

# 6-1

## Webサイトを実現する技術の基礎

Webサイトは、数多くの技術のうえに成り立っており、Webサイトにおける表現も機能もすべて、これらの技術を用いて実現される。また、これらの技術は日々刻々と進化し続けている。ここでは、まずそうした技術を俯瞰するため、全体像の概観を行う。

### 6-1-1 さまざまな技術

Webサイトとは、さまざまなIT技術の集合体であり、ハードウェア、回線、セキュリティ、データベース、各種言語、さまざまな標準仕様など、そこで利用されている技術は分野も内容も多岐にわたっている。1人の技術者がこれらすべての技術に精通するのは困難であり、各分野を担当するそれぞれの専門家がチームを組んで制作を行うのが一般的である。

しかし、Webサイト制作者として仕事をしていくためには、自分の専門以外の技術についても、どのような技術が存在し、それがどのような役割を担っているのか、その技術を使って何ができるのか、などの事柄を把握している必要がある。

たとえば、Webのデザインやインターフェースを考えるにあたって、それが技術的に実現可能であるか理解していなければ、そもそもデザインをすること自体が不可能である。せっかく魅力的なデザインをつくりあげたとしても、その実現が技術的に不可能であっては意味がない。

こうした技術が、日々刻々と変化し続けていることもWebサイト制作の世界の特徴である。技術の変化は、新しい技術の出現、既存技術の改良、規格の改訂などのかたちで現れることが多いが、それとは別に、既存の技術の新しい使い方、あるいは似たようなものだが既存技術の組み合わせによる新しい手法の出現なども重要である。この代表的な例がAjax<sup>\*1</sup>といえる。

技術の変化は、新しいビジネスやサービスの提供を可能にしたり、新しい表現をもたらしたり、ときにはWebサイトのあり方そのものに大きな影響を与えることもある。そのため、Webサイトの運営者はもちろん、Webサイトの制作を担う制作者もまた、技術動向に関する情報をつねに注視し続ける必要がある。

\*1 Ajaxについては、6-3-6を参照のこと。

### 6-1-2 Webサイトの基本構成

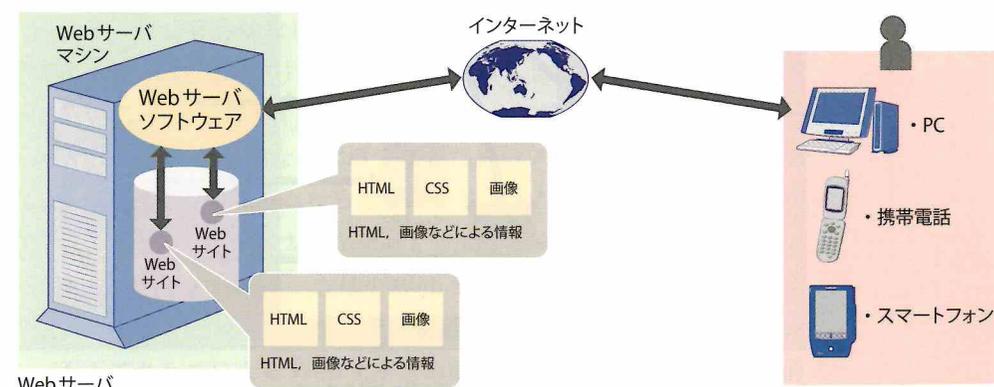
インターネットとは、世界中に展開している複数のコンピュータネットワークどうしを相互接続した巨大ネットワークの呼称である。インターネットではさまざまな通信プロトコルが用いられているが、なかでもTCP/IPはインターネットを代表する基本プロトコルとして重要である。

インターネットはただコンピュータネットワークどうしを接続するだけでなく、さまざまなサービスを提供している。そうしたサービスの1つが、HTTP(Hypertext Transfer Protocol)というプロトコルをベースとしたWWW(World Wide Web)サービスとよばれるハイパーテキストシステムである。WWWサービスを利用するためのソフトウェアはWWWクライアントとよばれるが、そのなかでもとくにWebページの閲覧を目的に作成されたソフトウェアをWebブラウザという。これらWWWクライアントからの要求を受けて情報の提供を行うシステムがWWWサーバまたはWebサーバである。

Webブラウザは、各OS向けにさまざまなものが提供されており、ユーザは自分の目的や好みに合ったWebブラウザを自由に選ぶことができる。しかし、こうしたWebブラウザの多様さが、Webブラウザ間の互換性問題の原因となっていることも事実である。

Webサーバは、物理的なコンピュータとしてのWebサーバマシンと、その上で実際の処理を行うWebサーバソフトウェアとで構成される。Webサーバマシンとして用いられるハードウェアには、汎用機とよばれる大型コンピュータや、Webサーバ専用設計されたコンピュータ、一般的なPCなど、さまざまなものがある。Webサーバソフトウェアも同様に、多様なものが用意されている。

Webサイトとは、このWebサーバ上に展開された情報群の総称といえることができる。図6.1は、WWWサービスの全体像とWebサイトを示した概念図である。



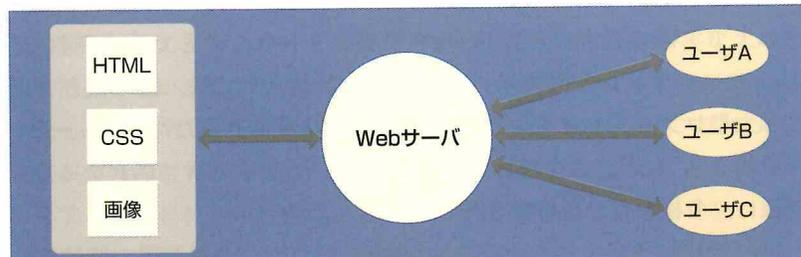
■図6.1—WWWサービスの全体像とWebサイト

### 6-1-3 静的コンテンツと動的コンテンツ

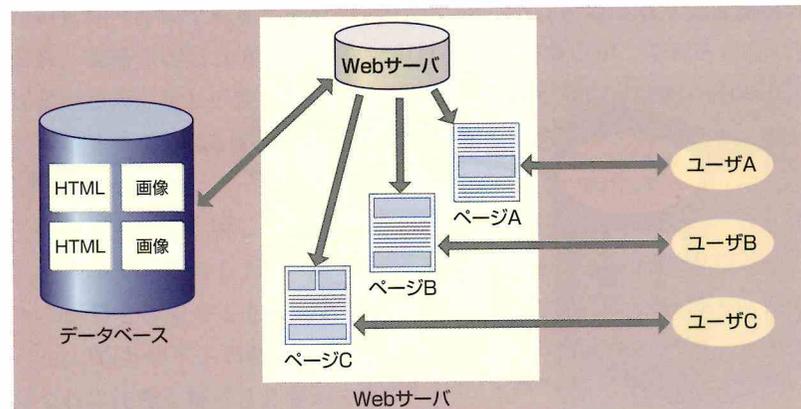
Webサイトのコンテンツは静的コンテンツと動的コンテンツに分類することができる。静的コンテンツとは、WWWクライアントからの要求を受けると、あらかじめ用意された情報を送信するものである。それに対して動的コンテンツとは、WWWクライアントからの要求内容やそのときの条件などに応じてコンテンツを生成、組み合わせたうえで送信するものである。多くの動的コンテンツは、データベースと連動するWebアプリケーション<sup>\*2</sup>によって生成されるのが一般的である(図6.2)。

たとえば、ECサイトなどで各ユーザごとに異なるおすすめ商品を表示する、といった機能は動的コンテンツの1つの例である。これは、各ユーザの購入履歴をデータベースに記録しておき、別のデータベースに記録された商品データのなかから類似の商品を検索し、コンテンツとして表示することで実現されている。

このように、データベースや各種Webアプリケーションなど複数の技術の組み合わせで成り立つ動的コンテンツは、Webサイトにおける情報提供のあり方の1つとしてきわめて重要なものであり、インターネットを利用するビジネスなどには不可欠な要素になりつつある。



[a] 静的コンテンツ  
あらかじめ用意しておいた同じ情報を各ユーザに送信する。



[b] 動的コンテンツ  
さまざまな条件に応じてデータベース内の情報を組み合わせて生成したページAをユーザAに、ページBをユーザBに、ページCをユーザCにそれぞれ送信する。

■図6.2——静的コンテンツと動的コンテンツ

\*2 Webアプリケーションについては、6-3-6を参照のこと。

### 6-1-4 ①側の処理とバックエンド側の処理

Webサイト制作で用いられる技術は、その処理が行われる場所によって、①側とバックエンド<sup>\*4</sup>側に分けることができる。一般に、それぞれの側ではつぎのような作業が行われる。

- ・ ①側の処理の例
  - デザインの作成
  - インタフェースの作成
  - インタラクティブ機能の作成
  - コンテンツの作成
- ・ バックエンド側の処理の例
  - 回線の手配
  - 各種サーバの手配
  - データベースの手配と設定
  - Webアプリケーションの作成

前述の作業は、フロントエンド側をデザイナーやコーダなどが、バックエンド側をシステムエンジニアやCGIプログラマが、それぞれ担当するのが一般的である。しかし、これらの作業は完全に分離して行われることはなく、設計の段階から実作業の終了まで、互いに密接に協議し、連絡を取り合って作業を進める。また、たとえばフロントエンド側の作業においても、高度なプログラミングの作業が必要とされる場合が多く、そうした部分についてはエンジニアが参加することも多い。

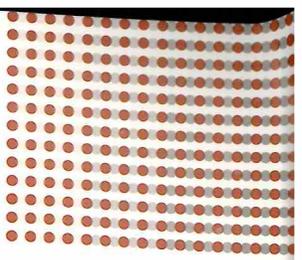
### 6-1-5 技術を超えて考えるべきこと

Webサイトの社会的な重要度が高まるにつれ、「技術的に可能なことを実現する」だけでは受け入れられなくなってきたのも事実である。近年はとくに、個人情報の保護など②の確保や、より多くのユーザがWebサイトを不便なく利用できるための③への配慮などが重視されるようになってきている。また、OSやWebブラウザの違いによる非互換性も大きな問題として残り続けている。

セキュリティの問題は、もはやWebサイトだけに限定できない、企業や公共機関などのネットワーク全体を含めた重要な課題となっている。不正アクセスによるWebサイトの改ざんや機能停止などは企業活動にも実害を及ぼすし、個人情報の漏えいなどは実害だけにとどまらず、運営企業や公共団体の社会的信用を失墜させる重大な問題である。Web

\*3 フロントエンドとは、おもにユーザ側のクライアント(端末)のことを指す。

\*4 バックエンドとは、おもにWebサイト開設者側のサーバのことを指す。



# 6-2

## Webサイト上の機能

さまざまなサービスを提供するWebサイトは、いまや単なる情報発信のみにとどまらない複雑なメディアとなっている。ここでは、そうしたWebサイトが提供するサービスを構成する基本機能を、いくつかのカテゴリに分類整理して解説する。

### 6-2-1 ビジュアルイゼーション機能

Webページにおける情報の視覚化、聴覚化のための機能には、つぎのようなものがある。

#### テキスト表示機能

HTMLで記述されるWebページの基本機能である。通常のテキスト文章または、リンク付きテキストを表示する。

#### 静止画像表示機能

Webページ上に静止画像を表示させる機能である。静止画像データにはさまざまな形式があるが、Webページで使われる代表的な形式にはGIF形式、JPEG形式、PNG形式があり、それぞれの特徴で使い分けられている。

#### レイアウト機能

Webページ上のコンテンツの要素をレイアウト表示する機能である。ページレイアウトは、CSSとよばれる言語を用いて行うことが推奨されている。なお、HTML言語の表組みデータ表示用の<table>タグを用いてレイアウトが行われることも多いが、この手法は推奨されていない。

#### 動画再生機能

Webページ上で動画を再生させる機能である。動画の再生には各動画形式に対応したプラグインが必要な場合もあるが、HTML5に対応したWebブラウザでは追加のソフトウェアなしで動画の再生や停止、早送り・巻き戻しなどの制御が行える。ただし、現在のところHTML5が制御できる動画の形式はWebブラウザによって異なる。また、動画データダウンロードしつつ逐次再生する方式は、ストリーミング配信とよばれる。

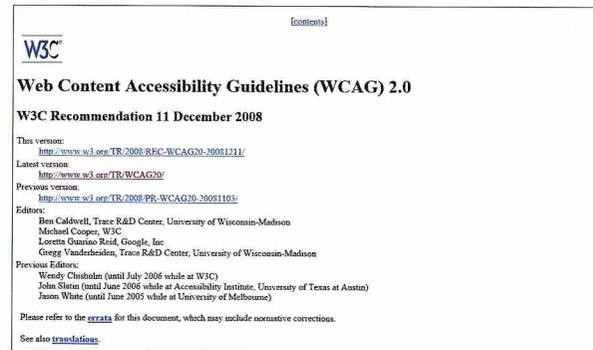
#### 3次元データ描画機能

Webページ上に3次元のコンピュータグラフィックス(CG)や動画を表示させる機能である。Webブラウザ上で3次元のコンピュータグラフィックスを表示させるための標準規格としてWebGLなどが存在する。WebGL対応のWebブラウザであれば、特別なソフトウェアなしで3DCGを表示することができる。

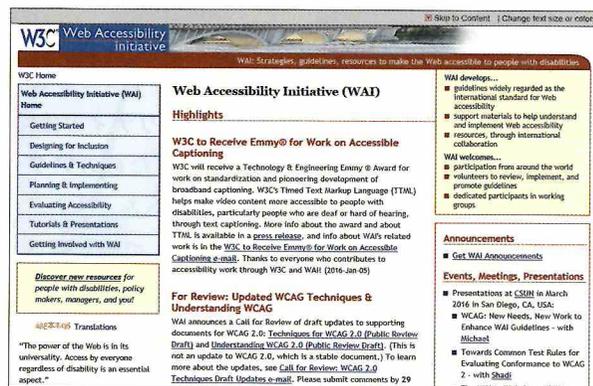
サイトの運営者にはこうした事態を招かないよう、事前の十分な対策が求められている。

アクセシビリティとは、視覚や聴覚、四肢などにハンディキャップをもつユーザや視聴覚能力の衰えた高齢者はもちろん、日本語の読解が不十分な外国人などにも同等の情報に接する機会を確保したデザインをするべき、という考え方である(図6.3)。アクセシビリティの確保は、公共機関のWebサイトはもちろんのこと、企業サイトにおいても社会的責任や機会損失の回避などの面で重視されてきている(図6.4)。しかし、アクセシビリティとデザイン的な表現とを両立させるためには高度な技術が必要であるため、アクセシビリティを考慮したWebサイト制作には広範な知識が求められ、制作工数とコストの増大を招きがちであるなど、課題も多い。

また、Webサイト制作者にとっては、異なるOSやWebブラウザ間の互換性も大きな問題である。現時点では、異なるOSやWebブラウザでまったく同じ機能、デザインを実現するためにはWebサイト制作者に広範かつ高度な知識と豊富な経験が求められることが多く、制作工数とコストの増大を招く一因ともなっている。



■図6.3—W3Cによるアクセシビリティのガイドライン(画像は2016年1月21日取得。Copyright © 2008 W3C® (MIT, ERCIM, Keio), All Rights Reserved. W3C liability, trademark and document use rules apply.)



■図6.4—W3Cではアクセシビリティ実現のためのヒントやアドバイスも多数公開している(画像は2016年1月21日取得。Copyright © 2014 W3C® (MIT, ERCIM, Keio, Beihang) Usage policies apply)

## 音再生機能

Webページ上で音データを再生する機能である。主たるコンテンツとして再生する場合はもちろん、インタフェースの補助機能としての利用や、バックグラウンドミュージックとして利用することも多い。音データの再生にはプラグインが必要な場合もあるが、HTML5に対応したWebブラウザでは追加のソフトウェアなしで音データの再生や停止などの制御が行える。ただし、現在のところHTML5が制御できる音データの形式はWebブラウザによって異なるため注意が必要である。

## 6-2-2 インタラクティブ機能

Webサイトのメディアとしての大きな特徴の1つが、**インタラクティブ性(双方向性)**である。これを実現するための機能には、つぎのようなものがある。

## ハイパーリンク機能

Webサイトにおける最も特徴的な機能である。リンクが設定されているテキストや画像などをクリックすることにより、ユーザはリンク先のWebページなどに移動できる。

## 検索機能

テキストデータを対象に検索を実行する機能が一般的である。ユーザが入力した任意のキーワードや、あらかじめ設定された選択肢を検索条件として検索処理を実行し、関連する情報を一覧表示する。また最近では、ユーザが入力したキーワードに関連した静止画像や動画像を検索する機能も一般的になってきている。

## インタフェース機能

Webサイト内での移動をわかりやすくするため、ユーザにナビゲーションを提供する機能である。多くの場合、**ハイパーリンク**の一部をメニューやボタンなどのかたちで表示することで実現される。また、**DHTML**やFlashなどを用いて、インタラクティブ性のあるナビゲーションを提供することも多い。この場合、マウスの操作に合わせてサブメニューを表示したり、説明を表示するといった機能を実装するのが一般的である。

## コミュニケーション機能

ユーザどうし、あるいはユーザと運営者などが双方向のコミュニケーションを行う機能である。**掲示板**など、任意の時間に各ユーザが書き込み/閲覧を行うタイプのものと、**チャット**などユーザどうしがリアルタイムにコミュニケーションを行うものに大別される。また、カメラとマイクを利用して映像や音声の**双方向通信**を行う機能も普及してきている。

## メール送信機能

会員登録や商品購入など、ユーザが行ったアクションに対して、内容の確認メールなどを自動的に対象となるユーザに送信する機能である。

## 6-2-3 データハンドリング機能

現在のWebサイトでは、データハンドリングによりさまざまな機能が実装されている。ここでは、通常よく利用されている機能として、代表的なものを紹介する。

## データ連携機能

通常Webサイトは複数のサーバによって構成されており、個々のサーバはさまざまなサーバとデータ連携することができる。たとえば、ECサイトにおけるクレジットカード決済などにおいては、ECサイトのサーバがクレジットカード会社のデータベースサーバへカード情報などを照会してから、決済手続きを行っている。

## ユーザ管理機能

個々のユーザごとに**ID**、**パスワード**などの登録処理を加えることで、ユーザ管理機能を実装することができる。ECサイトの多くは、**アカウント管理機能**を実装することで会員サービスを提供しており、個別会員の購入履歴や会員情報の管理機能などをもっている。

## コンテンツ生成・管理機能

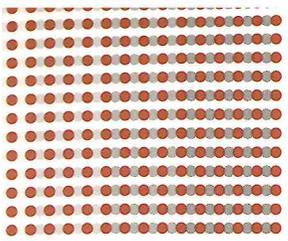
Webサーバ上へコンテンツの**ハンドリング機能**を実装することで、Webサイトのコンテンツを動的に生成することができる。たとえば、**パーソナライゼーション機能**とは、ユーザ管理機能より取得したアカウント情報でユーザの**プロフィール**を判別し、あらかじめ実装された**アルゴリズム**により動的にユーザに適したコンテンツを自動生成して提供する機能である。

## セキュリティ機能

**SSL**などのプロトコルを利用することにより、データを**暗号化**して送受信する機能である。Webサイト上の商取引などが増えていることから、**個人情報の保護**は、Webサイトが備えるべき重要な機能の1つとなっている。

## アクセスログ解析機能

Webサーバには、**アクセスログ**としてユーザが各ファイルへアクセスした日時やアクセス元の**IPアドレス**、訪問者のPCのOSタイプやWebブラウザの種類、処理にかかった時間などが記録される。これらのデータを解析することで、**エラー処理**の特定やWebサイトの効率化の指針、マーケティングデータの取得などが行える。



# 6-3

## Webサイト制作に用いられる言語

Webサイトを実現するための言語には多種多様なものが存在し、それぞれに特徴や役割がある。Webサイト制作に関わる者は、これらの言語の特徴や機能、限界を十分に理解し、使いこなすことが求められる。ここでは、そうした言語について解説する。

### 6-3-1 言語のもつ役割

どんなに優れたコンテンツやデザインがあっても、それを正しくWebブラウザに表示できなければ情報の発信はできない。どのような情報を、どのようなかたちで表示すればよいのか。それをWebブラウザに対して具体的に指示するのがHTMLやCSSのような言語である。Webサイト制作者はそれらの言語の特徴や機能、限界をきちんと理解し、場合によっては自ら記述を行えなくてはならない。

各言語の学習にあたっては、その言語がどのような用途に用いられるのか、その言語を使ってどのようなことができるのか、それはWebサイト上のどの部分を実現するものなのか、といったことを把握することが重要である。

Webサイト制作の現場で最も頻繁に用いられる言語は、HTML CSS, JavaScriptであるが、これらの言語は大まかにつぎのような役割を担っている。

• HTML	情報の記述および情報構造、文書構造の定義を行う。
• CSS	表示する情報の体裁を記述する。
• JavaScript	インタラクティブな機能や各種自動処理などを実装する。

さらに実際の制作現場においては、各言語の限界や問題点を認識することも実務上必須となってくる。各言語には、それぞれ標準化団体によって策定された標準仕様が存在する(図6.5)。しかし、各Webブラウザはそれら標準仕様にはない独自の機能拡張を行っていることが多く、それらの独自機能はほかのWebブラウザでは使用できない場合が多い。言語を習得していく過程では、それぞれの機能が標準仕様に基づいたものであるのか、特定のWebブラウザに固有の独自機能であるのかとい

う点の把握にも注意を払わなければならない。

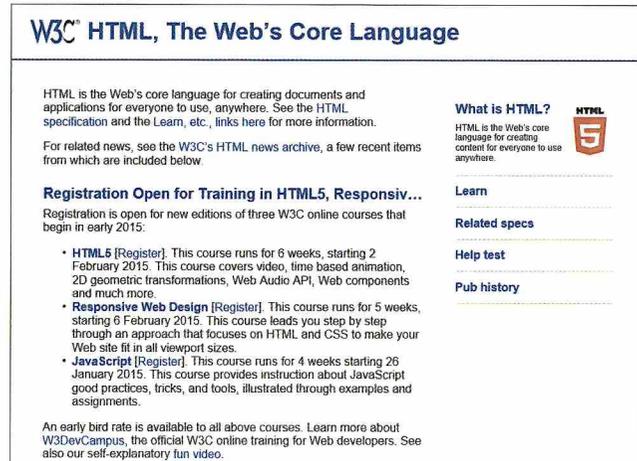


■図6.5——Webサイト制作に関わる各種標準仕様を定めるW3CのWebサイト(画像は2016年1月21日取得。Copyright © 2016 W3C® (MIT, ERCIM, Keio, Beihang) Usage policies apply.)

\*5 W3Cとは、World Wide Web Consortiumの略。WWWにおいて使用されている技術の標準化を行う国際団体のこと。

### 6-3-2 HTML

HTML (HyperText Markup Language) は、ハイパーテキスト機能を備えたマークアップ言語であり、WWWにおいて情報発信を行うために用いられる。いまなおWebサイト制作において中核をなす言語といえる(図6.6)。Webサイト制作におけるHTMLのおもな役割は、発信する情報の記述と、情報どうしの関連付け、文書構造の定義、およびユーザーインタフェースの作成である。



■図6.6——W3CによるHTMLの標準仕様や各種情報(画像は2016年1月21日取得。Copyright © 2016 W3C® (MIT, ERCIM, Keio, Beihang) Usage policies apply.)

### [1] 発信する情報の記述

文字情報はもちろん、表、写真やイラストなどの静止画像、動画や音声データ、Flashなど、Webサイト上で提供される情報のほとんどはHTMLによって記述されるか、または表示の指示が行われる。

表については、HTMLを用いて内容と構造を記述し、CSSを用いて体裁を記述することができる。

静止画像は、GIF、JPEG、PNGなどのビットマップ画像や、SVGなどのベクタ画像の表示を行える。また、JavaScriptなどと連携してアニメーションを作成・表示することも可能である。

動画と音声データについても、再生や停止、巻き戻し／早送りなどの基本的な操作を行うためのインタフェースをHTMLのみで作成することができ、JavaScriptと連携することで、より高度な操作も可能となる。また動画の上に文字や静止画像を配置することもできるため、たとえばユーザーが選択した言語でテロップや字幕を入れる、といったことも可能である。

### [2] 情報どうしの関連付け

情報どうしの関連付けは、**ハイパーリンク**とよばれる機能によって実現されている。ハイパーリンクにはつぎの3種類がある。

- ・同一Webサイト内のほかのHTMLファイルへのリンク。
- ・同一HTMLファイル内のほかの箇所へのリンク。
- ・ほかのWebサイト内のHTMLファイルへのリンク。

こうしたハイパーリンクによって関連付けられた情報のことを**ハイパーテキスト**とよんでおり、リンクをたどりながら関連する情報を容易に閲覧できる。これは、Webサイトというメディアがもつ大きな特徴の1つとなっている。

### [3] 文書構造の定義

文書は情報の主体である本文と、文書の内容を要約したタイトル、各章や段落を表す見出しなどといった要素から成り立つ構造をもっている。この文書構造を定義し、Webブラウザがそれぞれの要素をどのように処理／表示すればよいか指示することもHTMLの重要な役割である。一般的に行われているHTMLによる文書構造の定義には、たとえばつぎのようなものがある(図6.7)。

- ・文書のタイトルの指定。
- ・ヘッダおよびフッタ領域の指定。
- ・本文領域の指定。
- ・ナビゲーション領域の指定。
- ・段落の指定。
- ・見出しとその重要度の指定。
- ・画像の指定。
- ・表の作成や要素の指定。
- ・箇条書きの指定。

また、これとは別に、ユーザからの情報入力インタフェースとなる**HTMLフォーム**なども、双方向メディアとしてのWebサイトを実現するための重要な機能である。

前述のように、HTMLはWebサイト制作の中核的存在ではあるが、いくつかの問題点が存在するのも事実である。とくに、Webブラウザ間の互換性の問題と、Webサイトに求められる機能との乖離の2つは、Webサイト制作に関わる際に必ず意識すべき問題である。



■図6.7—HTMLによって定義された文章構造に従ってWebブラウザはWebページを表示する

WebサイトのようなHTMLで記述された文書は、メニューやナビゲーションといったインタラクティブな要素をもつ点が書籍のような通常の文書と異なる。そのため、HTMLには文章構造の一部としてナビゲーション項目を指定するための要素も取り入れられている。

### [4] ユーザインタフェースの作成

HTMLを用いた文書の重要な特徴の1つは、ユーザとの対話を行う双方向メディアであるという点である。HTMLにはそのための機能がいくつか用意されている。

1つは、ハイパーリンクを用いたインタフェースであり、メニューやナビゲーションというかたちでユーザに提供される。ユーザはこれらをクリックすることで新たなページを表示したり、別のメニューを開くなどのアクションを行うことができる。

2つ目はユーザからのさまざまな情報入力インタフェースとなるHTMLフォームである。文章や数値などの情報はもちろん、処理続行の許可を求める「はい/いいえ」ボタンや選択肢のなかから特定の候補を選んでもらうといった情報入力はHTMLフォームを用いて行われる。

ユーザインタフェースをより使いやすいものにするための機能も提供されている。たとえばマウスの動きに応じて色を変えることでクリック可能であることを示したり、機能の説明を表示したり、別のインタフェースを表示したりといった動きを与えることは、ユーザに使用上のヒントや利便性を与えるものとして重視されている。

また、Webページ上において、あるいはWebページとデスクトップ間で要素をドラッグ&ドロップするための機能も提供されており、HTMLは単なる文書記述言語の域を超えてデスクトップアプリケーション作成のための言語としても用いられている。

### 6-3-3 ①

HTMLが情報の記述や文書構造の定義を行うという役割を担う一方で、①はWebページの体裁を記述するという役割をもっている(図6.8)。CSSは、Webサイト側、Webブラウザ側、ユーザ側のそれぞれで定義でき、各定義の効果を積み重ねる(カスケードさせる)ことができるため、カスケードスタイルシートという名前がつけられている。



■ 図6.8——W3CによるCSSの標準仕様や各種情報(画像は2016年1月21日取得。Bert Bos, style activity lead Copyright © 1994-2016 W3C®)

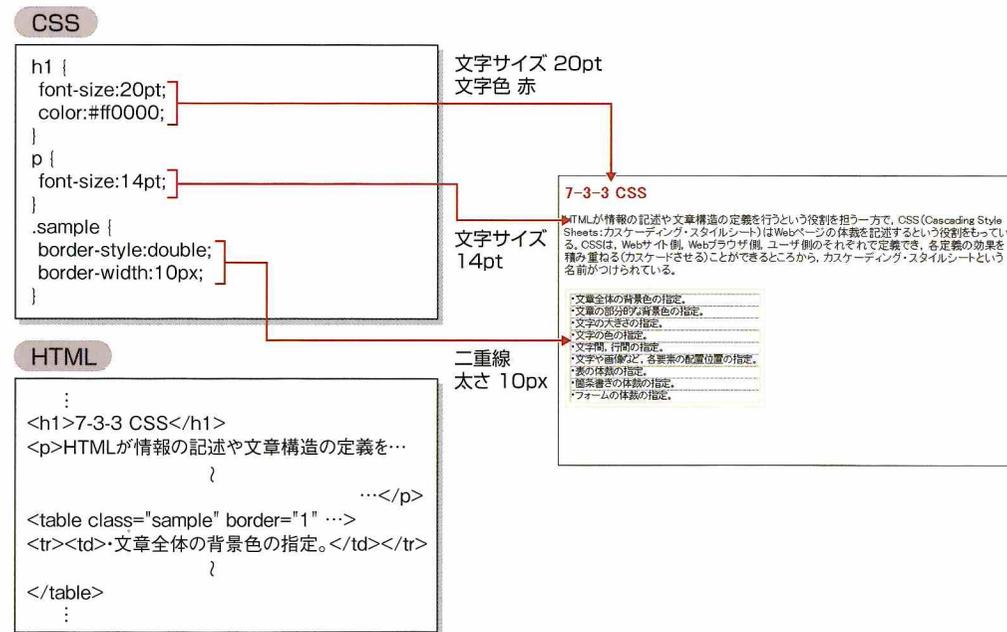
①は、HTMLで記述された情報をどのように整形し修飾するかという体裁の記述を行う。これにはたとえば、つぎのようなものがある。

- ・ 文書全体の背景色の指定。
- ・ 文書の部分的な背景色の指定。
- ・ 文字の大きさの指定。
- ・ 文字の色の指定。
- ・ 文字間、行間の指定。
- ・ 文字や画像など、各要素の配置位置の指定。
- ・ 表の体裁の指定。
- ・ 箇条書きの体裁の指定。
- ・ フォームの体裁の指定。

こうした体裁の指定は、HTML内などで定義した文書構造に関連付けるかたちで行うことができる。

たとえば、本文要素と見出し要素、それに表要素という3つの文書構造要素で構成されたHTMLがあったとする。①内に、本文要素の文字サイズ、見出し要素の文字サイズと色、表要素の枠線の種類や太さといった体裁の指定が記述してあった場合、これらのHTMLとCSSを読み込んだWebブラウザは、HTML内の各文書構造要素を、①内の体裁定義で指定されたとおりに整形し表示する(図6.9)。もちろん、こうした文書構造を基準にした関連付けではなく、任意の文字や表などの体裁を指定することも可能である。

①を利用するうえでもう1つ重要なことは、記述する場所である。これには以下のふたつがある。



■ 図6.9——HTMLとCSSで構成されたWebページ

### ①HTMLファイル内に直接記述する

HTMLファイル内にCSSを直接記述することが可能である。ただし、CSS本来の趣旨からすれば、手軽にCSSを利用するための便宜的手法ともいえる。

### ②別ファイルに記述する

HTMLとは別ファイルにCSSを記述する方法で、CSS本来の趣旨に合った使い方である。HTMLファイルにCSSファイルを取り込む指示を記述することで、HTMLファイルとCSSファイルの関連付けを行う。

このようにCSSとHTMLを別のファイルに分けて記述することは、**文書構造と体裁の分離**という、Webサイト制作における重要な概念を実現する手法の1つである。

HTMLファイル内にCSSを直接記述している、あるいはHTMLそのものを使って文書の体裁を記述しているWebサイトの場合、このWebサイト全体に影響がおよぶようなデザイン修正を行うとしたら、すべてのHTMLファイルをチェックして体裁に関する記述部分を修正しなければならない。一般的に、大規模とはいえないWebサイトであっても、数百から数千のファイルで構成されており、こうした作業にかかる工数は膨大なものとなる可能性が高い。

文書構造と体裁の定義を分離する理由の1つには、こうした作業の工数を大幅に削減できるということがある。HTMLファイルは情報の記述と文書構造の定義のみを行い、それらの体裁を定義したCSSファイルを別に分けておけば、デザイン修正などの体裁変更作業はCSSファイルのみを書き換えるだけで行うことができる。

文書構造と体裁の定義を分離する目的はこれだけではなく、つぎのようなことも想定される。

#### ①アクセシビリティの実現の容易さ

健常者用の通常のCSSファイルのほかに、たとえば高齢者や弱視者用に文字サイズを大きく定義したCSSファイルや、色覚異常のユーザのために色のコントラストに配慮したCSSファイルを用意し、ユーザ側で選択可能としておくことで、**アクセシビリティの実現が容易**となる。HTMLファイルが文書構造の定義を正しく行っていれば、読み上げソフトウェアや自動翻訳ソフトウェアにとっても適したデータとなるため、極弱視のユーザや日本語の読解が困難なユーザにとっても閲覧が容易なWebサイトを実現しやすくなる。

#### ②出力機器に応じた体裁の変更

通常のモニターで表示するためのCSSファイルのほかに、プリンタ出力

に適したCSSファイルや、テレビ画面、あるいは携帯電話に表示するためのCSSファイルを別に用意しておくことで、それぞれの出力機器に最適化した体裁を柔軟に実現できる。

### ③情報の共有や再利用の柔軟性

たとえばコンテンツフィード<sup>\*6</sup>のように、ほかのWebサイトとの情報共有を図る場合や、印刷物など他メディアでの情報の再利用を図る場合、共有あるいは再利用先で独自の体裁を用いることが多い。情報構造と体裁の定義が分離されていれば、こうした情報の共有や再利用が容易となる。

\*6 コンテンツフィードとは、コンテンツの概要をほかのWebサイトなどへの配信用に加工したデータのこと。7-3-5を参照のこと。

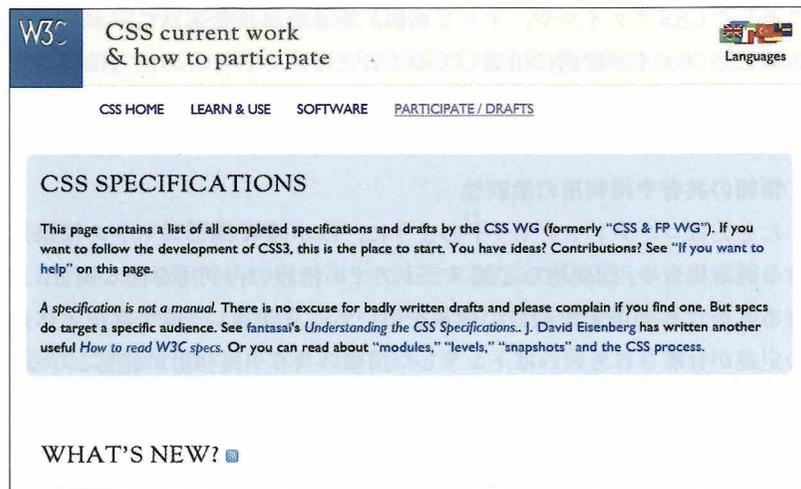
CSSはいまなお進化し続けている言語であり、バージョンが2から3へと移行しつつある。ただし、**CSS3**はいままでのようにひとつの大きな規格としてまとめられるのではなく、モジュールという概念で標準規格化が進められている。**モジュール**とは、セレクトタ、文字の装飾、背景色、ボーダーの扱いなど、CSS3が実現すべき機能をジャンルごとに分類したものである。

CSS3のもつべき機能は多岐にわたるため、そのすべてを1つの規格にしてしまうと、標準化作業も、Webブラウザにおける実装作業も、ともに時間と労力がかかりすぎてしまう。そこで、モジュールという小さな単位ごとに標準化や実装作業を進めよう、という考え方である。現在30以上のモジュールの標準規格化が作業中であり、最新の主要Webブラウザはすでにいくつかのモジュールを実装済みである。現在策定中のモジュールのうち、重要なものを表6.1に示す。

各モジュールの標準化作業の進捗状況は、**W3C**のWebサイトで公開されている(図6.10)。また、Webブラウザの各モジュールへの対応状況も、それぞれのWebブラウザの公式Webサイトで公開されている。

■表6.1—CSS3の主要モジュールと機能

モジュール	新機能やCSS2.1からの変更点
背景とボーダー	角丸ボーダーおよび画像を用いたボーダーが指定可能になる。
文章の段組み	文章を複数カラムに分ける段組みが可能になる。その際、カラム間の区切り線やそのスタイルなども指定可能である。CSS2.1では、擬似的にしか実現できないものである。
ボックスの段組み	複数のボックスを並べて配置する段組みを容易に行えるようになる。その際、各ボックスの高さを揃えるなどの処理も容易に行える。CSS2.1では、これらの処理はやや技巧的な手法を用いなくては実現できないものである。
変形	2Dおよび3Dの変形モジュールが用意されており、文字やボックス、図形などの変形が行える。また、時間変化モジュールと組み合わせることで、それらの変形のアニメーションを作成することも可能である。
透過色	文字やボックス、図形などに透明度を指定することが可能になる。時間変化モジュールと組み合わせることで、フェードイン・フェードアウトのような効果を作成することも可能である。



■図6.10——CSS3各モジュールの標準化作業の進捗状況（画像は2016年1月21日取得。Bert Bos, style activity lead Copyright © 1994–2016 W3C®）

### 6-3-4 JavaScript

\*7 JavaScriptは、ソースを直接HTML内に書き込むことができ、実行にあたってはWebブラウザが直接ソースを解釈し実行する。クライアントサイドで動作するスクリプト言語としてスタンダードな存在である。

②は、Webサイトにインタラクティブ性や操作補助などの機能を追加するために用いられるスクリプト言語である。プログラミング言語であるC言語に似た構文をもち、学習や開発が比較的容易であることから、Webサイト上の標準スクリプト言語として広く普及している。

JavaScriptの用途は非常に広く、さまざまな機能の実現に利用されているが、比較的多く見受けられる代表的な用途の例としては、つぎのようなものがある。

#### ①フォームなどの入力補助

ECサイトにおいて、商品の購入希望数を入力すると合計金額を自動的に計算するなど、データや情報を入力するための入力フォームの作成。

#### ②ナビゲーション補助

マウスの操作に反応してメニューを開くドロップダウンメニューやポップアップメニューなど、HTMLだけでは実現できないナビゲーション補助機能の作成。

#### ③インタラクティブな操作性の付加

画像にマウスを重ねると画像が変化したり、説明文を表示したりなど、Webサイトにインタラクティブな操作性を付加する。前述のナビゲーション補助のために用いられることも多いが、エンタテインメント的な使い方や、ユーザの情報理解を助けるために用いられることも多い。

\*8 ドロップダウンメニューとポップアップメニューについては、4-3-3を参照のこと。

#### ④Webページの動的な書き換え

通常、Webブラウザは、Webサーバから情報を受け取るとWebブラウザ全体の情報を再描画するが、②を用いることでWebブラウザ画面内の一部、あるいは全体を任意のタイミングで書き換えることができる。この機能は後述するDHTMLにおいて中核となるものの1つであり、またそれをさらに発展させたものとして③という手法も注目されている。

前述の内容以外にも、日付や時刻などの情報を利用した処理、買い物カゴに入った商品の合計金額を計算し表示する処理、ユーザの選択内容に応じて違うWebページを表示する処理などにも、②は利用される。

このように、②は、HTMLやCSSでは実現できない機能を補完する存在であり、ユーザに、より一層の利便性や娯楽性を提供する目的で利用される。

②は、1995年にNetscape NavigatorというWebブラウザの独自機能としてLiveScriptという名前でも実装されたのが始まりである。1996年には、Internet ExplorerでもJScriptという名称で実装され、さらにさまざまなWebブラウザに採用され普及したが、その過程で各Webブラウザ独自の拡張を施されたため、互換性がかなり低い時期があった。その後、2009年に、ECMA (European Computer Manufacturers Association: ヨーロッパ電子計算機工業会) によってECMAScriptとして標準化され、現在ではほとんどのWebブラウザがECMAScriptに準拠するかたちでJavaScriptを実装している。そのため、JavaScriptの互換性はかなり高いものとなってきている。ただし、まだ若干の機能の差異があること、ECMAScript準拠ではない古いWebブラウザが残っていることなど、問題は完全には解決されていない。

現在、よく用いられる処理などをライブラリ、あるいはフレームワー



■図6.11——さまざまなライブラリのなかでも人気の高いjQueryのWebサイト

クというかたちでまとめたものが開発／公開されており、Webサイト制作の現場でもよく利用されている。これらのライブラリは、高度なユーザインタフェースや後述の③を用いたWebサイトの開発などで工数を削減でき、Webブラウザ間の差異も吸収してくれるなどメリットが多い。有名なライブラリには、jQuery<sup>\*9</sup> (図6.11)がある。

\*9 jQueryとは、2006年に公開されたJavaScriptのライブラリである。軽量でありつつ、高度なユーザインタフェースに必要なさまざまなエフェクトやアニメーションを少ない工数で開発できるため人気が高い。jQueryのWebサイト(<http://jquery.com/>)から入手可能である。

\*10 APIとは、一般にはアプリケーションから、OSやサーバなどのソフトウェアの機能を利用するためのしくみのことを指す。この場合は、JavaScriptからWebブラウザの機能を利用するためのしくみである。

### 6-3-5 DHTML

DHTML(Dynamic HTML)とは、特定の言語の名称ではなく、Webページの内容を動的に書き換える手法、あるいは概念を表す言葉である。具体的には、HTMLとCSSで構成されるWebページを、JavaScriptなどのスクリプト言語によって部分的に、あるいは全体的に書き換えることで、Webページに動的な要素を付け加える手法である。

一般的なWebのしくみでは、WebブラウザとWebサーバが通信を行う場合、必ず画面遷移をとらなう。リンクなどがクリックされると、WebブラウザはWebサーバにリクエストを送る。Webサーバはリクエストに応じたHTMLファイルなどのデータを返信し、そのデータを用いてWebブラウザが画面全体の書き換え、つまり画面遷移を行う。このように、通信を行うと無条件に画面が遷移するというしくみのおかげで、簡単にハイパーテキストを実現できているわけだが、インタラクティブなコンテンツを制作する場合は、逆にこのしくみが障害になってしまう。

そのため、DHTMLではWebサーバとの通信を行わず、必要なデータをあらかじめダウンロードしておき、JavaScriptを利用してデータの一部を書き換えることでインタラクティブ性を実現している。具体的には、Webブラウザ内に読み込まれたHTMLとCSSの内容をDOM<sup>\*10</sup> (Document Object Model)というAPIを用いて逐次書き換えることでWebブラウザの表示内容を動的に変更する手法である。たとえば、マウスオーバーによって文字の色が変わったり、画像が切り替わったりする効果は、マウスオーバーを検知したらCSS内の文字の色を指定している部分や、HTML内の表示すべき画像を指定している部分を、DOMを介してJavaScriptで書き換えることで実現している。

DHTMLはおもに、リッチインタフェースの作成に利用されている。これは、ユーザの操作性を高める、ユーザの操作を補助する、情報の一覧性と詳細さを両立させる、などの目的がある。ドロップダウンメニューやポップアップメニュー、項目のドラッグ&ドロップといったインタフェースは、ユーザの操作性を高めるものとしてよく用いられている。また、ボタンや入力項目にマウスオーバーすると操作方法などを表示するインタフェースは、操作に不慣れなユーザを補助するものとして有効である。情報の一覧性と詳細さを両立させる例としては、たとえばECサイトなどで、商品の一覧表示は小さな写真と最小限の情報で行い、商品

にマウスオーバーすると大きな写真と詳細な情報がポップアップされる、といったものがある。

DHTMLを活用するためには、HTML、CSS、JavaScriptの知識が必要であり、Webブラウザごとの互換性も考慮しなくてはならないなど、とくに初学者にはやや敷居が高い。しかし、前述のJavaScriptライブラリなどを利用することで、作業の手間をかなり減らすことができ、互換性についてもほとんど考慮する必要がなくなっている。

### 6-3-6 ③

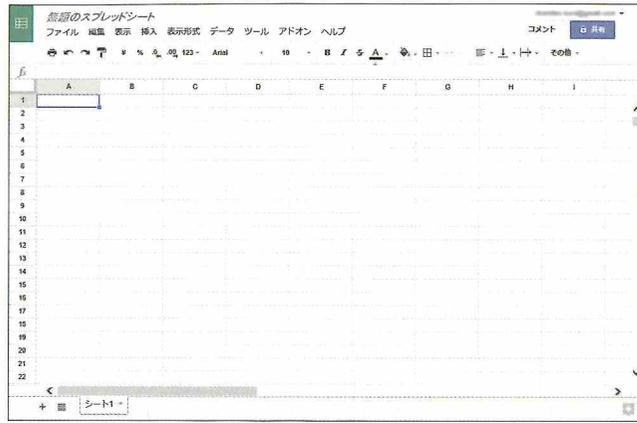
現在、Webサイト制作において重視されている制作手法の1つが③<sup>\*11</sup> (Asynchronous JavaScript and XML, あるいはAsynchronous JavaScript + XML)である。

\*11 Asynchronousとは、非同期という意味である。

Ajaxという手法の明確な定義は存在しないが、XMLHttpRequestというしくみを利用してWebサーバからデータを取得しつつ、DHTMLを用いて画面の制御を行う、という組み合わせが一般的である。XMLHttpRequestを用いた場合、WebブラウザはWebサーバからデータを受信しても画面の再描画を行わないため、DHTMLによる画面制御を邪魔することがない。そのため、あらかじめ読み込んでおけないような大容量のデータを用いたり、Webサーバ側で計算などの処理を行ってから改めて表示する、といった処理が可能である。また、XMLHttpRequestは非同期通信を行うため、通常のページリクエストのように、Webサーバからのデータ送信終了を待たなくてよい。そのため、Webサーバとの通信中であってもユーザの操作を中断させる必要がないというメリットもある。また、DHTML同様、既存の技術の組み合わせで実現されている手法であるため、プラグインを必要としないことも重要である。

DHTMLは、スムーズなポップアップメニューやマウスオーバーによる画像の拡大など、Webサイトのインタフェースをより魅力的で使いやすいものにしたが、情報を発信するメディアというWebの位置付けを変えるものではなかった。それに対してAjaxは、Webサーバ上で動くプログラムや、データベースと連携しつつ、インタラクティブな操作性を実現でき、デスクトップアプリケーションと変わらないアプリケーションをWebブラウザ上に構築できる(図6.12)。こうしたアプリケーションはWebアプリケーションとよばれ、メディアからアプリケーションプラットフォームへとWebを進化させるものである(図6.13)。

Webアプリケーションでは、アプリケーション本体はもちろん、作成するデータも利用者のPCではなくWebサーバ上に置かれる。そのため、利用者は、インターネットにアクセスできる環境であればどこからでも作業を行えるというメリットがある。また、企業にとっても社員数



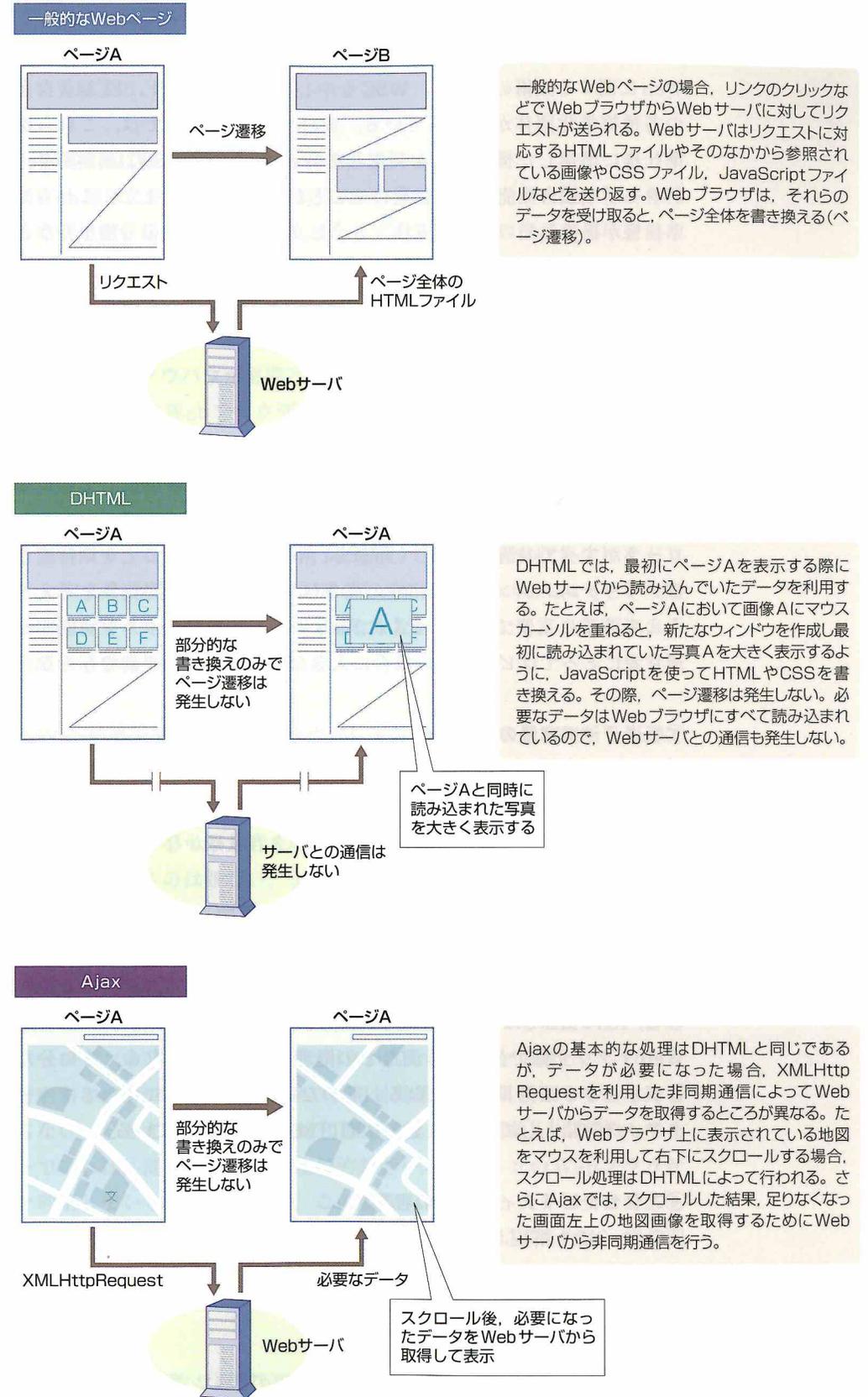
■図6.12——Webブラウザ上で表計算やワードプロセッサを利用できるGoogleドキュメント  
(©グーグル株式会社)

に応じて契約数を柔軟に変更でき、アプリケーションのバージョンアップの手間もかからないなどメリットは多い。このような利用形態はクラウド・コンピューティングとよばれ、コンピュータの利用形態の1つとして定着している。

ただし、XMLHttpRequestは、Webブラウザが発したリクエストにWebサーバが応えるというかたちでしか通信を行えない、片方向通信であるところに問題がある。つまり、Webサーバ側が何らかの情報を送りたくても、Webブラウザ側からのリクエストがない限り送ることはできないということである。そのため、たとえば為替や株価情報をリアルタイムで表示するといった、Webサーバ側の情報の変化に応じて通信を開始しなくてはならないWebアプリケーションの開発は難しい。この問題を解決するため、現在WebSocketというプロトコルが策定中である。これは、Webブラウザ側からだけでなく、Webサーバ側からも通信を開始できる、いわゆる双方向通信を実現するものであり、2016年1月現在、標準規格として勧告候補になったところである。

### 6-3-7 Web標準

これまでの各言語の解説でも触れてきたとおり、Webブラウザ間に完全な互換性がないという状況は、Web全体にとっても大きな問題点である。制作側にとって、複数のWebブラウザで同様の表示や機能を実現するための作業は工数の増大を招く大きな要因の1つになっている。また、各Webブラウザの仕様や特性に精通していなければ、こうした作業自体が行えないという点も問題である。一方で、Webサイトの運営側にとっては、より多くのWebブラウザで閲覧可能にするということは、ビジネスチャンスを広げるための要件の1つである。このような観点から提唱されているWebサイト制作の考え方がWeb標準である。



■図6.13——一般的なWebページとDHTML、Ajaxの違い

Web標準を大雑把に定義すると「国際的な標準団体によって作成された仕様に基づいてWebサイトを制作する」ということになる。現在、Webに関する技術の仕様は、W3Cを中心に、ISO、IETF、ECMAなどの各種標準化団体が策定している。Web標準の考え方では、これら標準仕様に準拠した機能だけを利用してWebサイト制作を行い、各Webブラウザの独自機能の使用は避けることが求められる。また、これら標準仕様が推奨している記述方法、とくに文書構造と体裁の分離を行うということもWeb標準では重視されている。

Web標準によって得られるメリットは、こうした互換性問題の回避以外にもつぎのようなものがあげられる。

#### ①サーチエンジンとの親和性のよさ

ロボット型サーチエンジンのクローラは、巡回先のWebサイトが正しいHTMLで書かれていることを期待してWebサイトの内容を解析し情報を収集している。Web標準にのっとったWebサイトであれば、クローラがすべての情報を正しく解析し、収集してくれることを期待できる。サーチエンジンとの親和性の高さはWebサイトの閲覧数を考えるうえで非常に重要であり、とくにWebサイト上でビジネスを展開する運営者にとってはビジネスの成否に大きな影響を与える要素でもある。

#### ②転送データ容量のスリム化

たとえばtableレイアウトの多用は転送データ容量の増大を招くことが統計的に指摘されている。こうした部分を標準仕様にのっとり、CSSレイアウトに変更するだけでも、転送データ容量はかなり低減できる。

#### ③Webブラウザにかける負担の低減

WebブラウザがHTMLを処理、表示する場合は、省略された終了タグの位置を推測するなど余計な処理も行わなければならない。それに対して、XHTMLではタグの省略を許さず、記述の規則も厳密であるため、Webブラウザ側で余計な推測などの処理を行う必要がなく、その分だけ表示にかかる時間が短くなる。そのため、Web標準においてはマークアップ言語としてHTMLよりもXHTMLが推奨されている。

#### ④アクセシビリティの実現の容易さ

6-3-3でも解説したとおり、文書構造と体裁を分離することによって、アクセシビリティの実現も容易になる。

#### ⑤メンテナンスの容易さ

これも6-3-3で解説したとおり、後々のデザイン変更やデータ共有などが容易となる。

このようにメリットの多いWeb標準ではあるが、つぎのような問題点も存在する。

#### ①各種標準に精通していなければならない

Web標準に準拠した記述を行うためには、Webサイト制作者が各種の標準規格に精通していなければならない、それぞれの機能が標準仕様に準拠しているものなのか、特定のWebブラウザに固有の機能であるのかを把握している必要がある。

#### ②経験則的なノウハウが必要とされる

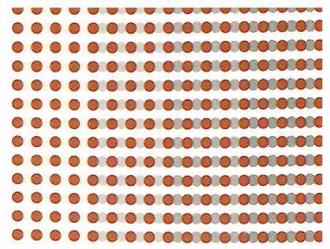
現在の主要なWebブラウザの最新バージョンは、みなWeb標準への準拠度が高いが、細かな機能面などでわずかな差異がある。こうした差異についての情報がWebブラウザの提供元から文書などで公開されているわけではないため、Web制作者はさまざまなテストを行って自分で調べたり、ほかのWeb制作者が公開している情報を収集するなどして、「こういう場合はこういう記述のしかたをしたほうがよい」という経験則的なノウハウを蓄積しなくてはならない。そのため、ある程度の手間暇と情報収集能力が求められる。

#### ③Web標準未準拠のWebブラウザにどこまで対応するか判断しなくてはならない

バージョンの古いWebブラウザはWeb標準に準拠していないか、していたとしても不十分である場合が多く、共通のソースから完全に同じ表示結果を得るのは難しい。より多くのWebブラウザに対応するほど、より多くのユーザーに閲覧してもらえ可能性が高まるわけだが、各Webブラウザごとに表示可能なソースを記述しなくてはならず、制作期間と制作コストの上昇を招くことになる。そのため、Webサイト制作者は、制作するWebサイトが対象としているユーザーのなかで、古いバージョンのWebブラウザが利用されている割合を予測し、これらに対応する必要があるのかどうか、費用対効果の面から判断しなくてはならない。たとえば、20代から30代前半の社会人を対象としたWebサイトであれば、閲覧者の多くは比較的新しいバージョンのWebブラウザを利用していると期待できる。この場合、古いWebブラウザに対応しなかったとしても、閲覧できない対象ユーザーは少数であるため、制作期間や制作コストを余分に掛けてまで対応する必要性は低いと判断できる。

#### ④最新WebブラウザのWeb標準未準拠機能について判断しなくてはならない

最新のWebブラウザであっても、Web標準に未準拠の機能を実装し



ている場合は多い。また、標準規格のなかには内容があいまいな機能もあり、こうした部分については現在のところ、各Webブラウザが独自の解釈で実装している。こうした機能を利用する場合は、③と同様、各Webブラウザごとに個別のソースを記述するか、対象とするWebブラウザを限定するかを判断しなくてはならない。

ただし、6-3-4で解説したjQueryのようなライブラリを使う場合は、ライブラリ内部で各Webブラウザへの対応を行うため、Webサイト制作者は、Webブラウザの差異をあまり考えなくてもよい。

このようなメリットと問題点を比べてみても、やはりWeb標準に準拠した記述を行うべきであるというのがWebサイト制作業界の大勢である。古いバージョンのWebブラウザはいつか使われなくなるはずであり、今後のWebブラウザのバージョンアップはWeb標準への準拠の度合いをより高めていく方向に進むはずである。各種標準規格もまた、Webサイト制作の最前線で必要とされる機能を取り込みつつ、より厳密に規格を定めていくものと期待されている。そのため、今後は、Web標準に準拠した記述を行うことのメリットがますます高まっていくと考えられている。

## 6-4

### バックエンドで活用する技術

Webサイト制作を行うためには、6-3までで解説したフロントエンドの技術だけでなく、バックエンドの技術に関する理解も必要である。ここでは、Webサイトのバックエンドがどのようなものであるのか、そこで使われている技術にはどのようなものがあるのか、具体的にどのように構成されているのか、などについて解説する。

#### 6-4-1 バックエンドとは

一般的に、Webブラウザ側でコンテンツを表示したり、リンクのクリックやフォームへの入力といったユーザの操作を受け付ける、といった処理を①とよんでいる。

それに対して、フロントエンドからのリクエストに応じてコンテンツデータを送信するのが②である。その際、バックエンド側に保存されているコンテンツをそのまま送信する場合もあるが、リクエストに応じて、さまざまなプログラムによってデータを加工してコンテンツを生成した後に送信する場合もある(図6.14)。

#### 6-4-2 バックエンドの一般的な構成

Webサイト構築におけるバックエンドとは、**Webサーバと回線**である。Webサーバは、ハードウェアとしてのサーバマシンと、その上で稼働する各種ソフトウェアによって構成されている。

サーバマシンとしてのハードウェアは、一般的なPCが利用されることもあるが、サーバ用途向けに特化したハードウェアが用いられることも多い。これは、Webサーバが24時間365日の連続稼働を要求されるためであり、とくに電源まわりや冷却システムが強化されたものが利用される。

Webサーバを構成するソフトウェアは、さまざまなものがあり、目的に応じて構成が異なる。これについては6-4-4で詳しく解説する。

#### 6-4-3 バックエンド選定時に考えるべきこと

サーバマシンおよび回線を選定する場合、Webサイトにどれだけのアクセス数が見込まれるかを考慮する必要がある。サーバマシンの処理

能力と回線の帯域の太さは、同時に処理できるアクセス数と個々のアクセスに対する転送速度に影響する。より多くのユーザからのアクセスに対応するには、より処理能力が高いサーバマシンと帯域の太い回線が要求される。

ただし、処理能力が高いほどサーバマシンの導入コストと維持コストは上昇し、回線も帯域の太さに応じてコストが上昇するため、見込まれるアクセス数に見合ったものを導入するべきである。

もう1つ考慮すべきことは、サーバを管理するための人件費である。サーバマシンを自社で保有する場合は、それを管理するための人員も確保しなくてはならず、この人件費もバックエンドを維持するためのコストとなってしまう。

前述のコストを下げ、また事業の展開に応じて柔軟にバックエンドの能力を増減するために、ホスティングサービスを利用するケースも多い。ホスティングサービスは、数台から数百台以上のサーバマシンと太い帯域を所有し、利用者に貸し出す専門業者である。基本的なサーバマシンのセットアップと管理も行ってくれる。

利用者は、見込まれるアクセス数に応じた最小限の処理能力と帯域でWebサイトを開設し、アクセス数の増加に応じて借り受けるサーバマシンの処理能力と帯域を増強していく、といった運用が可能になる。最近では、**仮想化技術**<sup>\*12</sup>を利用し、サーバマシンの処理能力やディスク容量をより柔軟に変更できるサービスも提供されている。

\*12 仮想化技術とは、プログラムによって仮想的なハードウェアを生成し、その上でOSなどのソフトウェアを稼働させる技術であり、たとえば、OS X上でWindowsを稼働させる、といったことが可能になる。ホスティングサービスにおいては、1つのハードウェア上に複数の仮想ハードウェアを生成し、貸し出すといったことが行われている。その際、各仮想ハードウェアに割り当てるCPUパワーやメモリ、ディスク容量などをソフトウェアで変更できるため、利用者の要求に応じた柔軟な処理能力の増減が可能になる。

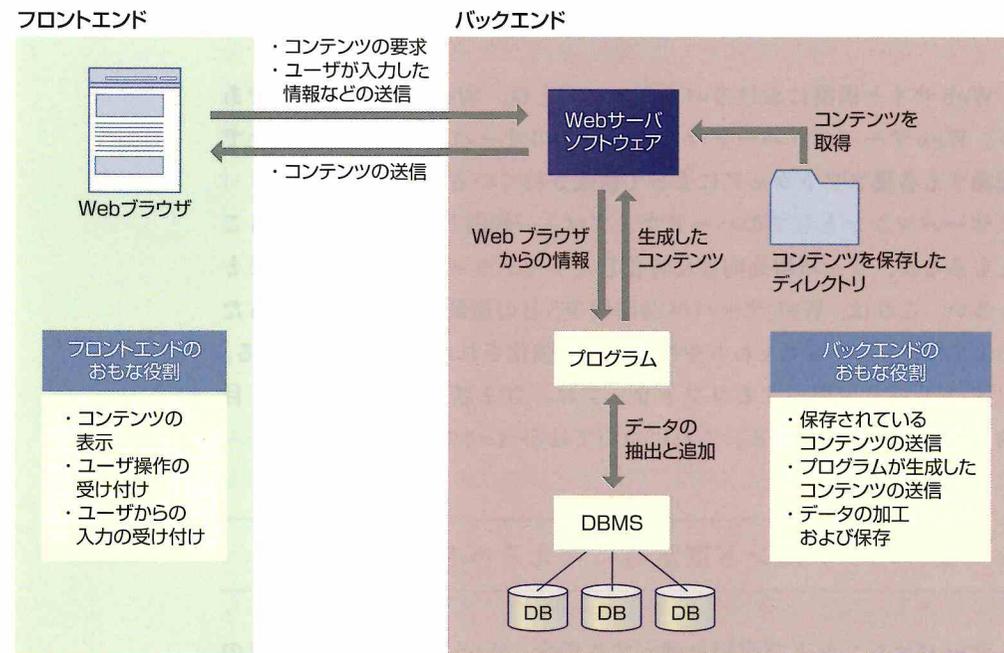


図6.14 フロントエンドとバックエンドの処理

## 6-4-4 Webサーバを構成するソフトウェア

Webサーバ上では、さまざまなソフトウェアが稼働している。そのなかでもおもなものは、OS、Webサーバソフトウェア、DBMS(データベース管理システム)、開発言語の4つである。

### [1] OS

Webサーバソフトウェア、DBMS、開発言語を稼働させるための土台(プラットフォーム)となるのが**OS**である。WebサーバのためにさまざまなOSが利用されているが、現在主流となっているのは**③**と、**Windows Server**の2つである。

Linuxは、1991年にリーナス・トーバルズが開発したUNIX系列のOSである。誰もが無料で入手して利用できるオープンソースソフトウェアであり、WebサーバやDBMS、各種開発言語といったバックエンドの開発環境までも、すべて無料で構築できることから人気が高い。こうしたLinuxによるバックエンド環境は、Linux(OS)+ Apache(Webサーバソフトウェア)+ MySQL(DBMS)+ PHP(開発言語。PHPではなくPerl, Pythonとされることもある)の頭文字をとって**LAMP**ともよばれる。GUI環境も整備されており、かなりの操作をGUIで行うことが可能だが、CUI環境による操作が必要な場合も多い。とくに遠隔地にあるWebサーバを操作する場合、CUI環境での操作が求められることがほとんどであるため、十分に使いこなすためにはさまざまなコマンドの書式を覚える必要がある。

\*13 Linux自体はカーネルとよばれる、OSの核となる部分だけが開発されている。実際に利用されているのは、各種グラフィカルインタフェースやツール類、サーバソフトウェアを組み込んでパッケージ化した、ディストリビューションとよばれるものである。おもなディストリビューションには、Red Hat Enterprise Linux, Fedora, CentOS, Debianなどがある。

Windows Serverは、マイクロソフト社が開発・販売しているサーバ向けOSである。企業内におけるファイルサーバやプリントサーバなど、さまざまなサーバ用途に用いられており、Webサーバとしての採用も多い。Windows Serverは、基本的にほとんどの操作をGUI環境で行うことができ、これは遠隔地にあるWebサーバを操作する場合も同様である。

### [2] ④

④は、リクエストを受け付け、コンテンツをWebブラウザへと送信するプログラムであり、Webサーバの中核である。

Webサーバソフトウェアの基本的な役割は、Webブラウザからのリクエストに応じて、あらかじめWebサーバ内に保存されているHTMLファイルや画像、動画などのコンテンツデータをそのまま送信することである(静的コンテンツ)。しかし、リクエストの内容や日時、ほかのWebサーバの状況など、さまざまな条件に応じて、外部プログラムと連携することでコンテンツを生成、送信するといったことも可能である(動的コンテンツ)。

\*14 GUIとは、グラフィックスを多用した表示を用い、マウスなどのポインティングデバイスによる操作を基本とする、ユーザーインタフェースのこと。Graphical User Interfaceの略である。

\*15 CUIとは、文字による表示を行い、キーボードからのコマンド入力による操作を基本とする、ユーザーインタフェースのこと。Character User Interfaceの略とされるが、これは和製英語であり、欧米ではCommand Line Interface(CLI)とよばれる。

\* 16 Apache は、さまざまなLinuxディストリビューションやMacOS Xに標準で付属している(インストールは別途行う必要がある場合が多い)。また、公式サイトである<http://httpd.apache.org/>などからも入手可能である。

\* 17 SQLとは、DBMSを操作するための簡易言語の国際規格であり、DBMSに対して、データの抽出、追加、削除などの操作を指示するために用いられる。なお、SQLは略語ではなく、「SQL」という単語である。エスケュールまたはシーケルと読む。

\* 18 MySQLとは、もともとMySQL AB社が開発し、現在はOracle社が開発、配布しているDBMSである。世界的には最もシェアの高いオープンソースのDBMSである。ただし、条件によっては有償の商用ライセンスになる場合がある。<http://www.jp.mysql.com/>から入手可能である。

\* 19 PostgreSQLとは、PostgreSQL Global Development Groupによって開発、配布されているDBMSである。日本においては、利用者数がMySQLと拮抗している。<http://www.postgresql.org/>から入手可能である。

\* 20 関係データベースとは、複数のデータを、キーとよばれる情報によって関係付け、管理するデータベース管理システムのこと。現在は、ほとんどのDBMSが関係データベースである。

Webサーバソフトウェアにはさまざまなものが存在するが、現在主流となっているのはApache HTTP Server(以下、「Apache」とIISである。

⑤は、Apacheソフトウェア財団によって開発、配布されているオープンソースソフトウェアである。Linux, BSD, Windows, MacOS Xなど、さまざまなOS上で動作すること、誰でも無料で利用できること、多機能であること、かなり初期から利用されてきたことなどから、Webサーバソフトウェアの事実上の標準的な扱いを受けている。

IIS(Internet Information Services)は、マイクロソフト社が開発しているWebサーバソフトウェアである。Apacheとは異なり、動作環境はWindows上でのみだが、Windows Serverには標準で付属しており、Windows 8.1やWindows 10にも機能制限版が標準で付属しているため、Windowsユーザであれば誰でも利用可能である。

[3] ⑥(データベース管理システム)のおもな役割はデータベースの管理とデータ操作の2つである。データベースの管理では、データベースの作成、バックアップ、削除といった作業が行われる。データ操作では、特定の条件に合ったデータの抽出、追加、削除といった作業が行われる。これらの操作は、DBMSに装備されているインタフェースプログラムから行われる場合もあるが、開発言語によるプログラム内からSQL文とよばれる特別な命令によって操作される場合が多い。

ECサイトにおける商品情報、会員サイトにおける会員情報、情報サイトにおけるコンテンツデータなどの管理はDBMSを用いて行われるのが一般的であり、Webサイト制作において非常に重要な役割を担っている。

Webサイト制作では、MySQL、PostgreSQLの2つのオープンソースDBMSがよく利用されている。どちらも、Linux, Windows, MacOS X上で動作し、誰でも無料で利用することができる。

DBMSのなかでも関係データベースを扱うものをRDBMS(Relational DataBase Management System)とよんでいる。Webサイト制作で用いられるDBMSは、ほとんどの場合RDBMSである。

#### [4] 開発言語

Webサイト制作では、一般的にスクリプト言語とよばれる、プログラミング言語よりも習得が容易な言語が利用される。代表的なものにPerl、PHP、Python、Rubyなどがある。これらの言語はすべてLinux, Windows, MacOS X上で動作し、オープンソースソフトウェアとして誰でも無料で利用することができる。

開発言語によるプログラムにはさまざまなものが考えられる。典型的

な例の1つに、Webサーバソフトウェアから何らかのキーワードを受け取り、そのキーワードに一致するデータをDBMSから抽出し、それらのデータをコンテンツの形式に加工した後Webサーバソフトウェアに返す、といったものがある。

もう1つの典型例は、Webサーバソフトウェアから何らかのキーワードを受け取り、そのキーワードに加工を施した後、DBMSを介してデータベースに保存する、といった処理である。

こうした処理の典型例については、6-4-5で紹介する。

#### [5] Webサーバソフトウェアと外部プログラムの連携方法

Webサーバソフトウェアと外部プログラムとのおもな連携方法には、CGI, FastCGI, モジュール化がある。

##### ① CGI

CGI(Common Gateway Interface)は、Webサーバソフトウェアが必要に応じて外部プログラムを起動し、その処理結果を受け取るためのしくみである。このとき起動する外部プログラムをCGIプログラムとよんでいる。WebサーバソフトウェアからCGIプログラムへの情報の受け渡しは、コマンドライン引数、環境変数、標準入力が使われる。また、CGIプログラムが標準出力に出力したデータがWebサーバソフトウェアを介してWebブラウザに送られる。

つまり、Webサーバソフトウェアから情報を受け取ったCGIプログラムが、その情報に基づいて何らかの処理を行い、その処理結果をHTML文書の形式で出力することで、動的にコンテンツを生成するというしくみである。しかし、Webサーバソフトウェアから要求があるたびにCGIプログラムの起動と終了を行うため、効率が悪いという欠点も存在する。

##### ② FastCGI

前述のCGIの問題点を改善したものがFastCGIである。FastCGIにおいては、Webサーバソフトウェアからの要求で起動したCGIプログラムは終了させず、つぎの要求が来たらそのまま処理を行うことになっている。そのためプログラムの起動と終了の時間が必要ない。また、FastCGIではWebサーバソフトウェアとの情報のやりとりをネットワーク経由で行うことも可能である。そのため、WebサーバソフトウェアとCGIプログラムを別々のハードウェア上で動作させることで、Webサーバ全体の高速化や耐久性の向上を図ることができる。

##### ③ モジュール化

モジュール化とは、開発言語によって開発したプログラムをWebサーバソフトウェアのなかに組み込んでしまうという考え方である。プロ

\* 21 Perlとは、ラリー・ウォールによって開発されているスクリプト言語である。もともとは、UNIX上でテキスト処理や簡単な自動処理を記述する目的で開発されたものだが、現在はWebサイト制作のための開発言語としても多用されている。<http://www.perl.org/>から入手可能である。

\* 22 PHPとは、The PHP Groupによって開発されているスクリプト言語である。当初からWebサイト制作のための言語となることを意図して開発されており、Webサイト制作の開発言語としては世界的に最も利用されている言語である。<http://www.php.net/>から入手可能である。

\* 23 Pythonとは、グイド・ヴァンロサムによって開発されたスクリプト言語である。Webサイト開発以外にも、数式処理やグラフィックアプリケーションなど、幅広い用途に利用されている。<http://www.python.org/>から入手可能である。

\* 24 Rubyとは、まつもとゆきひろによって開発されているスクリプト言語である。当初からオブジェクト指向を取り入れた開発が行われている。<http://www.ruby-lang.org/>から入手可能である。

\*25 コマンドライン引数、環境変数、標準入力、標準出力とは、OSとプログラムあるいはプログラムどうしにおいて情報をやりとりするためのしくみである。もともとUNIXにおいて発展した機能であるが、WindowsやOS Xなどでもサポートされている。

プログラムがWebサーバソフトウェア内部で動作しているため、起動と終了の時間もかからず、情報のやりとりも高速化するなどメリットは多い。モジュール化のためのしくみは開発言語ごとにさまざまなものが用意されており、CGIやFastCGIに比べて開発の手間が特別かかるといったことはない。ただし、モジュールとして組み込んだプログラムにバグや脆弱性が存在した場合、プログラムがハングアップするとWebサーバソフトウェア自体もハングアップしたり、プログラムの脆弱性をつかれてWebサーバソフトウェアそのものを第三者に乗り取られるなどの危険性がある。

### 6-4-5 Webサーバにおける処理の例

Webサーバソフトウェア、DBMS、開発言語によるプログラムが連携して行う処理にはさまざまなものがある。ここではいくつかの例を簡略化して示す。

これらの例において、Webサーバソフトウェアは、Webブラウザに対する窓口、プログラムへの処理の依頼、プログラムの処理結果に基づくWebブラウザへのコンテンツの送信といった役割を果たしている。

#### [1] キーワードに一致したコンテンツを表示

Webブラウザからキーワードを入力すると、そのキーワードに一致したコンテンツをデータベースから抽出し、表示するという処理である。情報提供サイトや知識データベース系のサイトでよく見られる処理で、ECサイトにおける商品検索なども、考え方はまったく同じである。

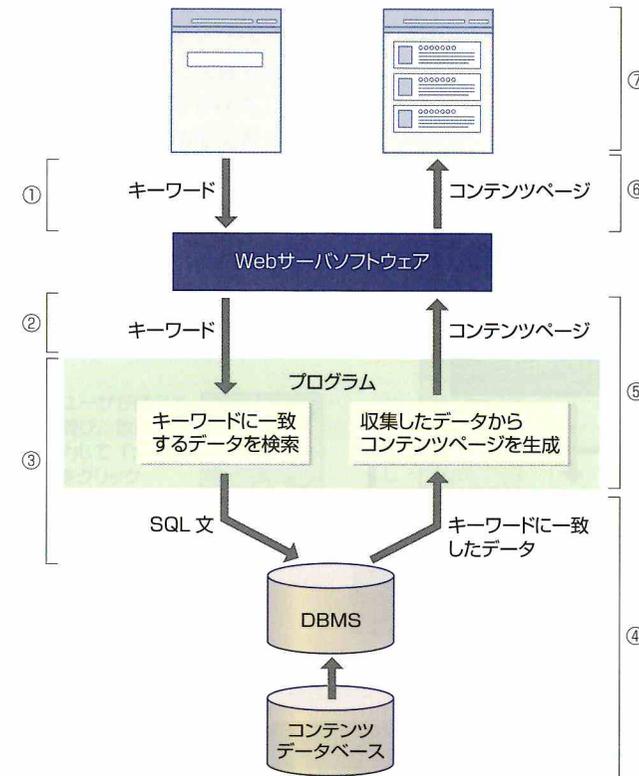
この処理のポイントは、キーワードに一致するデータの検索と、収集したデータからコンテンツページを生成する部分である。データの検索はDBMSに対してSQL文を発行することで行う。ここでは省略しているが、実際には一致するデータが見つからなかったときの処理も必要である。

コンテンツページを生成する処理では、収集したデータを表示するようなHTML文書の生成を行うことになる(図6.15)。

#### [処理の流れ]

- ① Webブラウザ上でユーザが入力したキーワードがWebサーバソフトウェアに送信される。
- ② Webサーバソフトウェアは、受け取ったキーワードをプログラムに渡す。
- ③ プログラムは、DBMSに対してキーワードと一致するデータを抽出するよう指示するSQL文を送る。
- ④ DBMSは、管理しているコンテンツデータベースからキーワードと一致するデータをすべて抽出し、プログラムに返す。

- ⑤ プログラムは、DBMSから返されたコンテンツデータを基に、HTML文書を作成する。
- ⑥ Webサーバソフトウェアはプログラムが生成したHTML文書をWebブラウザに送信する。
- ⑦ Webブラウザは、Webサーバから送られてきたHTML文書を表示する。



■図6.15—キーワードに一致したコンテンツを表示

#### [2] コンテンツの追加

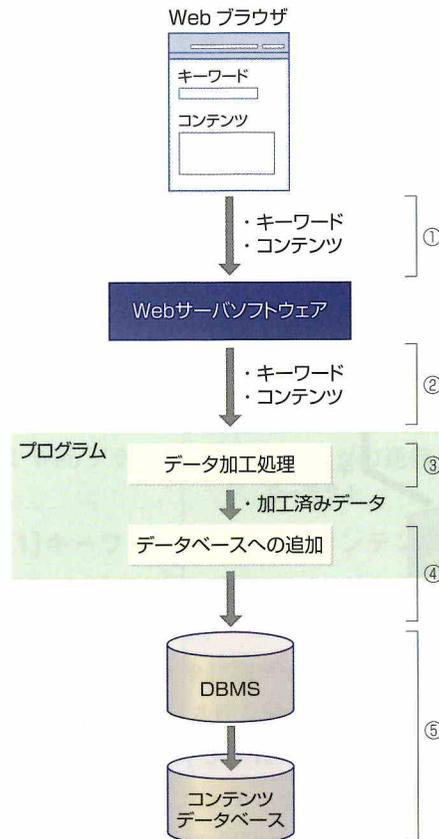
Webブラウザからキーワードとコンテンツを入力し、コンテンツデータに登録する処理である。<sup>\*26</sup>CMSのようなコンテンツ管理システム、掲示板、ブログシステムなどでよく見られる処理である。また、ECサイトにおける商品の登録なども、同様の処理を行っている。

この処理のポイントは、データ加工処理の部分である。ここでは、データベースに登録するためのデータの分割や加工を行うとともに、6-4-6で説明するSQLインジェクションのような、外部からの攻撃よりシステムを守るためのデータ加工も行われる(図6.16)。

- ① Webブラウザ上でユーザが入力したキーワードとコンテンツがWebサーバソフトウェアに送信される。
- ② Webサーバソフトウェアは、受け取ったキーワードとコンテンツをプログラムに渡す。

\*26 CMSについては、7-3-5を参照のこと。

- ③ プログラムは、受け取ったキーワードとコンテンツに何らかの処理を施す。
- ④ プログラムは、DBMSが管理しているコンテンツデータベースに、キーワードとコンテンツを追加するよう指示を出す。
- ⑤ DBMSは、コンテンツデータベースにキーワードとコンテンツを追加する。



■図6.16——コンテンツの追加

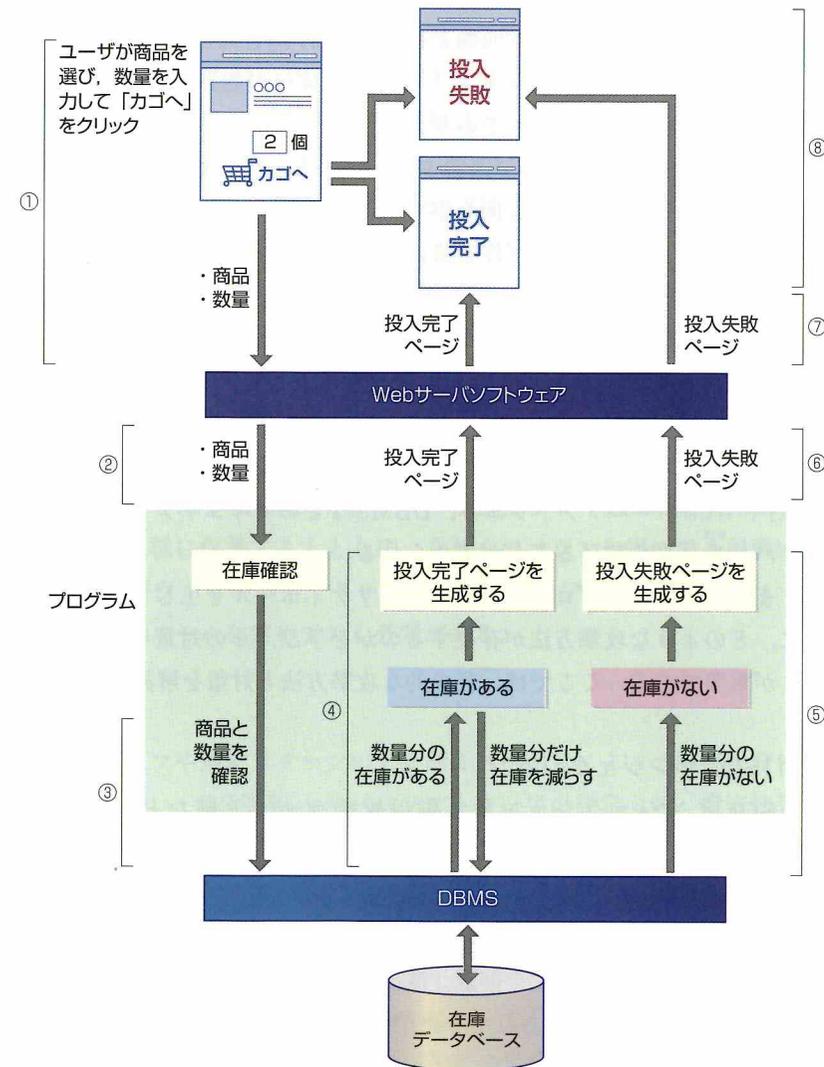
### [3] 買い物カゴ

ECサイトにおいて、選んだ商品を望みの数だけ買い物カゴに入れるための処理である。

この処理でポイントとなるのは、商品の在庫の確認と更新である。ユーザーが選んだ商品と希望する数量が存在するか、在庫データベースを確認し、存在するならば在庫から希望数量分を減らすという処理を行う。現実には、商品を発送しない限り在庫は減らないわけだが、この段階で一時的に在庫データベース上の在庫は減らす必要がある。たとえばある商品の在庫が5個あったとして、ユーザーAが買い物カゴに3個入れたとする。この段階で在庫を減らしていなかった場合、直後にユーザーBが同じ商品を買物かごにやはり3個入れることができってしまう。ユーザーA、

Bがともにこの後で決済を行ったら、商品発送の段階で在庫不足ということになってしまうわけである(図6.17)。

- ① ユーザが商品を選び、数量を入力して「買い物カゴへ」をクリックすると、商品と数量がWebサーバソフトウェアに送信される。
- ② Webサーバソフトウェアは、受け取った商品と数量をプログラムに渡す。
- ③ プログラムは、在庫データベースに商品が数量分存在するか確認する。
- ④ 数量分の在庫が存在する場合、買い物カゴへの投入が完了したことを表示するHTML文書を生成する。
- ⑤ 数量分の在庫が存在しない場合、買い物カゴへの投入が失敗したことを表示するHTML文書を生成する。
- ⑥ ④または⑤で生成したHTML文書をWebサーバソフトウェアに返す。
- ⑦ Webサーバソフトウェアはプログラムが生成したHTML文書をWebブラウザに送信する。
- ⑧ Webブラウザは、Webサーバから送られてきたHTML文書を表示する。



■図6.17——買い物カゴ。

## 6-4-6 セキュリティ

Webサイト制作においては、コンテンツを正しく表示し、ユーザにサービスを提供することも重要だが、Webサイトの安全性を確保し、安定した運用を続けることも重視される。これにはハードウェアのトラブルやプログラムのバグといった問題とともに、悪意ある第三者からの攻撃という問題も含まれている。こうした攻撃によって発生し得る障害には、システム停止、データ改ざん、データ流出などがある。

Webサイトを広報ツールとして活用している企業にとって、システム停止は広報活動の停滞を招き、ECサイトや情報提供サイトのような、Webサイトから直接利益を得ている企業にとっては、直接の売上減少をもたらす。Webサイト上のデータ改ざんは、企業やWebサイトへの信頼を損ねるものである。データ流出、とくに個人情報の流出は、補償問題に発展して金銭的にも大きな損失を招くとともに、企業の社会的信用を大きく失墜させる重大問題となる。また、障害発生後の調査や対策、場合によっては新聞やテレビなどのマス媒体への謝罪広告の掲載など、多大なコストが発生しがちである。

そのため、Webサイトを運営する企業にとって、Webサイトの安全性を確保することは重大な関心事であるとともに、社会的義務であるともみなされている。これは制作を請け負う制作者にとっても同様である。

こうした悪意ある第三者からの攻撃は、システムの **(7)** をつくものがほとんどである。セキュリティホールは、OSやWebサーバソフトウェア、DBMSなどに内在するものと、Webサイト制作者が開発したプログラムに生じてしまったものが考えられる。

Webサイト制作者は、最新のセキュリティ情報にはつねに目を配り、OSやWebサーバソフトウェア、DBMSなどのセキュリティホールを塞ぐ **パッチ**<sup>\*28</sup>を当て続けることが求められる。

また、開発するプログラムにセキュリティホールを生じさせないために、どのような攻撃方法が存在するのかを学び、その対策を理解することが重要である。ここでは、代表的な攻撃方法と対策を解説する。

### [1]SQLインジェクション

SQLインジェクションは、掲示板やコメント欄など、ユーザがHTMLフォームを利用してWebサイトに情報を送信する機能を利用して、Webサーバ側のデータベースを操作するSQL文を送り込む攻撃である。SQLを注入(インジェクション)することから、この名称でよばれている。データの改ざんや削除、流出といった被害が発生する。

Webブラウザから送られてきた情報はすべてエスケープ処理を行う、などの対策が取られる。

\*27 セキュリティホールとは、OSやソフトウェアのセキュリティ上の脆弱点のこと。とくに、管理者権限がなければ実行不可能なはずの操作を可能にさせてしまったり、そこから情報の改ざんや漏えいが起こるような欠陥のことをいう。設計ミスや攻撃に対する想定不足、単純なプログラミング上のミスなどによって生じる。

\*28 パッチとは、プログラムの一部を更新することで、バグやセキュリティホールへの対策を施すためのデータのこと。OSや各種サーバソフトウェアは、定期的なパッチの配布を行うことが多い。重大なセキュリティホールが発見されたときは緊急でパッチが配布されるので、情報にはつねに気を配っておく必要がある。

\*29 エスケープ処理とは、プログラムに対する命令や指示となっているような文字情報を加工して、命令や指示としての意味をもたないようにする処理である。処理内容は言語によって異なるが、一般的には文字情報全体をダブルクォーテーションで囲ったり、文字の前にバックslashを付加するなどの処理が多い。

[2] **(8)**

**(8)**は、掲示板のように、HTMLフォームから入力した情報がWebサイトのコンテンツとして反映されるWebサイトにおいて、入力内容に不正なスクリプトなどを埋め込んでおく攻撃である。ほかの閲覧者がそのWebサイトを表示するだけで、不正なスクリプトがWebブラウザ上で実行され、Cookie情報の流出や不正なページ遷移によるフィッシングが行われたりする。

SQLインジェクション同様、Webブラウザから送られてきた情報にエスケープ処理を行う、などの対策がとられる。

[3] **(9)**

**(9)**は、Webサーバなどに対して、大量のアクセスを集中させることによってWebサイトの応答を極端に遅くしたり、Webサーバをダウンさせる攻撃である。特定のPCから大量のアクセスが送り込まれている場合は、それらのPCからのアクセスを遮断することで攻撃を回避できる。

しかし、不特定多数のPCにウイルスを感染させ、それらのPCから大量のアクセスを発生させる**DDoS攻撃**(Distributed Denial of Service attack:分散型サービス拒否攻撃)の場合、それらすべてのPCからのアクセスを遮断することもできず、きわめて対応がとりづらい攻撃である。

### [4]コンピュータウイルス

コンピュータウイルスは、情報の流出、ファイルの消去やアクセス権の変更といったシステムへの破壊行為、実行権限の変更など、種類によってさまざまな障害をコンピュータにもたらす。

Webサーバにウイルス対策ソフトウェアをインストールするなどの対策を取ることはもちろんだが、サーバに接続する可能性がある作業者のマシンも同様に保護するとともに、それらのマシンを用いて不用意にファイルのダウンロードや、メールに添付されたファイルを開かない、といった運用ルールを定めておく必要がある。

### [5]人的問題

開発スタッフや運営スタッフの人為的ミスや悪意によって障害が発生することも多い。OSやWebサーバソフトウェアの設定ミス、機密情報や個人情報の記録された記録メディアやノートPCの紛失などは、不注意から生じる人為的ミスの典型である。また、故意に大量の個人情報を流出させるという事件も後を絶たない。

対策として、各種設定は必ず複数の作業者が確認する、機密情報は一度に大量のデータを閲覧できないようにするとともに、どのスタッフがどの情報にアクセスしたかという記録をすべて残す、作業に使用するノートPCの外部持ち出しを禁止する、などといった運営体制がとられる。

# chapter 6 keywords

## ■ Web サイトを実現する技術の基礎

- Web サイトは、HTTP プロトコルをベースとする WWW サービスと不可分であり、さまざまな技術や要素によって構成されている。
- Web サイトのコンテンツは、静的コンテンツと動的コンテンツに分類される。
- 動的コンテンツは、インターネットを利用するビジネスなどには不可欠な要素である。
- Web サイト制作で行われている作業は、フロントエンド側およびバックエンド側の各作業に分類される。
- 現在の Web サイトには、セキュリティの確保やアクセシビリティへの配慮などが強く求められている。

インターネット	通信プロトコル	TCP/IP
HTTP	WWW サービス	ハイパーテキストシステム
WWW クライアント	Web ブラウザ	WWW サーバ
Web サーバ	Web サーバマシン	Web サーバソフトウェア
静的コンテンツ	動的コンテンツ	データベース
Web アプリケーション	フロントエンド	バックエンド
セキュリティ	アクセシビリティ	

## ■ Web サイト上の機能

- 現在の Web サイトは、単なる情報発信にとどまらず多様なサービスを提供している。
- Web サイトが提供しているサービスを構成する各種の機能は、複数のカテゴリに分けることができる。

テキスト表示機能	HTML	リンク付きテキスト
静止画像表示機能	GIF 形式	JPEG 形式
PNG 形式	レイアウト機能	CSS
動画像再生機能	ストリーミング配信	3次元データ描画機能
WebGL	音再生機能	インタラクティブ性
双方向性	ハイパーリンク機能	検索機能
インタフェース機能	DHTML	コミュニケーション機能
掲示板	チャット	双方向通信
メール送信機能	データ連携機能	ユーザ管理機能
ID	パスワード	アカウント管理機能
コンテンツ生成・管理機能	ハンドリング機能	パーソナライゼーション機能
プロフィール	アルゴリズム	セキュリティ機能

SSL	暗号化	個人情報の保護
アクセスログ解析機能	アクセスログ	IP アドレス
エラー処理		

## ■ Web サイト制作に用いられる言語

- Web サイト制作のための言語には多種多様なものが存在し、それぞれに特徴や役割がある。
- HTML, CSS, JavaScript といった Web サイト制作の現場で頻繁に利用される個々の言語について、Web サイト制作者はできるだけその特徴や役割、問題点などを把握する必要がある。

HTML	マークアップ言語	文書構造の定義
ハイパーリンク	ハイパーテキスト	HTML フォーム
W3C	リッチインタフェース	HTML5
ドラッグ & ドロップ機能	CSS	カスケーディングスタイルシート
文書構造と体裁の分離	アクセシビリティ	CSS3
モジュール	JavaScript	スクリプト言語
ドロップダウンメニュー	ポップアップメニュー	ECMA
ECMAScript	ライブラリ	フレームワーク
DHTML	DOM	Ajax
Web アプリケーション	クラウド・コンピューティング	WebSocket
Web 標準	ロボット型検索エンジン	クローラ

## ■ バックエンドで活用する技術

- Web サイト制作で用いられる技術は、処理が行われる場所によって、フロントエンドとバックエンドの2つの技術に分類される。
- コンテンツを動的に生成するための代表的な技術が CGI である。
- データベースとの連携機能を実装した Web サイトは、高度なサービスを提供することができる。
- Web サイトは、システムの安全性やセキュリティの確保が行われて運用されることが重要である。

フロントエンド	バックエンド	Webサーバ
回線	ホスティングサービス	仮想化技術
OS	Linux系	Windows Server
LAMP	Webサーバソフトウェア	Apache
IIS	DBMS	SQL
MySQL	PostgreSQL	RDBMS
Perl	PHP	Python
Ruby	開発言語	CGI
FastCGI	モジュール化	CMS
コンテンツ管理システム	セキュリティホール	パッチ
SQLインジェクション	クロスサイトスクリプティング	DoS攻撃
DDoS攻撃	コンピュータウイルス	