1　Web システム

１．演習の目的

サーバを構築することで，ネットワークの基礎を身につけると同時に Web アプリケーション開発を体験し，Web システムの基本的な動作原理について学ぶ.

Linux のコマンドに関し復習し，SSH，vi エディタについて予習を行っておくこと.

２．サーバの概念

２．１　サーバについて

クライアント・サーバにおけるサーバとは一般的にクライアントからのリクエスト（要求）が来るのを待つソフトウェアを指す．一般的によく知られている例として， クライアントが Internet Explorer 等の Web ブラウザ，サーバが Apache 等の Web サーバである．クライアントからのリクエスト（Web ブラウザでは特定の URL）を入力することで，リクエストに応じたレスポンス（Web サーバでは特定の html ファイル等を返却）がある.

ただし，サーバと呼ばれるものにはコンピュータ（物理的ハードウェア）も含むため，コンピュータとソフトウェアを区別するためにソフトウェアをサーバソフトウェアやサーバプログラム，サーバサイドプログラム等と呼ばれる事もある．対し，サーバソフトウェアが導入されたコンピュータをサーバコンピュータやサーバマシンと呼ぶ.

２．２　サーバの IP アドレス，ポート番号について

ネットワークを用いたクライアント・サーバモデルでは必ず IP アドレスとポート番号が必ず必要となる．IP アドレスはコンピュータの OS（厳密には NIC）に与えられ，ポート番号はソフトウェアに接続するために準備する.

IP アドレスにはローカル IP とグローバル IP がある．グローバル IP とは世界で１つしか無い固有のアドレスを指し，ローカル IP は DHCP によって特定組織の中でだけで有効な IP アドレスである．本演習では学内ネットワーク管理者から提供される IP を与えた仮想マシンを１人１台提供する.

また，IP アドレスには対でサブネットマスクと呼ばれる数値が与えられる．サブネットマスクとは「どこのネットワークのどのコンピュータ」なのかを判別するための 32 ビットの数値である．例えば，本学の IP アドレスは 150.89.xxx.xxx がグローバルＩＰで有効であり，サブネットマスクは 255.255.0.0 となる．つまり，サブネットマスク 0（2 進数で）が

記述されている箇所 xxx.xxx 部分は大阪工業大学で利用することができる IP アドレスである．これが例えば 150.89.160.xxx 等となればサブネットマスクは 255.255.255.0 となる．また IP アドレスとサブネットマスクを表記する場合には／（スラッシュ）で表記される事があり，

150.89.xxx.xxx/255.255.0.0

もしくは

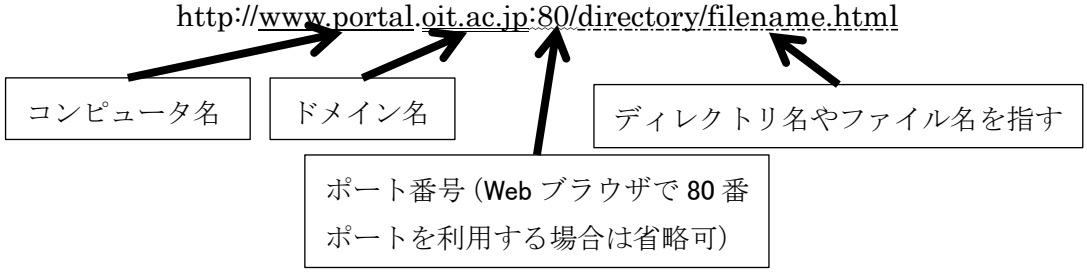
150.89.xxx.xxx/16

と表記される．/16 の意味は 2 進数でサブネットマスクを表記したものであり先頭から 16 桁が 1 を示す.

最後にポート番号は 1 つの IP アドレスに対し 0 番から 65535 番まで付与されており，ソフトウェアによって利用するポート番号が推奨されている．例えば，Web サーバ等は 80 番（http），443 番（https）ポートを利用しており，通称「http ポート」と呼ばれる事がある．その他にも ftp や smtp（メール）等のソフトウェアでも利用が推奨されているポート番号は存在するが，「推奨」されているだけであり，絶対では無い．つまり，サーバ運用者によって通信ポートは変更する事が可能である.

２．３　IP アドレスと DNS 名

一般的な利用で IP アドレスを利用することは無いが，先に述べた用に IP アドレスはネットワークにコンピュータを接続するためには必ず必要なものである．一般的に利用されない理由は，IP アドレスは DNS サーバによってコンピュータ名とドメイン名に変換するためである．例えば，学内のポータルシステム（https://www.portal.oit.ac.jp）の場合，



となり，コンピュータ名＋ドメイン名で一つの IP アドレスへ変換している．また，ドメイン名の後ろに：（コロン）＋数字を入力することで，ポート番号を指定することができる．なお，「https://」はスキームと呼ばれ，通信 の手法を示したものであり，他にも http，ftp，file 等がある.

２．４　IP アドレス，DNS 名の調べ方

自分が利用しているコンピュータの IP アドレスを調べるには端末（Windows の場合コマンドプロンプト）から以下のコマンドを実行すると調べる事ができる．

Windows の場合：　ipconfig
Linux　の場合：　ifconfig

また，IP アドレスから DNS 名や，DNS 名から IP アドレスを調べる事も可能である．調べ方は

Windows，Linux 共通　：　nslookup

その後，「>」と表示されるので，　DNS 名（コンピュータ名＋ドメイン名）や IP アドレスを入力すると DNS サーバに問い合わせた結果が返却される．

演習１　：　以下の URL や IP アドレスについて nslookup コマンドで調べよ．また，調べた IP アドレスや DNS 名でブラウザからアクセスするとどうなるかを確認せよ．

１．https://www.portal.oit.ac.jp　→
２．http://www.oit.ac.jp　→
３．121.83.137.12　→

３．サーバへの接続

サーバマシンの操作は基本リモート接続によって行われる事が多い．本演習でも，演習室からサーバ室にあるサーバマシンへ SSH ポートを経由しリモートで操作する．

３．１　SSH での接続

SSH とは Secure Shell の略称で，コマンドベースでコンピュータをリモート操作するためのプロトコルの事である．当然，サーバ側には SSH サーバが稼働されており，SSH のポート番号（22 番）がファイアーウォールで許可されている必要がある．
サーバマシンに SSH 接続するためには，Windows，Linux で少し異なる．Linux の場合は端末を開いて

ssh ユーザ名@サーバマシンの IP アドレス（DNS 名）

例：　ssh root@150.89.17.178

でパスワードを要求されるので，パスワードを入力することで接続できる．対し，Windows の場合は Tera term 等の SSH 接続のソフトウェアが別途必要となる．

先に述べたように，SSH はコマンドベースでのリモート操作となる．以下に本演習で必要最低限のコマンドを列挙しておく．

ls	ディレクトリ内にあるファイルを表示. -l オプションでファイルパーミッション等の詳細が表示される.
pwd	カレント・ディレクトリパスを表示する
cd フォルダ名	指定したフォルダに移動する
cat ファイル名	指定したファイルの内容を表示する
chmod ファイル名	指定したファイルのパーミッションを変更
chown ファイル名	指定したファイルの所有者を変更
cp ファイル名 コピー先ファイル名	ファイルコピーを行う．二つ目の引数がコピー先となる 例　：　cp file1 /home/backup_file1 ←file1 を home フォルダの中に backup_file1 という名前でコピーする.
mv ファイル名　移動先	ファイルを移動，もしくはファイル名を変更する. 例１：　mv file1 /home/　←　home フォルダに file1 を移動する 例２：　mv file1 file2　←　file1 を file2 という名前に変更する
touch ファイル名	指定したファイルを作成する.
mkdir フォルダ名	指定したフォルダを作成する.
rm ファイル名	指定したファイルを削除する.
rmdir フォルダ名	指定したフォルダを削除する．ただし，指定したフォルダの中身が空で無ければいけない．そのため，rm コマンドで-rf オプションを付けて削除する方が良い.
useradd ユーザ名	指定した名前のユーザを作成する. /home 配下にホームディレクトリも自動的に作成される.
userdel ユーザ名	指定したユーザを削除する. -r オプションでホームディレクトリ等も削除される.
passwd ユーザ名	指定したユーザのパスワードを設定する. ユーザ名を指定しなければログインしているユーザのパスワードを変更する.
su ユーザ名	指定したユーザに切り替える．ユーザを指定しなければ root に切り替わる.
exit	ssh を終了する.
grep 文字列	指定した文字列を検索する．他のコマンドと併用する事が多い. 例：　ls	grep sample.txt ←　ls コマンドの結果から sample.txt

	を検索する 例： cat file.txt	grep sample ← file.txt の内容に sample という文字が無いかを検索
reboot	再起動する

３．２　管理者アカウントと一般ユーザアカウント

管理者アカウントとは，全ての権限を付与したアカウントの事を指す．Linux ではこれを root アカウント等と呼ばれる．root アカウントはソフトウェアのインストールから，ハードウェアのドライバーの導入や，一般ユーザのファイル操作等，全ての作業が行う事ができる．Linux では必ず root というユーザ名でログインしなければ管理者になることができない．

管理者アカウントに対し，一般ユーザアカウントとは，ソフトウェア等のインストール権限が無いユーザで，自分が所有するファイルにだけアクセスすることができる．

３．３　vi エディタの使い方

vi エディタとは，コマンドベースで動作するテキストエディタである．かなり特殊な使い方ではあるが，コマンドベースで動作するため，動作が軽く，また SSH 等でも編集を行うことができる．

vi エディタのコマンドは

vi 編集したいファイル名

例： vi sample.txt

で vi エディタを起動することができる．vi エディタには大きく２つのモードがあり，１つが移動モード，もう１つが挿入モードである（図１参照）．vi エディタ起動後は移動モードになっており，中身を編集することができない．

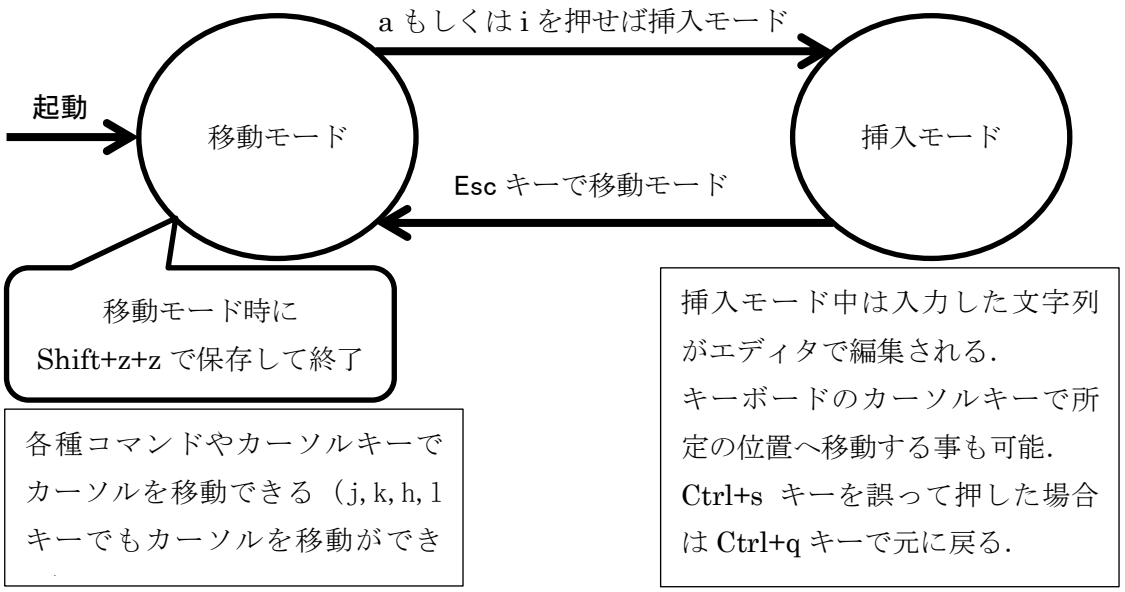


図 1　vi エディタの移動モードと編集モード

また，移動モードでは次の表のような便利なコマンドが存在する.

:q	vi エディタを終了する.
:q!	vi エディタを強制終了する.
:w	保存する．半角スペース入力後，ファイル名入力で別名で保存.
:wq (Shift+z+z と同じ)	保存して終了.
u	元に戻す.
yy	現在のカーソル位置の行をコピー.
p	下の行に貼り付け.
Shift + p	上の行に貼り付け.
/ 検索文字	指定した検索文字を先頭から検索する.
? 検索文字	指定した検索文字を末尾から検索する.
n	/ や ? で検索している時に次に一致する検索場所へ移動する
nnG	nn 行目に移動する

３．４　サーバとのデータ転送

サーバは SSH 等を用いてリモート接続することが多いため，クライアント PC で編集しているファイルをサーバへアップロード，またはダウンロードしたい場合がある．その時に利用するコマンドが SCP コマンドである．SCP コマンドは SSH ポートを利用しファイルを転送する Linux のコマンドである．Linux のコマンドであるため，Windows では利用できないが， WinSCP と呼ばれるソフトウェアを用いる事で scp コマンドと同様の事ができる．　Linux における scp コマンドの利用方法は以下である．

scp ローカルのファイルパス　ユーザ名@接続先ホスト名:コピー先のパス

例１：scp sample.txt root@150.89.17.178:/home/

入力すると，送り先のホストからパスワード要求され，パスワードを入力すると転送が始まる．また，上記のコマンド例では，クライアントのファイルをサーバにアップロードしているが，サーバにあるファイルをローカルの保存したい場合は，２つの引数を逆にすればよい．

Windows は WinSCP というソフトウェアを利用すると書いたが，基本 GUI があるため，画面の指示に従って設定するだけで SSH ポートを経由してファイルを転送することができる．

４．サーバインストール

サーバソフトウェアをインストールするために今回は yum コマンドを利用する．yum コマンドは Cent や Fedora と言った Red Hat 系の LinuxOS で利用できるパッケージ管理機能である． yum を利用することでソースコードのコンパイルが不要な他に，パッケージの依存関係等も勝手に考慮し，必要なソフトウェアを自動的に導入してくれることが可能である．yum コマンドの利用方法は以下である

yum install インストールしたいソフトウェア名

でインストールできる．コマンド実行にはインストールして良いかの確認が出るので y を押してエンターキーを押せば良い．また，導入できるソフトウェアの一覧を

yum list

で確認することもできるが，パッケージ量が非常に多いため，grep 等を用いて絞り込む必

要がある．

４．１　Web サーバのインストールと設定

注意：サーバ設定ファイルを編集する場合，必ず設定ファイルのバックアップを作成しておくこと（最悪，サーバソフトウェアが動作しなくなります！）

ここでは Web サーバソフトウェアとして有名な Apache のインストールとその設定を行う．yum によるパッケージ管理では Apache を httpd というソフトウェア名で管理されているため，yum コマンドを用いて Apache をインストールする時は

```
yum install httpd
```

となることに注意する．また，インストール後 Apache のサーバ設定を変更したい場合は，/etc/httpd/conf/　というフォルダの中に httpd.conf というファイルを編集する．設定のパラメータ等の情報は

```
http://httpd.apache.org/docs/2.2/ja/
```

にオンラインでドキュメントが公開されているので，そちらを参考すること．
また，Apache が初期で公開するルート・ディレクトリは

```
/var/www/html/
```

フォルダが指定されており，ルート・ディレクトリ配下に sample.txt を設置した場合，ブラウザから

```
http://IP アドレス/sample.txt
```

とするとサーバ起動後に sample.txt にアクセスすることができる．

４．２　サーバ起動，停止，再起動

Apache のインストールが完了したならば，次に Apache を起動する必要がある． yum コマンドでのインストールでは通常のプログラム実行とは異なる．実行には

```
service サービス名 オプション
```

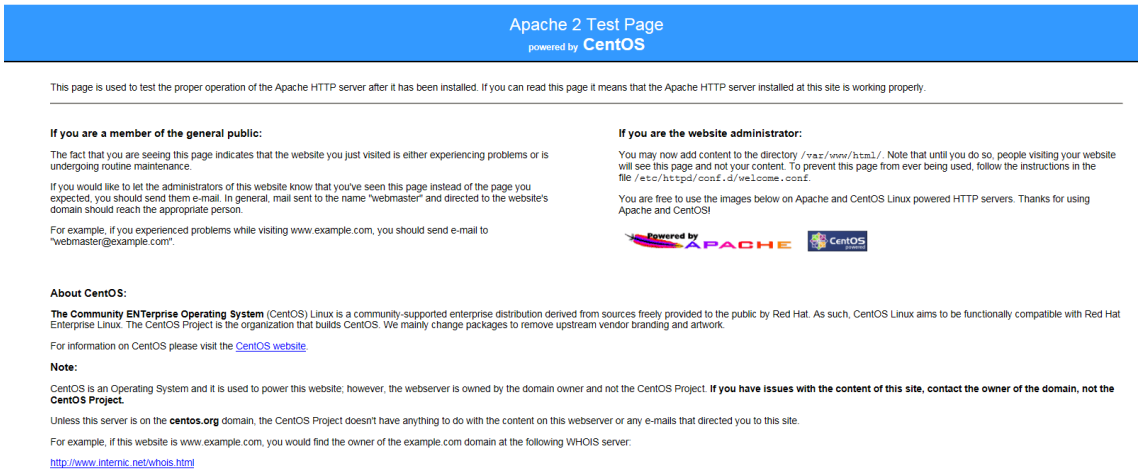
例：service　httpd start

となる．サービス名は yum でインストールしたソフトウェア名と基本同じで，オプションには start，stop，restart，reload 等がある．今回は Apache を起動するので，start コマンドであり，設定ファイルを編集した場合等は restart で再起動する必要がある．

ただし，デフォルトの CentOS ではファイアーウォールが起動しており，外部からは SSH のみが許可されている状態である．そこで，下記のコマンドを実行することで，ファイアーウォールを一時的に停止させることができる（サーバコンピュータ再起動後はまた起動しているので注意）．

service iptables stop

サーバ起動後，ブラウザでサーバの IP アドレスを入力し，次のような画面が出たら，無事に WEB サーバが起動できている．



４．３　サーバアクセスログ

管理者はサーバアクセスログを使ってどのようなユーザ（アドレスやドメイン）がアクセスしているかを管理する事も業務の一つである．サーバのアクセスログは

/var/log/

フォルダに集められている．例えば Apache のアクセスログは

/var/log/httpd/access_log

がある．他にも error_log や日付が付与されたログファイルがある．これらのファイルを cat コマンドで開くとアクセス日時や IP アドレス等の情報が記述されているので，それらの情報を基にどのようなユーザがアクセスしているかを推測することができる．また，デフォ

ルトの設定のアクセスログは下記のように記載されている.

150.89.244.23 - - [30/Mar/2016:16:47:33 +0900] "GET /ensyu.txt HTTP/1.1" 304 - "-" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E; GWX:QUALIFIED) "

最初の行頭がアクセスしてきた IP アドレスであり，次に日時，アクセス方法（GET なのか POST なのか）とアクセスしたファイル名等が記述されている．アクセスログの詳細については Apache のオンラインドキュメントに記載されているので, そちらを参考して頂きたい.

４．４　プロキシサーバへの対応

本学の演習室の環境ではプロキシサーバを経由し，ネットワークに接続されている．そのた，Web サーバのアクセスが全てプロキシサーバの IP アドレスになる．そこで，プロキシサーバを経由しているクライアント PC の IP アドレスを取得するように Aapche の設定を変更する必要がある．そこで，４．１でも述べたように httpd.comf の下記を修正することで，プロキシを経由した際でもローカルの IP をログに書き込めるようする.

（修正前）LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined

（修正後）LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" \"%{X-Forwarded-For}i\"" custom

末端の custom の前に半角スペースが入っている事に注意してください．更に，以下の部分も修正する

（修正前）CustomLog logs/access_log combined

（修正後）CustomLog logs/access_log custom

とすることで，下記のログファイルの末端のようにプロキシを経由しているクライアント PC の IP アドレスを判別することができる．IP アドレスは１つのログの末端に「”」で囲われて表示される

150.89.130.25 - - [31/Mar/2016:15:21:53 +0900] "GET /favicon.ico HTTP/1.1" 404 289 "-" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.3; WOW64; Trident/7.0; .NET4.0E; .NET4.0C; .NET CLR 3.5.30729; .NET CLR 2.0.50727; .NET CLR 3.0.30729; Tablet PC 2.0)" "150.89.217.75"

演習２　：

Apache のドキュメントルートに学籍番号だけを入力した「ensyu.txt」というファイルを作成し，演習のグループ内でアクセスができるかの確認を行う．またこの時アクセスがあったかを確認するために，httpd のログを見て，アクセスがあったかを確認すること．必ず４．４プロキシサーバへの対応を行った後に行う事.

IP アドレス	学籍番号

演習３　：

Apache のドキュメントルートを/var/www/rootdirectory/　に変更しなさい．変更後は演習２で作成した ensyu.txt を配置すること．今後の演習作業は変更したディレクトリで行うこと（var/www/html を使わない）.

演習４　：

Apache のポート番号を４４３番に変更しなさい．また，他のポート番号になるとどういう動作になるかも調べておくこと．今後は４４３ポートを使って演習を行うこと.

演習５　：

現在の設定では URL の末尾に演習３で作成したファイルを指定しなればいけない．そこで「http://IP アドレス/」で ensyu.txt（演習３で作成済み）にアクセスできるような設定にしなさい.

補足　：　Apache ではサーバマシンにブラウザからアクセスした場合，ファイルを指定しなくても（例：http://lss.oit.ac.jp/）ファイルを参照することができる．これは Apache の設定がファイルを指定してクライアントがアクセスしなかった場合，Apache が指定する

ファイルをクライアント返却するような設定になっているためである．なお，lss.oit.ac.jp では index.html がデフォルトで設定されており，多くの Web サーバが index.html となっていることが多い．

演習６　：

ファイアーウォールを起動させ，Web サーバへ接続できるように，ファイアーウォールの設定を行いなさい．ファイアーウォールの設定は

/etc/sysconfig/iptables

に記述されている．これを編集し，Web サーバもアクセスできるように修正せよ．ヒントは SSH は既に許可されていることである．

応用演習　１：

ファイルサーバである，samba のインストールし，チーム内でアクセスできる共有フォルダを作成しなさい．構築に関し，Web 検索を利用しても良い．

注意１：Samba は Linux 上のユーザと Samba 上のユーザの２つのユーザが必要となり，それぞれにパスワードの設定が必要である．

注意２：Windows の場合，Samba へのアクセスはブラウザ上から

file://IP アドレス

からアクセスすることができる．

他にも，「ファイル名を指定して実行（Windows キー＋Ｒ）」から￥￥IP アドレスからでも接続可能（￥は半角）

注意３：Linux（Ubuntu）の場合，デスクトップのファイルからサーバへの接続を選択し，Windows（共有）で接続することができる．

応用演習　２：

DropBox 等のクラウドストレージサービスのような owncloud を Web 検索を利用して導入せよ．導入に関し，php モジュールの追加や，場合によっては MySQL の導入が必要となるので, 演習中に編集した apache の設定ファイルと Log ファイル等のバックアップを必ずローカルマシンに保存しておくこと．

２日目

５．Web アプリケーションの概要

本来の Web サーバの役割はリクエストのあったファイルの内容を http ポートを利用してクライアントに返却する事である．そのためインターネット普及時の Web ページはクライアント等で HTML を作成し， FTP サーバ等を経由し Web サーバにアップロードするだけの「動きの無い（静的）な Web ページ」であった．

しかし，オンラインショッピング等の情報システムは状況（アクセスするユーザや，日時等）によって表示する内容が変わる「動きのある（動的）な Web ページ」が実現されている．これらは，サーバサイドで実行するプログラムによって，動的に HTML を生成し，生成した HTML をクライアントに返却しているだけである．つまり，一般的に言われている Web アプリケーションとは動的に生成される HTML であり，その HTML を動的に生成してくれるプログラムが CGI や PHP 等のサーバサイドスクリプト言語である．

６．　HTML の記述

HTML はブラウザ等が解釈するマークアップ言語である．HTML を用いる事で見栄えを良くしたり,データを入力するフォームを作成する等の GUI を良くするための言語である．HTML はタグと呼ばれる特別な文字で形成された要素から成り立っており，タグにも開始タグ，終了タグがあり，開始タグから終了タグまでに挟まれた文字列を修飾する．

リスト１は HTML の最も単純な書き方を示したものである. HTML は html タグ, head タグ，body タグの３つの要素から成り立っており，html タグ要素の中に head タグ要素と body タグ要素がある．html タグはこのファイルが html であることを示し，head タグはタイトルや作成者の情報，このページの検索用キーワード等の内部情報を持たせる事ができ，最後に body タグは本文を記述するタグとなる．

他にも，フォントの大きさを変えるや，画像を張り付ける等の様々なタグが用意されている．html に関する詳しいリファレンスは下記の Web ページを参考にして頂きたい．

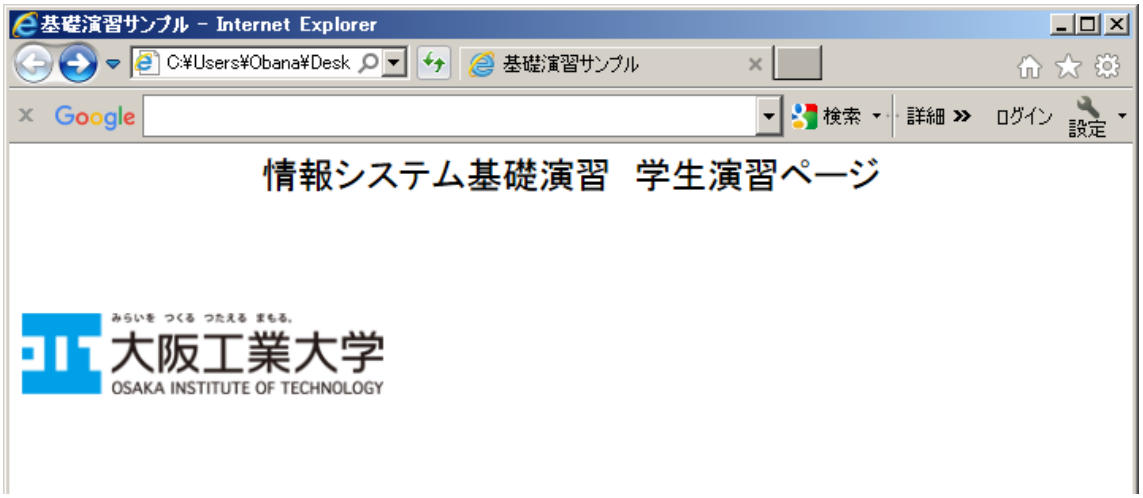
http://www.htmq.com/html/indexm.shtml

```
<html>
    <head>
        <title>基礎演習サンプル</title>
    </head>
    <body>
        本文
    </body>
</html>
```

リスト１　HTML の基本構成

演習７　：

下記のような html を作成しなさい．なお，画像には http://www.oit.ac.jp へのリンクが張られている．ファイル名は work1.html とする．



８．　PHP のよる Web アプリケーション開発

Web サーバで php を動作するためには php モジュールが必要となる．そこで，yum コマンド用いてサーバに php を導入する．

```
yum install php
```

８．１　PHP の基本

PHP はサーバサイドプログラムであり，動的に HTML を生成するものである．つまり，単純に print 文を使って HTML 文を出力することで，動的な Web ページを作成している．リスト２に PHP 文のサンプルを示す．

PHP は body タグの中に

```
<?php
処理内容
?>
```

と直接 PHP プログラムを記述することでスクリプトが実行される．記述したスクリプトは C 言語や Java 言語のように１ステートメント（１命令）ごとに「；」を入力する．また，php を実行する際にはファイルの拡張子を「.php」に変更する必要がある．「.html」では<?php>部分が単なる文字列として出力されてしまい，記述したスクリプトがそのまま出力されるので注意すること．また，実行時エラーは表示されず，以下のディレクトリにある Apache のエラーログに保存される．

```
/vat/log/httpd/error_log
```

```
<html>
    <head>
        <title>基礎演習サンプル</title>
    </head>
    <body>
        <?php
            print' Hello PHP ';
        ?>
    </body>
</html>
```

リスト２　PHP の基本

８．２　PHP の変数宣言

PHP はスクリプト言語のため，C 言語の int や double のような厳密な型宣言がありません．また，変数を宣言する際には必ず変数名の前に「$」を付ける必要がある．

例　：　$data；

PHP で数字を扱う場合には

$data = 10;

とＣ言語と同様に扱えば良いが，文字列を扱う場合には

$str = ' Hello ';

とシングルコーテションで挟む必要がある．また，演算も C 言語等と同様に

$data = $data + 10;

「+」と記述すれば加算を行うことができる．対し文字列を連結する場合は

$str = $str . 'PHP'

と連結する変数や文字列の間に「ドット」を入力する必要がある．また配列宣言は最初に要素数を設定する必要が無く，

$data[0] = 10;
$data[1] = 20;

と，添え字を増やしていくことで，要素数が可変長の配列を自動的に生成する．

８．３　PHP の制御文

PHP の制御文は基本Ｃ言語等と同じで，例えば if 文等の場合は

if($a < $b) print(' $b');

と記述することができる．他にも for 文や while 文等もあるが，C 言語とほぼ同等なので，今回は説明を省略する．

８．４　PHP の関数

PHP でも関数を宣言することができる．関数は<php　　>の中に以下のように記述すると宣言することができる．

```
function 関数名(引数){
        処理
        return 0;
}
```

関数宣言の際に戻り値の型を宣言する必要は無い．利用する際には従来の C 言語等と同じで

```
関数名(引数);
```

とすると，関数を呼び出すことができる．重要な点は関数の利用であり，PHP はデフォルトで様々な関数が用意されており，それらを利用することで用意に PHP プログラムを作成することができる．例えば，配列の大きさを取得したい場合等は

```
data[0] = 10;
data[1] = 20;
print( count($data) )
```

と count 関数を利用することで，容易に配列の大きさを取得することができる．

その他にも日付を取得する関数や，テキスト処理等の様々な関数が用意されている．詳しくは下記の URL の「関数リファレンス」の項目を参考にして頂きたい．

８．６　特殊な型について

８．２で述べたように php では int 型等といった厳密な型宣言がありません．しかし，php では様々な関数が用意されており，その中で特殊な戻り値が返却される場合がある．特殊な戻り値とは，完結に言えばポインタを指す．例えば，PHP にてファイル読み込みを行う際には fopen 関数を利用するが，PHP のリファレンスでは以下のように定義いる．

resource fopen (string $filename , string $mode [, bool $use_include_path = false [, resource $context]])

戻り値に resource と書かれているが，リファレンスの戻り値を参考すると，「成功した場合にファイルポインタリソース、エラー時に FALSE を返します。」と記述されている．このような場合でも，８．２のように

```
$fp = fopen(“sample.txt” , “r”);
```

とすれば，ポインタを参照することができる．このように，型が無いため，宣言している型が何の型が判別しにくい事もスクリプト言語の特徴である．なお fopen のあとは以下のようにすれば，オープンしたファイルの中身を読み取り，出力することができる

```
while (!feof($fp)) {
  $line = fgets($fp);
  print $line;
  print "<br>\n";
}
```

８．６　ページ遷移時のデータのやり取り（POST，GET）

HTML では入力フォームを生成し，入力されたデータによって HTML 内容を変えたり，ページ遷移時にデータを引き継ぐために，GET，POST と呼ばれる手法がある．

GET とは，URL の後ろに「?フォーム名=値」という形でデータ（これをクエリと呼ぶ）をやり取りする手法である．ブラウザから次のページをリクエストする際にブラウザが URL の末端にクエリを付けて次のページに遷移し，遷移先のページで値を受け取ると言う手法である. URL のクエリを付けるため, データ量に制限があることに注意が必要である．

これに対し，POST とはリクエスト情報（エンティティ）に記載される．リクエスト情報は一般的に利用する際には見ることが無いものであり, サーバにリクエストする際にブラウザがリクエスト内容と一緒にクエリをサーバに送信する手法である．GET と違い，リクエスト内容にクエリを記載するため，データ制限が無く，テキストフィールド等の長文のデータやり取りをする際に有効である．また，GET，POST 共にデータをサーバに保持するのでは無く，ブラウザから送信される．

８．７　入力フォーム

入力フォームは HTML で生成される．リスト３は入力フォームが２つとボタンがひとつあるサンプルである．入力フォームではまず，フォームタグで外枠を生成し，その後中にボタンやテキストフィールドといったコンテンツ（INPUT タグ）を記述する．フォームタグでは，入力されたデータをどこのページ（どの php ファイル）にどのような手段（POST なのか，GET なのか）をパラメータで設定する．サンプルでは sample.php に post を使ってデータを渡すというものである．

また，フォームタグ内のテキストフィールドでは，type="text"で１行が入力できる入力フォームを生成し，その名前を name とするという意味である．最後に type="submit"がボタンであり，ボタンが押された場合，form の action で指定したページに移動する．

```
<form action="sample.php" method="post">
        名前: <input type="text" name="name" />
        内容: <input type="text" name="data" />
        <input value="送信" type="submit" />
</form>
```

リスト３　入力フォームのサンプル

post されたデータを php で受け取る場合のサンプルをリスト４に示す．リスト４では $get_val1, $get_val2 で入力フォームから POST されてきたデータを受け取る事ができる．

```
<?php

        $get_val1 = $_POST['name'];
        $get_val2 = $_POST['data'];

        print($get_val1. '<br>');
        print($get_val2);
?>
```

リスト４　POST プログラムのサンプル

演習８　：

演習 7 で作成した work1.html をコピーし，リスト３の入力フォームを追加し，

work2.html を作成する．更にリスト４を実装した work2.php も作成すること．

演習９　：

演習８で作成したものを GET を用いた方法に変更した work3.html と work3.php を作成せよ．尚，PHP のリファレンスを参照すること．

演習１０　：

以下の仕様を満たす簡単な掲示板プログラムを作成せよ．

- work4.php の仕様
  - ２つの入力フォームがある
    - １つめは名前の入力フォーム
    - ２つめが書き込む内容を入力するフォーム
  - data.txt の内容を読み込み，表示させる
- kakunin.php の仕様
  - 戻るボタンがあり，押すと work4.php に戻る
  - work4.php から post されたデータを data.txt に保存する
- data.txt の仕様
  - 書き込まれた内容が書かれている
  - フォーマットは特に指定しません

ファイル操作があるが，この辺りは PHP のリファレンスページのファイル操作関数である fopen，fwrite，fread（もしくは fgets）関数を調べる．なお，排他制御などは考慮しなくてもよい．

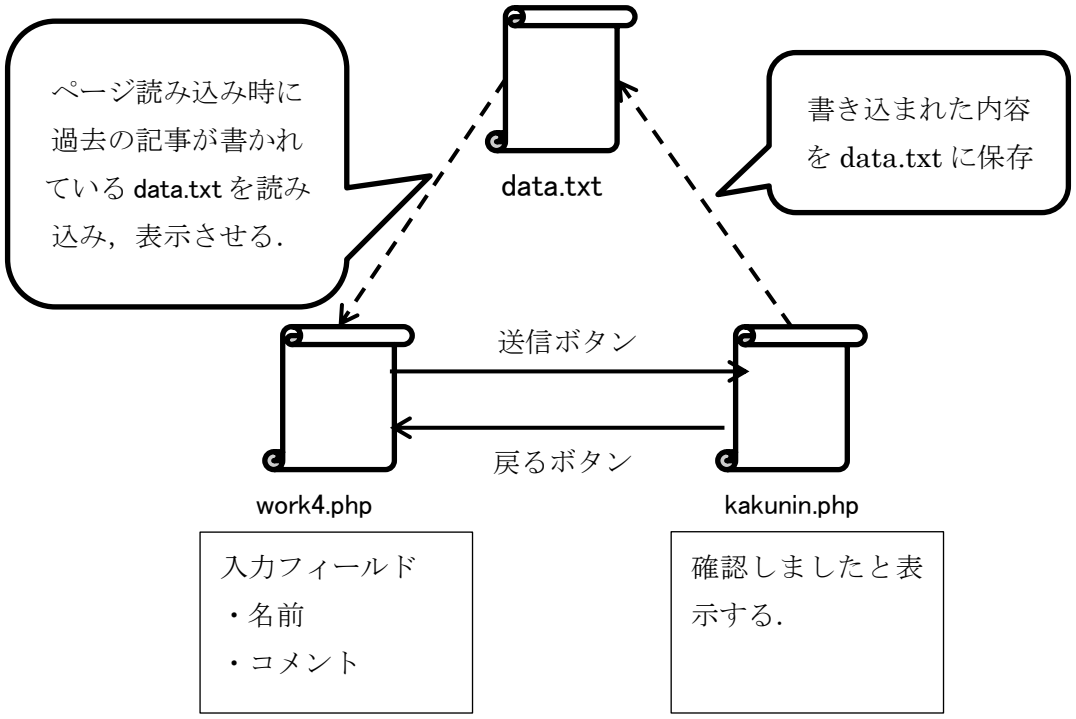


図２　演習９の仕様

演習１１　：

PHP で用意されている好きな関数を２つを利用して，演習１０で作成した掲示板プログラムに組み込み新たな機能を実装しなさい.