

インターネット数理科学第3回 ～ネットワークのこちら側を支える数理科学その1～

2006年10月19日

株式会社インターネット総合研究所代表取締役所長
東京大学大学院数理科学研究科客員教授

藤原 洋

目 次

1. ネットワークのこちら側を支える数理科学とは？
2. WWWと「ネットワークのこちら側」のブラウザとは？
3. WebブラウザのためのHTMLとは？
4. HTTPとは？

1. ネットワークこちら側を支える数理科学とは？

③(ネットワークの)あちら側

⇒「グラフ理論」「金融工学理論」に基づくデータベース、検索エンジン最適化、検索連動データベース、ネット金融サービス

①ネットワークそのもの

⇒「グラフ理論」による動的ルーティング、帯域制御、放送型ルーティング
「デジタル信号処理理論」に基づく変復調技術

②(ネットワークの)こちら側

⇒「デジタル信号処理理論」に基づくコンテンツ符号化技術

以下の3つの分野にわたって①②③⇒①②③⇒・・・順に

③ネットワークのあちら側を支える数理科学

⇒「グラフ理論」「金融工学理論」に基づくデータベース、検索エンジン最適化、検索連動データベース、ネット金融サービス

①ネットワークそのものを支える数理科学

⇒「グラフ理論」による動的ルーティング、帯域制御、放送型ルーティング
「デジタル信号処理理論」に基づく変復調技術

②ネットワークのこちら側を支える数理科学

⇒「デジタル信号処理理論」に基づくコンテンツ符号化技術

③(ネットワークの)あちら側

Web1.0(ポータル) ⇒ Web1.5(SNS) ⇒ Web2.0(ロングテール)

①ネットワークそのもの

ダイヤルアップ/2Gモバイル ⇒ ブロードバンド/3Gモバイル ⇒ IP放送/NGN/WiMAX

*NGN: Next Generation Network

*WiMAX: Worldwide Interoperability for Microwave Access

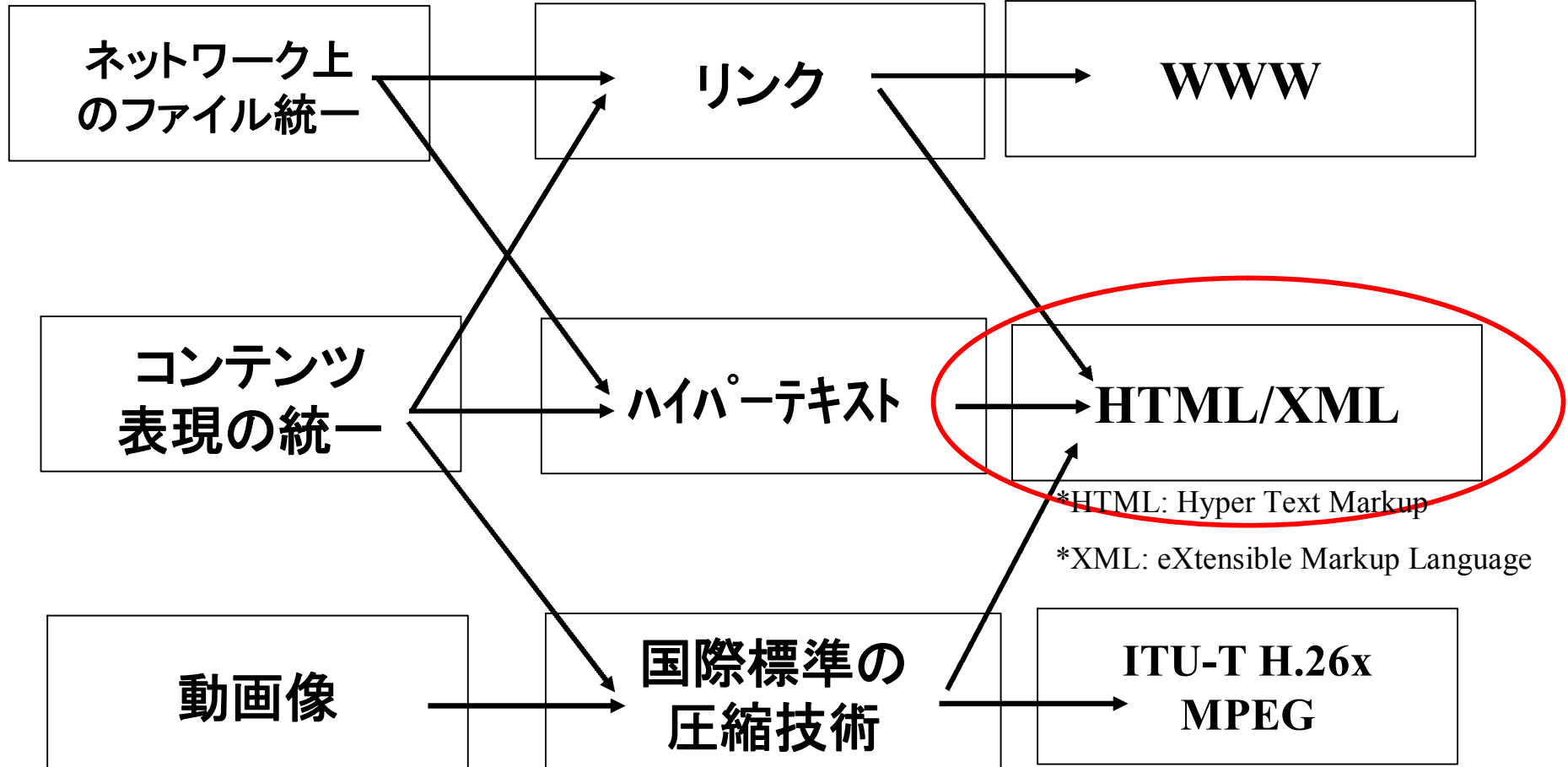
②(ネットワークの)こちら側

文字情報(Eメール) ⇒ HTML(ブラウザ) ⇒ 動画(デジタル符号変換)

課題

着眼点

具体策



*HTML: Hyper Text Markup

*XML: eXtensible Markup Language

*ITU-T: International Telecommunications Union-Telecommunication Sector

*MPEG: Moving Picture Experts Group

2. WWWと「ネットワークのこちら側」のブラウザとは？

私が、訪問研究員として米ベル通信研究所で第2回目のMPEG会合に出席していた頃、WWW (World Wide Web)が発明されました。この全く異なる2つの技術が世界を大きく変えることになったのでした。WWWは、欧州核物理学研究所 (CERN)のティム・バーナーズ・リーが論文閲覧システムとして1989年に考案、91年に一般公開された、データベース研究者も気づかないほどデータが存在する場所 (URL)にリンクを張るだけのシンプルなアイデアでした。商用化前のインターネットで論文検索が飛躍的に便利になったのを覚えています。92年、イリノイ大学の学生マーク・アンドリーセンが、HTMLで文書の論理構造やグラフィックスを記述し、画像や音声など文字以外のデータや、他の文書のある場所(ハイパーリンク)を埋め込み、自由に取り出せる仕組みを考えました。Webページを閲覧するためのアプリケーションソフト(ブラウザ:閲覧ソフト)で、92年から普及、93年のネットスケープ社の登場、マイクロソフト社による後発参入と席卷などが起こり、現在では世界中のWebサーバをつなぐ巨大網へ発展、激しいメディアビジネスの戦場を用意しました。

このように「ネットワークのこちら側」の要素技術である、「ブラウザ」が、その後、検索エンジン、金融、電子商取引、オークション、Web掲示板、SNSなどの多くの「ネットワークのあちら側」のサービスの出現を促しました。

数学の一分野。ノード(節点・頂点)の集合とエッジ(枝・辺)の集合で構成されるグラフの性質について研究する学問である。

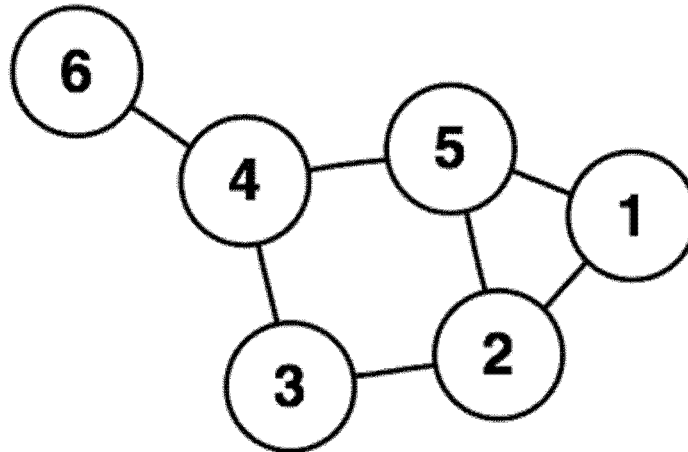
その応用分野としては、コンピュータのデータ構造、アルゴリズム、ネットワーク理論など広範囲にわたっている。

例えば電車の乗り換え案内図を考える際には、駅(ノード)がどのように路線(エッジ)で結ばれているかが問題であって、線路が具体的にどのような曲線を描いているかは本質的な問題でないことが多い。

事実、乗り換え案内図を書く場合には、駅間の距離や微妙な配置、路線の形状といったものは、地理的な実際のそれとは異なって描かれることが多い。電車で移動する人を対象とした乗り換え案内においては、駅と駅の「つながり方」が本質的に重要である。

このように、「つながり方」に着目して抽象化された「点とそれをむすぶ線」の概念がグラフであり、グラフが持つ様々な性質を探求するのがグラフ理論である。

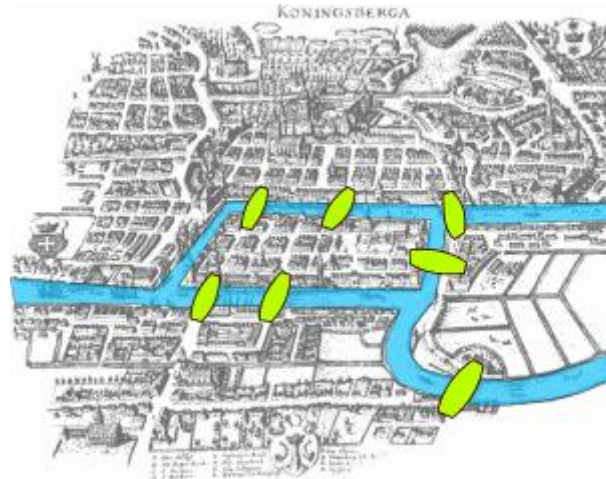
つながり方だけではなく「どちらからどちらにつながっているか」をも問題にする場合、エッジに矢印をつける。このようなグラフを有向グラフという。矢印のないグラフは、無向グラフという。



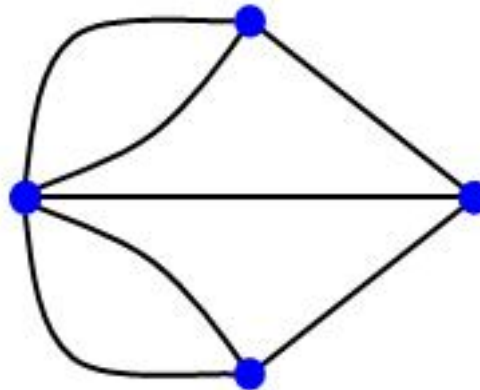
6つのノードと7つのエッジから成るグラフの一例

「1736年にレオンハルト・オイラーが解いたことが起源」

ケーニヒスベルク(現在のロシア領カリーニングラード)を流れるプレーゲル川には、市の中心地で大聖堂の建っている中州クナイプホーフ島ともう一つの大きな中洲があり、それらを中心に両岸へ下図のように7つの橋が架かっている。この7つの橋を各1度ずつ通って、元の場所に戻ってくることができるかどうか? ただし、同じ橋を2度以上通ってはならない。



1736年、レオンハルト・オイラーは、この問題を以下のグラフに置き換えて考えた。このグラフが一筆書き可能であれば、ケーニヒスベルクの橋を全て1度ずつ通って戻ってくるルートが存在することになる。そして、オイラーは、このグラフが一筆書きできないことを証明し、ケーニヒスベルクの問題を否定的に解決した。



一筆書き可能かどうかの判定法

ある連結グラフが一筆書き可能な場合の必要十分条件は、以下の条件のいずれか一方が成り立つことである。

- ・すべての頂点の次数が偶数 → 運筆が起点に戻る場合(閉路)
- ・次数が奇数の頂点の数が2で、残りの頂点の次数は全て偶数
→ 運筆が必ずしも起点にもどらない場合(路)

WWWにおけるウェブページの、リンク・被リンク関係がなす構造は、有向グラフの一種である。

World Wide Web (WWW、ワールド・ワイド・ウェブ) は、インターネットで提供されるハイパーテキストシステム。単にWeb (ウェブ) と呼ばれることも多い。インターネットは本来、TCP/IPで接続されたコンピュータ・ネットワークを指す言葉であるが、日常用語ではWWWのことだと解釈している人も多い。

プロトコル (通信プロトコル、あるいはネットワーク・プロトコルの意味では、ネットワーク上での通信に関する規約を定めたもの) は、主にHTTPが使用される。またドキュメントの記述には主にHTMLやXHTMLなどのハイパーテキスト記述言語が使用され、ドキュメントに別のドキュメントのURIへの参照を埋め込むことで (これをハイパーリンクと呼ぶ) インターネット上に散在するドキュメント同士を相互に参照可能にすることができる。分かりやすい例で言うと、主にマウスによるクリックなどによってページ間を移動することや、別のファイルである画像をドキュメント内に表示させることなどが挙げられる。そのつながり方が蜘蛛の巣を連想させることからWorld Wide Web (世界に広がる蜘蛛の巣) と名付けられた。尚、蜘蛛の巣は現実のケーブルの配線を表しているわけではない。HTMLの記述方式は比較的単純なため、急速に広く普及した。

WWWにアクセスするためのソフトウェア (ユーザーエージェント) はWWWクライアントと呼ばれる。そのうち、人間による閲覧を目的としたものは特にウェブブラウザ (WWWブラウザ、あるいは単にブラウザ) と呼ばれる。また、WWWサービスを提供するソフトウェアをWWWサーバソフトウェアという。

Webの根底にある考え方は1980年にティム・バーナーズ＝リーが Robert Cailliau と構築したENQUIREに遡ることができる(ENQUIREは一般に公表されるまでいかなかった。その名称は Enquire Within Upon Everything というビクトリア朝時代の日常生活のハウツー本に由来していて、バーナーズ＝リーが幼少のころを思い出して付けたものである)。それは現在のWebとは大分違うが、根本的なアイデアの多くを含んでいる(さらには、バーナーズ＝リーのWWW後のプロジェクトである Semantic Web の考え方も含んでいる)。

1989年3月、欧州原子核研究機構(CERN)のティム・バーナーズ＝リーは「Information Management: A Proposal(情報管理:提案)」を執筆し、ENQUIREを参照しつつさらに進んだ情報管理システムを描いた。彼は1990年11月12日、World Wide Web をより具体化した提案書を発表した。実装は1990年11月13日から開始され、バーナーズ＝リーは最初のWebページを NeXTワークステーション上に置いた。

その年のクリスマス休暇の間に、バーナーズ＝リーは Webに必要な全ツールを構築した。世界初のWebブラウザ(Webエディタでもある)と世界初のWebサーバである。

続いて、1991年8月6日、彼は後述のWorldWideWeb - Executive Summaryを alt.hypertextニュースグループに投稿した。この日が Webがインターネット上で利用可能なサービスとしてデビューした日となる。

ハイパーテキストの概念は1960年代まで遡ることができる。テッド・ネルソンの Project Xanadu、ダグラス・エンゲルバートの oN-Line System(NLS)などである。

ヴァネヴァー・ブッシュのマイクロフィルムベースの「memex」にインスパイアされたものであり、memex は1945年の論文「As We May Think」で描かれている。

テッド・ネルソンによる1965年のProject Xanadu、1987年アップルのHypercard。

バーナーズ＝リーのブレイクスルーはハイパーテキストとインターネットを結合したことである。彼の著書『Weaving The Web』では、このふたつの技術の結合は双方の技術コミュニティの協力によって成立することを強調しているが、誰もこの提案を取り上げることはなく、彼は最終的に自分でプロジェクトを実行したのである。この過程で彼はURIと呼ばれるグローバルな資源識別子を開発した。

World Wide Web は当時実現していた他のハイパーテキストシステムとはいくつかの点で異なる。ネットワークをまたがりコンピュータネットワーク上に実現したのであった。



バーナーズ＝リーがCERNで使用していたNeXTcube。最初のWebサーバとなった

In article <6...@cernvax.cern.ch> I promised to post a short summary of the WorldWideWeb project. Mail me with any queries.

WorldWideWeb - Executive Summary

The WWW project merges the techniques of information retrieval and hypertext to make an easy but powerful global information system.

The project started with the philosophy that much academic information should be freely available to anyone. It aims to allow information sharing within internationally dispersed teams, and the dissemination of information by support groups.

Reader view

The WWW world consists of documents, and links. Indexes are special documents which, rather than being read, may be searched. The result of such a search is another ("virtual") document containing links to the documents found. A simple protocol ("HTTP") is used to allow a browser program to request a keyword search by a remote information server.

The web contains documents in many formats. Those documents which are hypertext, (real or virtual) contain links to other documents, or places within documents. All documents, whether real, virtual or indexes, look similar to the reader and are contained within the same addressing scheme.

To follow a link, a reader clicks with a mouse (or types in a number if he or she has no mouse). To search and index, a reader gives keywords (or other search criteria). These are the only operations necessary to access the entire world of data.

The WWW browsers can access many existing data systems via existing protocols (FTP, NNTP) or via HTTP and a gateway. In this way, the critical mass of data is quickly exceeded, and the increasing use of the system by readers and information suppliers encourage each other.

Making a web is as simple as writing a few SGML files which point to your existing data. Making it public involves running the FTP or HTTP daemon, and making at least one link into your web from another. In fact, any file available by anonymous FTP can be immediately linked into a web. The very small start-up effort is designed to allow small contributions. At the other end of the scale, large information providers may provide an HTTP server with full text or keyword indexing.

The WWW model gets over the frustrating incompatibilities of data format between suppliers and reader by allowing negotiation of format between a smart browser and a smart server. This should provide a basis for extension into multimedia, and allow those who share application standards to make full use of them across the web.

This summary does not describe the many exciting possibilities opened up by the WWW project, such as efficient document caching, the reduction of redundant out-of-date copies, and the use of knowledge daemons. There is more information in the online project documentation, including some background on hypertext and many technical notes.

Try it

A prototype (very alpha test) simple line mode browser is currently available in source form from node info.cern.ch [currently 128.141.201.74] as

`/pub/WWW/WWWLineMode_0.9.tar.Z.`

Also available is a hypertext editor for the NeXT using the NeXTStep graphical user interface, and a skeleton server daemon.

Documentation is readable using www (Plain text of the installation instructions is included in the tar file!). Document

<http://info.cern.ch/hypertext/WWW/TheProject.html>

is as good a place to start as any. Note these coordinates may change with later releases.

Tim Berners-Lee Tel: +41(22)767 3755
WorldWideWeb project Fax: +41(22)767 7155
C.E.R.N. email: t...@cernvax.cern.ch
1211 Geneva 23
Switzerland

WWWは、双方向ではなく単方向のリンクを使用する。これにより、何らかの資源の所有者と連絡を取らなくてもリンクすることが可能となった。これによって Webサーバやブラウザの実装も簡単になっているが、同時にリンク先の資源がいつの間にか無くなるという問題も発生させることとなる。

HyperCardやGopherとは違い、World Wide Web は独占されておらず、サーバやクライアントを独自に開発し拡張するのも自由にできてライセンスを得る必要も無い。

開発当初、WWWは文字情報を扱うだけの比較的単純なものであった(NeXT上で開発されたためOS自身が文字以外を適切に扱うため、WWWは情報を区別しなくてもよかったというのが真相)。しかし1992年、現在のような画像なども扱えるWWWにしたのが、イリノイ大学に設置されている米国立スーパーコンピュータ応用研究所(National Center for Supercomputing Applications・NCSA)である。

この学生であったマーク・アンドリーセンらは、文字だけでなく画像なども扱える革新的なブラウザ「Mosaic」を開発。そしてこのソフトに改良を加えるために無料でソースコードを公開したため、Mosaicはたちまち普及し、WWWは誰でも手軽に使うことのできる世界的なメディアとなった。

1993年4月30日、CERNは World Wide Web を無料で誰にでも開放することを発表した。

日本最初のホームページを開設したのは、高エネルギー加速器研究機構所属の森田洋平氏である。

WWWを参照するにはウェブブラウザなどのソフトウェアを使用する。ウェブブラウザは、ウェブページなどのドキュメントをWebサーバから取得し、モニターなどの出力デバイスに表示する。ウェブページに含まれるハイパーリンクをたどることで他の文書を読んだり、情報をサーバに送ることで何らかの相互作用をもたらすこともできる。Webページはウェブサイトと呼ばれる関連情報を集めたものに含まれていることが多い。

「インターネット・サーフィン」という言葉は 1992年6月に出版された Wilson Library Bulletin の中の Jean Armour Polly(司書)の書いた文章から発祥しているという。Polly は独自にこの言葉を生み出したかもしれないが、1991年から1992年にかけて Usenet で同様の言葉が散見された。さらにそれ以前にハッカーのコミュニティで使われていたという証言もある。Polly はインターネットでは NetMom で有名である。

英語では、*worldwide* と一語で表記するのが普通だが、*World Wide Web* やその略記の *WWW* は、英語でも普通に使われるようになった。最初のころは、*WorldWideWeb* (プログラマが好む命名規則、つまり単語を連続して書いて単語の先頭だけを大文字にするインターキャップとかキャメルケースといわれる書き方)とか *World-Wide Web* (ハイフンが入っていて英語の本来の使用法に近い)と表記されることも多かった。

World Wide Web上のWebページなどの資源にアクセスするには、WebブラウザにURLを入力するか、Webページのリンクをたどればよい。

第一段階としてURIのサーバ名を表す部分が Domain Name System (DNS) と呼ばれるインターネットの分散データベースによってIPアドレス(TCP/IPネットワーク上で機器を判別するための重複しない特定の番号【アドレス】。なおIPとは、インターネットプロトコルの略)に変換される(IPアドレスが直接指定されている場合はこの変換は行われない)。

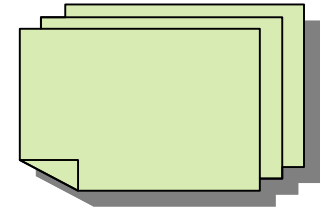
次に、そのIPアドレスに対応するWebサーバに対して、URIで指定されたスキームに従い接続を試みる。プロトコルとしては主にHTTPが使用される。一般的なWebページでは、ページを構成するHTMLドキュメントや画像ファイルが要求され、即座に要求元に転送される。

Webブラウザは、受け取ったHTMLファイルやCSSファイルにしたがってレンダリングし、画像をはめ込み、リンクをはめ込むなどの仕事を行う。これによって画面上の「ページ」が生み出される。

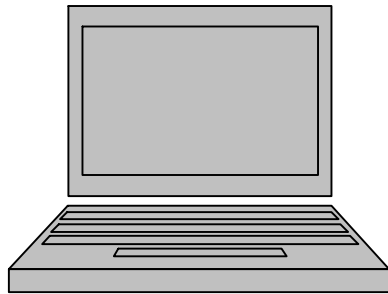
多くのWebページは他の関連するページとのハイパーリンクを含んでいる。それは例えばダウンロードのページだったり、ソース文書だったり、他の定義だったり、Web上の何かの資源だったりする。このハイパーリンクによって情報の「Web(網)」が形成される。これによって World Wide Web が構成されているのである。

WWWの本質：インターネットとハイパーテキストの融合！

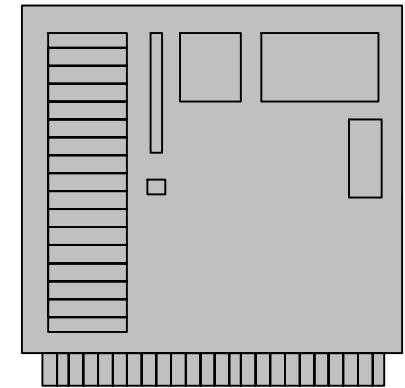
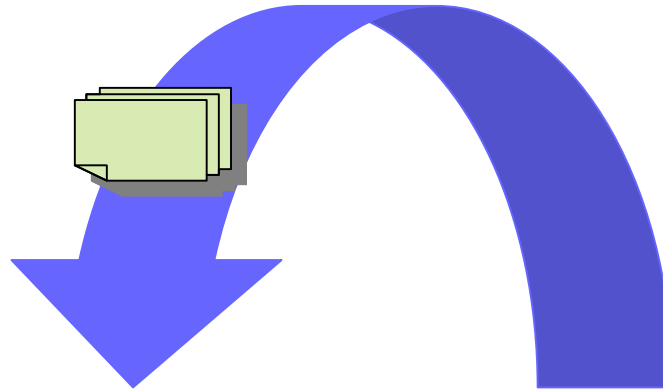
①記述言語:HTML



④送受信プロトコル:HTTP



③情報受信:Webブラウザ



②情報発信:Webサーバ

**WebブラウザにURLを入力するか、
Webページのリンクをたどればよい!**

OSI参照モデルの実際例

7	アプリケーション層	<i>HTTP</i> , SMTP, SNMP, FTP, Telnet, AppleTalk, X.500
6	プレゼンテーション層	SMTP, SNMP, FTP, Telnet
5	セッション層	NetBIOS, NWLink, PAP, 名前付きパイプ
4	トランスポート層	<i>TCP</i> , UDP, SPX, NetBEUI
3	ネットワーク層	<i>IP</i> , ARP, RARP, ICMP, DHCP, IPX, NetBEUI
2	データリンク層	イーサネット, トークンリング, PPP, フレームリレー
1	物理層	電話線, 無線, 光ケーブル

OSIモデルは仕様ではなく指針であるため、全てのプロトコルやネットワークがOSIモデルに沿って実装されているとは限らない。従って、一部のプロトコルやサービスに関しては、OSIモデルのどの層に属するかについて、幾つかの異なる見解が存在する。複数層に跨っている物もある。図示の例はあくまでも例に過ぎない。

Uniform Resource Identifier (URI) は、一定の書式によってリソースを指し示す識別子。1998年8月に RFC 2396 として規定され、2005年1月に RFC 3986 として改定された。URI は *Uniform Resource Locator (URL)* の考え方を拡張したものである。

URI は http/https や ftp などのスキームで始まり、コロン (:) による区切りのあとにスキームごとに定義された書式によってリソースを示す。

また、URI によって示されるリソースはネットワーク上にあるものに限らず、人や会社、書籍などを示すことも可能である。

URI スキームは IANA によって登録されたものが公式なものとされている。irc や javascript のように未登録ではあるが広く使われているスキームも存在する。

Uniform Resource Locator (URL) は、インターネット上のリソースを特定するための形式的な記号の並び。WWWをはじめとするインターネットアプリケーションにおいて提供されるリソースを、主にその所在を表記することで特定する。なお、ここでいうリソースとは(主にインターネット上の)データやサービスを指し、例えばウェブページや電子メールの宛先といったものがそうである。URLを含む一般概念としてURIがある。

URI には、以下の2つのサブセットがある。

Uniform Resource Locator (URL)

リソースの「場所」を識別する。ネットワーク内の位置を示してリソースを同定する。

Uniform Resource Name(URN)

リソースの「名前」を識別する。もしネットワーク上にリソースが無くなっても、一意で永続的な識別を行えるようにする。例えば urn:ietf:rfc:2648 という URN は、RFC 2648 への参照を示す。

ただし、W3C が2001年9月に発表した『URIs, URLs, and URNs: Clarifications and Recommendations 1.0』では上記を古典的な見解とし、現在の見解は URL や URN といった区別をせず、これらの語は非公式な概念だとしている。

注)

Hypertext Transfer Protocol (HTTP)

WebブラウザとWebサーバの間でHyperText Markup Language (HTML) などのコンテンツをやり取りする際に用いられる通信プロトコルであり、RFC 2616で規定されている。ハイパーテキスト転送プロトコルとも呼ばれる。

最初、HTTP/0.9ではURLのみの簡単なやりとりであったが、HTTP/1.0でNNTPやSMTPのような各種ヘッダが定義され、HTTP cookieなどの利用が可能になった。HTTP/1.1では複数データを転送するためのキープアライブ(keep-alive)機能やプロキシなどの利用も想定された仕様になった。

HyperText Markup Language (HTML)

WWW上のドキュメントを記述するためのマークアップ言語(タグなどを用いてデータの構造などを明記するための人工言語の1種で、ページ記述言語とも呼ぶ)であり、WWWの基幹的役割をもつ技術の一つ。HTMLでマークアップされたドキュメントはほかのドキュメントへのハイパーリンクを設定できるハイパーテキストであり、また画像・リスト・表など高度な表現力をもつ。HTMLは現在W3Cによる標準規格であるが、後継規格であるXHTML(Extensible HyperText Markup Language、SGMLで定義されていたHTMLをXMLの文法で定義しなおしたマークアップ言語であり、その仕様はHTMLと同じくW3Cによって勧告されている。XMLを採用したため、MathMLやSVGなどの他のXML文書を埋め込むことができるようになった。)への置き換えが進められつつある。

HyperText

複数の文書(テキスト)を相互に関連付け、結び付ける仕組みである。ダグラス・エンゲルバートによって考案され、「テキストを超える」という意味から hyper- (~を超えた) text (文書) と名付けられた。テキスト間を結びつける参照のことをハイパーリンクと言う。

注)

* **Domain Name System (DNS)** は、インターネットを使った階層的な分散型データベースシステムである。1983年に情報科学研究所(ISI)のポール・モカペトリスとジョン・ポステルにより開発された。現在ではおもにインターネット上のホスト名や、電子メールに使われるドメイン名とIPアドレスとの対応づけを管理するために使用されている。

* **ポート番号**

コンピュータがデータ通信を行う際に通信先を特定するための番号のこと。
ネットワーク上でコンピュータ同士がデータを交換するプロトコルの一つである

TCP/IPにおいては、IPアドレスにより通信相手のコンピュータを指定するが(IP・ネットワーク層)、そのコンピュータ上で動いている複数のプログラムのうちの一つを通信相手として指定するために、ポート番号を用いる(TCPやUDP・トランスポート層)。

指定できる番号の範囲はTCPやUDPなどの通信の種類毎にそれぞれ0から65535(16ビット符号無し整数)と定められている。これらの番号と、TCPやUDPなどの通信プロトコルの指定の組み合わせで特定される、論理的な通信の口をポートと呼ぶ。

IPアドレスを建物の住所にたとえるなら、ポート番号は部屋番号に相当する。

プログラムでポートを用いて通信するには、一般にソケット(BSD【Berkeley Software Distribution の略語で、カリフォルニア大学バークレー校において開発・配布が行われたソフトウェア群】由来のプロセス間通信のAPIで使われる記述子・IPアドレス・ポート番号・プロトコル種別の組み合わせ)と呼ばれる仕組みを用いる(ソケットはBSD系UNIXで開発されたため、特にBSDソケットとも呼ぶ)。ソケットとは通信の端点のことで、2台のコンピュータが通信しているとき、その通信路の両端、すなわちそれぞれのコンピュータにソケットが存在する。また、ソケットのインターフェースやソケットを扱うライブラリプログラムも併せてソケットと呼ぶこともある。ソケットを用いた通信は、以下のような手順で行われる。

- ①サーバ機でサービスを提供するプログラムは、ソケットを作成し、サービス固有のポート番号をソケットに割り当て (bind)、待ち行列を用意し (listen)、クライアントからの接続を待ち受ける (accept)。
- ②サービスを利用するクライアントプログラムは、ソケットを作成し、そのソケットの通信相手としてサーバ機のIPアドレスとサービスのポート番号を指定し (connect)、接続を行う。
- ③サーバは接続を受け付けると、新規にソケットを作成し、そのソケットとクライアントとの間に通信を確立する。もとのソケットは再び待ち受けに戻る(これは、会社などで、受付係が来客を担当者に引き合わせ、その後また受付に戻るようなものと考えることができる)。
- ④通信が終わると、②および③で作成したソケットは破棄される。

これらの手順に沿って通信を行うプログラムの形式は、ソケットプログラミングと呼ばれる。WWW (HTTP) の場合、原則としてHTMLファイルや画像ファイルを1件取得するたびに上記2.~4.の手順が繰り返されるため、アクセスが集中するとサーバの負荷が非常に高くなる(ただしHTTP 1.1では1回の接続で複数件のデータを取得する動作がデフォルトとなっている)。このように、ポート番号とはサーバ上のサービスを特定するための番号であるといえる。そのため、サーバ側のポート番号は重要であるが、クライアント側のポート番号は通常問題とされず、空いている番号を適当に自動で割り当てる。

なお、上記手順の3.で作成されるソケットは、待ち受け用のソケットと同じIPアドレス・ポート番号・プロトコル (TCP/UDP) を有する。すなわち、1台のサーバ機に、同じIPアドレス・ポート番号・プロトコルのソケットが複数存在し得る(ポートとソケットは一对多の対応となり得る)ことに注意が必要である。この様子は、同時に複数の相手と通信を行っているWebサーバなどで、netstatコマンド(ソケットの通信状況を表示するコマンド)を実行してみるとよくわかる。

```
$ netstat
```

```
tcp4 0 0 xxx.yyy.zzz.www.80 aaa.bbb.ccc.ddd.39321 ESTABLISHED
```

```
tcp4 0 0 xxx.yyy.zzz.www.80 eee.fff.ggg.hhh.56047 TIME_WAIT
```

```
tcp4 0 0 xxx.yyy.zzz.www.80 iii.jjj.kkk.lll.41358 TIME_WAIT
```

```
tcp4 0 0 xxx.yyy.zzz.www.80 mmm.nnn.ooo.ppp.3356 FIN_WAIT_2
```

```
tcp4 0 0 xxx.yyy.zzz.www.80 qqq.rrr.sss.ttt.3355 FIN_WAIT_2
```

ここでxxx.yyy.zzz.wwwはこのWebサーバ機のIPアドレス、80はWebサーバプログラムのポート番号、右側は各クライアントのIPアドレスおよびポート番号。

ポート番号は大別して3種類に分けられる！

ポート番号の種類 種類範囲内容

WELL KNOWN PORT NUMBERS 0番 - 1023番 一般的なポート番号

REGISTERED PORT NUMBERS 1024番 - 49151番 登録済みポート番号

DYNAMIC AND/OR PRIVATE PORTS 49152番 - 65535番

自由に使用できるポート番号上記のうち、

WELL KNOWN PORT NUMBERは、Internet Assigned Numbers Authority (IANA) が管理。

REGISTERED PORT NUMBERSに関しては、IANAが利便性を考慮して公開している。なお、クライアント側に割り当てられるポート番号など、ユーザが自由にポート番号を使用する場合は、49152番以降を使用する。

ただし、これはあくまで指針であり強制力はない。これに従わず勝手なポート番号を用いて通信することも一応は可能であるが、混乱を招くであろう。

なお、UNIX系のOSでは、0～1023のポート番号を使用するには、root権限が必要である (UNIX系のOSにおいて、システムを管理する権限を持つユーザ。ルートユーザ。ルートディレクトリをユーザディレクトリとしていたことから、このように呼ばれるようになった。スーパーユーザとも言い、多くのオペレーティングシステムでシステムアドミニストレータが管理する特殊なユーザーアカウントを指す。)。

- TCP/20 : FTP (データ)
- TCP/21 : FTP (制御)
- TCP/22 : SSH
- TCP/23 : Telnet
- TCP/25 : SMTP
- UDP/53 : DNS
- UDP/67 : DHCP (サーバ)
- UDP/68 : DHCP (クライアント)
- TCP/80 : HTTP
- TCP/110 : POP3
- TCP/119 : NNTP
- UDP/123 : NTP
- UDP/137~138 : NetBIOS
- TCP/139 : NetBIOS
- TCP/143 : IMAP
- TCP/443 : HTTPS
- TCP/445 : ダイレクトホスティングSMBサービス
- TCP/587 : Submission(メール送信)

* プロトコル名称とポートの対応表は、UNIX系OSやMac OS Xでは/etc/servicesに、Windows NT系では%SystemRoot%\System32\drivers\etc\servicesにテキストファイルとして存在する。ただしIANAに登録されているものがすべて記載されているとは限らない。

一般に、コンピュータ内には様々なプログラムが動いている。近年は通信処理を行うのに十分な能力を持つコンピュータが安価になったことから、家庭用のパソコン内でもいわゆるサーバ機能を有するプログラムが多数動作している。これらのプログラムはポートを作成して結合を待ち受けているが、プログラムの不備があると、ある特別なデータを受信した時にプログラムが誤作動することがある。悪意を持って巧妙に作られたデータの場合、本来通信データとして送信してはいけない情報(コンピュータ内に隠匿してあるはずのパスワードなど)を勝手に送信してしまうような誤動作をさせられることもある。このような誤作動の元となるプログラムの不備のことをセキュリティーホールと呼ぶ。

セキュリティーホールのないソフトウェアを用いるのが最善の策であるが、プログラムが人間によって作られるものである限り誤りを完全になくすことはできない。また、セキュリティーホールになりえる動作でも用途によっては便利な機能であることがあり、悪意のあるデータが送られてこないことが保証されていればそのソフトウェアを動かしたい場合も多い。

このような場合のために、安全を確保しやすい内部のLANと何者がいるかわからない外部のインターネット網との間の通信に介入し、不正な通信の一部を遮断するということが行われる。この遮断機能をファイアーウォールと呼ぶ。

ファイアーウォールにはいくつかの手法があるが、最も多く用いられているのはソケットのバインドを規制するものである(パケットフィルタ型)。特定のポート番号とのバインドができないようにしたり、逆に特定のポート番号とのバインドしかできないようにすることにより、ユーザが気づかないうちにコンピュータが外部と勝手に通信してしまうのを防ぐのである。また、TCPのように通信回線確立の手順があるプロトコルにおいては、LAN側からインターネット上にあるサーバに対して接続開始を要求するのは認めるが、インターネット側からLAN上のコンピュータへの接続開始の要求は通さないということもよく行われる。

ファイアーウォールはソフトウェアまたはハードウェアの形態を取り、その機能はルーター(LANとインターネットの接続点に位置する)に内蔵されたり、コンピュータ内部に導入されたりする(IDS)。パソコン向けのIDS(Intrusion Detection System の略であり、侵入検知システム)は、特にパーソナルファイアウォールと呼ばれる。

一般論として、ネットワーク経由で記憶媒体を共有する、いわゆる「ファイル共有」を実現するサービスは、処理の高速化などの必要性からプロトコルが単純化されておりセキュリティ面での脆弱性を持つ事が多い。特にMicrosoft Windowsがファイル共有などの実現に用いているNETBIOSと呼ばれるプロトコルは非常に広く用いられているにもかかわらずプロトコルの脆弱性が高いといわれている。

このため、NETBIOS用として定義されているポート番号137番～139番(TCPとUDP)については、インターネットとLANを接続するルータにおいて通信を遮断する事が強く推奨されている。市販のルータでは工場出荷状態でこれらのポートの通信遮断設定が入っている事が多い。

近年のインターネットの爆発的な拡大にともない、コンピュータ1台1台を識別するはずのIPアドレスの数がコンピュータの台数よりも遥かに少ないという状況が起こってしまった。IPv6を用いてIPアドレスの数を飛躍的に増加させるという動きもあるが、現状では一つのIPアドレスを複数のコンピュータで使い回すということが行われている。

この際、「電子メールの受信サービスを行うのはAというコンピュータ」「WWWのサービスを行うのはBというコンピュータ」というように、サービス毎に処理をするコンピュータが違いうようにするためには、「ポート番号25番の通信はAへ」「ポート番号80番の通信はBへ」というようにポート番号毎に通信先を振り分ければよい。

このポート番号による通信コンピュータの振り分けの機能を「ポートフォワード (port forward)」と呼ぶ。

一般に、ポートフォワードはIPマスカレード(ルータによるIPアドレス変換、コンピュータネットワーク構成時に使われる、ルータや、ファイアーウォールを通過するパケットのIPアドレスやポート番号を変換する技術。マスカレード(masquerade)は仮装の意)の機能の一部として実現される。IPマスカレードの内、ポートフォワードを行わないものをNAT、ポートフォワードを行うものをNAPTと呼ぶことが多い。

* IPマスカレードは、NAT(Network Address Translation)やNAPT(Network Address Port Translation)と同義に扱われる場合もあるが、もともとのIPマスカレードはNAPTのLinuxシステム上での呼び名。

注)

Network News Transfer Protocol (NNTP)

インターネット・アプリケーションプロトコルのひとつである。おもに、ネットニュース (Usenet) の記事を読むことと記事を投稿することのために使われる。記事はニュースサーバ間を相互に配送される。カリフォルニア大学サンディエゴ校の Brian Kantor と カリフォルニア大学バークレー校の Phil Lapsley が Network News Transfer Protocol の仕様である RFC 977 を 1986 年の 5 月に完成させた。他の貢献者として、Baylor College of Medicine の Stan Barber とアップルコンピュータの Erik Fair がいる。

Usenet

UUCP ネットワーク上での使用を前提として設計された。つまり、ほとんどの記事は電話回線で直接コンピュータ同士を接続して配送されていた。購読者と投稿者は同じニュースサーバにログインし、そのサーバのディスクにある記事を直接読んでいた。LAN とインターネットが一般に普及すると、パーソナルコンピュータ上で使用できるニュースリーダーと、インターネット上で記事を配送する手段が必要とされた。インターネットで互換性のあるファイルシステムがまだ広くは利用できなかったため、SMTP に類似した新しいプロトコルを作ることになった。

* Well-known TCP port である 119 番は NNTP のために予約されている。クライアントが SSL でニュースサーバに接続するときは TCP のポート 563 番が使われる。これは NNTPS と呼ばれることがある。

注)

SMTP (Simple Mail Transfer Protocol)

インターネットで電子メールを転送するプロトコル。通常TCPのポート番号25を利用する。転送先のサーバを特定するために、DNSのMXレコード(DNSで定義される情報の1つである。電子メールの配送先を決定する時に使われる。複数のレコードを定義できる。優先度を設定でき、優先度の高い配送先が障害になった場合、代替して処理を行なうサーバを指定できる)が使われる。RFC2821で標準化されている。

メールサーバのMTA(Mail Transfer Agent メール転送エージェント)間の転送だけでなく、MUA(*Mail User Agent*、電子メールソフト、メールクライアントとも呼ばれる)からメールサーバにメールを送信するときにも使われることが多い。ただし、この場合受信したサーバ側のふるまいがサーバ同士の転送と異なる点が多いため、サーバ側をMSAと呼びポート番号587を利用し、通常のMTAと分けることが多くなってきている。

SMTPは本来テキストベースのプロトコルで、要求/応答メッセージのみではなく、全ての文字が7bit ASCIIでなければならない、という制限があった。現在では拡張機能により8bit以上を要求する言語や添付ファイルに使われることの多いバイナリもそのまま転送することも可能であるが、互換性を考慮すると、MIME(Multipurpose Internet Mail Extension (MIME、マーム)は、規格上US ASCIIしか使用できないインターネットの電子メールでさまざまなフォーマットを扱えるようにする規格)という方式で、7bitに収まるようにすることが望ましい。なお、日本語は通常ISO-2022-JPという方式を使うことで、全てのバイトが7bitに収まるようにしている。

3. WebブラウザのためのHTMLとは？

- 1991年 HTML1.0 : 基本要素HTML,HTTP,URL
- 1995年 HTML2.0 : 画像を埋め込むタグ
ユーザー入力<form>タグ追加
- 1997年 HTML3.2 : 新機能を大幅に追加
Javascriptを記述する<script>
表を意味する<table>
Javaアプレットを埋め込む<applet>
- 1997年 HTML4.0 : カスケードスタイルシート(CSS)の導入
共通属性の推奨
⇒データ構造と見栄えの分離
マルチメディアファイル設定<object>
⇒Flash(Macromedia)、Media Player(Microsoft)

- 文書構造をタグと呼ぶ記号で表現したもの
- テキスト文では表現不可能な文字フォント、組版、参照先への関連付けを同一文書中に表現可能

- 記述例

<center>

緑色の文字を中央に 表示

<center>

- Webブラウザ

HTMLは、専用のビューワ(HTMLビューワ)によって表示

⇒「HTMLビューワ」をWebブラウザと呼ぶ:

例： Internet Explorer (Microsoft)

- **ドキュメントタイプの宣言： W3Cで定義(HTML4.01)**
- **いずれかのドキュメントタイプに準拠すること**
 - (1) **厳密型** : **Strict DTD**
 - (2) **移行型** : **Transitional DTD**
 - (3) **フレーム構造型** : **Frameset DTD**
- **記述例**
HTML4.01 Strict DTD
<!DOCTYPE HTML PUBLIC “- //W3C//DTD HTML4.01//EN”
“http: // www.w3.org/TR/html4 / strict.dtd” >

- ドキュメントタイプの基本構造： W3Cで定義(HTML4.01)
- いずれかのドキュメントタイプに準拠すること
 - (1) 厳密型 : Strict DTD
 - (2) 移行型 : Transitional DTD
 - (3) フレーム構造型 : Frameset DTD

- 記述例

HTML4.01 Strict DTD

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd" >
```

宣言

```
<html>  
<head>ドキュメントのヘッダー情報</head>  
<body>ドキュメント本文</body >  
</html>
```

ドキュメントタイプ
の基本構造

- HTMLドキュメント中のタグは、**<英数字タグ名>**の基本構造をとる
- HTMLでは大文字と小文字の区別ない
- 後述のXHTMLではタグ名は必ず小文字
- 開始タグ/終了タグ : **<タグ名> ~ </タグ名>**のペア形式
- 入れ子タグ :
 <タグ名a> <タグ名b> <タグ名c> ~ </タグ名c> </タグ名b> </タグ名a>
 の複合ペア形式
- タグ属性 : 開始タグの後に空白を置き英数字で記述
 <タグ名b 属性1="値1" 属性2="値2"> ~ </タグ名>

- HTMLの基本構造タグでヘッダ情報を記述するもの
- `<title>`タグ : ドキュメントにタイトルを付ける
- `<base>`タグ : URLに相対値を使用時、URL絶対値を指定
`<base href=“基準URL”`
`target=“基準となるリンク先のウインドウ名またはフレーム名”>`
`_blank` : 新ウインドウを開く `_parent` : 現在の親のフレームに表示
`_self` : 現在と同フレームに表示 `_top` : フレーム分割中止し全面表示

- **<meta>タグ** : HTMLドキュメントに関する種々の情報を記述
 - * **name属性**
<meta name=“ドキュメント情報名” content=“ドキュメント情報内容”>
例: author, copyright, keywords, description (該ページ概要),
date (更新日時), generator (作成ソフト),
robot (検索エンジン巡回ソフトへの指示)
 - * **http-equiv属性** (name属性の代わりにhttp経由でドキュメント指定)
<meta http-equiv=“デフォルト名” content=“デフォルト内容”>
例: content-type (ドキュメントの文字コード),
content-style-type (デフォルトのスタイルシート),
content-script-type (デフォルトのスクリプト言語),
refresh (指定URLへのジャンプ), expires (有効期限)
- **<link>タグ** : ドキュメント同士の関係を記述
- **<address>タグ** : ドキュメントの問合せ先を記述
- **コメントタグ** : ブラウザには表示しない内容を記述

- HTMLドキュメントの背景色指定や画像の添付を記述

『全体的概観』

- HTML4.01 で指定の色 (Black, Red, Navy, Aqua)
- RGB16進数 : #RRGGBBで各色8ビットで表現

『レイアウト』

- 見出し : `<h1> 見出しとなる文章</h1>`
- 段落 : `<p>段落内容</p>`
- 改行 : `
`
- 行揃え : `<div align="">行揃えの内容</div>`
例: left左寄せ reight右寄せ center中央揃え
- 横罫線 : `<hr>`
例: hr size=太さ width=長さ align=配置
noshadeの指定

- フォントの大きさ、色、タイプを記述 : ` ~ `
- 太字 : ` ~ `
- 斜体文字 : `<i> ~ </i>`
- 取り消し線 : `<strike> ~ </strike>`
- 等幅 : `<tt> ~ </tt>`
- 下線 : `<u> ~ </u>`
- 文字強調 : ` ~ ` さらに強調は ` `
- 整形済みテキスト表示 : 空白や改行をそのまま表示
`<pre> ~ </pre>`
- ドキュメント変更 : ` ~ ` `<ins> ~ </ins>`
- 記号表示 : タグの一部として認識される記号をそのまま表示
例: `>` : `>`

- HTMLドキュメントに画像を表現
- 画像の埋め込み : ``
- 画像の大きさ :
``
- 画像の枠線: ``
- 画像の配置: ``
⇒配置: 上・下・中央・左・右端へ揃える
- 画像周辺の余白:
``

- **JPEG: Joint Photographic Experts Group**

静止画像データの圧縮方式の一つ。ISOにより設置された専門家組織の名称がそのまま使われている。圧縮の際に、若干の画質劣化を許容する(一部のデータを切り捨てる)方式と、まったく劣化のない方式を選ぶことができ、許容する場合はどの程度劣化させるかを指定することができる。方式によりばらつきはあるが、圧縮率は1/10～1/100程度。

写真などの自然画の圧縮には効果的だが、コンピュータグラフィックスには向かない。

- **GIF : Graphics Interchange Format**

アメリカのパソコン通信CompuServeで画像交換用に開発された画像形式。JPEGと共にインターネットで標準的に使われる画像形式である。256色までの画像表現ができ、JPEGが苦手なイラストやアイコンなどの保存に向いている。動画を保存できるアニメーションGIFや、透明色を指定して背景イメージと重ね合わせることができるトランスペアレントGIF、全体をダウンロードしなくてもイメージの確認ができるインターレースGIFなどの拡張仕様がある。圧縮に使われているLZWというアルゴリズムはUnisys社が特許を持っているため、GIFに対応したソフトウェアを販売／配布するにはUnisys社にライセンス料を払う必要がある。このライセンス料を嫌って、PNGというライセンスフリーの画像フォーマットが策定された。

- 基本リンク

` 内容 `

- リンク先のウィンドウ指定

`<a href="リンク先のファイルの場所"`

`target ="リンク先のウィンドウ名"> 内容 `

* target 属性の指定 : `_blank, _parent, _self, _top`

- 特定の位置へのリンク

` 内容 `

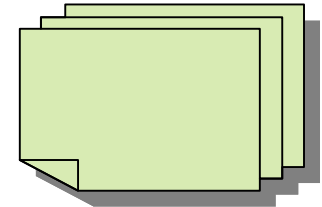
` 内容 `

*` アンカーの内容 `

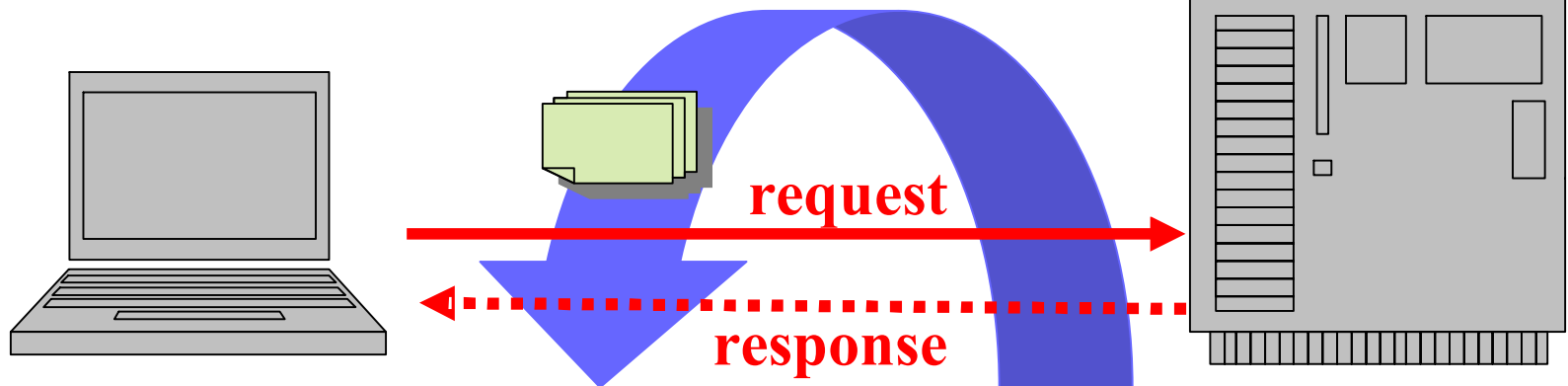
- HTMLにおけるフレーム：
ブラウザのウィンドウ内に分割して複数のHTMLドキュメントを表示する機能
- 基本形
<frameset rows=“縦分割長” cols=“横分割長”>フレーム内容</frameset>
- フレームのサイズ変更禁止
- フレーム境界線幅指定
<frameset border=“フレーム境界線幅”> ~ </frameset>
- スクロールバーの表示
- その他

4. HTTPとは？

①記述言語:HTML



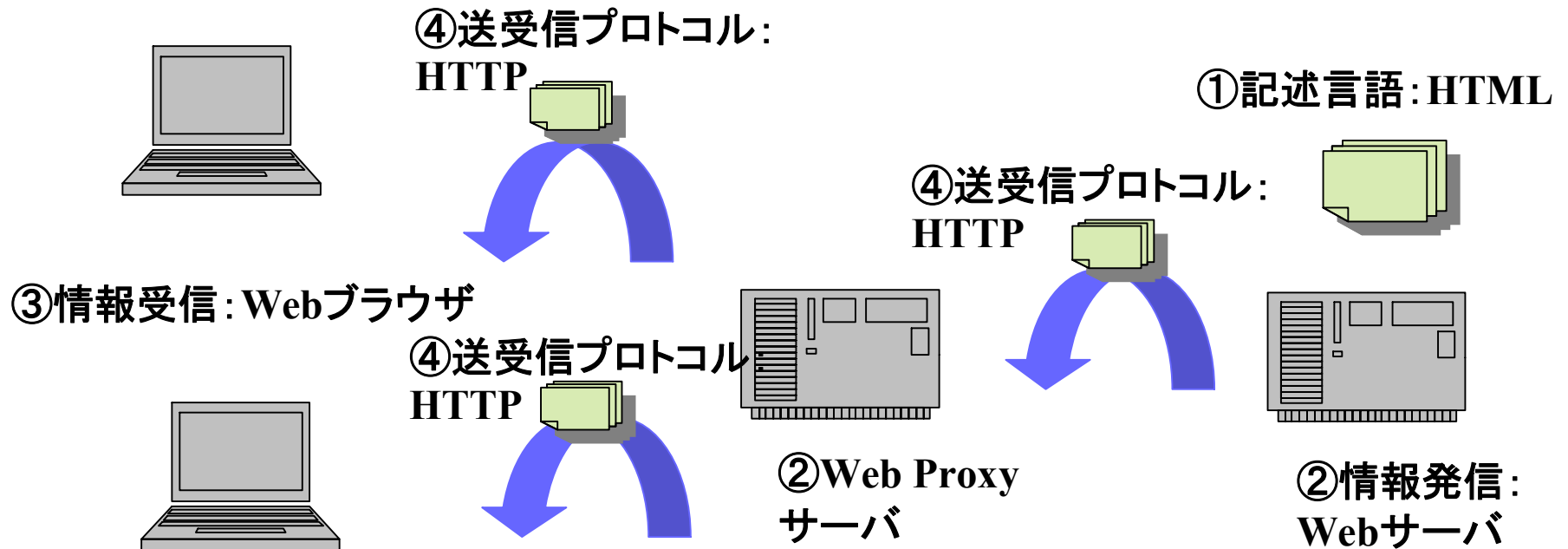
④送受信プロトコル:HTTP



③情報受信:Webブラウザ
(Internet Explorer, Navigator,etc.)

②情報発信:Webサーバ
(IIS, Apache,etc.)

**WebブラウザにURLを入力するか、
Webページのリンクをたどればよい!**



Web Proxyは、Webサーバから受信したデータをキャッシュ領域に保管しておく

HTTPリクエストの1行目に記述するMethods

Methods	Meanings
GET	オブジェクトをWebブラウザに返す。URIで示したコンテンツを欲している。
POST	オブジェクトをWebブラウザに送る。CGI等でサーバに情報送信。
HEAD	オブジェクトの情報をWebブラウザに返す。ヘッダー情報だけを授受するので負荷軽減効果あり。
PUT	オブジェクトをWebサーバに送り、サーバ上に保存。
DELETE	Webサーバ上のオブジェクトを削除。
OPTIONS	Webサーバが如何なるMethods、機能を持つか知る。
TRACE	Webサーバが受信したHTTPリクエストをHTTPレスポンスに含めて返す。Webブラウザは、途中のProxyやGatewayの変更経過を知ることができる。
CONNECT	SSLを通すためのMethod。

付加情報名	Meanings
Accept	Webブラウザが、受信可能なこと。
If-Modified-Since	指定した日付よりも新しいオブジェクトだけを返す。
Accept-Language	Webブラウザが受信可能な言語コードを複数優先順位をつける。
User-Agent	Webブラウザの種類。
Referer	直前にリンクの張られていたURL。

HTTPレスポンスの1行目の主なステータスコード

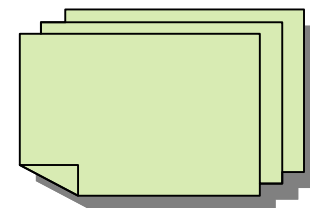
ステータスコード	理由	Meanings
200	Document Follows	リクエスト成功。
304	Not Modified	If-Modified-Sinceの指定日以降、オブジェクト更新がなされていない。
401	Unauthorized	オブジェクトが保護(特定者以外)。
403	Forbidden	オブジェクトへのアクセス禁止。
404	Not Found	Webサーバ上にオブジェクトが存在せず。
500	Server Error	リクエストMethodがWebサーバ上に未実装。

付加情報名	Meanings
Server	Webサーバ名とバージョン。
Date	GMT日付。
Last-Modified	オブジェクト更新日。
Content-Length	出力オブジェクト長。
Content-Type	出力MIMEタイプ。

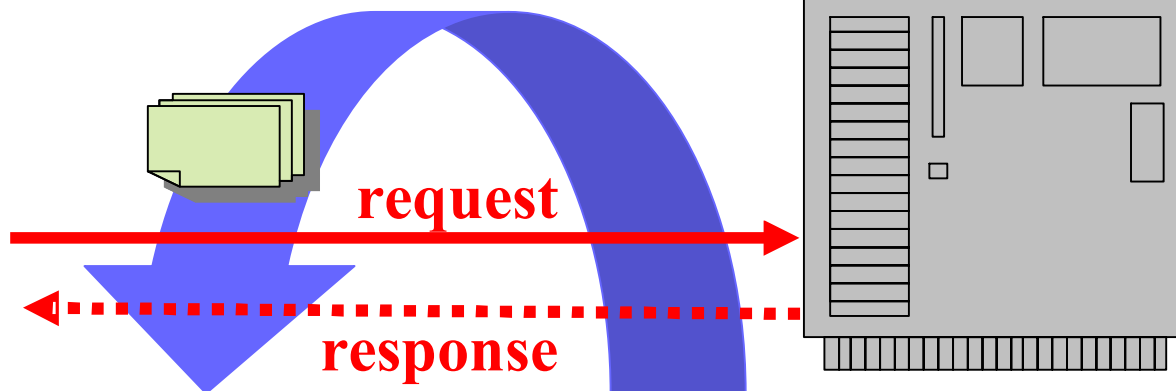
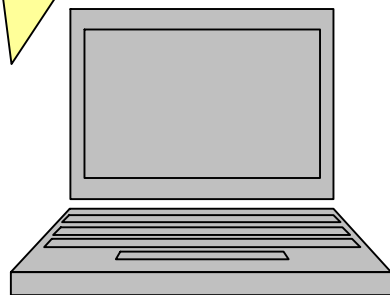
情報検索
予約サービス
電子商取引
Web掲示板

これが基本！

①記述言語:HTML



④送受信プロトコル:HTTP



③情報受信:Webブラウザ
(Internet Explorer, Navigator,etc.)

②情報発信:Webサーバ
(IIS, Apache,etc.)

WebブラウザにURLを入力するか、
Webページのリンクをたどればよい！

ご清聴ありがとうございました