

## 3.3 World Wide Web

# WWWの生い立ち

- 1989 CERN (European Center for Nuclear Research) で文書管理のために考案
  - 他の文書へのリンクを含むテキスト (HyperText)
  - HTML
- 1993 Mosaic
  - クリックするだけでリンク先にアクセスすることができるグラフィカルユーザインタフェース
- 1994 World Wide Web Consortium
- Netscape Navigator vs. Internet Explorer

# URL (Uniform Resource Locator)

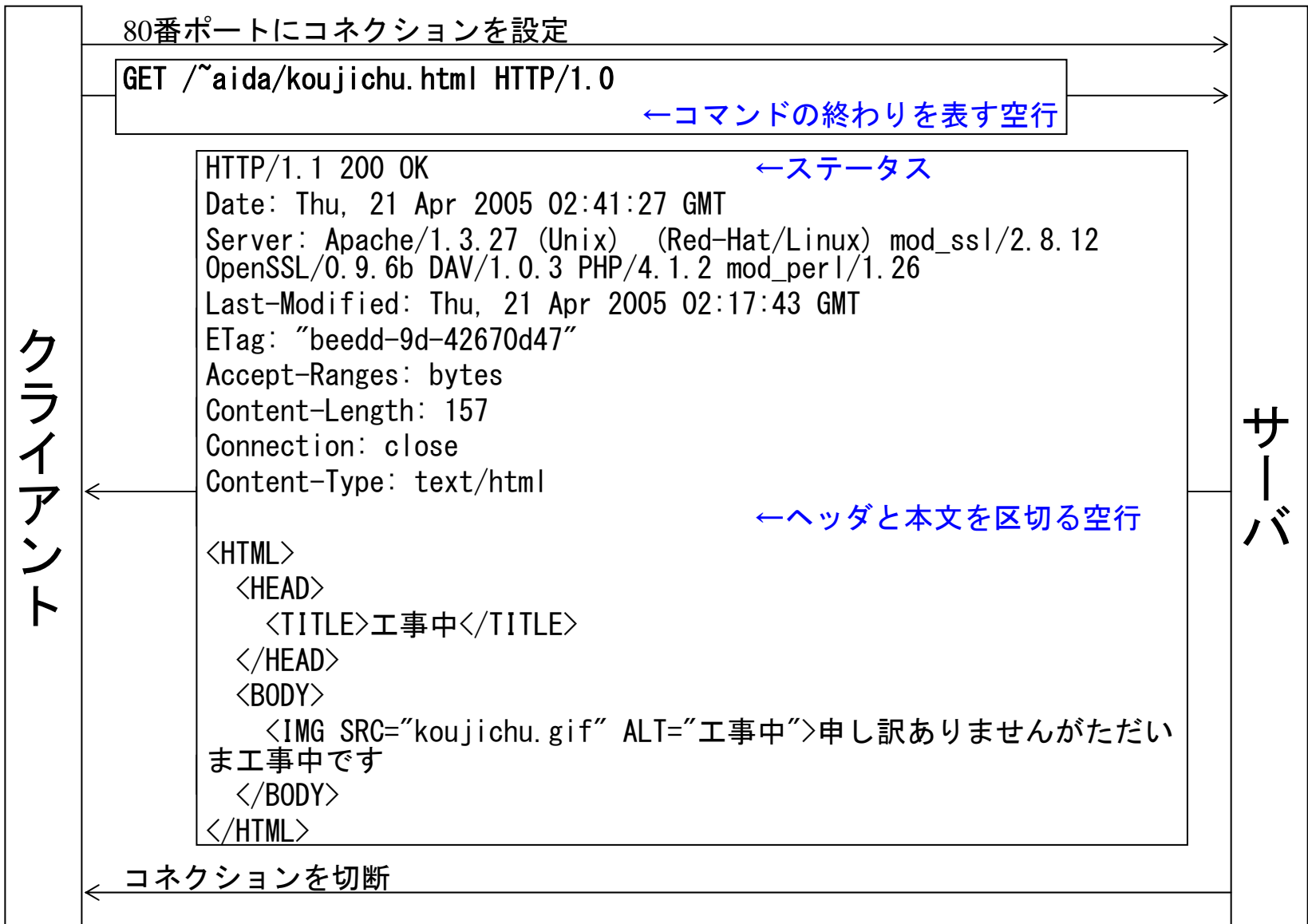
スキーム名://ドメイン名:ポート番号/ファイル名

例 : <http://www.aida.k.u-tokyo.ac.jp:8080/index.html>

- 「:ポート番号」、 「/ファイル名」 は省略可能
- 「:ポート番号」 を省略した場合にはスキームごとの既定値が使用される

# スキームの例

スキーム名	プロトコル	ポート番号
http	HTTP	80
https	暗号化されたHTTP	443
file	そのコンピュータ上の ファイル	-
ftp	ファイル転送プロトコル	21
mailto	電子メール	25

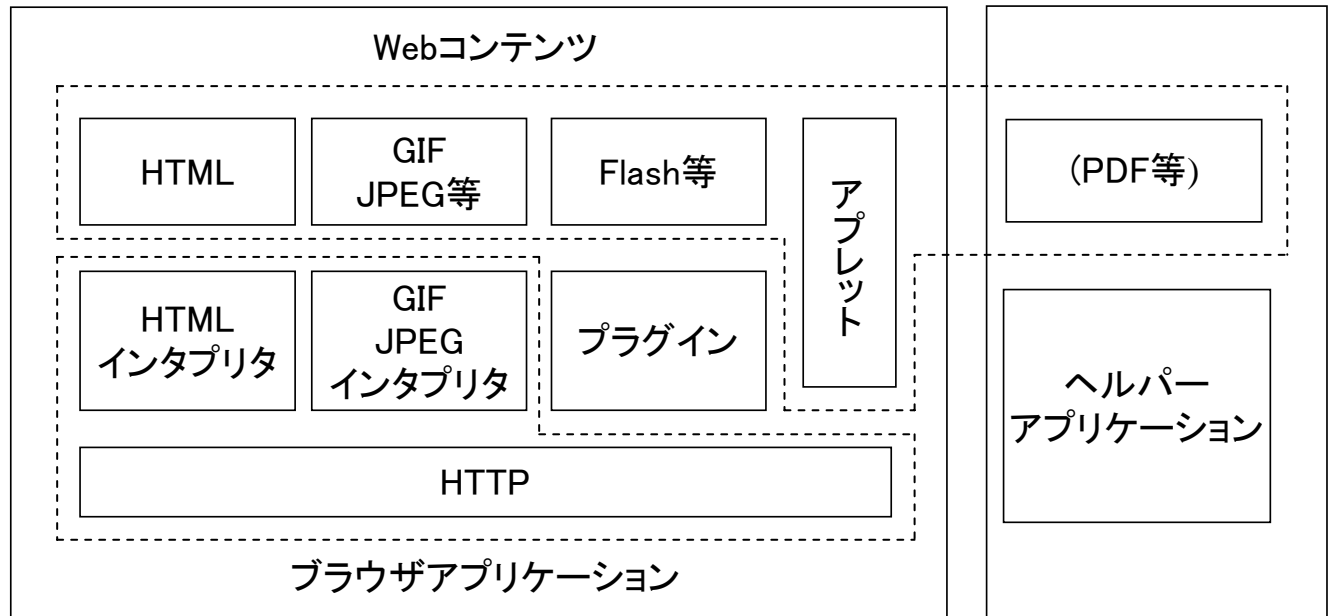


# ブラウザの構造

- 第1世代：全てのコンテンツをブラウザ自身が処理
- ヘルパーアプリケーション
  - 別ウィンドウで動作
- プラグイン
  - ブラウザウィンドウ内で動作
- アプレット
  - 実行可能なプログラムをコンテンツとしてダウンロード

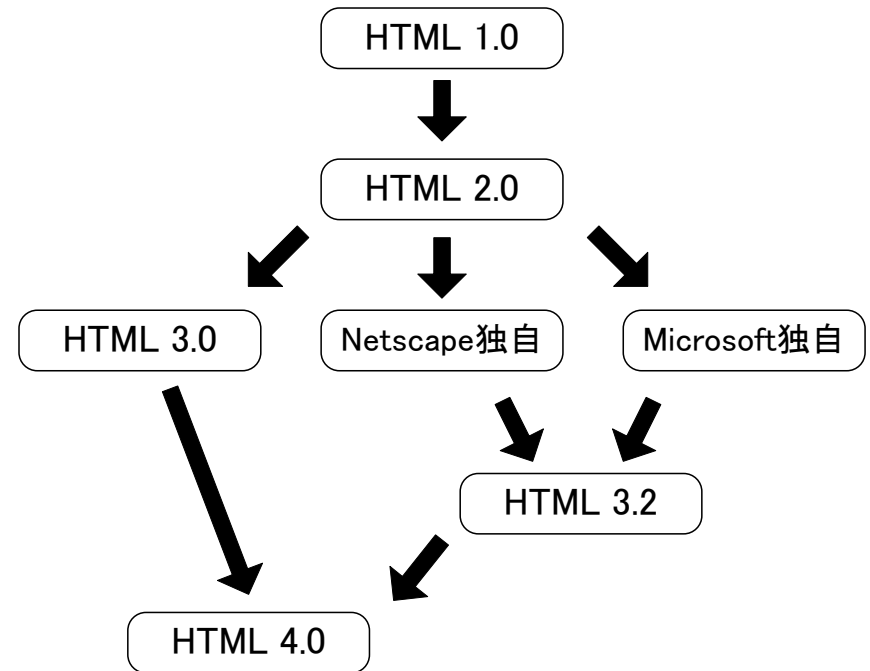
# ブラウザの構造

ユーザインタフェース



# HTML (HyperText Markup Language)

- 文書構造を記述するためのSGML (Standard Generalized Markup Language)をハイパーテキスト記述に応用





# タグ

<タグ名 属性名=“属性値”>文字列</タグ名>  
←開始タグ→ ←終了タグ→

- タグ名、属性名の大文字／小文字は区別しない
- 終了タグは省略されることもある

# 基本的なタグ

タグ名	意味	属性など
H1～H6	見出し	数字の小さい方が大きな見出し
P	段落	align=“right”など
BR	改行	終了タグなし
I	イタリック	
B	ボールド	
FONT	フォント	size, face, colorなど
IMG	画像埋め込み	src=“ファイル名” alt=“代替文字列” width, height、終了タグなし

# ハイパーリンク（アンカー）

- `<A href="URL">文字列</A>`
- `<A href="ファイル名">文字列</A>`
- 多くのブラウザでは文字列の部分の色を変えてアンダーラインを引くことでハイパーリンクの存在を意識させる

# HTML文書の全体構造

<HTML>

<HEAD>

<TITLE>タイトル</TITLE>

</HEAD>

<BODY>

文書本体

</BODY>

</HTML>

# リスト、表

タグ名	意味	属性など
UL	Unnumbered List	type=“disc”/”circle”/”square”
OL	Orderd List	type, startなど
LI	List Item	リストの項目
TABLE	Table	borderなど
TR	Table Row	rowspan(行をまたがる)など
TD	Table Data	align, valign, rowspan, colspan, nowrapなど
TH	Table Header	ボードでセンタリング

# ブラウザ独自のタグ

- BLINK
  - 点滅表示
  - Netscape Navigatorのみ
- MARQUEE
  - 横スクロール
  - Internet Explorerのみ

# XML (eXtensible Markup Language)

- 見かけではなく項目の意味づけを記述
  - 利用者が新たなタグを定義
  - 企業間でのデータ交換など多目的に利用可
- ブラウザに表示するためには見かけをXSL (eXtensible Stylesheet Language) によるスタイルシートで記述

# XMLによるコンテンツ記述例

```
<?xml version="1.0" ?>
```

```
<?xml-stylesheet type="text/xsl" href="book_list.xsl"?>
```

```
<book_list>
```

```
<book> <title> Computer Networks, 4/e </title> <author> Andrew S.  
Tanenbaum </author> <year> 2003 </year> </book>
```

```
<book> <title> Modern Operating Systems, 2/e </title> <author> Andrew  
S. Tanenbaum </author> <year> 2001 </year> </book>
```

```
<book> <title> Structured Computer Organization, 4/e </title> <author>  
Andrew S. Tanenbaum </author> <year> 1999 </year> </book>
```

```
</book_list>
```



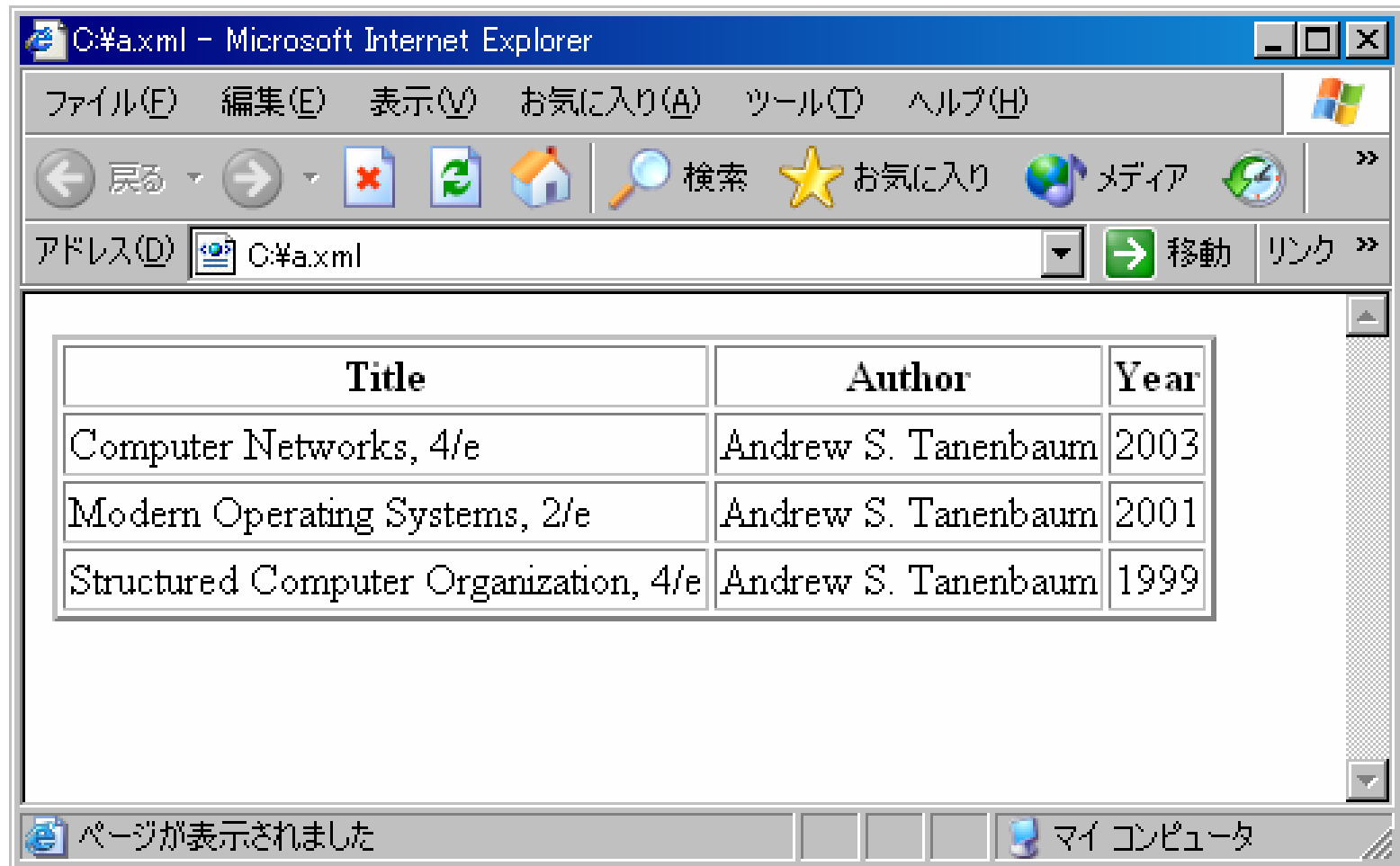
# XSLによるスタイルシート

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
<xsl:template match="/">
<html>
<body>
<table border="2">
  <tr>
    <th> Title</th>
    <th> Author</th>
    <th> Year </th>
  </tr>
```

# XSLによるスタイルシート(続き)

```
<xsl:for-each select="book_list/book">
  <tr>
    <td> <xsl:value-of select="title"/> </td>
    <td> <xsl:value-of select="author"/> </td>
    <td> <xsl:value-of select="year"/> </td>
  </tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

# 表示結果



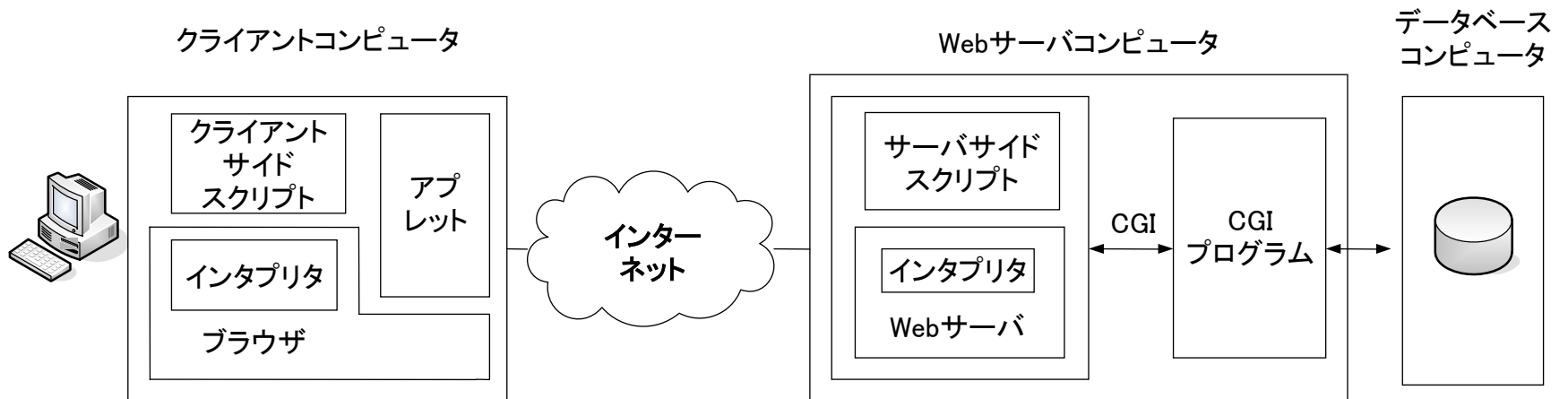
The screenshot shows a Microsoft Internet Explorer window with the title "C:\#a.xml - Microsoft Internet Explorer". The address bar contains "C:\#a.xml". The main content area displays a table with three columns: Title, Author, and Year. The table lists three books by Andrew S. Tanenbaum: "Computer Networks, 4/e" (2003), "Modern Operating Systems, 2/e" (2001), and "Structured Computer Organization, 4/e" (1999). The status bar at the bottom indicates "ページが表示されました" (Page displayed) and "マイ コンピュータ" (My Computer).

Title	Author	Year
Computer Networks, 4/e	Andrew S. Tanenbaum	2003
Modern Operating Systems, 2/e	Andrew S. Tanenbaum	2001
Structured Computer Organization, 4/e	Andrew S. Tanenbaum	1999

# 動的文書・アクティブ文書

- クライアント側スクリプト
  - JavaScript
  - VBScript
- アプレット
- サーバー側スクリプト
  - ASP (Active Server Pages)
- サーバー側プログラム
  - CGI (Common Gateway Interface)
  - Java Servlet

# 動的文書・アクティブ文書



# クライアント側スクリプトの例 (クリックなしにリンク先を表示)

```
<html>
<head>
<script language="javascript" type="text/javascript">
if (!document.myurl) document.myurl = new Array();
document.myurl[0] = "http://www.aida.k.u-tokyo.ac.jp/~aida/rat.jpg";
document.myurl[1] = "http://www.aida.k.u-tokyo.ac.jp/~aida/ox.jpg";
document.myurl[2] = "http://www.aida.k.u-tokyo.ac.jp/~aida/tiger.jpg";
function pop(m) {
    popupwin = window.open(document.myurl[m],"mywind","width=256,height=256");
}
</script>
</head>
<body>
<p> <a href="#" onMouseover="pop(0); return false;" > rat </a> </p>
<p> <a href="#" onMouseover="pop(1); return false;" > ox </a> </p>
<p> <a href="#" onMouseover="pop(2); return false;" > tiger </a> </p>
</body>
</html>
```

# クライアント側スクリプトの例 (階乗の計算)

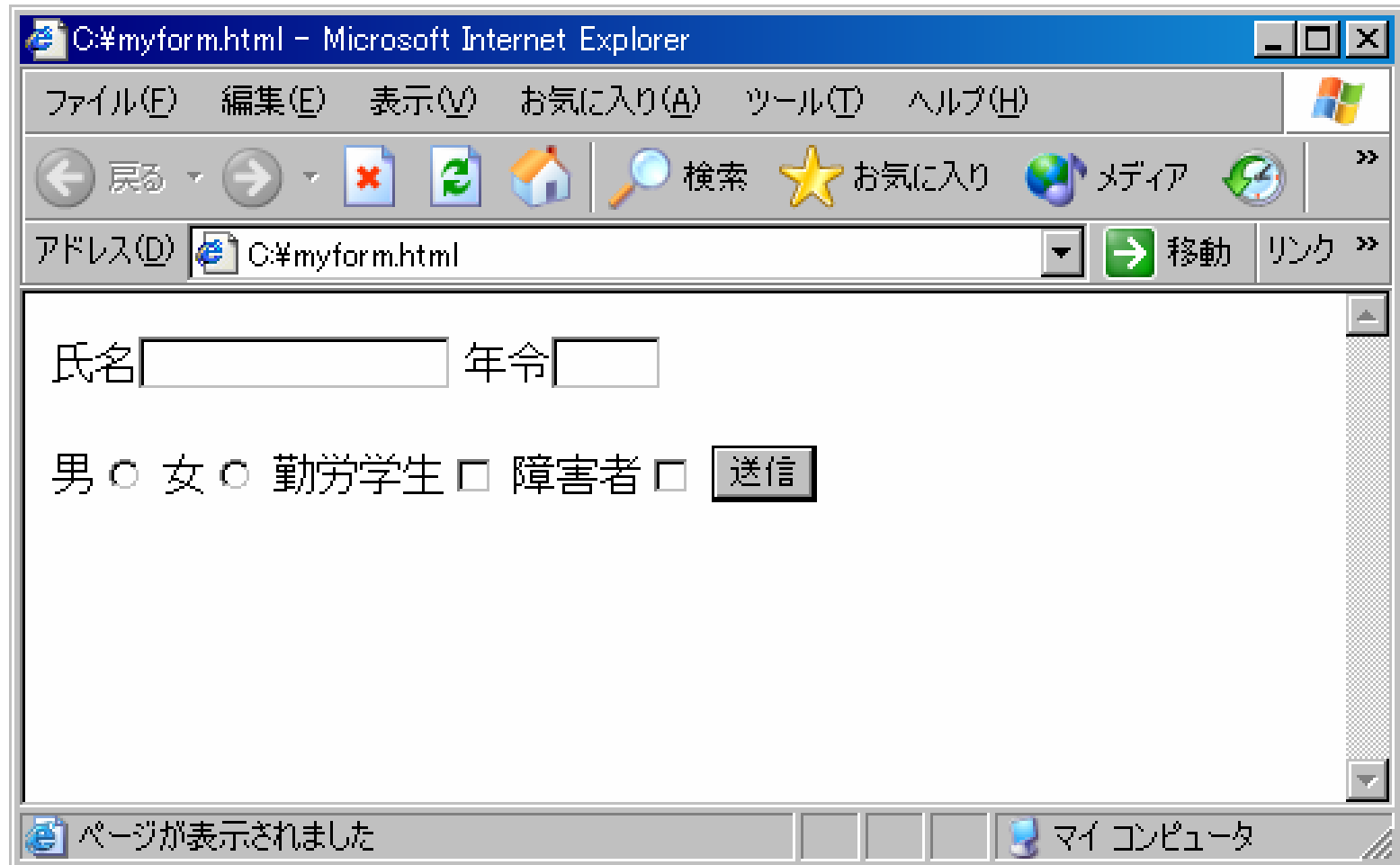
```
<html>
<head>
<script language="javascript" type="text/javascript">
function response(test_form) {
    function factorial(n) {if (n == 0) return 1; else return n * factorial(n - 1);}
    var r = eval(test_form.number.value);    // r = typed in argument
    document.myform.mytext.value = "Here are the results.¥n";
    for (var i = 1; i <= r; i++)            // print one line from 1 to r
        document.myform.mytext.value += (i + "! = " + factorial(i) + "¥n");
    }
</script>
</head>
<body>
<form name="myform">
Please enter a number: <input type="text" name="number">
<input type="button" value="compute table of factorials" onclick="response(this.form)">
<p>
<textarea name="mytext" rows=25 cols=50> </textarea>
</form>
</body>
</html>
```

# フォーム

```
<FORM METHOD="GET" ACTION="/cgi-bin/sample.cgi">  
氏名<INPUT TYPE="text" NAME="name">  
年齢<INPUT TYPE="text" NAME="age" SIZE="5">  
<P>  
男<INPUT TYPE="radio" NAME="sex" VALUE="male">  
女<INPUT TYPE="radio" NAME="sex" VALUE="female">  
勤労学生<INPUT TYPE="checkbox" NAME="student" VALUE="1">  
障害者<INPUT TYPE="checkbox" NAME="handicapped"  
    VALUE="1">  
<INPUT TYPE="submit" VALUE="送信">  
</FORM>
```



# 表示結果



# CGIプログラムへのデータの渡し方

- METHOD=“GET”の場合
  - URLのファイル名の後に付けられる  
GET /cgi-bin/sample.cgi?name=Hitoshi+AIDA&age=47&sex=male
  - CGIプログラムの環境変数  
QUERY\_STRINGに与えられる
- METHOD=“POST”の場合
  - CGIプログラムの標準入力に与えられる

# CGIプログラムの例と実行結果

```
#!/bin/sh
```

```
echo "Content-type: text/plain"
```

```
echo ""
```

```
echo '$QUERY_STRING'
```

```
echo $QUERY_STRING
```

```
echo 'stdin'
```

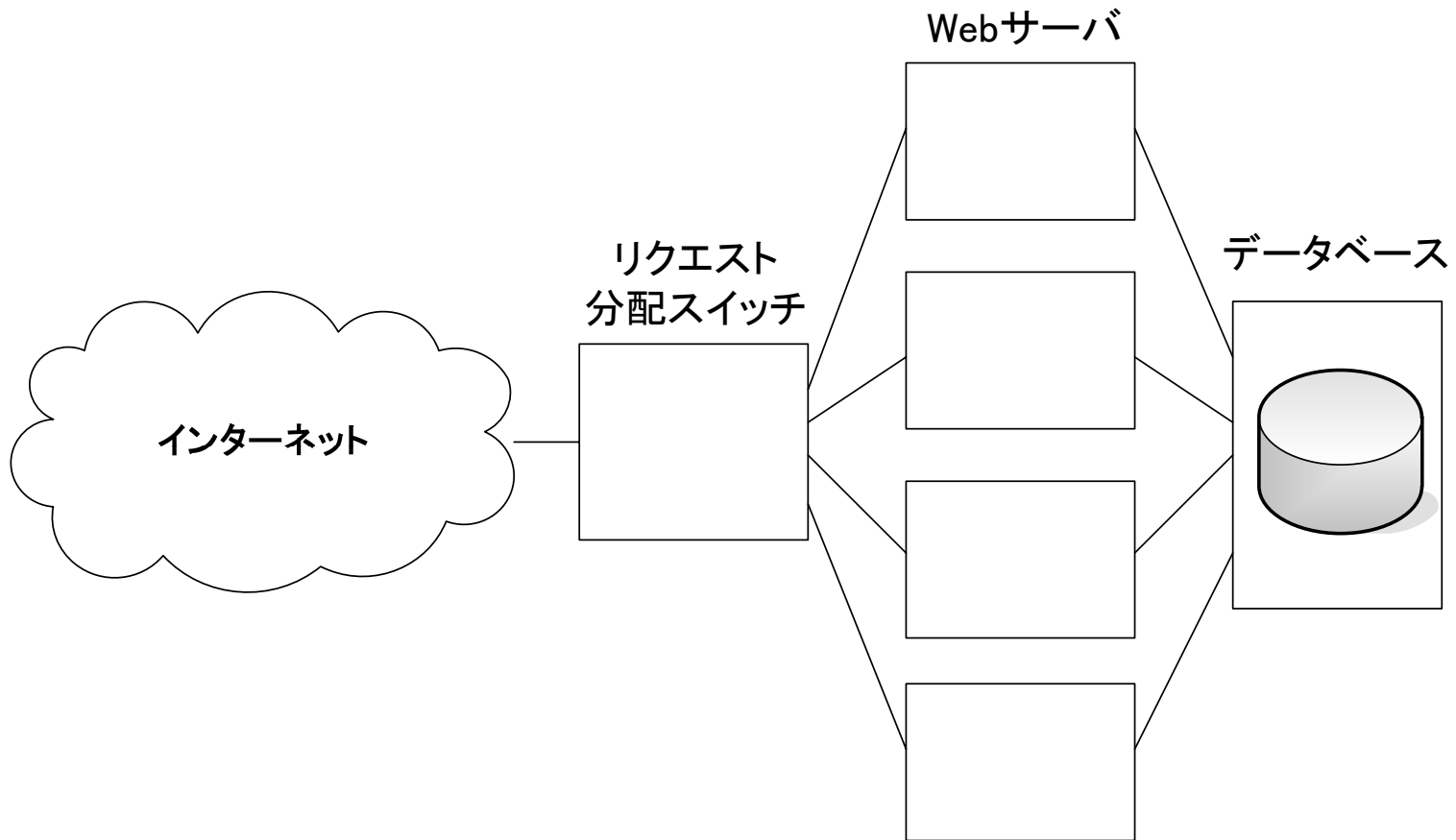
```
cat
```

```
$QUERY_STRING  
name=Hitoshi+AIDA&age=47&sex=male  
stdin
```

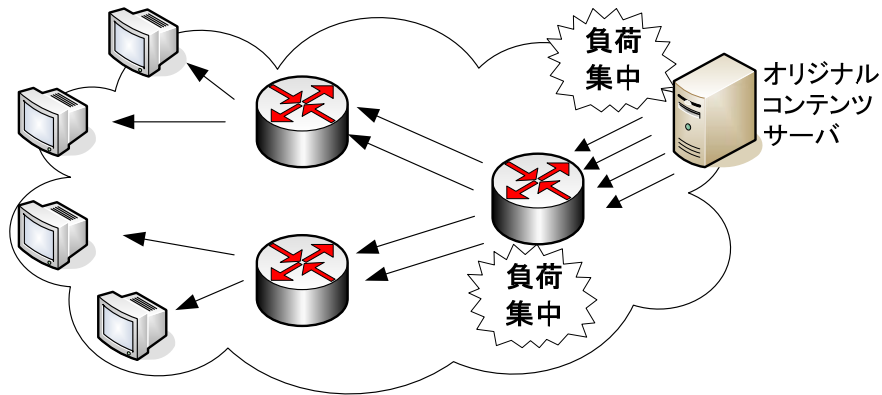
# Cookie

- HTTPリクエストをまたいで状態を保持する手段
- Set-Cookie/Set-Cookie2レスポンス
  - Set-Cookie: 変数名=値; Path=パス名 ...
- Cookieリクエストヘッダ
  - Cookie: 変数名1=値1; 変数名2=値2 ...
- CGIでは環境変数HTTP\_COOKIEで参照可

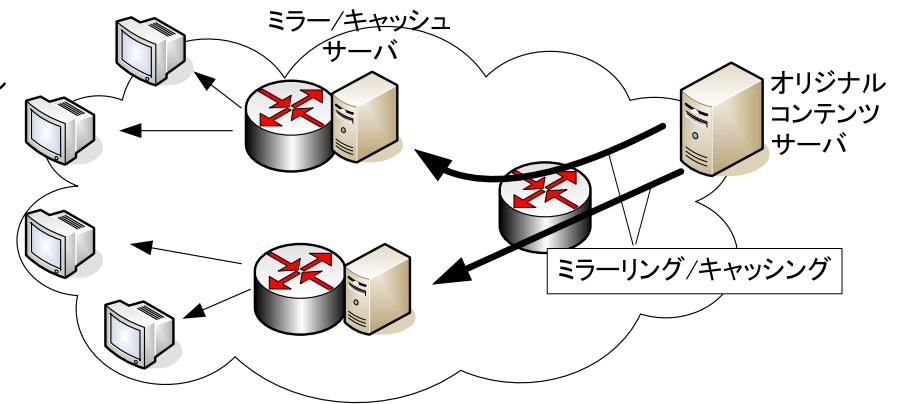
# 負荷分散



# Content Delivery Network



従来のWebサーバシステム



コンテンツ配信システム