

サービス提供型ホームページの技術

第1章 サービス提供型ホームページ

プロジェクトの全体像

このプロジェクトでは、ネット上に自分たちの考えるサービスを作って見る。サービスのイメージの例は、ブログ(記事を書き込む、他人が見られる)や、ショッピング(アマゾンとか楽天のように、商品を選んで購入できる)や、授業登録、検索、アンケートなど、いろいろなものがある。作って見る中で、サービスの動作のしかた、使われる技術を理解し、試してみる。後で何かサービスを作りたいくなった時に、どうやって作ればいいのかの筋道が分かる。



プロジェクトの前半では、上のようなサービスを作るための仕組み・構造を理解し、そのための技術の要素を使えるように学ぶ。後半では、自分たちで作ってみたいサービスを考え、そのための仕組み・システムを実現してみる。最後の回に、それについて売り込みの発表を試みる。

このメモは、前半で学ぶ、サービスのための仕組みと技術的な知識について説明し、かつ実際に試してもらうための教材である。

本プロジェクトで導入する技術要素を、簡単に紹介する。細かい内容は続く章で解説するので、本章では今後の話の前提と、話がどう展開するかの道筋を紹介する。

CGIプロジェクト1

プロジェクトの仕組みの考え方・前提

取り組み方は、

- * 基本的(原始的)な作り方で要素を作って見る。最近では実世界ではここまで原始的な方法では作らない。予め準備された「フレームワーク」(枠組み)の中に、サービス独自の記述を書きくわえることで、全体を容易に作り上げるのが主流らしいが、ここでは使わない。
 - * サーバー側でのプログラミングを中心に理解する。端末(クライアント)側での処理(たとえば javascript を動かしてスムーズにスクロールしたり入力値をチェックしたりするなど)は、時間の都合もあって省く。挑戦したい人は自分で勉強してほしい(たとえば ajax など)。
- のようにする。

本プロジェクトで利用する技術要素には、

- * ホームページ(技術用語でいえば Web ページ)の基本 ~ 「静的」なホームページ
具体的には HTTP という転送方法と、HTML という内容記述の方法(言語)
どんな画面を、どうやったら作れるのか
 - * CGIプログラミング ~ 「動的」なホームページの技術
どのようにサーバー側の処理(=プログラム)が動くのか
どうやって入力を画面からプログラムに渡すか、プログラムから画面に出力するか
プログラムの記述言語として何を使うか
 - * データベース
プログラムが受け取った情報(データ)を長期的に格納・保存するための入れ物
どのような形のデータを格納・保存できるのか、どんな操作ができるのか
プログラムからデータベースを使う(格納する・取り出す)にはどうするのか
がある。
- それぞれの要素について、初歩的な部分をざっと学んだ上で、簡単な使い方を試すことによって理解しようと思う。

Web サーバ

CGIプロジェクト1

静的ホームページと動的ホームページ

現在、多くのネットワークサービスは、「ホームページ」の仕組みを使って提供されている。まず、ホームページの技術を確認しておこう。

《静的なホームページと動的なホームページの違い》

静的なホームページは、ホームページの技術が発明された当時(1995年頃)に想定された使い方である。あらかじめ作ってある表示内容(コンテンツ)が、ユーザ端末上のブラウザ(Internet Explorer, Netscape, Firefoxなどの閲覧アプリケーション)に表示される。

コンテンツはいつでも同じ内容であり、サーバーに一旦アップしたら勝手には変化しない。端末のユーザが入力することはできないし、サーバーの側でその時に合わせたコンテンツを作ることもできない。静的なホームページと呼ぶことにする。

これに対して、動的なホームページは、ページの表示内容をサーバーが(プログラムで)作り出す。たとえば、今の時刻を表示したり、今までの訪問者数を表示したりできる。これらの値はサーバーが持っている情報であり、その場その場で変化する。更に、それに加えて端末のユーザがページに対して入力することができる。その入力を使って、サーバー内でプログラムが処理を行い、その結果を端末に表示する。このような形態を、動的なホームページと呼ぶことにする。

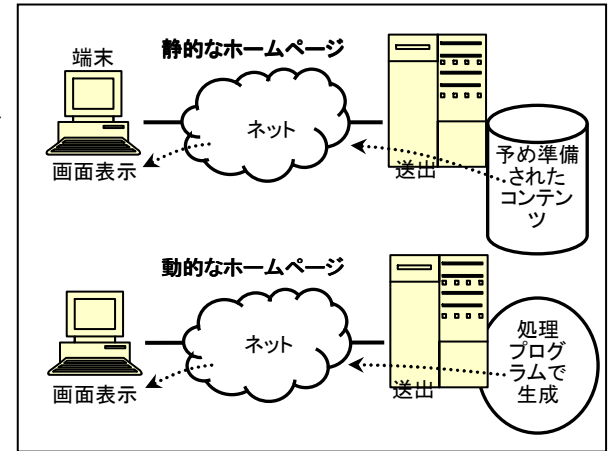
《静的なホームページのための技術》

静的なホームページで使われる技術は、

- (1) ページ内容(コンテンツ)を記述する仕組み(言語)HTMLと、
 - (2) そのページ内容をサーバーから端末へ送り届けるための転送の仕組み(プロトコル)HTTP、
- の2つから成る。

静的なホームページの場合、サーバー側では、ページ内容がデータとして保存されており(通常はファイルとして保存)、ユーザが端末からそのページを要求されると、

サーバーはネットへ送り出す。端末は、送られてきたページ内容(コンテンツ)を画面に表示する。



ページ内容(コンテンツ)は、それを記述するための仕組み(約束、「HTML言語」)に従って書かれているものであり、端末のブラウザ(=表示プログラム)は、送られてきた内容を、HTMLで決められた言語のルールに従って解釈し、画面に表示する。

単純な(ベタな)文字だけではなく、太字にしたり、図や絵を入れたり、項目リストを作ったり、表を作ったり、ということを示すことができる。

このHTMLの具体的な書き方は、第2章で簡単に学ぶ。

サーバーから端末まで、ネットを経由してページ内容を転送するための仕組みについては、本プロジェクトでは詳細には触れない。ネットワークの授業で勉強してほしい。ここではそういう仕組みが裏にあって送られてくる、とだけ理解しておけば済む。

《動的なホームページ》

「静的な」ページでは、ページ内容はあらかじめ準備された変化しないものだったのに対して、「動的」なページでは、サーバーがページ内容をその場で生成する。生成はサーバーにいるプログラムが行う。たとえば、ショッピングのサイトでは、商品の個数を入力すると、合計金額が表示されるが、これはユーザ入力からサーバー上のプログラムが計算し、その結果を表示している。ショッピングでは更に、購入したものを倉庫から送り出すための処理や、クレジットカードなどでの決済の処理も行われるが、これもサーバー上のプログラムが動作して指示を出す。このように、裏にプログラムが隠れていて、必要な処理を行い、結果の画面を生成して表示するようなページを、動的なホームページと呼ぶことにする。

この動的な仕組みは、最初に作られた静的なページの仕組みの上に追加する形で作られた。つまり、「静的」で決められていた、ページ内容を書くための言語 HTML と、ページ内容の転送の手順 HTTP は、多少の拡張をただけで、ほとんどそのまま使っている。追加したのは、端末側から入力できるようにする仕掛けである。

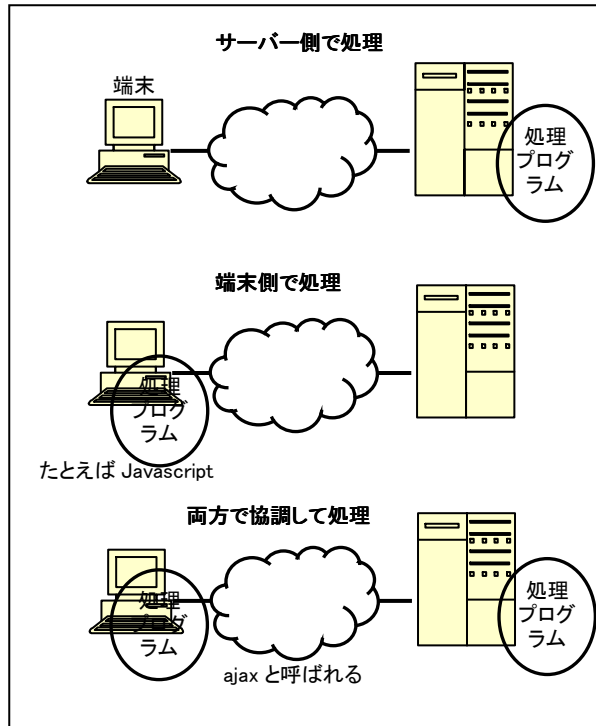
《サーバーで処理 vs 端末で処理》

動的ページのサービスを実現しようすると、図にあるように、処理をするプログラム処理をする場所が、

- ①サーバー側にプログラム処理がある場合、
- ②端末側にプログラム処理がある場合、
- ③両方で協調して処理する場合、

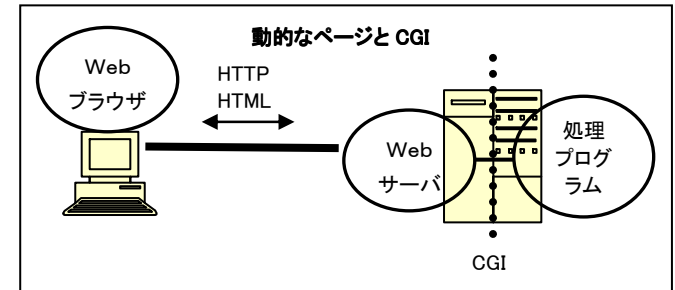
が考えられる。

本プロジェクトでは簡単な①を試す。最近では、ユーザの体感応答を改善などの理由から、③が主流だが、技術要素が増えるのでここでは取り上げない。各自の今後の勉強に委ねる。



CGI (Common Gateway Interface)

動的ページを提供する場合、処理プログラムと Web サーバ(ページデータを端末へ送り出す部分)は、分けて考えるのが普通である。Web サーバは静的ページと同じ形のまま



として、処理プログラムとやり取りするための入り口を右側に付け足した形になる。その、処理プログラムとやり取りするための境界(インターフェース)を CGI (Common Gateway Interface) と呼んでいる。

CGI は、右側の処理プログラムの部分が何の(プログラミング)言語で書かれているかには依存しないので、処理プログラムを書くのにいろいろな言語を使うことができる。たとえば、Java でプログラムしてもよいし、Perl というスクリプト言語を用いることも多いし、本プロジェクトで使う PHP という言語も広く用いられている。

プロジェクトで使う枠組の予備知識

動的なホームページの仕組みを実現する枠組みは、いろいろと提供されているが、このプロジェクトでは、広く使われていることと、簡便であることから、(Apache+PHP+MySQL)の枠組みを利用する。大規模なサービスで性能を要求される場合や、きっちりしたセキュリティを要求される場合、またサービスの開発をする業者が慣れているなどの理由で、他の枠組みを用いることもよくある。また、もう少し抽象度の高いレベルで記述ができる「フレームワーク」も多数提供され、使われている。ここで使う組合せが決して唯一ではないこと、条件によっては他の方法の方がよいこと、を心に留めておいて欲しい。

《Apache Web サーバー》

Apache は Web サーバプログラムであり、静的ページと動的ページの両方をサポートしている。動的ページの実現には CGI (Common Gateway Interface) を提供している。

本プロジェクトでは、apache の Web サーバーが既に動作しているサーバーホスト上で、ページを作る作業に終始する。apache Web サーバー自体のインストールや設定は自分では行わない。もちろん、興味のある学生は自分の PC へインストールしてみるのもよいだろうし、その環境が自分で用意できないが是非やってみたいという学生には、(少数ならば)インストールできる PC を研究室内で一時的に提供することも考えるので、相談して欲しい。

《PHP プログラミング言語》

動的なページは、サーバーホスト上で動作するプログラムによって生成する。そのプログラムを記述する言語として PHP (正式名称は PHP: Hypertext PreProcessor) という言語を用いる。動的ページのためのプログラミング環境は多数あり、たとえば Java、Perl、Ruby などがよく使われる。このプロジェクトでは、比較的簡単に学習が容易、また同一ページにホームページ記述 (HTML) とプログラム (PHP) が同居可能、かつ、世の中ではかなり使われている(いた)、という理由から PHP 言語を選んでいる。

PHP の大きな特徴として、ホームページ記述 (HTML) と同じ場所 (ファイル) に PHP のプログラムを置くことができ、実行できることが挙げられる。つまり、静的なホームページを作るのと同じ環境で簡単に作れる。他の言語を使う場合は、ホームページ記述とプログラムを別の場所 (ディレクトリ) のファイルに用意する必要があり、そのメンテナンスはそれなりに面倒になる。このように混在できることは非常に便利なのだが、他方でそれによる欠点もある。たとえば、セキュリティ設定が甘いとプログラムを読まれてしまう。なので、簡便さとの兼ね合いで、賛否は分かれる。

PHP の書き方の規則やその意味などは、最低限必要な要素は紹介するが、それ以上の詳細は PHP のホームページ <http://www.php.net> を参照のこと。本プロジェクトでは PHP を使ってどんどんプログラムを書いて欲しいが、プログラムの書き方については、第4章で学習する。

PHP はインタプリタ (逐次解釈) 型のプログラミング言語である。PHP 単体でプログラムを作ることも可能であるが、本プロジェクトでは Apache Web サーバーの後方に居て、ユーザからの入力データを受け取って起動され、処理結果をホームページに書き出す形の使い方をする。PHP で書いたプログラムと Apache Web サーバーとの間のやりとりのインターフェースは、CGI に相当するものを使う。このやりとりは、PHP プログラムから見ると、端末から入力されたデータはいくつかの特別な変数によって伝えられる。また端末で「送信」(submit) ボタンをクリックすると、指定した PHP プログラムが起動される。出力側は、普通に文字列を print すると端末のブラウザにそのまま送られて、HTML として解釈される。これらは、後から実例を持って試してみる。

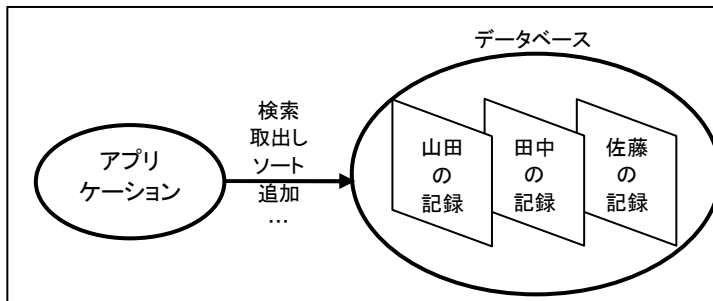
PHP の処理環境、つまり PHP の実行エンジンや Apache Web サーバーとの間のインターフェースモジュール (mod_php) は、本プロジェクトでは予め設定されて提供される。これらのインストールや設定は自分では行わない。興味のある学生は、難しくはないので、Apache Web サーバー同様に、自分の環境へインストールしてみると、よい経験になる。

《MySQL SQL データベース管理システム》

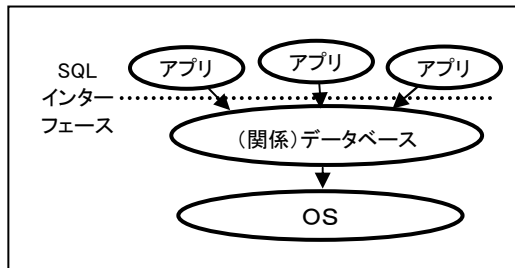
最後に紹介するのが SQL 対応のデータベース管理システム MySQL である。

Apache Web サーバーと PHP だけで動的 Web ページを作成できるが、本プロジェクトでは、ひとつの定番である、データベースと組合せた利用についても経験して置く。

データベースは、同じ形のデータをたくさん保持し、効率よく検索したり並べ替えたりできる、データ管理のソフトウェアである。たとえば学生に関する情報(氏名・学籍番号・生年月日・現住所…)を1人の学生の記録(レコード)とし、東邦大学全体の学生の記録を集めたデータベースを考えると、図のようになる。



オペレーティングシステムとユーザアプリケーションの間に位置するこのようなソフトウェアシステムを、ミドルウェアと呼ぶこともある。つまり、いろいろなアプリケーションがデータベース管理システムを使いに来るというモデルである。



データベースは、授業ではもっぱら理論的な側面を学ぶのだが、黒板だけではおよそ実感が湧かないと思う。データベースを体感する(Apache や PHP を抜きにしても)意味でも、プロジェクトの要素にデータベースを入れておくことにした。

データベースの理屈やデータ操作の詳細は授業で学んでもらうとして、ここでは広く使われている「関係データベース」(リレーショナルデータベース)の実際の使い方について学んでおく。ユーザのプログラムが、関係データベースを使う時の操作言語として、SQLが使われている。SQLの仕様自体は「標準」(ISO, JIS)として定められており、どの環境でも同じものが使える。MySQLはこのような関係データベースを提

供する管理ソフトウェアの1つで、アプリケーションに対して SQL 言語を提供する。管理ソフトウェアは MySQL 以外にもいくつも製品化されており、たとえば PostgreSQL、SQLite、Oracle、IBM DB2、Maria DB などの名前を聞くことがあるだろう。MySQLの詳細はホームページ <http://www.mysql.com> を参照のこと。なおホームページは有料版(MySQL Enterprise)と無料版(MySQL Community Server)があり、わかりにくいですが、学生が自分の環境で実験するには無料版で十分であろう。

MySQLはサーバー内に既にインストールされているので、学生は本プロジェクト内で、SQLの操作コマンド(インターフェース)を使った(データベースの)簡単な操作を試して、データベースとはどういうものかを実感して欲しい。

では、そのような関係データベースが利用できるとして、なぜ Apache Web サーバー+PHP CGI という「Web アプリケーション」と呼ばれるような枠組みの中に出てくるのか? Web アプリケーションの大半が、サーバーにデータを保存し、検索し、表示するような目的を持っているからである。たとえば、皆さんがいつも使う成績のシステムを考えると、成績のデータをサーバーに保存しておき、学生は学期の初めに授業登録するときに変更し、教員は学期の終りに成績の値を追記する。成績データをデータベースに保存し、必要に応じて検索したり情報を抽出したり並べ替えたりして、表示する。また、別の例として掲示板がある。掲示板に今までに書き込まれた内容をデータベースに保存し、ユーザが更に追加するときにはデータベースに追記し、表示するときには検索したり抽出したり並べ替えたりして表示する。これもデータベースがあれば簡単にできる。ショッピングも然り、アンケートも然りであろう。検索・抽出・並べ替えなどを自分でプログラムするのは面倒だし、安定して安全にデータを保存できる仕組みを作るのも経験が要る。それを、データベースシステムはまとめて提供してくれているので、便利に使おうということになる。

Web アプリケーションからデータベースを利用するためには、

- (1) データ操作のための SQL 言語、
- (2) PHP と MySQL との間のやり取りの仕方(インターフェース)、具体的には、PHP プログラムからデータベースへ SQL 言語のコマンドを出す方法と、データベースからの結果を取り出す方法、

の2点を知っている必要がある。幸いなことにいずれもそれほどややこしくない。後から具体的に紹介する。

[余談 Apache Web サーバー+PHP 以外の、動的 Web サーバーを作る枠組み]

《CGI を使う方法》

CGI のインターフェースを使うのだが、①Web サーバー(ネットワークとのやり取りをする側)に使うソフト、②アプリケーション(CGI インターフェースの裏側で動くプログラム)を作るのに使うプログラミング言語、の2つの要素が変更可能である。

①の Web サーバー側は、Apache Web サーバーや、Microsoft 社の IIS (Internet Information Server) が有名であるが、それ以外のソフトもある。環境や必要性によって選択することになる。

Web アプリケーションのプログラム言語は、ここで挙げた PHP の他、Perl、Java、Ruby などが多く使われている。本質は、CGI の約束さえ守れば何の言語で記述してもよい。たとえば C 言語でも作ることが出来る。

《専用の Web サーバー(コンテナ)を使う方法》

裏側でアプリケーションを走らせることを最初から念頭に置いた構造のサーバーが作られ、使われている。たとえば、Java のアプリケーションを想定したサーバーとして TomCat が代表であろう。TomCat は Apache と同じ組織 (Apache プロジェクト) 内で作られている。TomCat は Web アプリケーションを記述した Java プログラム(クラスになる)を、サーバー内に取り込んで実行する形になる。「コンテナ」などと呼ばれることがある。

コンテナにするメリットの1つに、実行効率がある。CGI 方式の場合は裏のプログラムのためのプロセスを「作って立ち上げる」のが標準的なスタイルで、立ち上げの時間がかかったり、メモリが余分に必要になったりする。立ち上げ時間を短縮するために予めプロセスをプールしておいて与える方式も使われているが、プールの数が固定されるなど完全に解決しているわけではない。コンテナ方式の場合はサーバーとして動いているプロセスがそのまま変身するので、プロセス立ち上げなどのOS操作に伴うオーバーヘッドは小さくなる。この方式については自分で勉強して欲しい。

まとめの設問

1. 「静的」なホームページと「動的」なホームページの違いが分かるように、両者を説明して下さい。

2. 「動的」なホームページを作るために本プロジェクトで用いる、3つのソフトウェア要素を挙げています。それぞれの役割を説明して下さい。

①Apache Web サーバー

②PHP 言語

③MySQL データベース管理ソフト

第2章 ホームページとHTML(タグ)

本章ではホームページのコンテンツを記述する言語 HTML（その記述要素である「タグ」）について、概要を簡単に説明する。この説明は、今まで自分で「タグ」を用いてページコンテンツを書く機会が無かった人を対象にする。既に使ったことのある人は、書かれている要素を使ったことがあれば、読み飛ばしてよい。初めての学生は、説明を読むだけでなく、自分でタグを使ってみて、慣れて欲しい。

HTML

通常の(静的な)ホームページは、Web サーバーから手元の端末に、ページ画面の情報が転送され、それを端末上で動いているブラウザ(Microsoft Internet Explorerとか Netscape Navigator とか Mozilla Firefox とか Opera ブラウザ)が解釈して、画面に

表示している。これは PC でもスマホでも同様である。

サーバーから端末に送られてくるページ画面の情報は、たとえば下記のような形のデータになっている。

このような情報の書き方(の規則、言語)を、HTML(HyperText Markup Language)と呼んでいる。



三角カッコ(angle bracket)に囲まれている<html>とか<head>などのことをタグと呼ぶ。別の授業や実験などでタグを見ているかも知れない。右記の例で

```

<html>
<head>
  <title>私のページ</title>
  <meta http-equiv=Content-Type
    content="text/html; charset=shift_jis">
</head>
<body>
  <p>あーたらどーたら</p>
</body>
</html>

```

CGIプロジェクト2

使われているタグは、次のような意味を持っている。

開始タグ	終了タグ	意味	ブラウザの解釈
<html>	</html>	これは HTML 文書	ページを始める
<head>	</head>	これはヘッダー部分	タイトルはブラウザの枠に表示
<title>	</title>	この部分がタイトル	同上
<meta ...>		メタ情報	メタ情報はブラウザが記憶。表示はしない
<body>	</body>	この部分が本文 (body)	タグに指定されたように表示
<p>	</p>	この部分がパラグラフ	パラグラフとして表示。上下に1行ずつ空白行。あとは文字列を流し込み

この他にもいろいろな種類のタグが定義され、使われている。たとえば、箇条書き(リスト形式)を作る一連のタグ (ul, ol, dl など、その中に項目を書く li タグ) や、表(テーブル)を作る一連のタグ、文字の大きさや色を指定する一連のタグ(文字の大きさ、色、フォントなど)、文字の修飾をする一連のタグ(太字、強調、上付き、下付き、下線など)などは、すぐにでも使うかもしれない。また、上の表にある meta タグは、タグの中で文書のメタ情報を指定する様々なパラメータを書くことができる。

このテキストはタグの参考書ではないので、細かい点は省略する。参考書が多数出版されているし、ホームページを探してみると解説が多数あるので、参照のこと。

なお、細かく言うと、HTML にもバージョンがあり 1998 年 4 月の HTML4.01 を基にしたものが JIS/ISO で標準化されているが、2014 年に勧告された HTML5 も広く受け入れられている。但し、(1)ブラウザの種類によって機能のサポートに差がある、(2)ホームページデータも厳密に最新の規約に従っていないものが散見される、など、細かい点ではあいまいな部分もある。最近はバージョン番号をコンテンツに明示することが勧められている。プロジェクトで紹介する HTML はごく一部で、バージョンやブラウザに依らず共通に使える部分である。

CGIプロジェクト2

例題演習 簡単なホームページデータを作って、アップしてみよう

まずは、いろいろなタグを試してみても慣れることにしよう。試すには、

- (1) 簡単なホームページデータをファイル(テキスト)として作り、
 - (2) ホームページとしてサーバーにアップ(転送・保存)する、
- の2段階を行う必要がある。

```
<html>
<head>
  <title>私のページ</title>
</head>
<body>
  <p>これは私のページです。</p>
</body>
</html>
```

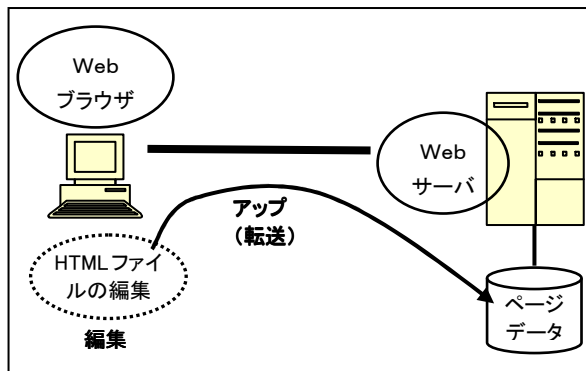
《アップのやり方》

テキストをタイプインする前に、(2)の点について先に説明しておこう。

今作ろうとしているのは、静的なWebページである。右図にあるような、HTMLで書かれたファイルを用意し、Webサーバーのディスク上に置くと、端末からの閲覧要求に応じてWebサーバーソフト(Apache Webサーバー)がこの内容を端末側に転送し、端末側でブラウザが表示する。

このHTMLデータを編集するのは、ユーザの手元の端末PCでも、Webサーバーマシン上でもよい。

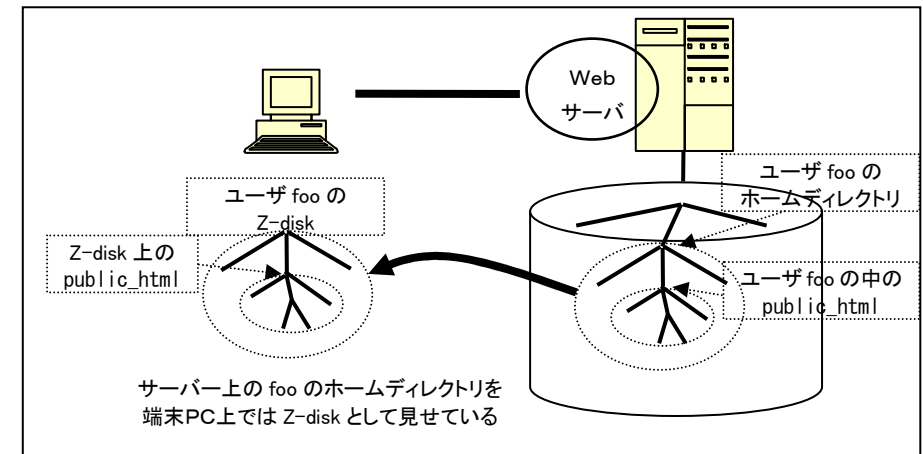
手元の端末PCで編集した場合は、作ったコンテンツをWebサーバーへ転送(コピー)する必要がある。コンテンツを手元の端末PCからサーバーのディスクに移すことを、「アップする」と呼ぶ。



《実験室でのファイル共有環境》

プロジェクトで使う情報科学科実習室のシステム環境は、通常的环境と少し異なる。実習室で使う手元のPCは、実験室サーバー(同時にWebサーバーでもある)のディスクを共有している(正確には、サーバーのディスクを手元のPCが《遠隔参照》~リモートマウント~している)。

具体的には端末PC上のZ:ディスク(Zドライブ)として見えているものが、実体はサーバー上のディスク(正確にはそれぞれのユーザのホームディレクトリ)になっている。であるから、手元の端末PC上でZ:ディスクにファイルを置けば、そのファイルはサーバー上のファイルになっている。



ルはサーバー上のファイルになっている。

《特別なフォルダ「public_html」》

更に、各ユーザが自由にホームページを持てるように、Webサーバーは各ユーザのホームディレクトリの中の特別な名前「public_html」のフォルダ(ディレクトリ)を、各ユーザの持つホームページとして公開するという設定になっている。

したがって、Z:ディスク上の直下にフォルダ(ディレクトリ) public.htmlを作成し、そのフォルダ内にホームページとして公開したいHTMLファイルを置けば、Webサーバーを介してそのファイルが公開される。

なお、このホームページは外部(端末PCを含む)からURL

http://venus.is.sci.toho-u.ac.jp/~*****/ファイル名

として見えている。

但し、半角 ~ は波線（正式名称ティルダ tilda）であり、その後の*****の所にメールのユーザ名（本当はこのシステムのユーザID）8桁（55??????、最後の英字を含む）を入れたものである。波線 ~ の入力は、キーボードではシフトキーを押しながら上線（キーボード最上段のひらがな「へ」の上）でできるはずである。（場合によってはシフトを押しながら ~ キーのこともある）

まとめると、手元の端末PCの Windows 上で、

- ① Z:ディスクを選び、
 - ②その中に public_html という名前のフォルダを作り（もし既にあればそれを使うので、新たに作る必要はない）、
 - ③その下にホームページとして公開したいファイル（たとえば index.html）を置く。
- そうすると、上記の URL として公開される。

《HTML ファイル作成・ブラウザからアクセスの実習》

まず、上に掲げた HTML テキストを、Windows のテキストファイルとして入力・作成せよ。たとえば「秀丸」や、メモ帳 (notepad) などのテキストエディタで作ればよい。ファイル名は、Z:ディスク上の public_html の下で、index.html という名前を付けておこう（ Z:%public_html%index.html ）。

この時、ネットワーク越しにブラウザからアクセスする時に使う URL は

```
http://venus.is.sci.toho-u.ac.jp/~*****/index.html
```

となる。但し、*****のところはユーザ ID（メールのユーザ名、学番＋英字）55??????を入れる。

もし、既に public_html フォルダを作っていて、そこに index.html などのファイルがある学生は、このプロジェクト用のフォルダ myproject を、public_html の中に作成してもよい（ Z:%public_html%myproject%index.html ）。この時は URL は

```
http://venus.is.sci.toho-u.ac.jp/~*****/myproject/index.html
```

となる。上と同様に、*****のところはユーザ ID 55??????である。

《脱線》

ところで、この index.html というファイル名はちょっと特別なものである。普通は URL でページを指定するときは

```
http://venus.is.sci.toho-u.ac.jp/~5599999z/index.html
```

のように、ファイル名 index.html まで記述しなければならない。だが、よくファイル名を書かず、ディレクトリ名（フォルダ名）までで終りになっている URL を見かける。たとえば

```
http://www.toho-u.ac.jp/
```

のように、ファイル名を書かないとき、Apache Web サーバーの指定で、自動的に index.html というファイル名が使われるようにできる。実験室のサーバーもこの設定がしてあるので、実は

```
http://venus.is.sci.toho-u.a.c.jp/~55*****/ (55*****はユーザID)
まで書くだけで、その下の index.html ファイルが表示される。
```

《注意1 ファイル名は英数字のみに》

venus サーバー (Windows での名称は isvenus) は UNIX（正確には Linux）を使っている。OS によってどの漢字コード体系を使うかが違うので、注意を要する。Windows 系は Shift-JIS が標準的に使われているし、UNIX 系は原則的に EUC か UTF-8 を使っていることが多い（最近では UTF-8 が増えてきた）。上で説明したようにディスク空間を共有していると、たとえば Windows で書いた漢字を含むファイルは、Unix では漢字コードが異なるために文字化けして読めない。

特に、URL の一部となる

```
ファイルの名前については、絶対に漢字(かなを含む2バイトコード)を使わないこと
```

さもないと、相手側の環境ではうまくファイル名が指定できないということになる。

《注意2 ファイル一覧の表示の設定を「全部表示」にする》

Windows では、ファイル一覧を表示するときに、ファイル名の拡張子（ファイル名の最後につく、ピリオドから後の文字列、たとえばワープロファイルなら「.docx」、エ

クセルファイルなら「.x|sx」など)のうち、一部または全部について「表示しない」というモードに、初期値が設定されている。たとえば.txt 拡張子を表示しないと、ユーザがindex.html というファイルを作ったつもりでも、

index.html.txt

となっていたりすることがある。(これは、テキストエディタが保存する時に自動的に.txt を追加した名前を作ってしまうことがあるためである。)

これは混乱の元になるので、原則は「表示する」モードにしておくこと。

「全部表示」モードにするには、ファイルの一覧表示で、上のほうにある「ツール(T)」から「フォルダオプション(O)」を選び、「表示」タブを開いて、次の2つの設定を変更する。

(1) 「ファイルとフォルダの表示」で、「すべてのファイルとフォルダを表示する」を選ぶ。

(2) 「登録されている拡張子は表示しない」のチェックマークを外す。

ここでOKを押せば、このフォルダだけでこの設定がされるが、後のためには、まず「適用(A)」をクリックした後、上から1/3ぐらいにある「すべてのフォルダに適用(L)」をクリックする。

《注意3 ネットで公開⇒世界中から見えている》

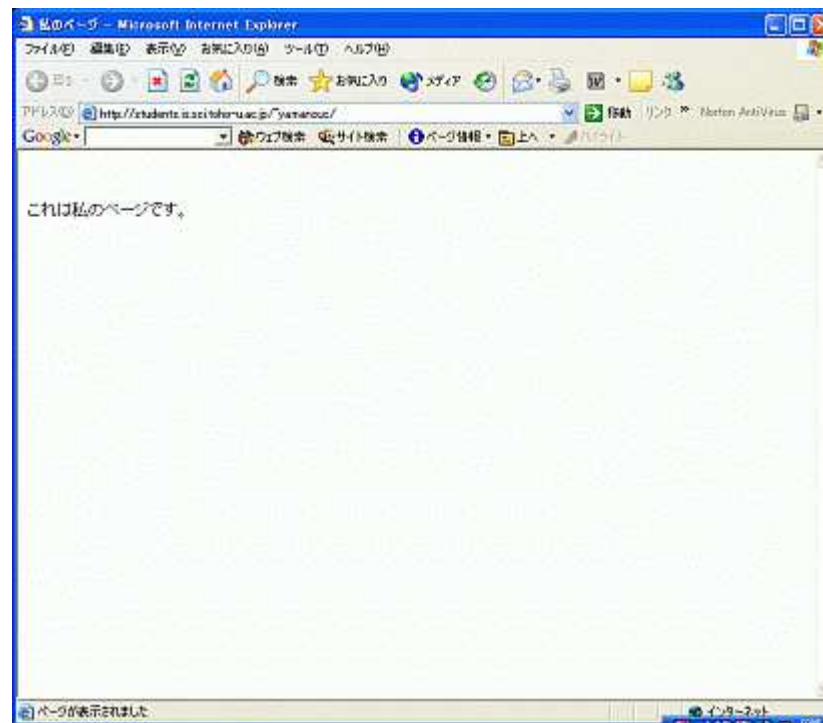
public.html フォルダの内容は、外部(学外つまり全世界)からホームページとして見えている。もし個人情報を書けば、それが世界中に公開されるし、他人の誹謗中傷などを書けば、世界中にそれを発信したことになる。従って、ファイルの内容に注意すること。具体的には、

- ① 住所・電話など公開したくない個人情報が含まれないように気をつけること
- ② 公開された内容について責任を問われるので、いい加減な情報や虚偽の情報、誹謗中傷などを掲載しないこと
- ③ 大学名を持つサイトであるから、大学を傷つけるような内容を含まないこと

また、この《public.html フォルダ》の中に、作りかけのファイル等を不用意に置かないこと。そのままの形で公開されてしまっている。

《チェック ~ ブラウザから見えたかな》

コンテンツのファイルを所定の(public.html)フォルダ内に置いたら、それでホームページは世界中から見えるはずである。自分の作ったファイルのページが、手元の端末PCのブラウザで見えただろうか？ ページの中には「これは私のページです。」という文字が入り、枠の部分に「私のページ」というタイトル(<title>タグで指定した内容)が入っているはずである。



《見えない場合のよくあるミス ~ ファイル名(特に拡張子)がおかしい》

作成したファイル名(特に拡張子の部分)がおかしいことがある。《注意2 ファイル一覧の表示の設定を「全部表示」にする》で触れたように、拡張子を表示しないモードになっていると、拡張子が間違っていることに気づかない。自分でindex.html というファイルを作ったつもりで、実はindex.html.txt というファイルを

作ってしまっていることがある。

確認は、「ファイル一覧」で、拡張子を表示するモードにして、ファイル名を確認する。または、ファイルのリストを表示している中で、そのファイルを右クリックし、最下段の「プロパティ」をクリックすると、そのファイルの属性情報が表示されるので、その中に表示されるファイル名が変でないか、確認することも出来る。

《注意4 html ファイルを Web サーバー経由で見えていますか?》

実は、Windows のファイル一覧の中で、index.html ファイルをダブルクリックすると、その内容がホームページのように(ブラウザで)表示される。これは、我々が今意図している見え方(Web サーバーから端末へ送られて見える)ではないので、注意して欲しい。

これは、手元 PC 内にある index.html ファイルの内容を、手元 PC にあるブラウザで直接見ているだけであって、Web サーバーによる転送を経っていないのである。

<同じに見えるのだからいいじゃないか>と思うかも知れない。この節で扱っている静的なページは、どちらで見ても同じになるので、今はいいのだが、次の節で PHP による動的ページを表示したときに、PHP のプログラムが動いていないように見える。PHP を実行しないと結果が出ないような動的なページでは、結果が全くおかしくなる。PHP プログラムを実行させるには、Web サーバーを経由して表示しなければならない。後のことを考えて、面倒でも、ちゃんとブラウザの URL 欄に venus の名前を入れてアクセスすること。

例題演習 少し複雑なタグの演習

これ以降は、いろいろな HTML タグを試してみる演習である。この段階であまり凝らずに、ざっと流すことを勧める。プロジェクトの後半で、自分たちの独自サイト、独自画面を作りたい時に、タグを書くことのできることをいろいろと研究してみたらよいだろう。

<body> ... </body>の中身については、いろいろ変えると画面表示が変わる。

(1) ヘッダー

「ヘッダー」は、章・節・パラグラフのタイトルであり、レベルが1から6までである。最も大きい単位はレベル1で、

```
<h1>この章のタイトル</h1>
```

のように書く。<h1>が開始タグ、</h1>が終了タグである。

```
<h1>章タイトル</h1>
```

```
<h2>節タイトル</h2>
```

```
<h3>パラグラフタイトル</h3>
```

```
<p>パラグラフの本文を書く。
```

```
<h3>パラグラフタイトル2</h3>
```

```
<p>パラグラフの本文を書く
```

```
<h2>節タイトル</h2>
```

```
<h3>パラグラフタイトル</h3>
```

のように使える。なお、パラグラフ本文は<p>で始める。

(2) 箇条書き

箇条書きは、番号をつける Ordered List (左側)と、番号をつけない Unordered List (右側)がある。番号付きの Ordered List はで囲み、番号無し Unordered List は で囲む。内容項目は共通で、1つ1つをで始める。

このほか、<dl>も使われる(後で調べてみよ)。

<p>家で飼う動物には3種類ある</p> <ol style="list-style-type: none"> 1. 犬 2. 猫 3. うさぎ <pre> 犬 猫 うさぎ </pre>	<p>家で飼う動物には3種類ある</p> <ul style="list-style-type: none"> ・ 犬 ・ 猫 ・ うさぎ <pre> 犬 猫 うさぎ </pre>
---	--

(3) 絵(イメージ)

絵(イメージ)をホームページに載せるには、イメージファイルをサーバーに置いておき、

``のように、``タグを使う。src=の後にイメージファイル(画像ファイル)の URL を指定する。

この例では同じフォルダに画像ファイルを置くこととすれば、自分の ID のところには ~5510123z のようなユーザIDが入る。まったく別のサイトの画像でも URL さえ書ければここに参照できる。

HTML 文書と同じサーバー内にイメージファイルを置く場合は、URL の代わりにファイル名だけを書くことも出来る。特に HTML 文書と同じディレクトリにおく場合には、ファイル名で書くと簡単になる(相対パス指定)。

``
この例では、イメージファイルのファイル名は henoheno.PNG である。その他に絵の表示の寸法、つまり width(幅)が 212ピクセル(画素)、height(高さ)が 200ピクセルであること、border(周りの線)の太さが 0つまり枠を描かないこと、最後に align(位置合わせ)が right(右そろえ=画面の右側のふちに合わせる)であることを指定している。指定しているそれぞれの「値」の部分(212とか 200とか 0とか right)を二重引用

符「”」で囲んでいることに注意。

なお、イメージファイル本体は予め適当な形式(GIF、JPEG、PNG、TIFF など)で作成して、サーバーに載せておかなければならない。

《イメージを作って表示してみたい人は》 (時間がなければスキップせよ)
img タグを簡単に試してみる目的で、手書きの絵をイメージファイルとして作成して、ホームページに表示してみるのも面白いかもしれない。

具体的には、

1) Windows のアクセサリの中にあるペイントプログラム

(「スタート」⇒「すべてのプログラム」⇒「アクセサリ」⇒「ペイント」)

を使って、手書きの絵を描く。描き終わった後で保存する時に GIF、JPEG、PNG、TIFF などの形式で保存する。

(「ファイル」⇒「名前を付けて保存」で、「ファイルの種類」欄で JPEG や PNG を選ぶ)。保存場所(フォルダ)は HTML ファイルと同じフォルダにする。

2) 前に作った HTML ファイルの中に



《絵が見えない場合のよくあるミス ファイル名(特に拡張子)の大文字小文字》
作った絵のファイルは、拡張子が大文字で付いていることがある。HTML ファイルでの `img src="..."` で書くファイル名は、大文字・小文字を画像ファイル名と同じにする必要がある。画像ファイル名の拡張子が henoheno.PNG のように大文字であれば、HTML での src のファイル名の指定も、src="henoheno.PNG" のように大文字でなければならない。小文字にすると、違うファイルだと思って、絵が表示されない。

(4) 表(テーブル)

表を書きたい時がある。罫線の入った表だけでなく、単に縦の位置をそろえたい時に罫線を入れない表を使うこともできる。

表は右のように書く。

1行の単位を<tr>...</tr>で囲み、その中に1項目(1カラム)ずつ<td>...</td>で囲んで書く。このサンプルは右端の図のようになる。

罫線(枠)をつけるには、<table border="1">のように指定する。"1"は線幅で、数字が大きくなると太くなる。左側のサンプルは枠なし、右図は枠幅「1」の例である。

また、項目の内容としてイメージを入れることも可能である。

```
<TABLE>
  <TBODY>
    <TR>
      <TD>1行目の1項目目</TD>
      <TD>1行目の2項目目</TD>
    </TR>
    <TR>
      <TD>2行目の1項目目</TD>
      <TD>2行目の2項目目</TD>
    </TR>
  </TBODY>
</TABLE>
```

[枠なしの場合]

1行目の1項目目	1行目の2項目目
2行目の1項目目	2行目の2項目目

[枠=1の場合]

1行目の1項目目	1行目の2項目目
2行目の1項目目	2行目の2項目目

《表を試してみる実習》 (時間がなければスキップせよ)

下記のような表を作ってみよ。

	東京都	大阪府
面積	2200平方 Km	1900平方 Km
人口	1億3千万人	8千9百万人

(5) ハイパーリンク

Web ページでは、文字列や絵をクリックすると別のページにジャンプできる「ハイパーリンク」を使うことができる。

これは

 標的文字列

のように書く。但し "…(URL)…" の部分は、ジャンプ先のページの URL を書く また、標的文字列(もしくはイメージ)は表示される文字列(もしくは絵)を書き、その部分(色が変わる)をクリックするとジャンプする。

<p>それではここをクリックしてください

このサンプルを表示すると、

では[ここ](#)をクリックしてください

「ここ」の部分をクリックすると、nextpage.html にジャンプする。実際にジャンプさせるには、飛び先のページも用意しておかなければならない。

《リンクを試してみる実習》 (時間がなければスキップせよ)

上記の、《表を試してみる実習》で作った表で、「東京都」をクリックすると東京都のホームページ、「大阪府」をクリックすると大阪府のホームページへ飛ぶようにせよ。

東京都のホームページは www.metro.tokyo.jp

大阪府のホームページは www.pref.osaka.jp

終了の確認

《例題演習 簡単なホームページデータを作って、アップしてみよう》 が正しくできたことを、TAに確認してもらって下さい。これによって、第2章を修了したことになります。

第3章 フォームによる入力

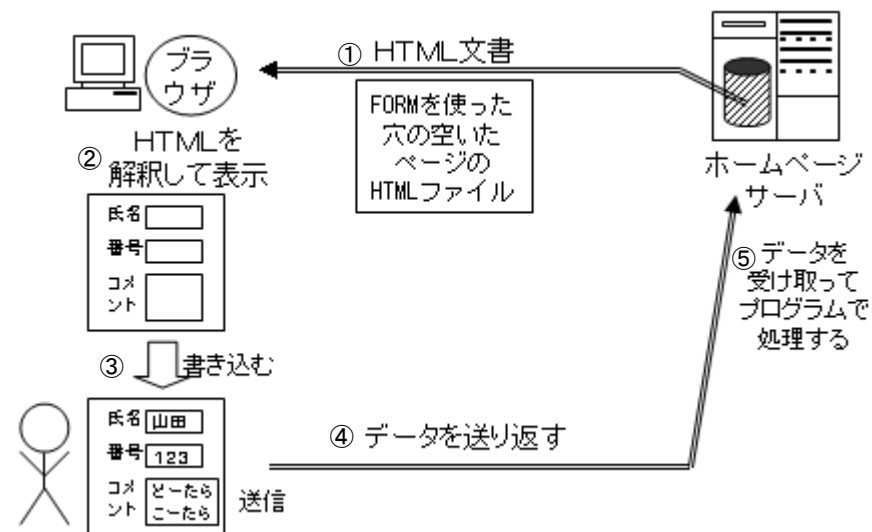
フォームとは = 入力できるホームページ

フォームとは、入力欄のあるホームページのことで、ユーザが端末で情報を入力し、それをサーバーが読み取るための仕掛けとして使われる。たとえば、ログインをする画面でユーザ名やパスワードを入力する、会員登録をする画面で会員情報を入力する、インターネットショッピングをする画面で欲しい商品にチェックしたり、支払いのためのクレジットカード番号を入力したりする、といった形でよく見かける。また、ネット検索の画面も、検索したいキーワードを入力するのに、この形で実現される。また、Web によるテスト(WBT=Web-Based Testing)、つまりサーバーが問題を出してユーザが解答し、サーバーが採点する、といった使い方もある。



フォームを使って入力をするためには、

- 1) 入力する場所(欄・枠)を用意した画面を、サーバーから端末に送り(図中①)、表示させる(②)。(入力の欄の表示は HTML のタグを使って行う。)
 - 2) その入力画面に対して、ユーザが端末から情報を入力し(③)、「送信ボタン」でサーバーに送り返す(④)。サーバーは受取った情報を入力として使い、あらかじめ決めた処理(プログラム)を行う(⑤)、
- という2ステップの動作が必要になる。



その仕組みを簡単に説明する。

《入力画面の表示》

まず、入力用の穴(枠)の開いた画面を表示させる。具体的には Web サーバーから、普通の静的なページと同じように HTML で書かれたページを送出し、端末はそれを受け取って解釈して表示する。

ただし、送られるページ画面データに、入力用の欄を表示するために、<FORM> タグと、その関連タグが、書かれている。端末は、この <FORM>タグとその関連タグを解釈して、

- 1) 入力枠(欄)を作って表示すると同時に、
 - 2) その欄に対する識別名(区別するための名前)を与える。
- 1)の入力欄には、単純なテキストを書くための欄だけでなく、選択肢を選べる欄(チェックボックス)や、(プルダウン)メニュー型の欄などを作るタグが、用意されている。後で具体的な例を通じてもう少し詳しく見ることにする。
- 2)の識別名は、入力をサーバーに伝える時に、たくさんの入力欄があるとそれぞれ

を区別する必要があるので、区別するために名前(識別名)を与える。たとえば入力画面にユーザIDとパスワードの2つの欄があるとき、サーバー側へ渡された入力データの上で、それらを区別する必要があることは、容易に想像できるだろう。

氏名

自由にコメントを書き込んでください

パスワード

好きな果物 りんご オレンジ バナナ

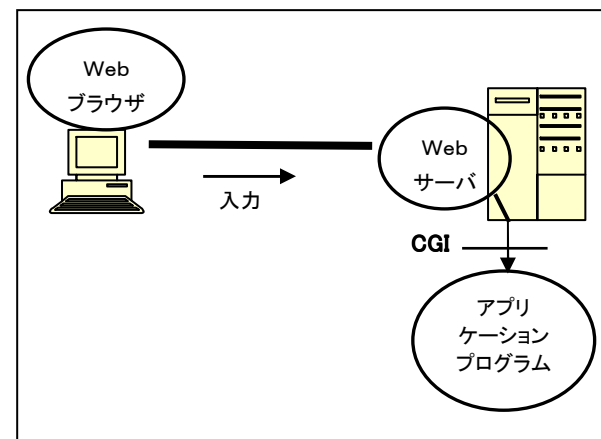
動物がすき? はい いいえ どちらでもない

また、入力タグの一種として、「送信ボタン」タイプを指定することができる。この場合は画面上に送信ボタンが作られる。ユーザは、端末画面上に表示された入力欄にデータを入力し、最後に「送信ボタン」をクリックする。このクリックした時点で初めて、入力データはサーバーに送信される。それ以前では、端末上で入力データを扱っているだけであり、サーバーは関知していない。

「送信ボタン」の形や文字はある程度自由に作ることができる。たとえば「送信」とか「了解」とか「終了」といったラベルを貼ることができる。

複数の入力欄がある場合にそれらを区別しなければならないので、各々の入力欄につけた名前(識別名)とキーインされた入力データをペア(対)にして、サーバーに送られる。

サーバーでは、端末で送信ボタンを押したことが伝えられ、そのボタンに対応するアプリケーションプログラムを起動させる。更に、端末から送られてきた入力データを読み取ってアプリケーションプログラムに渡す。



サーバー側でこのデータを処理するプログラム(図中で「アプリケーションプログラム」と書いたもの)を準備する人は、送信ボタンによって起動されて、CGI を介して入力を受け取って必要な処理をするプログラムを作ることになる。

本プロジェクトでは、プログラムを記述するために PHP と呼ばれる言語を用いる。要するに端末から送られてきた入力データが、決められた形式でプログラムの入力に渡されるので、それを解釈して必要な処理ができるプログラムを用意すればよい。

《脱線》

CGI を使うアプリケーションプログラムは、もちろんカスタムメイドで作ることが出来るが、一方で既に出てきているプログラムが提供されることもある。ショッピングやユーザ登録の仕組み、たいていカスタムメイドで作られているだろう。また典型的な(つまり、そのままの形で導入したいユーザが多い)CGI アプリケーションプログラムは、(有料の)製品として提供されているものもあるし、またフリーソフトとしてネットに公開されているものもある。典型的な例としては、掲示板システム(BBS=Bulletin Board System)やそれを日記として扱うブログ(WebLog)、メールをWeb上で読んだり書いたりするWebメール(キャンパスで提供されているActive Mail)などのソフトが出回っている。たとえばVectorのサイト <http://www.vector.co.jp> や窓の杜 <http://www.forest.impress.co.jp> などを覗くとこの種のソフトを見つけることが出来る。

このプロジェクトの前半で扱うのは、この種のアプリケーションプログラムの内部のうちでも、画面とやり取りする部分に限定しよう。具体的には、入力画面の出し方と、アプリケーションでの入力データの受け取り方に限定する。入力データを使って何か役に立つことをする部分(たとえばショッピングの機能だったり検索の機能だったり)は、ここでは追求しない。プロジェクトの後半で自分たちのサービスを作る時には、入力の端末・サーバー間でのやり取りだけではなくて、受取った入力を処理する部分や、ユーザに画面をどう見せるかの画面デザインも、一緒に考える必要がある。

《ここまでの内容を整理する》

1. 入力の仕組み

- 1) <FORM>を使った(入力欄を含む)ページを準備して、それを普通のページと同様に表示する、
- 2) 端末で入力欄にキーインした情報は 送信ボタンを押すと、まとめてサーバーに送られる、
- 3) サーバーでは送られてきた入力データを CGI インターフェースを経由してアプリケーションプログラムに渡す、アプリケーションプログラムは、所定の手続き(後で説明する)で入力の情報を受け取る
といった手順で行われる。

2. 入力データは、サーバー上に用意した CGI アプリケーションプログラムで受け取ることができるので、それを使って、たとえばデータベースへのユーザの登録なり、クレジットカードからの引き落としなりという、アプリケーション処理をすることが出来る。

CGI アプリケーションプログラムは Perl, Ruby, Java, PHP やそのほかの言語で書くことができ、端末からの入力 データを受け取って、それを処理する。

プロジェクト前半でやることは、このような CGI アプリケーションプログラムの簡単なものを書いてみて、プロジェクト後半で本格的な仕組みを作るための前提知識を得ることである。

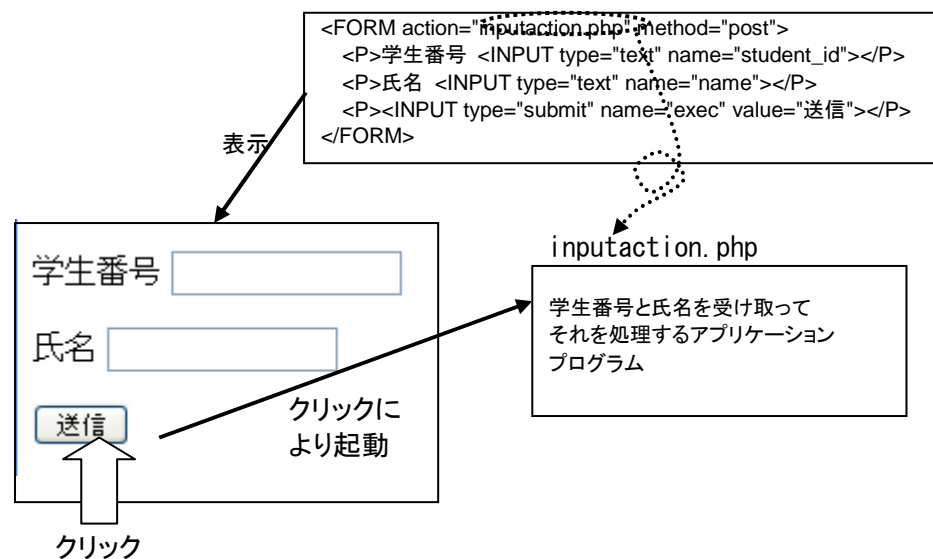
フォームの書き方

まずはサンプルから見てみることにしよう。

```
<FORM action="inputaction.php" method="post">
  <P>学生番号 <INPUT type="text" name="student_id"></P>
  <P>氏名 <INPUT type="text" name="name"></P>
  <P><INPUT type="submit" name="exec" value="送信"></P>
</FORM>
```

これを、ページの一部として作る。つまり、この前に<HTML><BODY>があって、この後に</BODY></HTML>を置く(<HEAD>…</HEAD>部分は必要なものを自分で書くこと)。

この中で、<P>タグ (パラグラフ タグ、パラグラフの先頭を示す) は既に知っているとする。他の<FORM>タグ、<INPUT>タグは初めてなので、説明を要するだろう。



《FORM タグ》

<FORM>タグは、ここからフォーム(記入用紙)が始まるという意味で、</FORM>で閉じられる。

<FORM>タグの中に、引数として、action=とmethod=が書かれている。actionはこのフォームを端末からサーバーへ送信した時(ユーザが送信ボタンをクリックした時)、サーバー側の後ろでフォームから入力データを受け取って動作するプログラム(アクションスクリプトと呼ぶことがある)の名前(そのプログラムが置かれているファイル名またはURL)を書く。ここではinputaction.phpという名前のファイルに書かれているプログラムが実行される、と指定している。

(注) action には本来は URL つまり http://foo.bar.toho-u.ac.jp/xyz/abc.php のような形を書くように規定されているが、相対パス表現も許されるので、同じディレクトリにあるプログラムファイルを指定する場合はそのファイル名だけを書いてよい。また相対パスを用いて、たとえば../cgi/abc.php のように「自分の1つ上(..)の下のcgiの下のabc.php」を指定することもある。規定に従えば、フルのURLを書いていいのだが、他のサーバーのプログラムを呼び出せるのだろうか? 試したことがない。

実行するプログラムのファイルは、大抵の場合は Perl、PHP、Java などのプログラムである。ここでは、拡張子が php なので、PHP のプログラムが実行されることになる。

(注) Web サーバーの設定で、拡張子 php のファイルに対しては PHP のインタプリタが起動されるようにしてある。

method 引数は、FORM のデータを端末からサーバーへ送る時の「送り方」の指定で、選択肢は POST と GET の2種類があるが、このプロジェクトの目的では POST しか使わない。GET を使う場合については、下記に公式な解説があるので参考にして欲しい。ここでは一切サボることにする。

http://www.w3.org/MarkUp/html-spec/html-spec_8.html#SEC8.2.2

《INPUT タグ》

次に<INPUT>タグが3回使われている。<INPUT>タグにはいろいろなタイプがあって、type パラメータで指定する。たとえば、最初の<INPUT>タグは、type="text"と指定しているが、これはテキスト型の入力欄を作る(表示する)ということである。type には

- text テキスト入力欄(1行のみ)を表示
- textarea 複数行にまたがるテキスト入力欄を表示
- password テキスト入力だが画面には入力文字を表示せず *などで隠すような入力欄を表示
- checkbox チェックボックスを表示(チェックボックスは1つだけ選択)
- radio ラジオボタンを表示(ラジオボタンはいくつ選択してもよい)
- (プッシュ)ボタン類 submit ボタン(送信ボタン)、reset ボタン(入力リセットボタン)、image ボタン(イメージボタン)など

などがある。この例では text という指定なので、1行のみのテキスト入力欄を作る。複数行が必要なら textarea にしなければならない。

次のパラメータ name="student_id" は、この入力欄を(他の入力欄と)区別するためにつける名前(識別名)である。CGI プログラムからアクセスする時は、複数の欄の入力が一度に入力として入ってくる。それを区別するために、student_id という識別名をつけたのである。この名前はプログラマの好きなものを選んで構わない(Java の変数名と同じように、最初の文字が英字とか、英数字以外の記号は一定のもの以外はダメとかの制限がある)。たとえば、次の行の <input>タグは、タイプは同じ text であるが、識別名は name="name" となっている。CGI プログラムから見ると、識別名 name として入ってくるデータが、画面上の「氏名」と書いたところの入力欄の値になる。

《送信ボタン》

最後に、同じ<INPUT>タグだが type="submit" としたタグが書かれている。これは「送信ボタン」を表示する。submit 型は、これを含む<FORM>…</FORM>に含まれるすべての入力データを、端末から送出するとともに、<FORM>タグの action パラメータで指定したファイルのプログラムを起動する。このタグのパラメータ name は、もし submit のタグが複数あれば(つまり2つ以上の送信ボタンがあれば)、どちらを押されたかを区別するために使うことが出来る(あまり必要ないかもしれないが)。次のパラメータ value は、画面上に表示されるボタンの上に貼り付けられる文字列を指定する。この場合は「送信」と書かれたボタンが表示される。

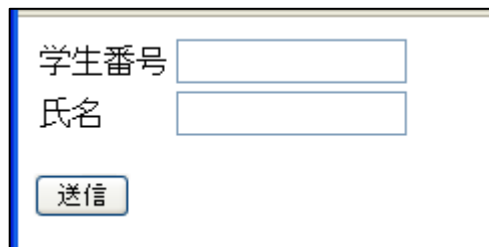
なお、他の INPUT タグ、たとえば TEXT などでは、value パラメタは、パラメタで指定した文字列が、枠内に初期値の形で表示される。そのまま送信ボタンを押せば初期値が送出されるし、端末上で値を書き換えれば書き換えた値が送出される。

いろいろなフォーム

これをまねすれば、いろいろなフォームを書くことが出来る。次の例を見て欲しい。

```
<FORM action="inputaction.php" method="post">
  <TABLE>
    <TR>
      <TD>学生番号</TD>
      <TD><input type="text" name="student_id"></TD>
    </TR>
    <TR>
      <TD>氏名</TD>
      <TD><input type="text" name="name"></TD>
    </TR>
  </TABLE>
  <P>
    <input type="submit" name="exec" value="送信">
  </P>
</FORM>
```

前の例と殆ど同じだが、フォームの中に<TABLE>タグを使っており、フォームの内容を「表」にしてある。その結果、1行目と2行目の入力欄の位置が揃っている。



The screenshot shows a web form with a table structure. The first row contains the label '学生番号' and a text input field. The second row contains the label '氏名' and another text input field. Below the table is a '送信' (Submit) button.

一般に、フォームの内容はどのような形でもよく、<P>を使ってパラグラフにしようと、やを使って箇条書きにしようと、この例のように表を使おうと、構わない。なお、参考資料として、HTML 言語の公式の定義をしている <http://www.w3.org/TR/html4/> がもっともオフィシャルなページなので参考にすると

よい。英語をいわずに読む努力をすること。FORM などを含めて難しめの内容に関する解説ページとして面白いと思っているものとして、

<http://www.cs.tut.fi/~jkorpela/forms/>

を挙げておく。

Google などではたとえば「HTML フォーム タグ」をキーワードに検索してみると、日本語のお手軽な解説ページがたくさん出てくるだろう。これらは上記の w3.org 公式ページと異なり、公式な正確さは下がるかも知れない。

例題演習： フォームのページを作る

上記の例を参考にして、簡単なフォームを含むホームページを作り、どのように見えるか試してみよう。試すのは、(a)テキスト、(b)テキストエリア、(c)パスワード、(d)チェックボックス、(e)ラジオボタン、及び (f)送信ボタン をやってみること。ファイルを作る場所は、たとえば **Z: ディスク中の public_html フォルダの下に** 適当なファイル名 (testform.html など) で作成 (保存) し、ブラウザで (URL)

<http://venus.is.sci.toho-u.ac.jp/~ユーザーID/testform.html>

をアクセスして試せ。

書き方は以下の例を参照。

```
<P>氏名<INPUT size="20" type="text" name="yourname" value="山田太郎" maxlength="30"></P>
<P><TEXTAREA rows="3" cols="20" name="freecomment">自由にコメントを書き込んでください</TEXTAREA></P>
<P>パスワード<INPUT size="20" type="password" name="yourpassword" maxlength="15"></P>
<P>好きな果物
<INPUT type="checkbox" name="fruit1" value="apple">りんご
<INPUT type="checkbox" name="fruit2" value="orange">オレンジ
<INPUT type="checkbox" name="fruit3" value="banana">バナナ
</P>
<P>動物がすき?
<INPUT type="radio" name="animal" value="yes">はい
<INPUT type="radio" name="animal" value="no">いいえ
<INPUT type="radio" name="animal" value="neither">どちらでもない
</P>
<P><INPUT type="submit" name="send" value="送信"></P>
```

1行目はテキストの例。size は欄の大きさ、name は入力欄を識別するための名前、value は初期値(どう現れるか確かめてみよ)、maxlength はここに書き込むことの出る最大の文字数で、欄の大きさ size より大きく出来る。

(注) 一般に、maxlength は指定しておくのがよい。理由は、受け取る CGI アプリケーションプログラムの側で、入力用に用意するバッファが足りなくてあふれてしまうことを防ぐためである。このバッファがあふれると、(場合によって)あふれた部分が悪さをして、CGI アプリケーションを壊したり、最悪時には悪いプログラムを意図的に書き込まれたりする可能性がある。これは攻撃に対する脆弱性(この場合はバッファオーバーフローの脆弱性)を提供する可能性があるということなので、実際に何かのトラブルが起こらない場合でも、なるべくあらかじめ避けておくのが望ましい。というわけで、実際に使う程度の文字数を maxlength として指定しておくのがよいだろう。

2行目はテキストエリアの例で、rows は欄の行数、cols は欄の長さを指定する。INPUT タグではないので注意。name は入力欄の識別。初期値は value ではなくて、この例のように書くので注意。

3行目はパスワードの例。テキストと同じ形だが、ユーザが書き込んだ時に文字を隠してくれる。試してみよ。

4~6行目はラジオボタンの例である。ラジオボタンはクリックして選択できるタイプの入力で、同時に複数のものを選ぶことができる(選択肢で複数選択可)。name は入力欄を識別する名前だが、4行目だけでなく5、6行目も同じ識別名を使っていることに注意。これによって、4・5・6行目は同じ選択群であることがわかる。value はそのボタンにチェックを入れたときに入力として戻ってくる値。この例では、ユーザがりんごにチェックを入れると、apple という値が戻ってくるようにしてある。もう1つ、「好きな果物」というテキストは INPUT のタグの外側に置かれている。つまり選択群の範囲外である。FORM の要素はこのように普通のテキストやイメージと混ざって使うことが出来る。

7~11行目はチェックボックスの例である。チェックボックスは、選択肢だが、ラジオボタンと違って唯一選択を強制される。つまりどれか1つをクリックすると、もし他のものにチェックをしてもそれが消されて、新たにチェックしたものだけが残る。8~10行目にチェックボックスを書くための INPUT タグがある。name は入力欄の識別で、

8~10行目に共通の名前にしているのは、ラジオボタンと同じである。value はこのボックスにチェックをしたときに入力として戻ってくる値で、yes か no か neither の3つの値のうちの1つが戻ってくるようにしてある。

12行目は送信ボタンの例で、引数は前に説明した。

この送信ボタンをクリックすると、このフォームのすべての入力の値が(まとめて)端末からサーバーに送られる。その結果、サーバーのプログラムから見ると、yourname には名前が、yourpassword にはパスワードが、fruits には好きな果物の選択値が、サーバーに送られるはずである。

サーバーに送られてくる値を入力として処理するためには、「CGI プログラム」を作らなければならないが、それは次の章で説明する。それが準備できていないので、とにかく HTML 上でフォームを書いてみて、どのように見えるかだけを試してみる。

⇒ つまり「送信」ボタンを押しても、(フォームページだけでは)何も起こらない。

The screenshot shows a web form with the following elements: a text input field for '氏名' (Name) containing '山田太郎'; a text area for '自由にコメントを書き込んでください'; a password input field for 'パスワード'; radio buttons for '好きな果物' (Favorite fruit) with options 'りんご', 'オレンジ', and 'バナナ'; checkboxes for '動物がすき?' (Do you like animals?) with options 'はい', 'いいえ', and 'どちらでもない'; and a '送信' (Submit) button.

終了の確認

直前で説明した《例題演習: フォームのページを作る》が正しくできたことを、TA に確認してもらって下さい。これによって、第3章を修了したとします。

《第1回目授業時間内で、最低限でここまでを終わらせてください》
時間に余裕のある人は、どんどん先へ進んで構いません。