



[Home Page](#)

[Title Page](#)



Page 1 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

# Linux, Apache, Mysql, PHP

Sameer D. Sahasrabuddhe

KReSIT, IIT Bombay

<http://www.it.iitb.ac.in/~sameerds/>



[Home Page](#)

[Title Page](#)



Page 2 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## The World Wide Web



[Home Page](#)

[Title Page](#)



Page 3 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## Client-Server Architecture

- Server: provides services.
- Client: generates requests for services.
- May or may not run on the same machine.
- Typically, they use a standard protocol for communication.
- For example, Apache is a webserver, the browser is a client and the protocol used is HTTP.



## What's a webserver?

- A process that responds to requests for resources on port 80.
- Typical requests are for hypertext files and related objects.
- The client uses a Universal Resource Locator (URL).
- Example - <http://www.kernel.org/pub/>
- A typical HTTP transaction:
  1. The browser (client) makes a request to the website.
  2. The webserver looks up the resource specified.
  3. The webserver reads/generates the corresponding content and sends it to the browser (client).

Home Page

Title Page



Page 4 of 39

Go Back

Full Screen

Close

Quit



## The Apache HTTP Server

- The number one http server on the Internet - 63% websites, as of August 2002.
- A project of the Apache Software Foundation.
- Provides full source code and comes with an unrestrictive license - the Apache Software License.
- Runs on Windows NT/9x, Netware 5.x and above, OS/2, and most versions of Unix, as well as several other operating systems.
- Can be customised by writing “modules” using the Apache module API.

Home Page

Title Page



Page 5 of 39

Go Back

Full Screen

Close

Quit



## What are modules?

Apache's functionality is contained in modules that you can choose to include or exclude.

## Types of modules

**static modules** compiled into the httpd binary.

**dynamic modules** stored separately and can be optionally loaded at run-time.

**third-party modules** modules that are not included in the standard Apache HTTP Server distribution.

Home Page

Title Page



Page 6 of 39

Go Back

Full Screen

Close

Quit



[Home Page](#)

[Title Page](#)



Page 7 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## Some common modules

**mod\_cgi** Execution of CGI scripts.

**mod\_imap** Server-side imagemap processing.

**mod\_include** Server-parsed html documents (Server Side Includes).

**mod\_php** Server-side scripting in PHP (PHP Hypertext Preprocessor).



Home Page

Title Page



Page 8 of 39

Go Back

Full Screen

Close

Quit

## Controlling Apache

**/etc/httpd/conf/httpd.conf** The configuration file for the Apache HTTP server.

**.htaccess** File used to control Apache's behaviour on a per-directory basis.

**/etc/init.d/httpd** Script used to control the httpd process - start, stop, restart, reload, status.

**/var/log/httpd/** Contains the Apache logs - error.log, access.log



Home Page

Title Page



Page 9 of 39

Go Back

Full Screen

Close

Quit

## Apache content

`/var/www/html/` The “DocumentRoot” - all websites are served from here by default.

`$home/public_html/` User’s directory for serving webpages. For example, `http://www.it.iitb.ac.in/~sameerds/` is served from `/home/sameerds/public_html/`

`index.html` “DirectoryIndex” that is used by default when a request specifies a directory.



## Dynamic Content Generation

- Client-Server decoupling: the client never knows the physical format of a resource.
- The reply to a request is a data stream from the server.
- The server may transparently create or modify data. Even images can be generated on the fly!
- This can be done in several ways:
  1. CGI - Common Gateway Interface to external programs.
  2. SSI - Server-Side Includes.
  3. Server Side Scripts - the most flexible way to use dynamically generated webpages.

Home Page

Title Page



Page 10 of 39

Go Back

Full Screen

Close

Quit



[Home Page](#)

[Title Page](#)



Page 11 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

# PHP: Hypertext Preprocessor



[Home Page](#)

[Title Page](#)



Page 12 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## What is it?

- A general purpose scripting language that is especially good for server-side scripting.
- A project maintained by the Apache Software Foundation.
- All major operating systems - Linux, many Unix variants, Microsoft Windows, Mac OS X, RISC OS, and probably others.
- All major web servers - Apache, Microsoft IIS, PWS, Netscape and iPlanet servers, O'Reilly Website Pro server, Caudium, Xitami, Omni-HTTPd, and many others.



[Home Page](#)

[Title Page](#)



Page 13 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## What can PHP do?

- Server-side scripting.
- Command line scripting.
- Writing client-side GUI applications: PHP-Gtk
- Text processing features such as PCRE's, XML documents, XSLT transformation.
- Abilities also included generating images, PDF files and even Flash movies, on the fly.



## External services

- Database connectivity
  - A wide range of databases - MySQL, PostgreSQL, Informix, ODBC, Oracle ... too many more to list.
  - DBX and ODBC interfaces, that allow scripting independent of the database backends.
- Network services such as IMAP, POP, SMTP, DNS, LDAP, SNMP etc. as well as raw TCP sockets!
- Support for instantiating Java objects, remote CORBA objects, and COM on windows.

Home Page

Title Page



Page 14 of 39

Go Back

Full Screen

Close

Quit



## PHP on the Web

PHP can be directly embedded in HTML markup.

```
<html>
  <head>
    <title>Hello World!</title>
  </head>
  <body>

    <?php
      echo "Hi, I'm a PHP script!";
    ?>

  </body>
</html>
```

Home Page

Title Page

◀ ▶

◀ ▶

Page 15 of 39

Go Back

Full Screen

Close

Quit



[Home Page](#)

[Title Page](#)



Page 16 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## PHP on the Web (continued)

```
<html>
  <head>
    <title>Hello World!</title>
  </head>
  <body>

    Hi, I'm a PHP script!
  </body>
</html>
```



[Home Page](#)

[Title Page](#)



Page 17 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## Language features

- Variables - pre-defined, external, user-defined, variable variables, references.
- Standard data-types such as booleans, integers, floating point numbers, strings and arrays.
- Control structures - if-then-else while, do-while, switch, foreach.
- Functions - user-defined functions similar to C, variable functions.



## Variables in PHP

### Variable variables!

```
$foo = "bar";  
$$foo = "value";
```

The result is that the variable \$bbar, gets the value value.

### References

```
<?php  
$foo = 'Bob' ;  
$bar = &$foo ;  
$bar = "My name is $bar" ;  
echo $bar ;  
echo $foo ;  
?>
```

[Home Page](#)

[Title Page](#)



Page 18 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



## Arrays

```
$a = array( 1 => 'one',  
           2 => 'two',  
           3 => 'three' );
```

```
$a[1] = 'one';  
$a[2] = 'two';  
$a[3] = 'three';  
$a[] = 'four';
```

```
$a = array( 'color' => 'red'  
           , 'taste'  => 'sweet'  
           , 'shape'  => 'round'  
           , 'name'   => 'apple'  
           ,  
           4  
           );
```

```
$a[] = 'four';
```

Home Page

Title Page

◀ ▶

◀ ▶

Page 19 of 39

Go Back

Full Screen

Close

Quit



[Home Page](#)

[Title Page](#)



Page 20 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## Arrays functions

```
unset ($a['color']);
```

```
reset ($a);
```

```
while (list($key, $value) = each ($a)) {  
    echo "Key: $key; Value: $value<br>\n";  
}
```

```
foreach ($a as $key => $value) {  
    echo "Key: $key; Value: $value<br>\n";  
}
```



## Functions

```
<?php
function foo()
{
    echo "In foo()<br>\n";
}

function bar($arg = '')
{
    echo "In bar(), with '$arg'.<br>\n";
}

foo();
$func = 'foo'; $func();

bar('test');
$func = 'bar'; $func('test');
?>
```

Home Page

Title Page

◀ ▶

◀ ▶

Page 21 of 39

Go Back

Full Screen

Close

Quit



## OOP in PHP

```
<?php
class Cart
{
    var $items;
    ...

    function add_item ($artnr, $num) { ... }

    function remove_item ($artnr, $num) { ... }
}

$cart = new Cart;
$cart->add_item("10", 1);

$another_cart = new Cart;
$another_cart->add_item("0815", 3);
?>
```

Home Page

Title Page

◀ ▶

◀ ▶

Page 22 of 39

Go Back

Full Screen

Close

Quit



## HTML Forms

```
<form action="foo.php" method="post">  
Name: <input type="text" name="username"><br>  
Email: <input type="text" name="email"><br>  
<input type="submit" name="submit">  
</form>
```

```
<?php  
    print $_POST['username'];  
    print $_REQUEST['username'];  
  
    // g, p and c type variables:  
    import_request_variables('p', 'p_');  
    print $p_username;  
  
    print $HTTP_POST_VARS['username'];  
    print $username;  
?>
```

Home Page

Title Page

◀ ▶

◀ ▶

Page 23 of 39

Go Back

Full Screen

Close

Quit



[Home Page](#)

[Title Page](#)



Page 24 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## Session handling

- A way to preserve certain data across subsequent accesses.
- A visitor accessing your web site is assigned a unique session id, which is used in two ways.
- Cookies that are stored on the client, to be read on subsequent accesses.
- Session ID's that are propagated in the URL - more reliable.



## Session example

```
<?php  
session_register ("count");  
$count++;  
?>
```

Hello visitor, you have seen this page  
<?php echo \$count; ?> times.<p>

To continue,  
<A HREF="nextpage.php?<?=SID?>">  
click here</A>

Home Page

Title Page

◀ ▶

◀ ▶

Page 25 of 39

Go Back

Full Screen

Close

Quit



## Cookies

- A mechanism for storing data in the client.
- Any cookies sent from the client will automatically be turned into a PHP variable.

```
setcookie ("cookie[three]", "cookiethree");  
setcookie ("cookie[two]", "cookietwo");  
setcookie ("cookie[one]", "cookieone");  
if (isset ($cookie)) {  
    while(list ($name, $value) = each($cookie)) {  
        echo "$name == $value<br>\n";  
    }  
}
```

Home Page

Title Page

◀ ▶

◀ ▶

Page 26 of 39

Go Back

Full Screen

Close

Quit



[Home Page](#)

[Title Page](#)



Page 27 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

# MySQL



[Home Page](#)

[Title Page](#)



Page 28 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## What is it?

- The most popular Open Source SQL database.
- Developed, distributed and supported by MySQL AB.
- Several different client programs, libraries and administrative tools..
- A wide range of programming interfaces (APIs).
- Works on a wide range of operating systems including Windows, Unix variants, Mac OS X, etc.



## Installation

- Installation on Red Hat Linux is a breeze, using rpms for `mysql`, `mysql-server` and `mysqlclient`.
- Control script - `/etc/init.d/mysqld`.
- Users on the server are distinct from those on the system.
- The user accounts and database details are stored in a database called `mysql`.
- All other users are created and controlled by the `mysql root`.
- Users can create databases, tables, other users, etc depending on privileges provided by the `root`.

Home Page

Title Page



Page 29 of 39

Go Back

Full Screen

Close

Quit



## Using the mysql client

Use `mysql` to connect to a MySQL server.

```
shell> mysql -h host -u user -p
```

```
shell> mysql -h host -u user -p < batch-file
```

### Configuration files

**/etc/my.cnf** Global options

**DATADIR/my.cnf** Server-specific options

**defaults-extra-file** The file specified with `--defaults-extra-file=#`

**/.my.cnf** User-specific options

Home Page

Title Page

◀ ▶

◀ ▶

Page 30 of 39

Go Back

Full Screen

Close

Quit



## User administration

```
shell> mysql -u root mysql
mysql> SET PASSWORD FOR
    ->   root@localhost=PASSWORD('new_password');

shell> mysqladmin -u root password new_password

mysql> GRANT ALL PRIVILEGES ON *.* TO monty@localhost
    ->   IDENTIFIED BY 'some_pass' WITH GRANT OPTION;
mysql> GRANT ALL PRIVILEGES ON foo.* TO bar@"%"
    ->   IDENTIFIED BY 'some_pass';
mysql> GRANT USAGE ON *.* TO simpleton@some.server
    ->   IDENTIFIED BY 'some_pass';
```

Home Page

Title Page



Page 31 of 39

Go Back

Full Screen

Close

Quit



[Home Page](#)

[Title Page](#)



Page 32 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## MySQL Privilege System

Managed by tables in the database called `mysql`.

**user** global privileges of each user.

**db** which user can access which database from which host.

**host** extension to `db` for multiple hosts.



## Data, Databases and Tables

- SQL - Structured Query Language
- Queries such as create, delete, select, update, insert, etc.
- Numeric Datatypes - INT, BIGINT, TINYINT, FLOAT, DECIMAL, etc
- String Datatypes - CHAR, VARCHAR, TEXT, BLOB
- ENUM and SET - “one”, “two”, “three”
- DATETIME, DATE, TIMESTAMP

[Home Page](#)

[Title Page](#)



Page 33 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



## MySQL databases

```
mysql> SHOW DATABASES;
```

```
+-----+
| Database |
+-----+
| mysql   |
| test    |
| tmp     |
+-----+
```

```
mysql> CREATE DATABASE menagerie;
```

```
mysql> USE menagerie
```

```
Database changed
```

```
mysql> SHOW TABLES;
```

```
Empty set (0.00 sec)
```

[Home Page](#)

[Title Page](#)

◀ ▶

◀ ▶

Page 34 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



[Home Page](#)

[Title Page](#)



Page 35 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## Putting it together



Home Page

Title Page



Page 36 of 39

Go Back

Full Screen

Close

Quit

## Resources

The resources listed here are usually installed on the system, under `/usr/share/doc/`

- The Apache Manual <http://httpd.apache.org/>
- The PHP Manual <http://www.php.net/manual/en/>
- The MySQL Manual  
<http://www.mysql.com/documentation/>



[Home Page](#)

[Title Page](#)



Page 37 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

**RTFM**



[Home Page](#)

[Title Page](#)



Page 38 of 39

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

Thanks..



*Home Page*

*Title Page*



*Page 39 of 39*

*Go Back*

*Full Screen*

*Close*

*Quit*