最終更新　2023.11.9

# Webのしくみ

## Webブラウザ表示のしくみ

　Edge, Chrome, Safari, Firefoxなどのブラウザに以下のURLを与えたときに、何が起こるかを考えてみましょう。

<http://www.nara-edu.ac.jp/guide/list/index.html>

　ブラウザはwww.nara-edu.ac.jpというWebサーバと通信を行い、そのサーバが保持する /guide/list/index.htmlというファイルを取得します。次に、そのファイルの内容を解釈して表示します。拡張子がhtmlのファイルはテキストファイルです。テキストファイルなのでWordやメモ帳などで読むことができます。ブラウザの表示領域で「右クリック」→「ページのソースを表示」と操作して下さい。テキストファイルの内容を見ることができます。

　urlとして

<http://www.nara-edu.ac.jp/guide/list/>

のようにファイル名を省略した場合、多くのWebサーバはデフォルトのファイル名としてindex.htmlを補完します。

## htmlとは

　最も単純なhtmlファイルを次に示します[[1]](#footnote-1)。この例のようにhtmlファイルは「タグ」と呼ばれる < > で囲んだ文字列によって構造化されたテキストファイルです。

<!DOCTYPE html>

<html lang="ja">

<head>

<title>ここにタイトルを書く</title>

<meta charset="utf-8">

</head>

<body>

ここに何か書く

</body>

</html>

練習

　メモ帳などのエディタに上記の内容をコピペし、tmp1.htmlというファイル名で保存しなさい。保存するときにファイルの種類を「テキスト文書(\*.txt)」ではなく「すべてのファイル(\*.\*)」を選択してください。また文字コードをUTF-8に設定してください。

　次に、そのファイルをダブルクリックするか、Webブラウザに「ドラッグ＆ドロップ」して確認して下さい。

　この例のように、htmlファイルは全体が <html lang="ja"> </html> で囲まれています。その中に <head> </head> で囲まれたヘッダと呼ばれる部分と <body> </body> で囲まれたボディと呼ばれる部分があります。ヘッダの中に <title> </title> で囲まれた部分があります。この中の文字列が、ブラウザのタイトルバーに表示されます。また「お気に入り」や「ブックマーク」に取り込んだときの見出しにもなります。bodyの部分がブラウザの表示領域に表示されます。

　htmlファイルの中でタグ（< > で囲まれた文字列）の前後の「改行」は無視されます。また、文章中での改行は半角スペース1個と同等です。

　bodyの部分を次のように変えてみましょう。

<body>

<h1>これはレベル1の見出しです。</h1>

<hr>

<p>

一つ目の段落。

</p>

<p>

二つ目の段落

</p>

</body>

実習

　上記のように書き換えて保存し、ブラウザの「このページを再読み込みします（Edgeは「更新」）」のアイコンをクリックして確認しなさい。

　この例から分かるように、Webブラウザはhtml形式で書かれたファイルを解釈して表示する働きを持っています。

　<h1>　</h1>　をタグと呼びます。h1はレベル1の見出しを作る命令です。<h1> は開始タグでその後に内容を書き、</h1> が終了タグです。<p> は段落の開始を示す命令で、</p> で段落の終了を表します。次の段落との間で改行を行い、縦方向のスペースを挿入します。<hr> は単独で使われるタグで終了タグはありません。水平線を引く働きをします。単独で使われるタグを「空の要素 (empty elements)」と呼びます。

　htmlは様々な命令を持っています。いくつか紹介しましょう。先ほどのファイルのbodyの部分を次のように書き換えて下さい。

<body>

<h1>レベル1の見出し</h1>

<h2>レベル2の見出し</h2>

<hr>

<p>ここに文章</p>

<ul>

<li> 番号なしリストです </li>

<li> unordered list </li>

<li> 3行目 </li>

</ul>

<hr>

<ol>

<li> 番号付きのリストです </li>

<li> ordered list </li>

<li> 3行目 </li>

</ol>

</body>

実習

　上記のように書き換えて保存し、再読み込みして確認しなさい。

　htmlには「表を作る」「画像を表示する」「リストを作る」など様々なタグがあります。詳しくは

　　　　　<http://www.htmq.com/>　　　htmlクイックリファレンス

　　　　　<http://www.tohoho-web.com/www.htm>　　　とほほのWWW入門

などを参照してください。

　htmlのタグの中で一番重要なタグを紹介します。それはリンクを表すタグです。まず、もう1つファイルを作って下さい。ここでは例として tmp2.html というファイルを作り、tmp1.htmlと同一のフォルダに置いてください。次のような内容にしましょう。

<html lang="ja">

<head>

<title>もう一つのファイル</title>

</head>

<body>

これはもう一つのファイルです。

</body>

</html>

　そして、先ほどの tmp1.html の body の部分に次の部分を挿入してください。

<p>

この文字は普通の文字<br>

<a href="file:tmp2.html">ここを押す</a><br>

<a href="http://www.google.com/">ここを押す</a><br>

この文字も普通の文字

</p>

　ここで <br> は単独で使われるタグであり、改行する命令です。「更新」ボタンを押して tmp1.html を再読み込みして下さい。<a> と </a> で囲まれた部分の文字の色は他の文字とは異なり、下線が引かれます（ブラウザの設定によっては異なるかも知れません）。そして、その文字をクリックすると href="...." で示されたファイルへジャンプします。これをハイパーリンクと呼びます。このハイパーリンクがhtml形式の一番大きな特徴であり、html形式のファイルをハイパーテキストと呼びます。

　リンクの上にマウスカーソルを持っていくと、ブラウザのステータスライン（一番下の行）にリンク先のurlが表示されます。

　href="...." のリンク先と <a> と </a> で囲まれている文字列は個別に設定可能です。学習用サイトにわざと表示文字列とリンク先文字列を違えた例があるので、体験してみて下さい。

実習

　上記のtmp2.htmlのファイルを作成し、tmp1.htmlのリンクをクリックしてtmp2.htmlが表示されることを確認しなさい。

　ホームページビルダーなどのWebサイト作成ツールを使うと、htmlのタグについての知識がほとんどなくてもhtmlファイルを作成することができます。しかし、htmlの仕組みを理解した上で使うと、より能率的にサイトを作成することができるでしょう。

## スタイルシート

　前節でタグ <h1> を使った例を示しました。スタイルシートの機能を使うと、タグh1の見栄えをカスタマイズすることができます。<head> ブロックに以下のようなスタイルの定義を入れると、h1 の外見が変わります。

<style>

h1{

 color: darkblue;

 margin-left: 0px;

 margin-right: 0px;

 margin-top: 0px;

 margin-bottom: 20px;

 padding: 5px;

 width: 600px;

 border-style: solid;

 border-width: 0px 0px 2px 20px;

 border-color: #0000ff;

}

</style>

　この例ではhtmlファイルの中にスタイルを記述しましたが、<style> ～ </style> の部分を別ファイルに分離することができます。各々のhtmlファイルからスタイルを記述したファイルをインクルード（include: 取り込む）すると、サイトに統一感ができます。

　上記のスタイル記述部分から、最初の <style> と最後の </style> を削除した部分をmystyle.cssというファイルに入れてセーブし（スタイルシートの拡張子はcssです）、head ブロックに次のように書くと、スタイル部分を別ファイルに入れることができます。

<link rel="stylesheet" type="text/css" href="mystyle.css">

　段落中のある文字列だけにスタイルを設定するには

<p>文章中の<span style="color:red; background-color:black;">ここの部分だけ赤字で黒背景</span>です</p>

のようにspanというタグを使います。ある段落だけスタイルを適用するには

<p style="color:red; background-color:black;">この段落だけ赤字で黒背景です。</p>

のように書きます。いくつかの段落に対してスタイルを適用するには

<div style="color:red; background-color:black;"><p>文章1</p><p文章2</p></div>

のようにdivというタグでスタイルを設定したい範囲を囲みます。

実習

　スタイルシートを使ってH1を定義し、スタイルシートを書き換えるとH1の体裁が変わることを確認しなさい。

## 1つのページを構成する要素

　1つのページは1つのファイルで構成されているとは限りません。1つのページが複数のファイルから構成され、そのファイルが複数のサイトに分散している場合もあります。「引用を含むページ」を表示して下さい。別ファイルに記載した演習において、問に答えて下さい。

　この例のように、あるサイトのページの一部が別のサイトから引用されていることがあります。自分のサイトを作るときに別のサイトを引用する場合は、引用であることが明白であるようにしましょう。

## サーバのログ

　ブラウザがWebサーバに対してファイルの送信を要求する場合、最小限で済ませるには

GET /~yabu/index.html HTTP/1.0

(空行)

でファイルを取得することが出来ます。しかし、Chromeなどのブラウザはそれ以外にも情報を送っています。「ブラウザの種類とバージョン名」などの情報です。例えば、私の机上のWindowsパソコンのChromeがサイトにアクセスするとき、次のような情報を送っています。これはネットワーク通信の学習のときに利用したserver.pyを80番ポートで待ち受けさせ、urlにhttp://localhost/index.html　と入力することで得られます。

GET /index.html HTTP/1.1

Host: localhost

Connection: keep-alive

sec-ch-ua: "Google Chrome";v="119", "Chromium";v="119", "Not?A\_Brand";v="24"

sec-ch-ua-mobile: ?0

sec-ch-ua-platform: "Windows"

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.7

Sec-Fetch-Site: none

Sec-Fetch-Mode: navigate

Sec-Fetch-User: ?1

Sec-Fetch-Dest: document

Accept-Encoding: gzip, deflate, br

Accept-Language: ja,en-US;q=0.9,en;q=0.8

　sec-ch-ua-platformでOSがWindowsであること、sec-ch-uaでブラウザがGoogle Chromeであることを送信しています。Edgeは以下のような情報を送っています。

GET /index.html HTTP/1.1

Host: localhost

Connection: keep-alive

sec-ch-ua: "Microsoft Edge";v="119", "Chromium";v="119", "Not?A\_Brand";v="24"

sec-ch-ua-mobile: ?0

sec-ch-ua-platform: "Windows"

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36 Edg/119.0.0.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.7

Sec-Fetch-Site: none

Sec-Fetch-Mode: navigate

Sec-Fetch-User: ?1

Sec-Fetch-Dest: document

Accept-Encoding: gzip, deflate, br

Accept-Language: ja,en;q=0.9,en-GB;q=0.8,en-US;q=0.7

　sec-ch-uaで自分がMicrosoft Edgeであることを送っています。

　初期の頃はHTTPのプロトコルとしてバージョン1.0を使っていましたが、現在のブラウザはバージョン1.1を使います。バージョン1.1ではGETの行に加えてHost: の行が必須です。

　Webサーバのログを見てみましょう。私のスマホで「arduino　覚え書き」で検索して、私のサイト内のarduinoに関するページを閲覧したとき、apacheと呼ばれるWebサーバは以下のログを記録します。

sp49-104-23-253.msf.spmode.ne.jp - - [13/Nov/2023:18:23:09 +0900] "GET /~yabu/soft/arduino.html HTTP/1.1" 200 20800

　docomoのスマホがarduino.htmlというファイルにアクセスしに来たことが分かります。

　サーバとクライアントはネットワーク通信を行いますので、サーバはクライアントのIPアドレスを知っています。Webサーバのログには、クライアントのドメイン名（この例では sp49-104-16-50.msf.spmode.ne.jp）あるいはIPアドレスのどちらかが記録されます。この例の場合、キャリアであるspmode.ne.jp（NTTドコモ）が個人情報を開示すれば、アクセスした人物を特定することができます。

　ocnのようなプロバイダ経由でアクセスが来たときは、アクセス元のルータ（住宅内に設置されたONU）までは特定できますが、そのルータに接続した機器までは分かりません。通常、ONUにはパケットのログを蓄積する機能を持っていないと思われます。

## クッキー（cookie）とは

　現在のクッキーは複雑ですが、ここでは最も単純な場合を例としてクッキーの説明をします。

　クッキーといえば食べるお菓子ですが、ここで取り上げるクッキーはWebサーバとWebブラウザの間でやりとりされます。まずは次の図をご覧下さい。



図1　クッキーの動作

　図のように1回目のアクセスの時、サーバからクライアント（ブラウザ）へクッキーをセットせよという命令が送られます。具体的には次のような行です。

Set-Cookie: ans=yes; Expires=......; Path=/abc; Domain=denki.nara-edu.ac.jp

　この例ではansという名前のクッキーの内容をyesに設定しています。Expires= の部分はそのクッキーの有効期限であり「○○年○月○日○時○分○秒」を表す値がエンコードされています。Path= がそのクッキーの返送対象となるディレクトリ、Domain= がクッキーの返送対象となるドメインを表します。

　ブラウザ側ではこのクッキーの値を保持しておき、「domainが後方一致するサーバ（ex. denki.nara-edu.ac.jpあるいはwww.denki.nara-edu.ac.jpのように右側が一致するホスト名を持つサーバ）」の「pathが前方一致するフォルダより下にあるファイル（ex. /abc/def/g.htmlのように左側が共通するファイル）」をGETするとき、保持していたクッキーの値を送ります。

　クライアントがアクセスするページがphpファイルあるいはcgiファイルの場合、phpやcgiのプログラムは送られてきたリクエストに含まれているcookieの値を読みとることができます。これによりクライアントの情報（例：前回いつ訪れたか、パスワードの認証済か否か）などの情報をサーバ側で得ることができます。たとえばGoogleで1回認証された後、2回目以降はパスワードなしで接続できるのは、cookieを使用しているからです。

　Chromeでcookieを確認するには以下のように操作します。知らない間に大量のCookieを受け入れていたことが分かります。

・ブラウザ上で右クリック → 検証

・デバッグウィンドウの ≫ マークを押しApplicationを選択

・左のフレームのStorage -- Cookies で名前と値を閲覧できる

・ページを切り替えるたびに、そのページをGetするときに送るクッキーが表示される

・Filterという文字の右隣のアイコンをクリックすると、そのサイトのクッキーを削除する

　全てのCookieを削除するには以下のようにします。

・右端の3点マーク → 設定 → プライバシーとセキュリティ → 閲覧履歴データの削除　Cookieにチェックを入れて「データを削除」

　Edgeの場合、次のように操作します。

・「…」→「その他のツール」→「開発者ツール」→「≫ を押してアプリケーションを選ぶ

・「左側のCookie」の▼をクリックし、サイトの行をクリック

　全てのCookieを削除するには以下のようにします。

・「…」→ 設定 → プライバシー、検索、サービス → 今すぐ閲覧データをクリア

実習

　以下のサイトにアクセスしなさい（ブラウザのurl欄に入力するか、ctrlを押しながらリンク先をクリックしなさい）。

<http://denki.nara-edu.ac.jp/~yabu/edu/jyoho/html/cookie-send2.php>

　クッキーの名前、コンテンツ、ドメイン、有効期限を調べなさい。

## 動的なサーバ

　多くのWebサイトはアクセスするユーザによって異なる挙動を示します。これはサーバにおいて、アクセス時にプログラムが実行され、サーバの応答を変えているからです。

　Webサーバでプログラムを実行するしくみはの一つとしてphpというプログラミング言語を用いる方法があります。ブラウザにページが表示されるとき、

1. ブラウザがサーバに対してファイルを要求する

2. サーバが要求されたファイルを送り返す

という処理が背後で行われています。サーバが送り返すhtmlファイルの中にphpというプログラミング言語で書いたプログラムを含めておくと、phpで書かれた部分は、プログラムを実行して得られる文字列に置換されます。

実習

(1)「phpの例（時刻表示）」と「その中身」をクリックし、「phpで書かれた部分」と「そのプログラムの実行結果」を確認しなさい。

(2)「python cgiの例2（2の累乗）」と「その中身」をクリックし、表を生成しているのはpythonのコードであることを確認しなさい。

　phpやpythonのプログラムはutf-8という文字コードで書かれています。そのままChromeやEdgeで表示すると文字化けしてしまいます。Chromeの場合、Charsetというアドオンを入れることで文字化けせずに読むことができます。アドオンの入れ方は以下の通りです。「googleで「Chrome　拡張機能　Charset」と入力」→「ChromeウェブストアのCharsetを「Chromeに追加」のボタンを押す。

　文字化けした状態で「☆の一つ右のアメーバのような形のアイコンをクリック」→「Charset」→「Unicode (UTF-8)」

## ブラウザ側で実行されるプログラム

　多くのWebページのhtmlファイルの中にJavaScript（プログラミング言語名）で書かれたプログラムが含まれています。ブラウザはJavaScriptを実行します。その結果、動的なページ（見る人、見るタイミング、ユーザの操作などによって異なるページになること）になります。

実習

　「JavaScriptの例（2の累乗）」をクリックして下さい。「ソースを表示」でhtmlファイルの内容を見ることができます。JavaScriptのプログラムが表の部分を生成していることが分かります。

　2の累乗を表示するページはphpのページとJavaScriptのページの2通りがあり、見かけ上の結果はほぼ同じです。しかし、そのページが生成される方法は全く異なります。phpを含むページではサーバ上でプログラムが実行されたのに対して、JavaScriptを含むページはクライアントのWebブラウザ上でプログラムが実行されました。

　Webブラウザ上におけるクリックやマウスの移動などのイベントに対応してJavaScriptが動作し、ページの内容を動的に書き換えることができます。

実習

　「JavaScriptの例（ボタンクリックのイベント）」「JavaScriptの例（マウスのイベント）」を表示して下さい。イベントによって画面が動的に変わることを確認して下さい。

　「JavaScriptの例（マウスのイベント）」を開いて下さい。「表示」→「ソース」で内容を見て、「鹿」「夕暮れ」の文字に設定されているタグを確認して下さい。それらのタグの上にマウスカーソルが来ると、関数が実行されます。関数を確認しなさい。

　「JavaScriptの例（ポップアップウィンドウもどき）」を開いて下さい。ウィンドウもどきが移動します。ソースを表示し、「この部分が移動します」という文字列に設定されているタグ <div> を確認しなさい。<div> はいくつかの段落をまとめて取り扱うためのタグです。

　JavaScriptがサーバと通信して動的にページを書き換えることができます。これはAjaxと呼ばれる技術です。Googleの検索ウィンドウに文字を入力すると、1文字入力するたびに候補を表示します。これは1文字入力するたびにサーバと通信して候補を取得しているからです。Googleの地図機能もAjaxを利用しています。

　「JavaScriptの例（サーバとの通信　同期）」を開きなさい。テキストボックスで「キーを押した」というイベントが発生するたびに、テキストボックスの内容をサーバに送信します。サーバ側のプログラムxml\_manage2.phpは送られてきた文字列について、大文字を小文字に変換し、小文字を大文字に変換し、その他の文字はそのまま返します。

## サーバへ情報を送る

　ここではブラウザ側からサーバへ情報を送る方法を学習します。

　「引数付きリンク1」と「引数付きリンク2」は同じリンク先です。マウスカーソルを上に持って行って確認して下さい。ただしリンク先のファイル名の後が異なります。それぞれ以下のようになっています。

　　　　html/action2g.php?key1=yamada&key2=tarou

　　　　html/action2g.php?key1=suzuki&key2=hanako

　"?" より左側がurlです。その後ろにデータが記述されています。「キー」と「値」の組み合わせで表されています。"key1", "key2" がキーで = の後が値です。"&" は区切りを表します。

　action2g.phpはこれらの情報を受け取ります。ここでは、action2g.phpは送られてきた情報を表示するだけのプログラムですが、実際のWebサイトでは送られてきた情報を記録したり、処理したりします。

実習

　Webサーバへ情報を送る（Get）をクリックして下さい。

　「3年生」を選び、名前に「abc」と書いて「送信」ボタンを押したときにアクセスするurlを確認しなさい。キーと値が含まれています。確認しなさい。

　サーバにデータを送る主な方法は2通りです。1つはこの課題で用いたGETメソッドを使う方法です。もう1つはPOSTメソッドを使う方法です。以下のような違いがあります。

1. GETメソッドでデータを送る場合、データ量の制限があります。大きなデータを送るときはPOSTメソッドを使います。

2. GETメソッドで送るとブラウザのurlの欄に送るデータが表示されます。すなわちデータがブラウザの履歴やサーバのログに残ります。

発展学習

　「Webサーバへ情報送る（Get)」（form-like.html）をデスクトップなどに保存します。次に、メモ帳などのエディタで開き、url = の部分を以下のように書き換えて保存します。

url = "http://localhost/action2g.php" + 以下同じ

　localhostは自マシンを意味するので、「送信」ボタンを押すと、自マシンの80番ポートに接続します。

　自マシンでserver.pyを起動し、80番ポートを待ち受け状態にします。server.pyはWebサーバのふりをすることになります。

　書き換えて保存したファイルをEdgeに読み込み、学年を選択し、名前を入れて「送信」ボタンを押します。

　受信文字列の一番先頭行を確認して下さい。入力した情報を入れてGETしています。

　Edgeはサーバからの応答を待つ状態になります。server.py側でソケット（一番番号が小さいもの）を指定して「切断」するか以下の応答を返しなさい。

HTTP/1.1 200 OK

Content-Length: 2

OK

1. 以下は省略しても表示に支障はないようです。

<!DOCTYPE html>

<html lang="ja"> のうち lang="ja" の部分

<meta charset="utf-8"> [↑](#footnote-ref-1)