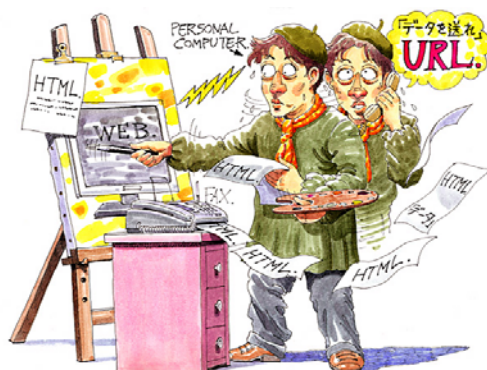


第3回 Web システムの仕組み

画面上の文字タイトルやボタンをマウスでクリックすれば、インターネット上の膨大な情報源（サーバ）にアクセスし、文字や画像によるグラフィカルな情報、音声や動画などを交えた豊かな情報を引き出してくれるのが Web システムです。Web システムはビジネスの様々な場面で、また家庭や個人レベルにおいても広く利用され、爆発的普及を遂げました。最近では PC 上だけでなく携帯電話など個人情報端末上でもウェブを利用したサービスも急速に普及しています。今回はすっかり身近となった情報ツール、Web システムの基本的仕組みについて説明してまいります。



【今回登場するキーワード】

- 「WWW (World Wide Web ワールドワイドウェブ)」
- 「HTTP (Hyper Text Transfer Protocol エイチティーティーピー)」
- 「HTML (Hyper Text Markup Language エイチティーエムエル)」
- 「URI (Uniform Resource Identifier ユーアールアイ)」
- 「URL (Uniform Resource Locator ユーアールエル)」
- 「HTML 解析機能」
- 「Content-Type (コンテンツタイプ)」
- 「レンダリングエンジン (rendering engine)」

■Web の歴史

Web が開発されたのは 1989 年、スイス・ジュネーブの CERN (Conseil Europeen pour la Recherche Nucleaire ヨーロッパ合同素粒子原子核研究機構 セルン) においてでした。開発したのは粒子加速器の制御プログラムの制作に携わっていたティム・バーナーズ＝リー氏です。

開発のきっかけは粒子加速器制御プログラムの研究が進むにつれ、実験に関するメモや印刷物が膨大となり、文書や論文などを効率的に管理する方法を模索していた状況だったといえます。こうした状況に頭を抱えていたバーナーズ＝リー氏は CERN 研究所内の文書を管理するシステムの開発を始め、1984 年、「ハイパーテキスト」と呼ばれる、全ての文書データ中のキーワード同士が互いに参照情報としてリンクする仕組みによって、研究者たちが相互に研究成果を交換するシステム作りに着手したのです。そして 1989 年 3 月、考え出されたシステムは、まだインターネットに接続するものではありませんでしたが現在の Web の原型となるものでした。

その後、同氏は同僚のロバート・カイヨー氏と Web の実現化に取り組み、自ら Web サーバ用プログラム、Web ブラウザ用のプログラムを組み上げ、また Web の基本となる HTTP、HTML、そして URI の 3 つのプログラムを組み、1990 年 11 月、Web ブラウザソフトを、同年 12 月に

Web サーバを立ち上げ Web システムの原型「WWW」を完成させたのです。

原型システムは OS や機種を問わずに機能するように作られ、CERN 内で本格的に使われ、パブリック・ドメイン（公有）化され、やがて CERN 以外でも使われるようになりました。

この当時の WWW は文字情報だけが表示され、画像データを見るためには別のソフトをダウンロードして表示する必要がありました。しかし 1992 年、米・イリノイ大学アーバナ・シャンペーン校の研究所 NCSA（National Center for Supercomputing Applications）の学生、マーク・アンドリーセン氏は WWW について独自に研究し UNIX 用の Web ブラウザ「Mosaic」を作ります。「Mosaic」は文字だけでなく画像を張り込むことができ、見栄えは雑誌のページデザインのように見やすくなりました。そして NCSA の他の学生たちの手により 1993 年 9 月、Windows 版、Macintosh 版が作られ同 11 月に公開されたのです。

「Mosaic」は文字と画像を同時に表示できる機能のほか、それ自身インストーラーによって簡単にインストールできるようになったため、プログラミング知識を持たない一般のパソコンユーザーにも利用され始め、急速にその利用者の数を増やしていきました。

「Mosaic」開発に携わったマーク・アンドリーセン氏らイリノイ大 NCSA の学生たちは 1994 年、カリフォルニア州シリコンバレーに移り、ネットスケープ・コミュニケーションズ社を立ち上げ Web ブラウザの開発に専念します。そして同年 12 月にリリースされたのがページ内の複数の画像を同時にダウンロードできる速くて使い易い「Netscape Navigator 1.0」でした。

ネットスケープ・コミュニケーションズ社はその後サン・マイクロ社との提携を続け、同社が開発した Java 言語によってブラウザの表現力をさらに向上させようとしています。こうして誕生したのが、Java のスクリプト言語（後に「JavaScript」と命名）を搭載した「Netscape Navigator 2.0」β版で、正式版は専門の知識を持つプログラマー以外の一般ユーザーにも広く使われるようになったのです。

1995 年までには、プロキシサーバ、「Cookie」技術などが開発され、現在の Web システムの基盤が確立しました。そしてマイクロソフト社が独自の Web ブラウザ「Internet Explorer」をリリースし、ネットスケープ・コミュニケーションズ社とで熾烈な競争を繰り広げる「ブラウザ戦争」と呼ばれる時代に入っていきます。その後は徐々にマイクロソフト社がシェアを伸ばし、やがて「Internet Explorer」が Web ブラウザの代名詞となるまでに普及していくのです。

1989年	ティム・バーナーズ＝リーによって「WWW」の原型が考案される。
1990年	バーナーズ＝リーとロバート・カイヨー氏、CERNにおいて「WWW」を正式に提案 バーナーズ＝リー、Web ブラウザを開発。
1991年	バーナーズ＝リー、カイヨー氏ハイパーテキスト学会で「WWW」をデモ
1992年	インターネット標準化組織 IETF (Internet Engineering Task Force) において URI (URL) 標準化のセッションが開かれる。
1993年	「WWW」パブリック・ドメイン化される。 「WWW」を見たイリノイ大学 NCSA の学生、マーク・アンドリーセン氏「Mosaic」開発。 「Mosaic」の Windows 版、Macintosh 版リリース (11月)
1994年	アンドリーセン氏、シリコンバレーでネットスケープ・コミュニケーションズ社設立。 「Netscape Navigator 1.0」リリース。
1995年	マイクロソフト社「Internet Explorer 1.0」リリース (8月)。 Java スクリプト、Cookie 搭載の「Netscape Navigator 2.0」β版公開 (9月)
1996年	「Netscape Navigator 2.0」リリース (3月) 「Netscape Navigator 3.0」リリース (8月) 「Internet Explorer3.0」リリース (8月)
1997年	「Internet Explorer4.0」リリース (10月)
1998年	ネットスケープ・コミュニケーションズ社「Netscape Navigator」ソースコードを公開しオープンソースのブラウザ「Mozilla」開発スタート ネットスケープ・コミュニケーションズ社、AOL に買収される。
1999年	この頃「Internet Explorer」が「Netscape Navigator」のシェアを追い抜く
2000年	Mac OS 用の「Internet Explorer」発表
2002年	オープンソースのブラウザ「Mozilla 1.0」公開 (6月) 「Netscape Navigator 7.0」公開
2004年	「Mozilla Firefox 1.0」公開。「Internet Explorer」のセキュリティ上の脆弱性が指摘される。
2005年	「Netscape Browser 8.0」公開
2006年	「Mozilla Firefox 2.0」公開

【図1】ウェブ関連技術の歴史

●Webシステムのルーツ「ハイパーテキスト」四半世紀前の提案

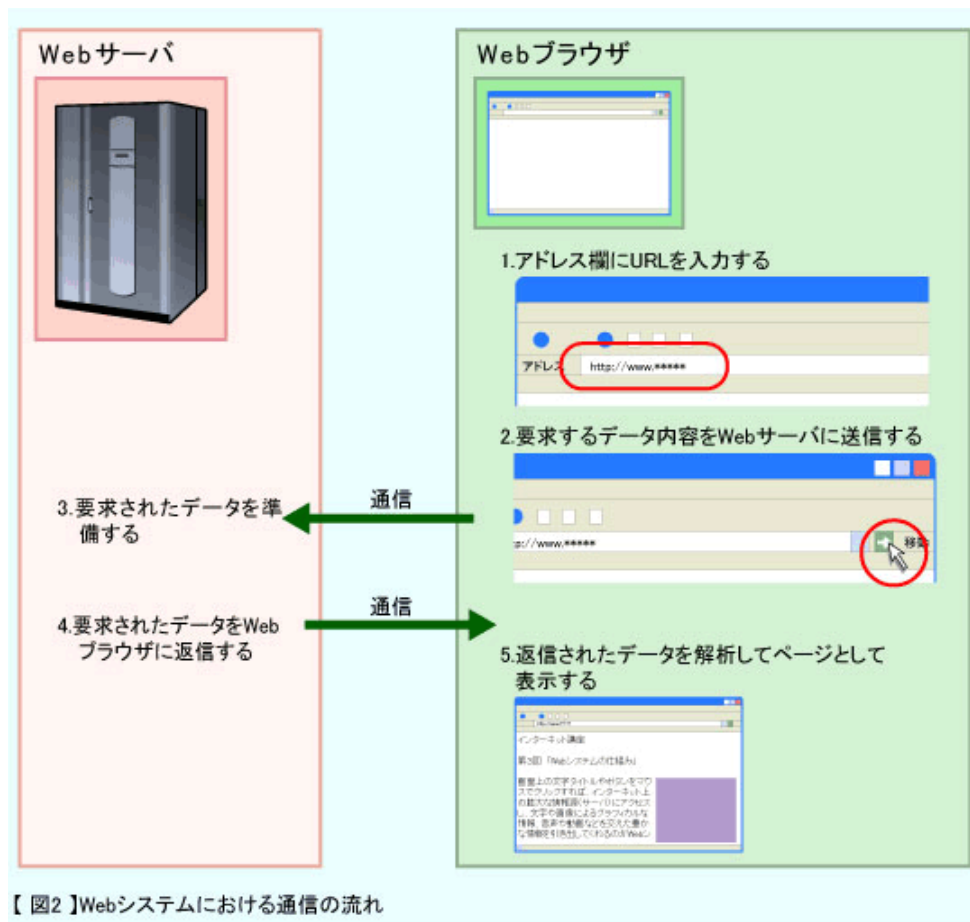
Webの基本的な仕組みともいえるユーザー・インターフェースの「ハイパーテキスト」を考え出したのはアメリカのコンピュータサイエンティストのダグラス・エンゲルバート氏です。1963年、エンゲルバート氏は戦時中、海軍においてレーダー技術者としてレーダー操作に従事した経験から、ディスプレイ画面を介して人間があたかもコンピュータと対話をするように操作できないかと考えたのです。さらに彼は、ある論文（膨大なマイクロフィルム1枚1枚に識別コードを振って効率よく呼び出し参照する機械装置を紹介）をヒントに、コンピュータをより使いやすいものにしようと取り組み、1968年コンピュータディスプレイ上に表示された文章の一部をクリックすると、関連する別の文書が即座に表示される「ハイパーテキスト」と呼ばれる全く新しいユーザー・インターフェースを発表し、コンピュータ技術者を始め多くの人々に衝撃を与えたのです。

この時発表されたシステムはマウス、そして1つの画面に複数の「ウィンドウ」を表示する機能を備え、ほぼ現在のPC（パーソナルコンピュータ）の原型を成すものでした。エンゲルバートが構想した「ハイパーテキスト」が現実のものとなったのは、インターネット、TCP/IPプロトコルとこれらの技術の上に成り立つWebシステムが実現した1980年代後半、四半世紀後でした。

■Web通信の流れ——5つのステップ

「Webサイトを見る」「Webで調べる」という場合、WebブラウザにURL（Uniform Resource Locator）を打ち込むとたちどころにブラウザ上に文字や画像が表示されます。この時、WebブラウザとWebサーバは次のように働き、やり取りをしています。

1. Webブラウザのアドレス欄にURLを入力する
2. Webブラウザが、要求するデータ内容をWebサーバに送信
3. Webサーバが要求されたデータを準備する
4. Webサーバが要求されたデータをWebブラウザに返信
5. Webブラウザが送られたデータを解析してページとして表示する



●安全なアクセス法として開発された「プロキシサーバ」

「proxy」（英語）は「代理」という意味を持っています。もともと「プロキシサーバ」は代理機能を果たすサーバで、LAN内クライアントの発信時にローカルIPアドレスをグローバルIPアドレスに置き換え、また受信時には逆にグローバルIPアドレスをローカルIPアドレスに置き換える、アクセスの集約・代理の役割を担うものでした。この機能によって外部からLAN内のローカルIPアドレスが見えないようになり、結果的にセキュリティ機能をはたすことになりました。Webシステムは当初、研究所（CERN）内の文書管理システムとして考案されたもので、インターネットとは接続されていませんでした。しかしWebシステムの利便性が認知されるにつれ、外部のWebサーバに自由にアクセスするニーズは高まっていきました。プロキシサーバはこの問題を解決し、自由にWebサーバへアクセスすることが可能になったのです。多くの企業内ネットワークではファイアーウォールサーバ内に「プロキシサーバ」を設置しています。

「プロキシサーバ」のもう一つの役割は、頻繁にアクセスするWebページのデータを蓄積するキャッシュ機能です。次のアクセス時に蓄積したデータを直接Webブラウザに返信する（キャッシング）ことでWebブラウザへの表示が速くなり、またインターネット回線の混雑が軽減されるのです。

■Web ブラウザの仕組み

これら Web 通信の 5 つのステップを支える Web ブラウザの基本機能は次の 3 つです。

1. 通信機能

Web ページ作りに必要なデータがインターネット上のどこにあるかを示す住所 URL から通信先の Web サーバを特定し、ドメイン名に対応する IP アドレスのデータを持つ DNS サーバを検索。求めるデータの内容を Web サーバに知らせ、データを受け取ります。HTTP (Hyper Text Transfer Protocol エイチティーティーピー) と呼ばれるプロトコルにしたがってデータの送受信をします。

2. HTML (Hyper Text Markup Language エイチティーエムエル) 解析機能

HTML とは文書の記述方式で、異なる種類・性能の PC や OS でも問題なくやり取りできるよう文書の構造や文字のサイズやフォントの種類などを伝えるスタイル (マークアップ言語) になっています。HTML 解析機能とは Web サーバから送られた HTML データの意味を読み取ったり、必要に応じて Web サーバ内の画像ファイルのダウンロードを通信機能に要求したりすることです。

3. 描画機能

HTML の解析結果に基づいて文字の大きさ、フォントの種類、画像や表、フレームなどの配置を決め描画する。

どのような Web ブラウザもこれらの機能を使ってサーバと通信し、情報を取り寄せ、文字や画像を表示しているのです。1989 年にティム・バーナーズ＝リー氏が WWW のシステムを考案して以来この方式は基本的に変わっていません。また、携帯情報端末、携帯電話に搭載された Web ブラウザも同様です。Web システムが Java スクリプトを搭載し進化し、データベースと連携して多機能化し、また PC 以外の情報端末機器に搭載しこれほどまでに普及してきたのも、URL、HTTP、HTML の比較的シンプルな要素技術によって成り立っているからであるともいわれています。

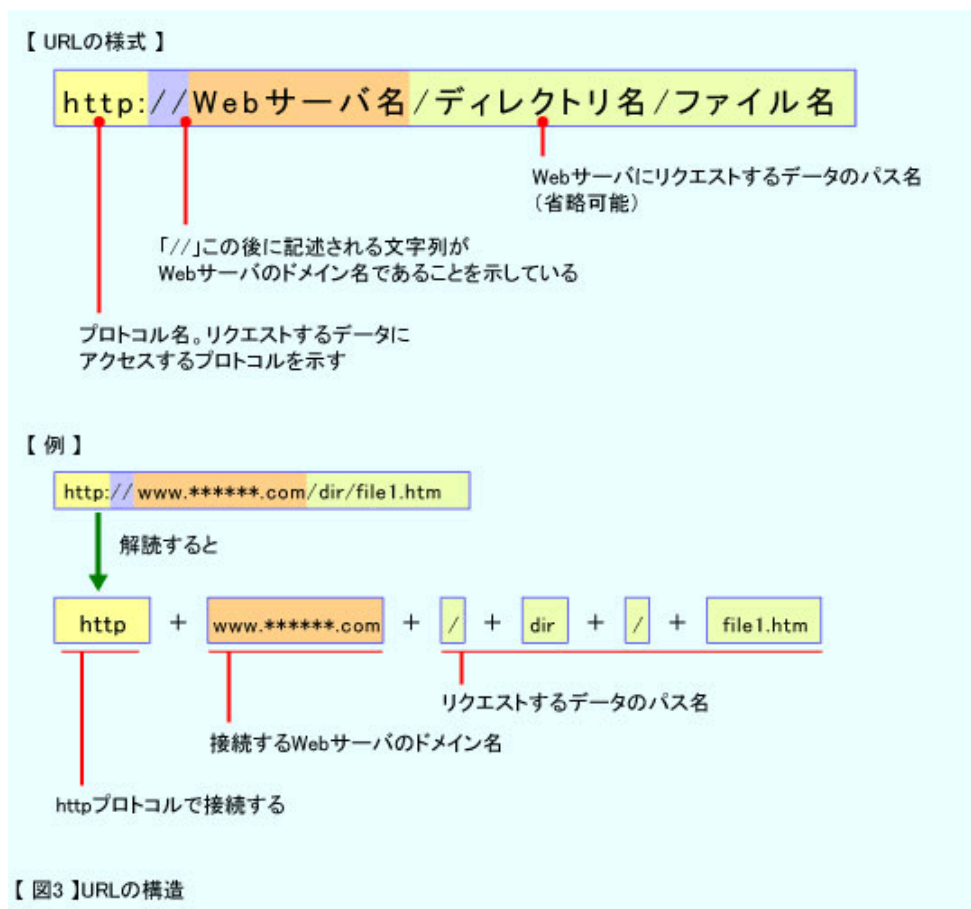
ではユーザーが URL を打ち込んでから Web サーバのデータが Web ブラウザ上に表示されるまでを、3 つの基本機能に触れながら詳しく見てみましょう。

■Web ブラウザの通信機能

・ URL の解説

ユーザーがアドレス欄に URL を打ち込むと、Web ブラウザの通信機能に URL の内容が知らされます。Web ブラウザはこの URL を解説し、どの Web サーバにアクセスしどのデータを要求すべきか判断しています。

よく私たちがアドレス欄に打ち込む URL は、実は図 3 のように 3 つの要素から成り立っているのです。



「http:」の部分は、Web ブラウザと Web サーバのやり取りで使用されるプロトコルを示しています。参考までにいえば URL は「http:」で始まるとは限りません。例えばファイルを転送する場合はファイル転送プロトコルを示す「ftp:」となるのです。

「www~」に続く部分はドメイン名で、Web サーバに割り振られた固有の住所です。WWW が開発された当時、すでに e メールシステムでは DNS が利用されていました。Web システムもインターネット上に散在する Web サーバから目的のサーバを見つけ出すために DNS を利用することになったのです。

「/」に続く「dir/file1」は Web サーバ内のどのデータかを示す部分です。

URL を解読し「HTTP プロトコルによって www.*****.com というサーバに接続し、file1 というパス名のデータを読み出す」と理解した Web ブラウザは、次に Web サーバのドメイン名から IP アドレスを探します。IP アドレス探しは e メールシステムと同じように、ドメイン名がどの IP アドレスに対応しているかのデータを持つ DNS サーバに問い合わせる方法です。

● 「リゾルバ」

Webブラウザは、接続するWebサーバのIPアドレスを探し出すためにDNSサーバに問い合わせを行います。より詳しくいえばDNSサーバへの問い合わせと、返送されたメッセージを受け取るのは「DNSリゾルバ」というソフトで、TCP/IPの一部としてOSが標準で備えています。

「DNSリゾルバ」はWebブラウザに限らずいろいろなネットワークアプリケーションによって利用されます。各ネットワークアプリケーションが「DNSリゾルバ」機能を持つより、TCP/IPの一部として用意し、各ネットワークアプリケーションで共用する方が効率的というわけです。

「リゾルバ」という呼び名は、DNSによってIPアドレスを調べることを「ネーム・リゾリューション」（名前を解決=resolutionするの意味）と呼ぶところから「解決するもの=resolver」を語源にしたものです。

・通信の開始

通信するWebサーバのIPアドレスが分かるといよいよ通信が始まります。具体的にはWebブラウザが「HTTPリクエスト」を作成しWebサーバに送信します。

「HTTPリクエスト」は図4のように3つの部分からなっています。最も重要な内容は1行目のリクエスト。ここにはWebサーバに対して何をしたいかが書かれています。図の例では、

【GET /file1.html HTTP/1.1】

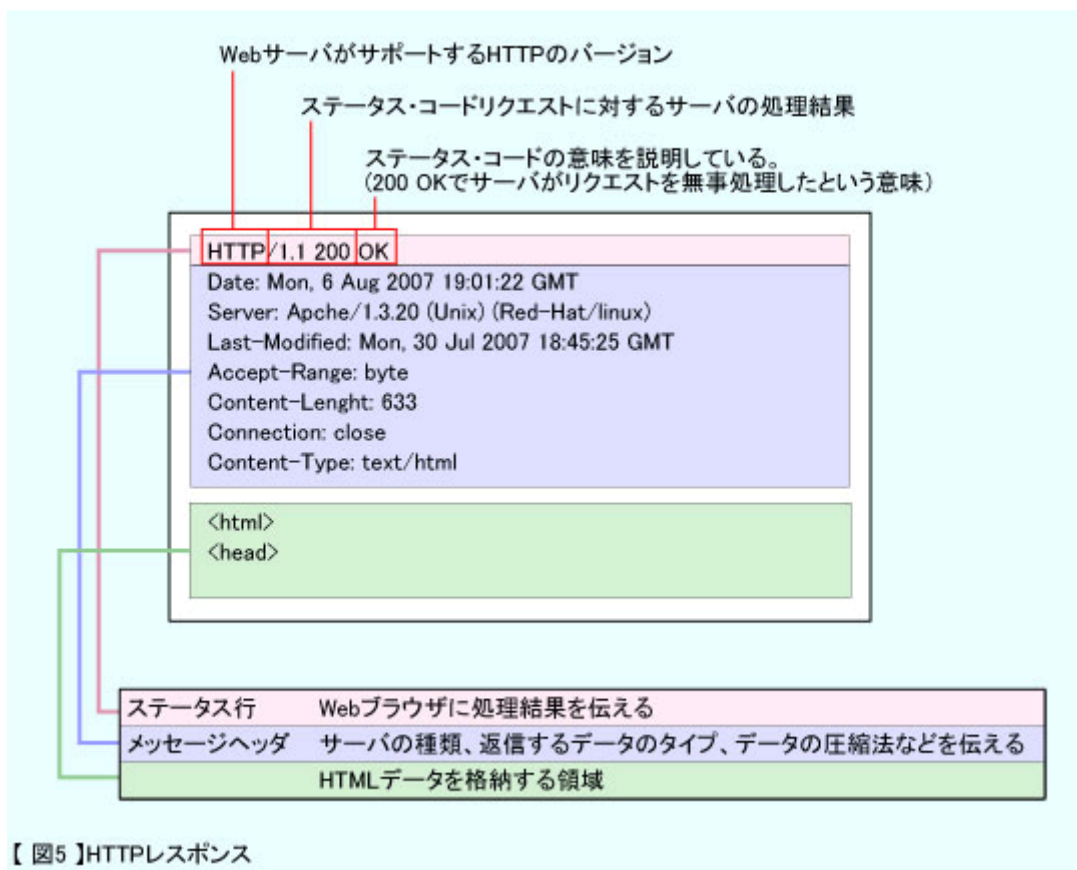
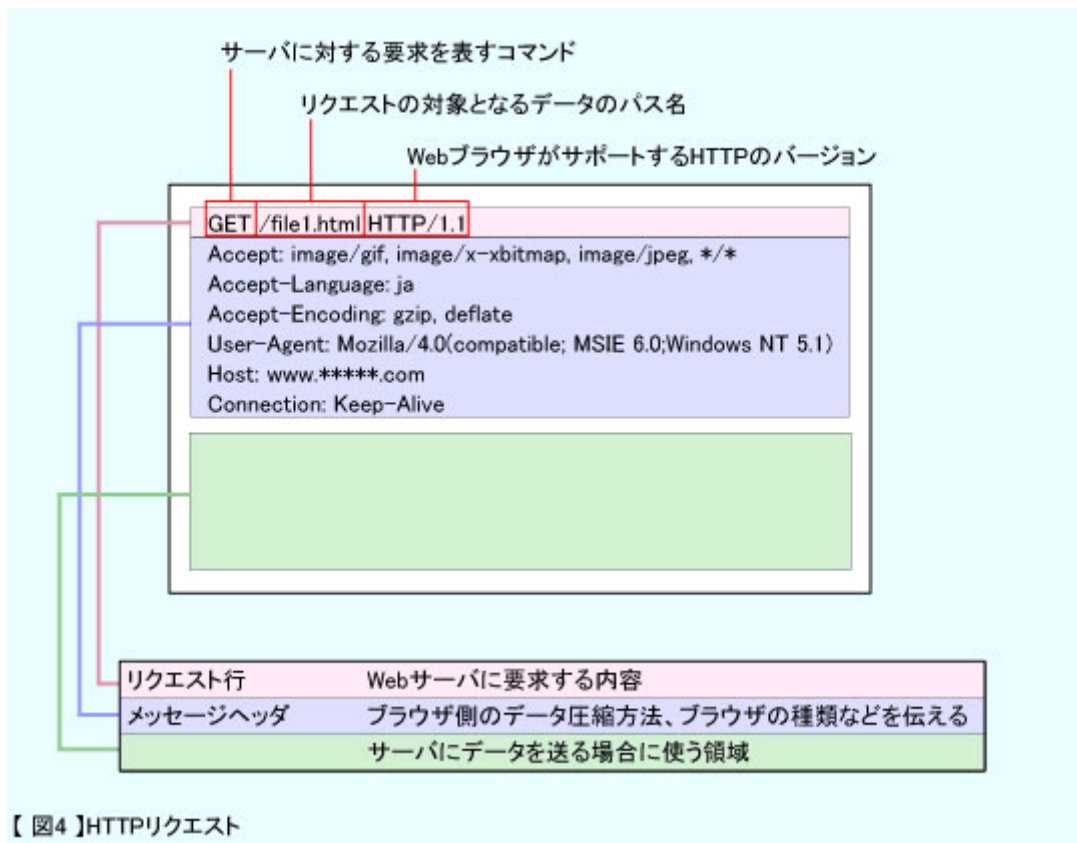
と書かれています。サーバに対する要求のコマンド「GET」は「メソッド」と呼ばれています。主に使われるのはWebサーバにデータ送信を依頼する「GET」、Webブラウザ側から大きなデータを送信する場合に用いる「POST」ですが、他にも指定したURLにファイルをアップロードする「PUT」などいろいろな「メソッド」があります。例に示したリクエストは「file1.html」のパス名で指定されるデータを送れ」ということ伝えています。

一方、WebサーバはWebブラウザからリクエストを受けると、HTMLデータを返信します。このデータは「HTTPレスポンス」と呼ばれています。構造的にはステータス行、メッセージヘッダ、HTMLデータを格納するボディ部分の3つの部分からなり「HTTPリクエスト」によく似ています。やはり、最も重要な内容は第1行目に書かれていて、図5の例の、

【HTTP/1.1 200 OK】

は、WebサーバのHTTPバージョンは1.1で、サーバがWebブラウザのリクエストに対して問題なく処理を終えたことを意味しています。3ケタの番号で示された「200」は「ステータス・コード」と呼ばれ、リクエストに対する処理結果を表すもので、100番台は「追加情報がある」、200番台は「サーバがリクエストの処理に成功した」、300番台は「別に指定したURLにリクエストし直すように」、400番台は「クライアントあるいはリクエスト内容に問題がある」などを伝えています。

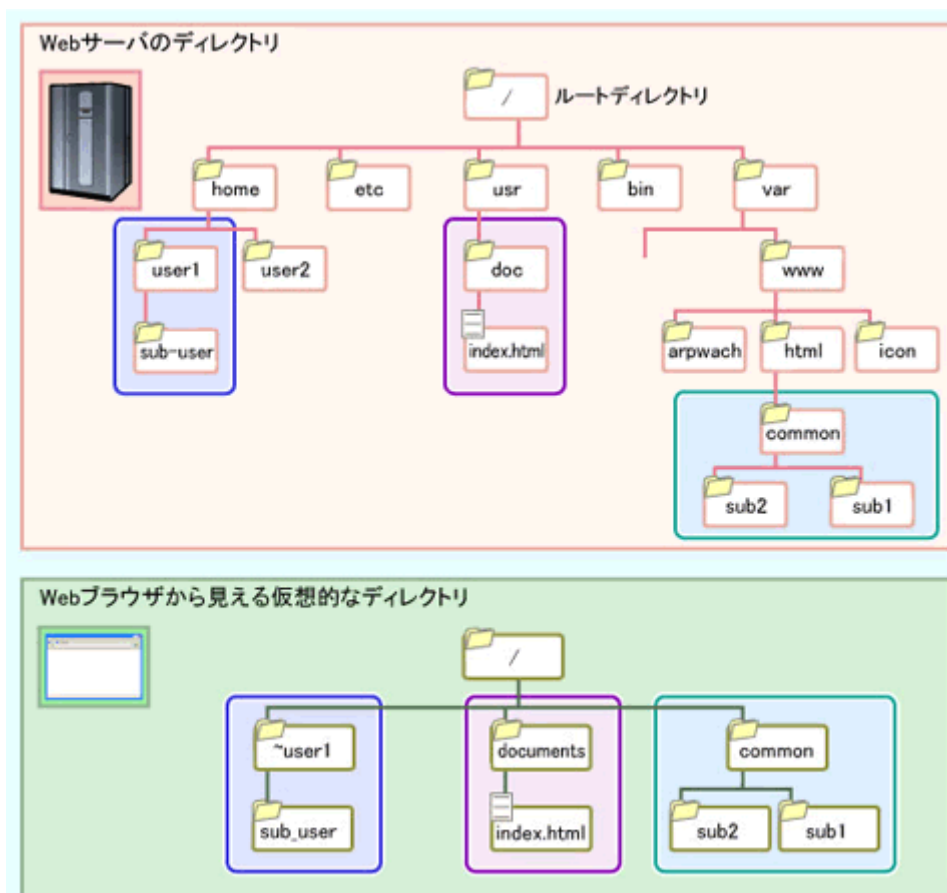
ステータス行の下には、返信データの形式や大きさなどに関する詳細な情報が記されたメッセージヘッダが続き、その後にWebブラウザに返信するデータが入る領域が続きます。



実際の Web ブラウザと Web サーバのやり取りは何度も繰り返して行われ、1回で Web ページが完成することはまれです。例えば画像が張り込まれている Web ページの場合、Web サーバから送信された HTML データ 1 行 1 行に応じてページを描きながら、<画像を張り込む>という指示（タグ）にさしかかると、Web ブラウザは指示された Web サーバにアクセスして画像データを送信してもらうことになります。画像データを数多く含む Web ページではその都度画像データのある Web サーバへのアクセスを繰り返してページを作ることになるわけです。

●仮想ディレクトリでリクエストに応える Web サーバ

Web サーバは Web ブラウザからリクエストを受け取ると、HTTP リクエストの第 1 行目に書き込まれた「要求されたパス名」に対応するデータを探して読み出します。しかしこのパス名は実際のディスク上のパス名を直接指し示すものではありません。ほとんどの Web サーバは Web ブラウザからのアクセスに対して仮想ディレクトリを見せています。サーバ内の実際のディレクトリとの対応関係を調べ、仮想ディレクトリのパス名を実際のディレクトリのパス名に変換してから読み込み、データを送り返しているのです。なぜなら、要求されるパス名とディレクトリのパス名が同一であると、Web サーバ側のディレクトリの全てのファイルにアクセスできるようになりセキュリティ上きわめて危険な状態になってしまうからです。



【図6】仮想ディレクトリの概念

■Web ブラウザの HTML 解析機能

・「Content-Type」のメッセージヘッダを解析する方法

Web ブラウザは Web サーバから送られた HTTP レスポンスを読み、HTML データ格納エリア内のデータの種別を調べます。これは Web ページが文章、画像、音声、映像などいろいろな種類のデータを取り扱い、それぞれに適した表示方法で対応するためです。

データの種別を判断するために、Web ブラウザはまず HTTP レスポンスのメッセージヘッダの「Content-Type」に示された値を確認します。図5の「Content-Type」では、

【Content-Type : text/html】

となっていますが、「/」左部分を「主タイプ」、右部分を「サブタイプ」としてデータ形式を分類しています。こうして確実にデータの種別を把握することで Web ブラウザは適切な表示方法で対応するのです。この場合は、HTML 仕様でタグを埋め込んだ HTML ドキュメントであることを表しています。「Content-Type」に示される主タイプ・サブタイプの意味は HTML 仕様によって定められています（図7）。

「Content-Type」のフィールドを参照してデータの種別を把握する方法は一般的ですが、この方法が正しく機能するためには Web サーバの「Content-Type」値が正しくセットされていなければなりません。しかし実際には必ずしも全ての Web サーバにおいて正しくセットされているわけではなく、「Content-Type」を参照するだけではデータの種別を把握できない可能性もあります。そのため最近の Web ブラウザでは「Content-Type」の値だけに頼るのではなく、返信されてきたファイルの拡張子や HTML データ中のタグやフォーマットから多角的に判断するようになっています。

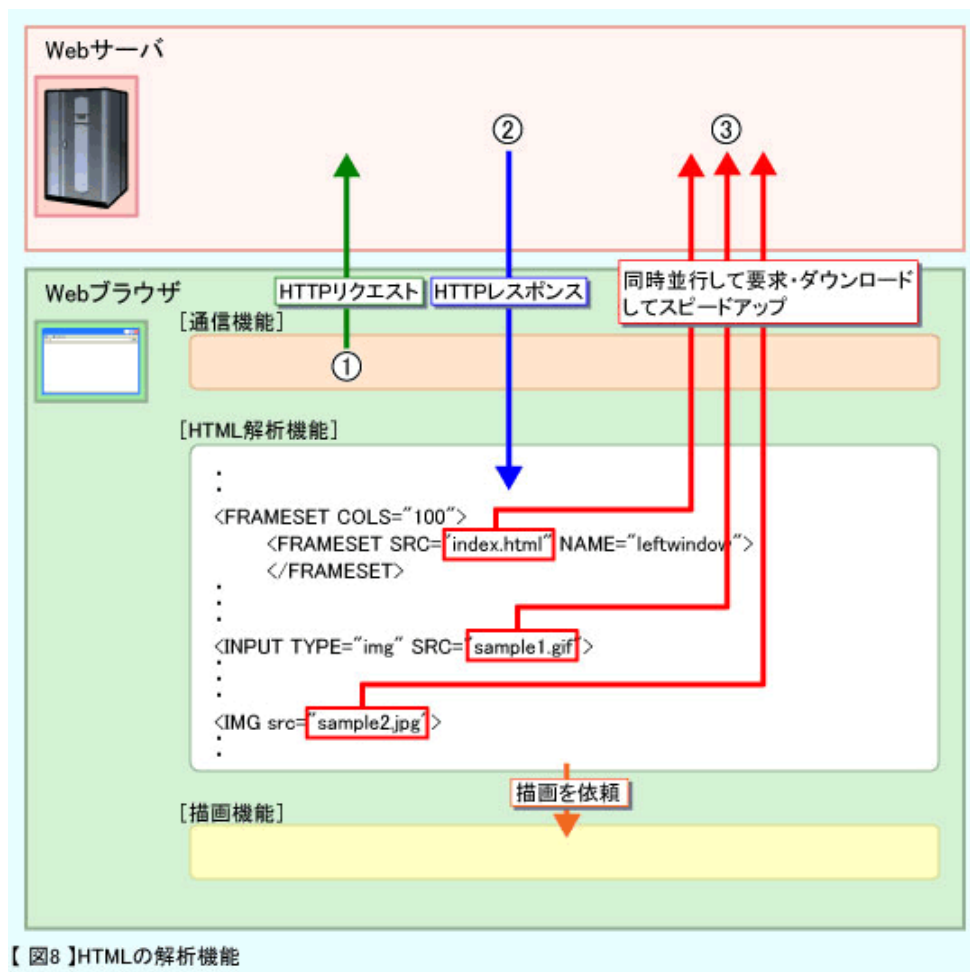
主タイプ	意味	サブタイプの例	意味
text	テキストデータを表す	text/html text/plain	HTML ドキュメント プレーンテキスト
image	画像データを表す	image/jpeg image/gif	JPEG 形式の画像 GIF 形式の画像
video	映像データを表す	video/mpeg video/quicktime	MPEG 形式の映像 Quicktime 形式の映像
audio	音声データを表す	audio/mpeg	MP2,MP3 形式の音声
application	上記以外のデータを表す。 Excel、Word などのアプリ は全てこのタイプ	application/pdf application/msword	PDF 形式の文書データ MS-WORD の文書データ
message	メールメッセージを別のメ ッセージに格納する場合に 使うタイプ	message/rfc822	通常タイプのメールデータ。 「From:」「Date:」などのヘ ッダを含む
multipart	メッセージボディ領域に複 数のデータが入っているこ とを表す	multipart/mixed	異なる形式の複数のデータ がメッセージボディ領域に 格納され、それぞれにメディ アタイプが記載されている

【図7】 Content-Type で指定される主要なデータ形式

・小プログラムが集合体化した Web ブラウザ

Web ブラウザは当初、テキストデータしか扱えず、Web サーバから受け取った HTML データを最上段行から逐次処理しながら画面を作成していくシンプルなプログラムでした。しかしその後の Web ブラウザは画像、音声や動画の表現機能を備え、1995 年頃から激しくなるブラウザ開発競争（「ブラウザ戦争」）を経て JavaScript を搭載、プラグインによって一段と多機能化するなど、いわば小さなプログラムの集合体となってきました。

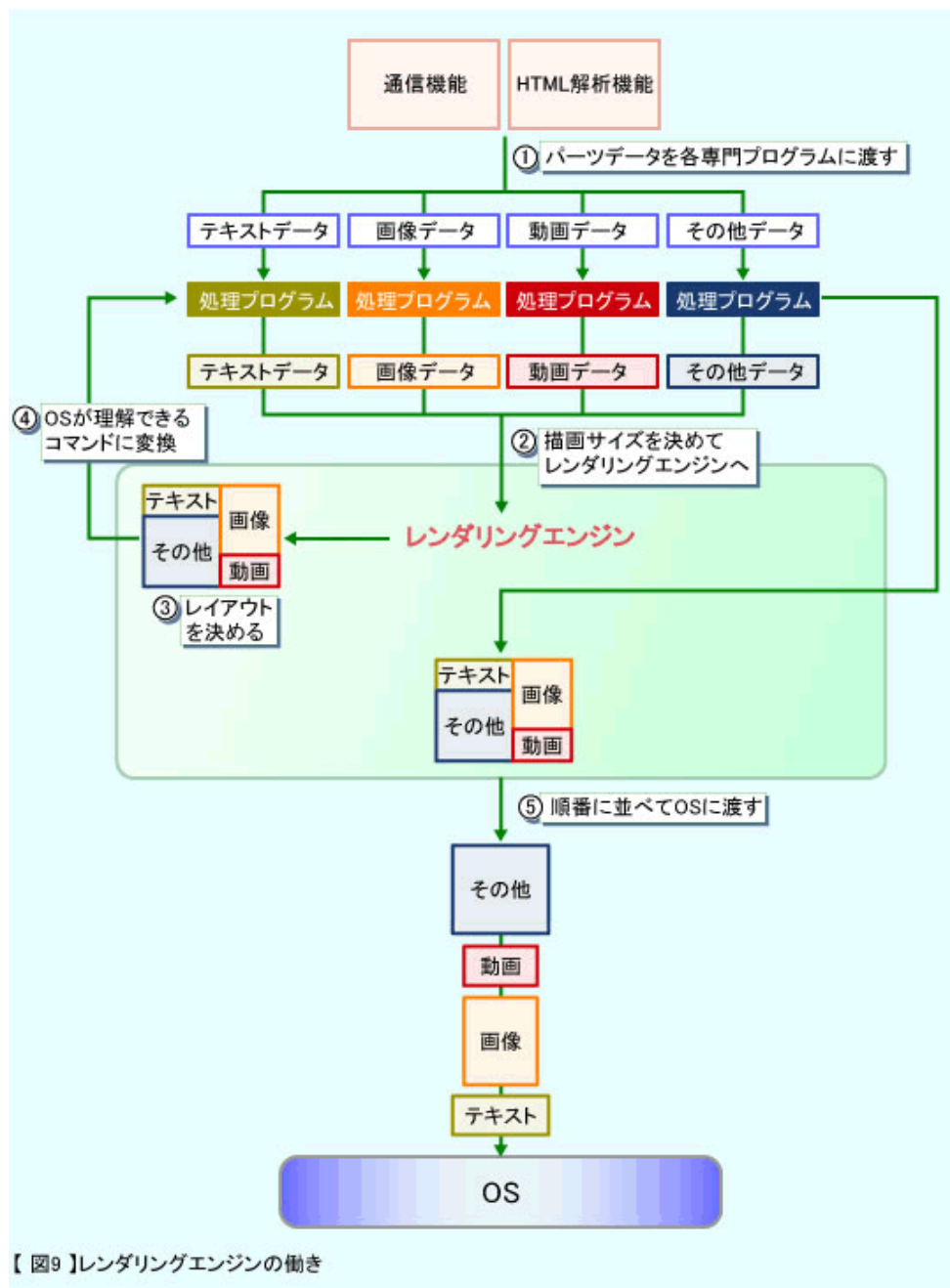
その結果、Web ブラウザは受信した HTML データを調べ、適切な小プログラムへ手渡す解析機能も向上し、複数のプログラムでデータを並行処理させるなどで Web ページの表示速度は大幅に向上しました（図8）。



■ 「レンダリングエンジン」による Web ページ描画

HTTP 解析機能によってデータの種別が特定されると、各データを Web ページに表示するプログラムが表示の処理を進めます。プレーンテキストや画像などの基本的なデータは Web ブラウザ自身が処理しますが、それ以外のデータは専門プログラムが処理します。この時、各専門プログラムが勝手に処理を始めると混乱するため Web ブラウザの描画機能を司る「レンダリングエンジン (rendering engine)」（レイアウトエンジンともいう）と連携して描画処理を調整します。

「レンダリングエンジン」と各専門プログラムは図9のように連携していきます。まず HTML 解析機能や通信機能からそれぞれの専門プログラムへ Web ページを構成する各種データが送られます。専門プログラムはデータを解析して描画サイズを決めそのデータを「レンダリングエンジン」へ渡します。「レンダリングエンジン」は解析データを読み込み、Web ページの枠内に収まるようにレイアウトを決めます。こうして決められたレイアウト用に処理された各種データは専門プログラムによって OS が理解できるコマンドに変換され、「レンダリングエンジン」から順番に OS に渡され、適切な位置に文字や画像を表示するのです。



■おさらい

インターネットを利用した、もっとも便利で身近な Web システムの基本的システムについて見てきました。主なポイントは次の通りです。

- 1) Web システムは膨大な文書データを多くの人々で効率的に共有するシステムとして誕生した。
- 2) Web システムは「Web ブラウザが Web サーバに対してデータを要求する」、「Web サーバが要求されたデータを Web ブラウザに返信する」という流れになっている。
- 3) Web ブラウザは、「通信機能」「HTML 解析機能」「描画（レンダリング）機能」3つの技術によって成り立つ比較的シンプルな構造。
- 4) Web ブラウザはシンプルな構造ゆえに、数々の機能が盛り込まれ高機能化してきた。

今回は Web ブラウザを高機能化、便利にしている数々の拡張機能、ヘルパーアプリケーション、プラグインの仕組みなどを紹介してまいります。