure Sockets Layer) や TLS (Transport Layer Security) を用いて、サーバ認証・クライアント認証やメッセージの暗号化を行う。これに伴い、盗聴やなりすましからの攻撃を保護し、プライバシーに関わる情報などを安全にやり取りすることができる。
- \* SSL 通信を実現するために、サーバは SSL サーバ証明書を公開する必要がある。SSL サーバ証明書には、サーバ所有者の情報や証明書を発行した機関の署名データ、暗号化に必要な鍵情報が含まれる。

スキル区分	OSS モデルカリキュラムの科目	レベル
セキュリティ分野	4-5 Web アプリケーションに関する知識	基本
習得ポイント	4-5-基-3. HTML/CSS	
対応する コースウェア	第 2 回 Web の表現技術 (HTML と CSS)	

### 4-5-基-3. HTML/CSS

そもそも Web コンテンツはどのような技術から構成されているのかを説明し、Web の最も基本的な要素である HTML について解説する。さらに、CSS とは何か、なぜ CSS を使うのかということを説明する。

#### 【学習の要点】

- \* 一般的な Web コンテンツは、HTML や CSS、その他のメディアファイル (画像、音声、動画、PDF など) によって構成されている。
- \* HTML (HyperText Markup Language) は W3C が仕様をまとめているマークアップ言語で、文章の構造や見栄えなどを記述でき、また、他の Web コンテンツへのリンクを埋め込むことができる。
- \* CSS (Cascading Style Sheets) は、Web コンテンツのレイアウトやフォントの大きさや文字飾りといった見栄えに関する情報を定義する規格。CSS を使うことで、文章の論理構造と見栄えの定義を分離させることができるので、HTML 本体が簡潔になり、また CSS を切り替えることで見栄えだけを一括して変更することが可能となる。

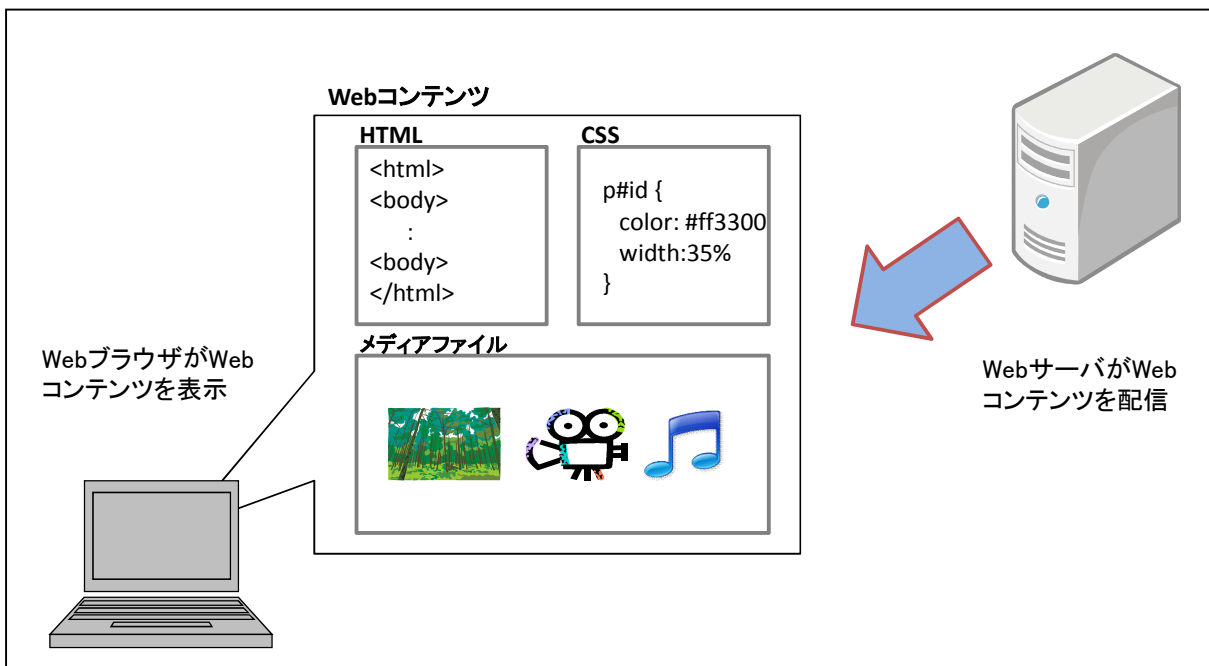


図 4-5-基-3. Web コンテンツの構成



## 【解説】

### 1) Web コンテンツとは

- \* Web コンテンツとは一般的には Web ブラウザから参照する情報を指し、HTML や CSS、メディアファイル(画像、音声、動画など)から構成される。
- \* Web コンテンツには従来の Web ブラウザ機能の範囲に収まらない、Web ブラウザにプラグイン機能を追加することによって参照することが可能となる Flash コンテンツや PDF ファイルなども含まれる。

### 2) HTML (HyperText Markup Language)

HTML は Web ページを記述するためのマークアップ言語であり、文書の表示方法や論理構造を表現できることが特徴である。また、他の Web ページやメディアファイルなどの Web コンテンツにリンクを埋め込むことでコンテンツ同士を関連付けることができる。

#### \* HTML 仕様

HTML 仕様は W3C によって標準化がおこなわれている。代表的なブラウザであれば HTML 文書を解釈し、表示することができる。

#### \* HTML の記述方法

HTML 文書を記述する際には「`<`」「`>`」で囲まれたタグという文字列を用いる。タグには開始タグ、終了タグがあり、それぞれ`<要素名>`、`</要素名>`形式で記述する。

また、タグには要素に特性を付加する属性情報を追加することができ、`<要素名 属性名=属性値>`形式で記述する。一つのタグに複数の属性を追加することが可能である。

なお、タグの要素名および属性名には大文字と小文字の区別がなく、通常のブラウザであればどちらで記述しても同様の結果が表示される。

### 3) CSS (Cascading Style Sheets)

CSS は Web ページの装飾やレイアウトを定義するための仕様であり、HTML の文書構造と装飾やレイアウトの情報を分離させることを目的としている。CSS を利用することで、文字のフォントやサイズ、装飾や行間などの文書の見栄えに関する情報を HTML 文書から分離し、文書の論理構造を表すという HTML 本来の目的を実現することが可能となる。

#### \* CSS 仕様

CSS 仕様は W3C によって標準化が行われている。HTML 同様に代表的なブラウザであれば装飾やレイアウトに関する情報を解釈し、表示することができる。

#### \* CSS の記述方法

CSS を記述する際には「`セレクタ {宣言、...}`」形式で記述する。

- セレクタとはスタイルを適用する対象のタグ要素を特定する記述である。特定の種類のタグ要素全体や特定のクラス属性をもつタグ要素、特定の ID 属性を持つタグ要素などの指定方法がある。
- 宣言には適用するスタイルを記述する。「`プロパティ:値`」形式で記述し、「`{`」「`}`」で囲まれた宣言ブロックの中に複数の宣言を記述することができる。

スキル区分	OSS モデルカリキュラムの科目	レベル
セキュリティ分野	4-5 Web アプリケーションに関する知識	基本
習得ポイント	4-5-基-4. DOM, JavaScript	
対応する コースウェア	第3回 Web の表現技術(DOMとJavaScript)	

#### 4-5-基-4. DOM, JavaScript

動きのある Web サイトを作る技術である JavaScript と DHTML について解説する。また、文書構造やデザインの構成を指定するために用いる DOM (Document Object Model) の意義を説明し、実際に動的な変化を可能とするページの作り方を示す。

##### 【学習の要点】

- \* DHTML (Dynamic HTML) は、JavaScript や CSS を使用することで HTML を Web ブラウザ側で動的に変化させ、Web コンテンツに動きをつける技術である。
- \* DOM (Document Object Model) は、HTML や XML 文書をプログラムから操作するためのインターフェースである。DOM によって HTML をオブジェクトとして JavaScript から扱うことができ、ID や要素名を指定して、任意のノードを探索し、ノードの作成や削除をすることができる。



図 4-5-基-4. 動的な Web ページの変更

## 【解説】

### 1) DHTML(Dynamic HTML)

DHTMLとはJavaScriptやCSSを利用することで静的なHTMLに動的な動きをつける技術である。DHTMLを用いることによりWebページに視覚効果や対話性をもたせることができる。

#### \* JavaScriptとは

JavaScriptはスクリプト言語であり、従来は静的な表現しかできなかったWebページに動きや対話性を持たせるために開発された言語である。

JavaScriptを解析するエンジンは、ほとんどの主要なWebブラウザに搭載されているため、Webシステム上でJavaScriptが利用されている頻度は他のブラウザ上で動作する言語より高い。

また、近年JavaScriptを使って非同期に(画面遷移を行わないで)サーバとの通信を行うAJAXという技術が登場したことで一躍脚光を浴びた。

#### \* CSSの動的な変更

JavaScriptでHTMLタグ要素のstyle属性やclass属性を変更する事で動的に装飾やレイアウトの情報を変更する事ができる(図を参照)。また、CSS3を使うとスライドやズームなど多彩な視覚効果をJavaScriptはほぼ使用しないで実現することができる。現状すべての主要なブラウザで利用できるわけではないが、HTML5と並んで今後普及していく技術だと考えられる。

### 2) DOM(Document Object Model)

DOMとはHTMLやXML文書をアプリケーションから操作変更するためのインターフェース(API)であり、W3Cから勧告されている。DOMのもっとも重要な目標は、さまざまな環境(ブラウザ)で使うことができる中立なインターフェースを提供することである。

DOMではXMLおよびHTML文書をノードオブジェクトの階層に構造化する。それに伴い、文書をオブジェクトとしてJavaScriptから操作することができ、IDや要素名を指定して任意のノードを探索し、ノードの作成や削除をすることができる。

#### \* DOMレベル

現在、DOMには3つのレベルが存在する。

##### - Level1

文書の基本的な構造化方法や操作インターフェースを定義したものである。  
XML文書を扱う「Core」とHTML文書を扱う「HTML」の2つのパートからなる。

##### - Level2

level1の「Core」で規定されたものに機能を追加したものである。  
操作対象をより使いやすくするための仕様「Traversal and Range」、スタイルの変更を可能にする「Style」、イベント処理に対応した「Events」などの仕様が追加され、大幅な機能強化が図られた。

##### - Level3

従来の仕様に加え、読み込みと保存を行うための「Load and Save」、文書検証方法の手順を提供する「Validation」などが追加されている。

スキル区分	OSS モデルカリキュラムの科目	レベル
セキュリティ分野	4-5 Web アプリケーションに関する知識	基本
習得ポイント	4-5-基-5. XML/XSL	
対応する コースウェア	第 2 回 Web の表現技術 (HTML と CSS)	

## 4-5-基-5. XML/XSL

構造的なデータを記述するための汎用的な言語である XML (Extensible Markup Language) の概念を説明し、XML の構造や定義、XML により記述されたデータの表示を XSL によりデザインする方法を解説する。

### 【学習の要点】

- \* XML (Extensible Markup Language) は、拡張可能なマークアップ言語のことで、W3C により標準化されている。使用者が独自にタグを定義し、特定用途向けのマークアップ言語を実装することが可能で、XHTML、RSS、SVG などが存在する。
- \* XML 文章は、XML 宣言、DTD (Document Type Definition)、XML インスタンスの 3 つの部分で構成される。
- \* XML インスタンスは、ルート要素を基点としたツリー構造。上位の要素を親要素、下位を子要素と呼ぶ。各要素は要素名の持つタグで、要素の値を囲んでいる。要素は属性を持つことができる。
- \* XSL (Extensible Stylesheet Language) は、XML 文章を加工するのに使われる。

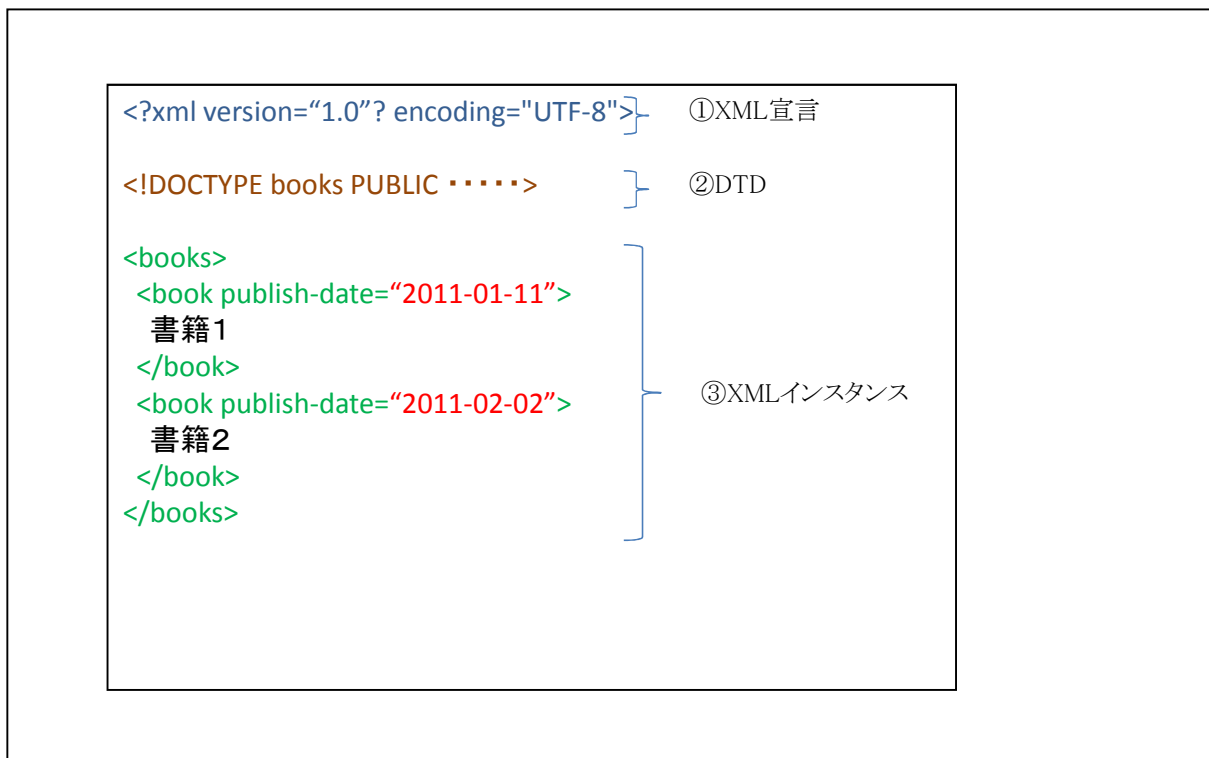


図 4-5-基-5. XML 文章の構造

## 【解説】

### 1) XML(Extensible Markup Language)

XML とは、特定用途で使用するマークアップ言語を作成するための仕様および作成された言語のことを指し、XML 仕様は W3C により標準化されている。

使用者は独自にタグや属性およびそれらの構造を定義する事で、用途に合わせたマークアップ言語を独自に作成することができる。

なお、XML は既に多く普及している技術であり、XML を適用したマークアップ言語としては、XHTML、RSS、SVG などが有名である。

#### \* XML の目的

XML の最も重要な目的は、異なるシステム間のデータや文書の通信・情報交換を容易にする事である。XML 仕様は特定のプログラム言語には依存しておらず、異なる言語で作成されたシステム間でも情報交換を可能とする。

#### \* XML 構文

XML 文章は、XML 宣言、DTD (Document Type Declaration)、XML インスタンスで構成される。XML 宣言は図①のように記述する。

DTD にはこの XML が準拠している DTD を記述する(図②を参照)。なお、DTD は XML が利用できるスキーマ言語で、この XML の妥当検証を行うことを想定していない場合は、記述を省略しても構わない。

XML インスタンス部分には、扱うデータや文書を記述する(図③を参照)。

なお、XML インスタンスは、ルート要素を起点としたタグ要素によるツリー構造を成し、タグ要素は必ず閉じる必要がある。

#### \* XML 関連技術

XML をさまざまな局面で利用できるように、XML 関連技術の規格も多く存在する。

XML の構造を定義するための XML Schema や RELAX NG、XML をアプリケーション側から了するための API である DOM や SAX などが存在する。

また、次項で説明する XSL もその1つと考えられる。

### 2) XSL(Extensible Stylesheet Language)

XSL とは、XML 文章の内容の見栄えを変更することなどを目的に XML を変換する技術である。

XSL を構成する要素としては下記がある。

#### \* XSLT(XSL Transformations)

XML を構造のことなる別の XML などに変換するためのマークアップ言語。

#### \* XPath(XML Path Language)

XML 内の特定の要素(タグ、属性、テキストなど)を検索する技術。

#### \* XSL-FO(XSL Formatting Objects)

人間に理解しやすい形式の文書の組版を記述するためのマークアップ言語。

スキル区分	OSS モデルカリキュラムの科目	レベル
セキュリティ分野	4-5 Web アプリケーションに関する知識	基本
習得ポイント	4-5-基-6. 3層アプリケーション、デザインとロジックの分離	
対応する コースウェア	第7回 Web アプリケーションの仕組み	

#### 4-5-基-6. 3層アプリケーション、デザインとロジックの分離

開発効率を高めるために必須の技法であるデザインとロジックの分離について説明する。さらに一般的な Web アプリケーションの構成である 3 層アプリケーションとは何か、その概要を解説する。

プログラムをモデル(Model)、ビュー(View)、コントローラ(Controller)に分けて設計することで、システム構成の抽象化を高めて説明する際に利用される MVC モデルについて解説する。

##### 【学習の要点】

- \* Web サービスを実現するサーバ構成について説明する。サーバ側は、HTTP サーバ、アプリケーションサーバ、DB サーバなどから構成される。
- \* 「プレゼンテーション層」「アプリケーション層」「データ層」の 3 層について説明する。
- \* 2 層の C/S 型との比較、ロジックをクライアントから切り離したことによる拡張性や保守性の向上について説明する。
- \* MVC モデルとは、プログラムを Model, View, Controller という 3 つのコンポーネントから構成する設計手法。業務ロジックと画面表示を分離させ、プログラムの汎用性、メンテナンス性を高めることができる。

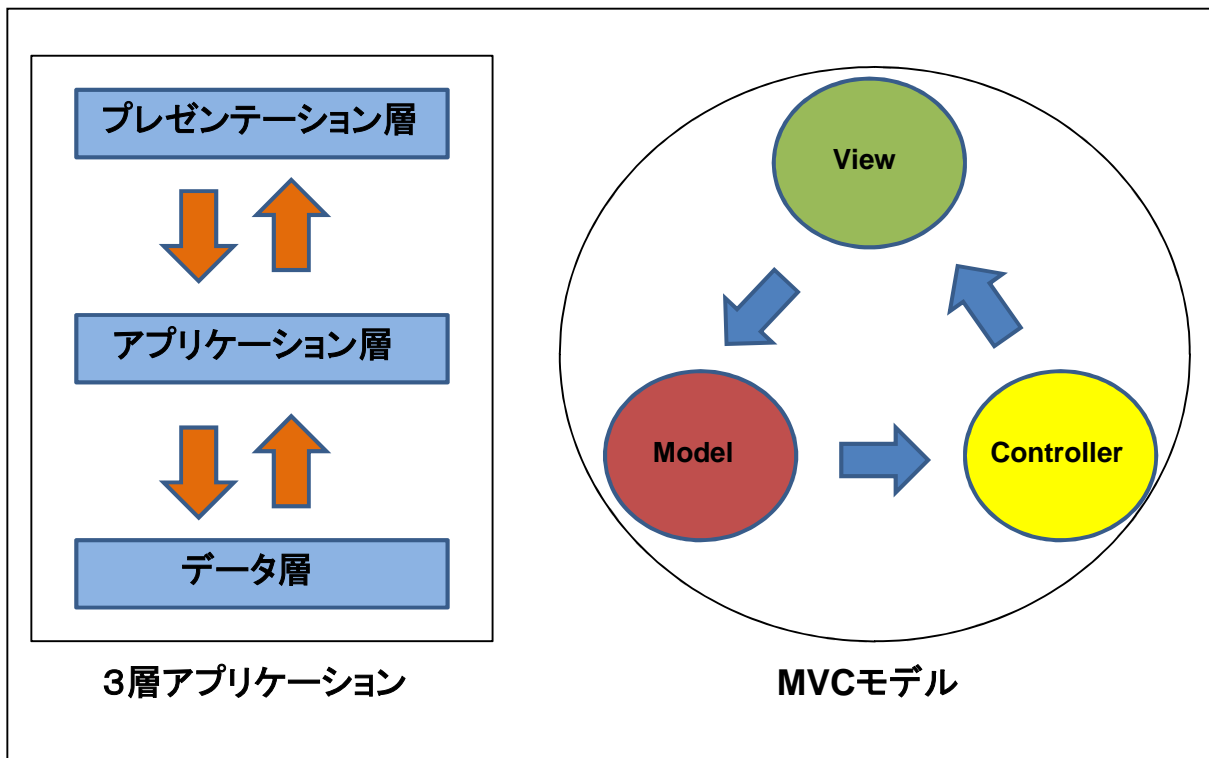


図 4-5-基-6. 3層アプリケーションおよび MVC モデル

## 【解説】

### 1) 3層アプリケーション

Web サービスは通常 HTTP サーバ、アプリケーションサーバ、データベースサーバなど複数のサーバで構成される。

3層アプリケーションとは、Web システムの構成要素であるサーバをプレゼンテーション層、アプリケーション層、データ層の3層に分割し、独立したモジュールとして設計するアプリケーション(アーキテクチャ)である。多くの場合、3層はそれぞれ別のハードウェア、別の OS で動作可能となるように設計され、実際にそのように運用される場合が多い。

メリットとしては、3層を独立したモジュールとして設計することにより相互の依存性が抑えられるため、要求に応じた機能変更や拡張が容易となる。

### 2) 各層の役割と構成要素

#### \* プレゼンテーション層

ブラウザから HTTP リクエストを処理し、必要に応じてアプリケーション層に処理を委譲する。

また、アプリケーション層から処理結果を受け取り、ブラウザに返す。

#### \* アプリケーション層

プレゼンテーション層から要求に応じたビジネスロジックを処理する。必要に応じてデータ層にアクセスし、データの検索、保存などを行う。

#### \* データ層

データを管理し、アプリケーション層に対してデータの検索、抽出、保存などのインターフェースを提供する。

### 3) 2層の C/S 型との比較

C/S 型のアプリケーションでは、ビジネスロジックがすべてクライアント側に記述されており、ビジネスロジックに問題が発生した場合はクライアント側を修正し、すべて再配置する必要があった。

3層アプリケーションであれば、ビジネスロジックの修正が発生してもサーバ側を改修するだけで済むので、システム全体におよぼす影響は少なくなる。また、ビジネスロジックをサーバ側に移動することにより、ネットワークを介した C/S 間での不要なデータのやり取りを軽減する効果もある。

### 4) MVC モデル

MVC モデルでは、プログラムを Model、View、Controller に分割して設計する。一見、3層アプリケーションに似ているが適用分野が異なり、物理的な概念を表す3層アプリケーションに対し、MVC モデルはソフトウェアの設計手法の1つである。

2つのアーキテクチャの違いとしては、3層アーキテクチャではプレゼンテーション層とデータ層が直接やり取りを行うことなく必ずアプリケーション層を介するが、MVC モデルでは各層が相互に連携する。

MVC を適用することにより、分業開発の容易化や相互の仕様変更の影響を軽減する効果がある。

スキル区分	OSS モデルカリキュラムの科目	レベル
セキュリティ分野	4-5 Web アプリケーションに関する知識	基本
習得ポイント	4-5-基-7. スクリプト言語	
対応する コースウェア	第 5 回 アプリケーションの記述言語	

## 4-5-基-7. スクリプト言語

Web アプリケーション開発でしばしば利用される OSS のスクリプト言語を紹介し、その特徴や Web アプリケーション開発特有の使い方、表現方法などを説明する

### 【学習の要点】

- \* スクリプト言語は、インタプリタ型、動的型付けなどの特長をもつスクリプト言語で、C/C++/Java などのコンパイル言語と比較して、プログラムの作成や修正が容易である。
- \* 代表的なスクリプト言語である Perl, PHP, Python, Ruby について説明する。

PHP	C言語
<pre>&lt;html &gt; &lt;head&gt; &lt;title&gt;PHP&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;p&gt;今日は、&lt;?php echo date("Y/m/d"); ?&gt; です。 &lt;/p&gt; &lt;/body&gt; &lt;/html&gt;</pre>	<pre>#include &lt;stdio.h&gt; #include &lt;time.h&gt;  void main() {      time_t now;     char *strNow;      now = time(NULL);     strNow = ctime(&amp;now);      printf("Content-type: text/html\n");     printf("\n");     printf("&lt;html&gt;\n");     printf("&lt;head&gt;\n");     printf("&lt;title&gt;C言語&lt;/title&gt; \n");     printf("&lt;body&gt;\n");     printf("&lt;p&gt;今日は、%sです。&lt;/p&gt; \n", str);     printf("&lt;/body&gt;\n");     printf("&lt;/html&gt;\n"); }</pre>
<p><b>PHPの特徴</b></p> <ul style="list-style-type: none"> <li>・少ないソース記述量で開発が可能</li> <li>・習得が容易</li> <li>・実行前のコンパイルが不要</li> </ul> <p style="text-align: center;">↓</p> <p style="text-align: center;"><b>開発効率が良い</b></p>	

図 4-5-基-7. PHP と C 言語の比較



## 【解説】

### 1) スクリプト言語とは

スクリプト言語とは、一般的にコンパイル言語である C/C++ や Java などの言語と比較して習得が容易で手軽に扱える言語群の総称である。

近年 Web アプリケーションの開発が増加する中、プログラマの Web アプリケーション開発や修正に対する負担を軽くするという意味で「軽量」という言葉が使われており、高速に動作したりメモリ消費量が少ないなどの性能的なことを意味するものではない。

言語群の特徴としては、インタプリタ型でコンパイルが不要、動的型付けが可能、などがあり、一般的にコンパイル型言語より開発効率が良いと言われている。

### 2) Perl

Perl は 1987 年に初期リリースされたスクリプト言語で、スクリプト言語の初期の代表的な言語である。言語的な大きな特徴としては強力な正規表現を利用できる点が挙げられ、正規表現以外にもテキスト操作の関数が豊富に用意されている。この点から Perl はテキスト処理を記述しやすい言語であると言え、CGI スクリプトやシステム管理系のプログラムを記述する際に広く利用されている。

### 3) PHP

PHP は 1995 年に誕生したスクリプト言語で、動的な Web ページを簡単に作成するために開発された言語である。言語の大きな特徴としては埋め込みプログラムが可能で、HTML 内に `<?php ~?>` を記述することで PHP コードを埋め込むことができる。

PHP は元来小規模 Web アプリケーション向けに開発された言語だが、普及に伴いオブジェクト指向に関する言語仕様の拡張が行われるなど、近年では大規模 Web アプリケーションにも多く採用されている。

### 4) Python

Python は 1990 年代初頭に誕生したスクリプト言語で、特に欧米で広く利用されている。多くの Linux ディストリビューションで採用されており、OS をインストールした時点で利用が可能となる。また、その優れた言語仕様や機能から Web アプリケーション、GUI アプリケーション、3D モデリング、数式処理など幅広い分野で利用されている。

### 5) Ruby

Ruby は 1993 年に日本で誕生したスクリプト言語で、Perl の代替として利用される事を意識して開発されている。Ruby は数値型も含めすべての型をオブジェクトとして扱うため、純粋なオブジェクト指向言語として考えられる。

近年、従来より少ないコードの記述で Web アプリケーション開発を可能とするフレームワークの「Ruby on Rails」の登場が話題となり、一躍脚光を浴びた。

スキル区分	OSS モデルカリキュラムの科目	レベル
セキュリティ分野	4-5 Web アプリケーションに関する知識	基本
習得ポイント	4-5-基-8. 開発フレームワーク	
対応する コースウェア	第7回 Web アプリケーションの仕組み 第8回 PHP を用いた Web アプリケーション開発の概要	

## 4-5-基-8. 開発フレームワーク

Web アプリケーションは通常、スクラッチから開発されることは稀であり、数多くの開発フレームワークが利用される。ここでは開発フレームワークとは何か、開発フレームワークの種類や OSS による代表的な実装を紹介する。

### 【学習の要点】

- \* 開発フレームワークとは、データベースアクセスや、セッション管理など、Web アプリケーション開発に共通する機能を提供し、効率的な開発を助ける基盤ソフトウェアである。
- \* 代表的な開発フレームワークである、Ruby on Rails(Ruby), Django(Python), symfony(PHP), Catalyst(Perl)について概要を説明する。

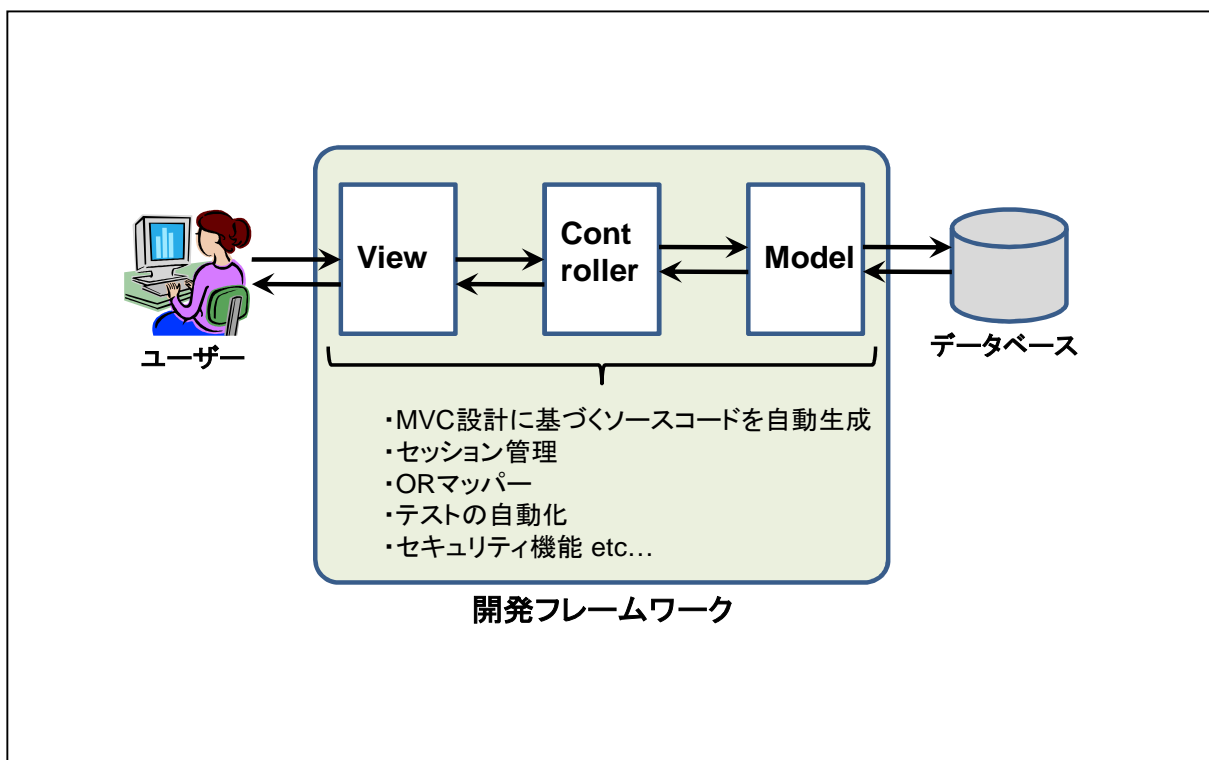


図 4-5-基-8. 開発フレームワークが提供する機能例

## 【解説】

### 1) 開発フレームワークとは

近年、Web アプリケーション開発は大規模化する反面、開発にかけることができる工数は短くなる傾向にある。必然的に開発に関わる人数も増える傾向にあり、重複した機能を作成してしまったり、個別に作成した機能同士が正しく結合できなかつたりなどと開発時に多くの問題が発生する。

このような背景から、Web 開発で共通する機能を提供する開発フレームワークが注目され、近年開発の効率化や品質の向上のために多く利用されている。

#### \* メリット

Web アプリケーション開発で共通するデータベースアクセスやセッション管理、MVC 分離など機能がフレームワーク側で提供されており、開発者はビジネスロジックにのみ集中して開発を行える。

また、フレームワークの制約に従い開発を行うため、開発者のスキルレベルに依存しない品質が確保できる。

#### \* デメリット

開発フレームワークの利点はルール化により自由度を制約し、開発効率や品質を向上する事なので、実現したい機能によっては開発フレームワークの制約により困難になったり、実現不可能になったりする場合がある。要件に適した開発フレームワークを選定することが重要である。

### 2) 代表的な開発フレームワーク

Web アプリケーション開発において採用されるフレームワークは MVC に基づくフルスタックフレームワークが多く、下記の代表的な開発フレームワークもフルスタックな機能を有する。なお、フルスタックフレームワークとは画面作成、データベース連携、セキュリティ対応など Web 開発に必要な機能作成を全体的にわたって支援するフレームワークである。

#### \* Ruby on Rails

Ruby on Rails は Ruby で記述された Web アプリケーションフレームワーク。「RoR」「Rails」などとも呼ばれている。

特徴としては、他の開発フレームワークより少ない記述で開発できるように設計されていることが挙げられる。特にデータベースを使った Web アプリケーション開発では他言語を使った開発よりも高い生産性を誇る。

「同じことを繰り返さない (DRY – Don't Repeat Yourself)」「設定より規約 (CoC – Convention over Configuration)」などの理念は有名である。

#### \* symfony

PHP の Web アプリケーションフレームワーク。「RoR」を意識して作成されている。

#### \* Catalyst

Perl のフレームワーク。CPAN モジュールを組み合わせることで開発効率の向上を実現。

#### \* Django

Python のフレームワーク。Google App Engine でもサポートされている。

スキル区分	OSS モデルカリキュラムの科目	レベル
セキュリティ分野	4-5 Web アプリケーションに関する知識	基本
習得ポイント	4-5-基-9. DB インターフェース	
対応する コースウェア	第 10 回 ファイルとデータベースへのアクセス	

## 4-5-基-9. DB インターフェース

アプリケーションからDBの操作を行う際に、SQLによるアクセスを隠蔽し生産性を高めるORマッパーなど、データベース操作のインターフェースについて説明する。

### 【学習の要点】

- \* OR マッパーとは、オブジェクト指向プログラミングで扱われるオブジェクトと、リレーショナルデータベースを結びつける仕組み。レコードの操作をオブジェクトのメソッド呼び出しで扱えるよう、SQL によるDB操作を隠蔽する。
- \* オブジェクトから透過的にDB操作できるため、SQLベースのDBアクセス処理よりもスマートで、柔軟なプログラムを記述できる。

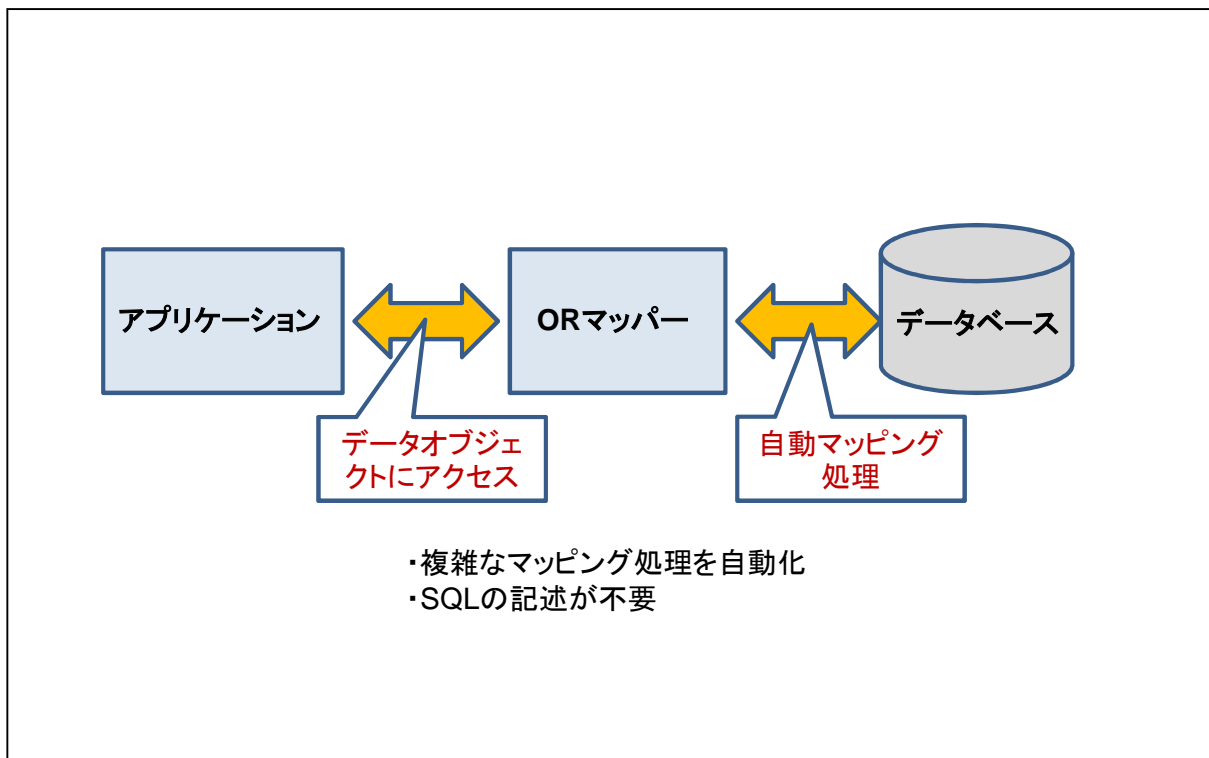


図 4-5-基-9. OR マッパーの役割

## 【解説】

### 1) OR マッパーとは

OR マッパーとはデータベースに格納されているデータとオブジェクト指向プログラミングで扱われるオブジェクトを結びつける技術およびそれを実現するフレームワークである。

近年の Web アプリケーション開発においてデータベース連携はほぼ必須であり、データ操作に関する開発効率化や品質向上のために OR マッパーは大規模 Web 開発などで多く採用されている。

### 2) 背景

Web アプリケーション開発で多く利用されるオブジェクト指向プログラミングではデータを現実世界のモデルに即した形で定義するのに対し、リレーショナルデータベースでは正規化などを基準とした数学的な考え方でモデルが設計されるため、データを結びつける際に問題(インバーダンスミスマッチ)が発生する。

通常はこの問題の解決のため、オブジェクト形式のデータとテーブル形式のデータを都度結びつける作業を行うが、正確さを要求されるこのマッピング作業は人為的なミスが発生しやすい。

OR マッパーは、この作業を軽減および自動化する事をもっとも重要な目的として作成された。

### 3) メリット

#### \* オブジェクトとDBのマッピング自動化

読み込み、保存などのデータベースに関する操作をオブジェクトのメソッド越しに行うことができ、複雑なデータベース操作に関するプログラム記述を簡略化することができる。

#### \* SQL 不要

SQL 記述なしでデータベースの操作を行うことが可能なため、SQL に関する深い理解が不要である。

#### \* SQL インジェクション防止

SQL 発行はフレームワークに隠蔽されるため、SQL インジェクションによるセキュリティホールを意識する必要がない。

### 4) デメリット

#### \* パフォーマンス

SQL を利用してデータベースを直接操作するより動作が遅くなる傾向がある。また SQL の発行が隠蔽されているため、SQL 記述の最適化によるパフォーマンスチューニングが困難。

### 5) 代表的な OR マッパーフレームワーク

- \* Hibernate (Java フレームワーク)
- \* Active Record (Ruby on Rails 付属)

スキル区分	OSS モデルカリキュラムの科目	レベル
セキュリティ分野	4-5 Web アプリケーションに関する知識	基本
習得ポイント	4-5-基-10. セッション管理	
対応する コースウェア	第 6 回 クッキーとセッション	

## 4-5-基-10. セッション管理

セッション管理の必要性について触れ、クッキーを用いたセッション管理やセッション管理を乱す要素(戻るボタン)の回避方法など、Web アプリケーション特有のセッション管理方法について解説する。

### 【学習の要点】

- \* HTTP は一回のリクエストとレスポンスで通信が完結するステートレスなプロトコルである。ログインの状態や、ショッピングサイトでのカートの状態などを複数の画面に渡って引き継ぎたい場合には、COOKIE を利用したセッション管理を行う。
- \* COOKIE はサーバによって任意のデータを設定しブラウザへ送られる。ブラウザは再びリクエストを発行する際に、ローカルに保存した COOKIE の情報も一緒にサーバへ送信する。これにより、複数のリクエスト/レスポンスにまたがって状態を管理することができる。
- \* セッションは、ログインの状態や、ショッピングサイトでカートの状態などの管理に影響するので、セッション情報が盗聴されないように注意する。また、セッションに有効期限を正しく設定する。

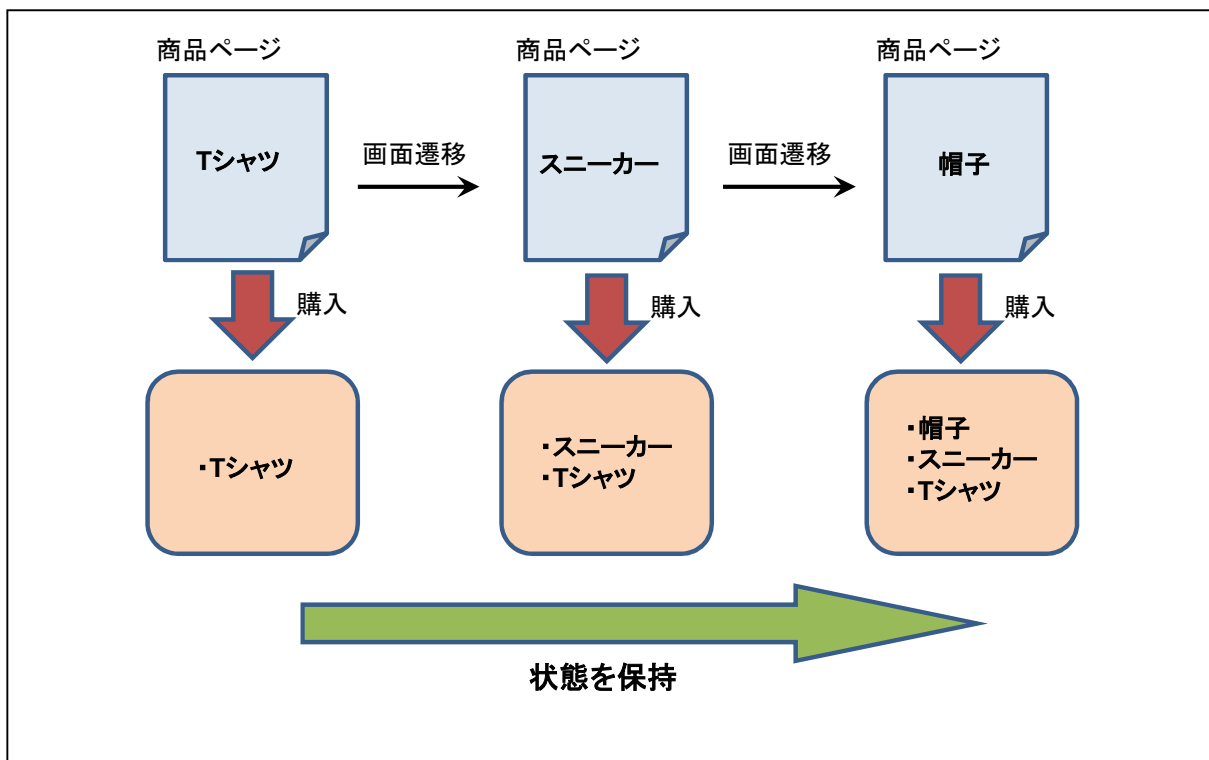


図 4-5-基-10. EC サイトにおけるセッション管理

## 【解説】

### 1) セッション管理の必要性

近年、Web にアクセスして EC サイト上で商品を購入したり、銀行口座から振り込みを行うなどの行為は一般的である。ただ、Web ブラウザと Web サーバの通信に使われる HTTP プロトコルは一回のリクエストとレスポンスで通信が完了するステートレスなプロトコルのため、プロトコルの機能だけでは次回アクセス時に購入途中の商品の情報や口座にログインしている状態などの情報を引き継ぐことができない。

この問題を解決するためにセッションという概念が存在する。セッションとはユーザ特定のための仕組みであり、特定ユーザに紐づくユーザ情報を管理する機能をセッション管理と呼ぶ。

### 2) COOKIE

COOKIE とは Web サーバが Web ブラウザに一時的にデータを保存させる仕組みで、そのプロトコルおよび保存された情報を指す。W3C などの団体が標準化されているわけではないが、代表的なブラウザはこのプロトコルをサポートしており、実質業界標準である。このプロトコルを利用する事により、本来ステートレスな HTTP プロトコル上でステートフルな機能が実現でき、Web サーバと Web ブラウザ間の情報共有が可能となる。

この COOKIE の機能を利用し、セッション管理は実現される。

### 3) セッション管理の仕組み

セッション管理は下記の Web サーバと Web ブラウザのやりとりで実現される。

- \* Web サーバは、Web ブラウザに対するレスポンスの際にブラウザを特定するための識別子である「Session ID」を COOKIE として送る。
- \* Web ブラウザは、次回リクエストの際に、前回 Web サーバから送られた COOKIE の情報を送る。
- \* Web サーバは、送られた COOKIE に含まれる「Session ID」を元にブラウザ(ユーザ)を特定し、サーバ側に保持しているユーザ情報を特定する。
- \* アプリケーションは特定されたユーザ情報を利用する。

### 4) セッション管理の安全性

セッションにはログイン状態など認証情報とほぼ等価である機密情報が含まれる可能性があるため、盗聴防止などを考慮して慎重に扱う必要がある。

できれば、セッション管理を必要とするアプリケーションや機能は HTTP over SSL などの暗号化されたプロトコル上で通信を行うのが望ましい。

また、アプリケーション機能に適したセッション有効期間を設定することで、乗っ取りやなりすまし攻撃のリスクを軽減することも重要である。