

PHP

Hypertext Preprocessor

- Server scripting language
- powerful tool for making

roots in C, Perl
resembles Java
dynamic & interactive

widely used, free, and efficient alternative to ASP

PHP can generate dynamic page content

create, open, read, write, delete and close files on the server

collect form data

send and receive cookies

add, delete, modify data in db

used to control user-access

can encrypt data

can output

HTML

images

PDL files

XHTML

XML

features

runs on various pfs

Compatible with almost all servers (Apache, IIS)

supports wide range of db

free.

easy to learn

runs efficiently on the server side

loosely typed language

Syntax

Simplicity
Efficiency
Security
flexibility

```
<?php
```

```
?> // php code goes here
```

→ PHP code ends with ;

→ keywords, classes, fun, user-defined functions are not case-sensitive.

→ variable names are case sensitive

comments

```
// single-line
```

```
# single-line
```

```
/* multiple-line
```

```
*/
```

Variables

start with \$

no declaring state in php

Created when assigned a

variable

A-z, 0-9, -

variable scopes : local (inside)
global : declared outside, accessed only from outside
static

data types

String
Integer
Float
Boolean
Array
Object

var_dump(\$varname)
returns datatype(val)

<?php

```
class Car {  
    public $color;  
    public $model;  
    public function __construct($color, $model)  
    {  
        $this->color = $color;  
        $this->model = $model;  
    }  
    public function message()  
    {  
        echo "My car is ". $this->color . " " . $this->model . "!";  
    }  
}  
  
$mycar = new Car("A", "B");  
echo $mycar->message();  
echo <br>
```

Conditional statements

if, if...else,
if...else if...else, Switch

looping

while, do...while, for, foreach

Arrays : can store multiple values in 1 single Variable

access using indexing

array() keyword

3 types ← Indexed with num index
Associative (named)
Multidimensional (one or more keys)

count() → length

array_sum()

array_shift()

array_search()

array_reverse()

end()

prev()

, reset()

array_unshift()

array_unshift()

asort()

arsort()

Arrays

<?php

// create indexed array

\$fruits = array("A", "B", "C");

// indexing

echo "First ele is: " . \$fruits[0] . "ln";

// adding elements

\$fruits[] = "D";

\$fruits[] = "E";

echo "Newly added: " . end(\$fruits) . " and " . prev(\$fruits)

// associative array

\$person = array('f-name' => "John", 'l-name' => "Doe");

// MultiD array

```
" " ( " " )
" " ( "John" , "Doe" )
```

Functions

without parameters

with parameters

with default parameter

variable-length parameter

with return statement

access global var

recursive fun

Strings

strlen()

strtoupper()

strtolower()

substr()

• str-replace()

explode()

implode()

string interpolation

escape char

multiline string = <<< EOD

String

EOD;

File

- open and read document → readfile("filename" .txt)

fopen() - open (create a file)

fwrite()

fclose()

feof(\$filename)

fgets()


```

$filepath = "example.txt"
$file = fopen($filepath, "w");
fwrite($file, "Hello, world!");
fclose($file);

// Read file
$file = fopen($filepath, "r");
while(!feof($file)) {
    echo fgets($file);
}
fclose($file);
?>

```

Cookie : → small file that server embeds on user's computer
 → often used to identify user
 → Each time same comp requests a page with a browser, it will send the cookie too.

→ With PHP, you can create and retrieve cookie values

Create : `setcookie(name, value, expire, path, domain, secure, httponly);`

→ must appear before HTML

```

<?php
$cookie-name = "user";
$cookie-value = "John Doe";
setcookie($cookie-name, $cookie-value, time() + (86400*3), "/");
?>

```

↑
1 day

```

<html>
<body>
<?php
if(!isset($_COOKIE['cookie-name'])) {
    echo "is not set";
} else {
    echo "name is set";
    echo "value is " . $_COOKIE['cookie-name'];
}

```

modify

use setcookie() again

delete

use setcookie() with date in past

setcookie("user", "", time()-3600);

check if cookies are enabled

first create a test cookie with setcookie()

```
<?php  
setcookie("test-cookie", "test", time()+3600, '/');
```

```
?>
```

```
<html>
```

```
<body>
```

```
<?php
```

```
if(count($_COOKIE)>0)
```

```
echo "cookies are enabled.";
```

```
else
```

```
?>
```

disabled

Session

→ when you work with an application, you
open it, do some changes and then displays it.
close

Session

→ comp knows who you are

On internet

web server doesn't

and what you do

because HTTP address doesn't maintain state.

→ session variables store this info to be used across
multiple pages

→ last until user closes the browser

start

session variables are set

session_start()

session_destroy()

with

\$_SESSION

before html tags

```
<?php
```

```
session_start();
```

```
?>
```

```
<html>
```

```
<body>
```

```
<?php
```

```
$_SESSION['favcolor'] = "green"
```

```
$_SESSION[''] = "b"
```

```
echo "session variable set"
```

```
print_r($_SESSION);
```

```
</body>
```

```
</html>
```

OOP about creating objects that contain data and functions.

faster and easier to execute
provides clear structure for programs

DRY

class: template for objects
object: instance of a class

variables - properties
functions - methods

```
<?php
class Fruit {
    // prop
    public $name;
    public $color;

    function set-name($name) {
        $this->name = $name;
    }
    function get-name() {
        return $this->name;
    }
}
```

```
<?php
class Fruit {
```

```
}
!>
```

keywords
class

\$this

refers to current obj
only available inside methods

instance of

checks if an obj belongs
to specific class

final to prevent inher,
overriding

— construct()
— destruct()

access specifiers
public
private
protected

constants

→ `const constant = "Bye"`

`echo $constant;`

→ case sensitive

can access outside class
by

`classname :: constant`

resolution

abstract classes

inheritance

interfaces

MySQL using PHP

<?php // db connection and op"

① // db config

```
$servername = "localhost"
$username = "root"
$password = "root"
```

② // create a connection

```
$conn = new mysqli()
```

③ // check connection

```
if ($conn->connect_error) {
    die("Conn failed: " . $conn->connect_error);
}
```

④ // create a new db

```
$database = "ex-db";
```

```
$sqlCreateDb = "CREATE DB IF NOT EXISTS $database";
```

```
if ($conn->query($sqlCreateDb) == TRUE) {
```

```
    echo "created or exists!"
```

```
else
```

```
    die("DB conn error")
```

⑤ // list db

```
$dbs = $conn->query("SHOW DBS");
```

```
if ($dbs->num_rows > 0) {
```

```
    echo "\n DBs: \n";
```

```
    while ($row = $dbs->fetch_assoc())
```

```
        echo $row["DB"]. "\n";
```

```
    else
```

```
        echo "no db found";
```

⑥ // select db

```
$conn->select_db($database)
```

⑦ // create a table

```
$sqlCT = "CREATE TABLE IF NOT EXISTS users (
```

```
    id INT AUTO-INCREMENT PK,
```

```
    username VARCHAR(30));
```

```

if ( $conn → query( $sqlCT) == TRUE )
    echo "T created ";
else
    die( "Error C, T : " . $conn → error );

```

⑧ // List tables

```

$tables = $conn → query( "SHOW TABLES" );
if( $tables → num-rows > 0 ) {
    echo "In Tables in db : \n";
    while( $row = $tables → fetch-assoc() ) {
        echo $row[ "Tables_in_db" ] . "\n";
    }
} else {
    echo "No tables ";
}

```

⑨ // Insert data into table

```

$sqlID = "INSERT INTO users( username, email ) VALUES
( 'A' , 'A@gmail.com' ) ";
if ( $conn → query( $sqlID) == TRUE )
    inserted
else
    no

```

⑩ // alter table add new column

```

$var = "ALTER TABLE users ADD COLUMN age INT ";
if ( $conn → query( $var) == TRUE )
    altered
else
    die( $conn → error )

```

⑪ // query

```

$sqlSelectData = "SELECT * FROM users";
$result = $conn → query( $sqlSelectData )
if( $result → num-rows > 0 ) {
    echo "Users : ";
    while( $row = $result → fetch-assoc() ) {
        echo "ID : " . $row[ "id" ] . " , Username : " .

```


⑫ // Delete data from table

```
$sqlDD = "DELETE FROM users WHERE username='john';"
if( $conn->query( $sqlDD) === TRUE)
    echo "In Data Del successfully"
else
    die
```

⑬ // Delete table of DB

```
$sqlDT = "DROP TABLE IF EXISTS users";
if( $conn->query( $sqlDT) === TRUE) {
    del
} else {
    error
```

⑭ // close connection

```
$conn->close()
```

phpmyadmin

- popular web based dbm tool for MySQL db
- provides user friendly if for performing tasks like creating, modifying, querying database as well as managing db users and privileges.
- used by dev & db administrators
- simplifies dbm tasks through web brows

index

Welcome to phpMyAdmin

Language

English ☒

Login

username

password

next page

enter cred of mysql server > log in

Databugs

Data Entry error

Data duplication

Data validation failures

Data Integrity violations - when rs btw data are broken, causing inconsistencies and errors.

Data loss : accidental deletion / overwriting of imp data, resulting in permanent loss of info.

Data security Breaches