

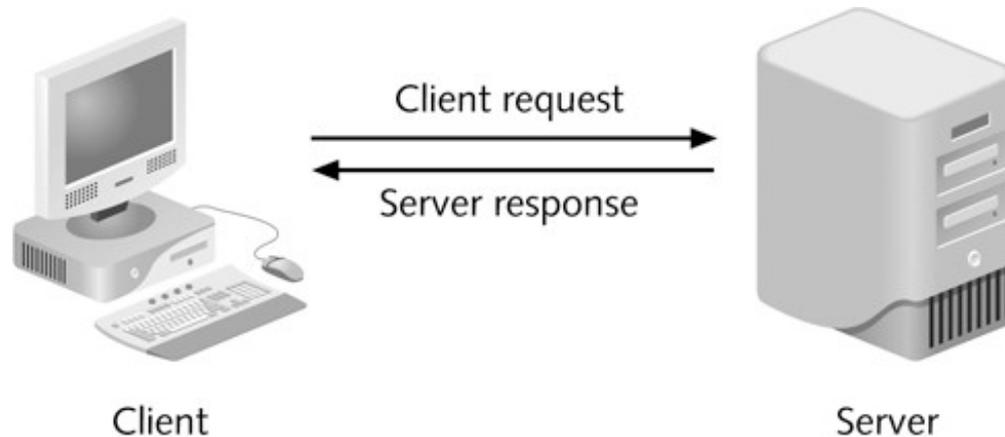


Basic 1

Table of Contents

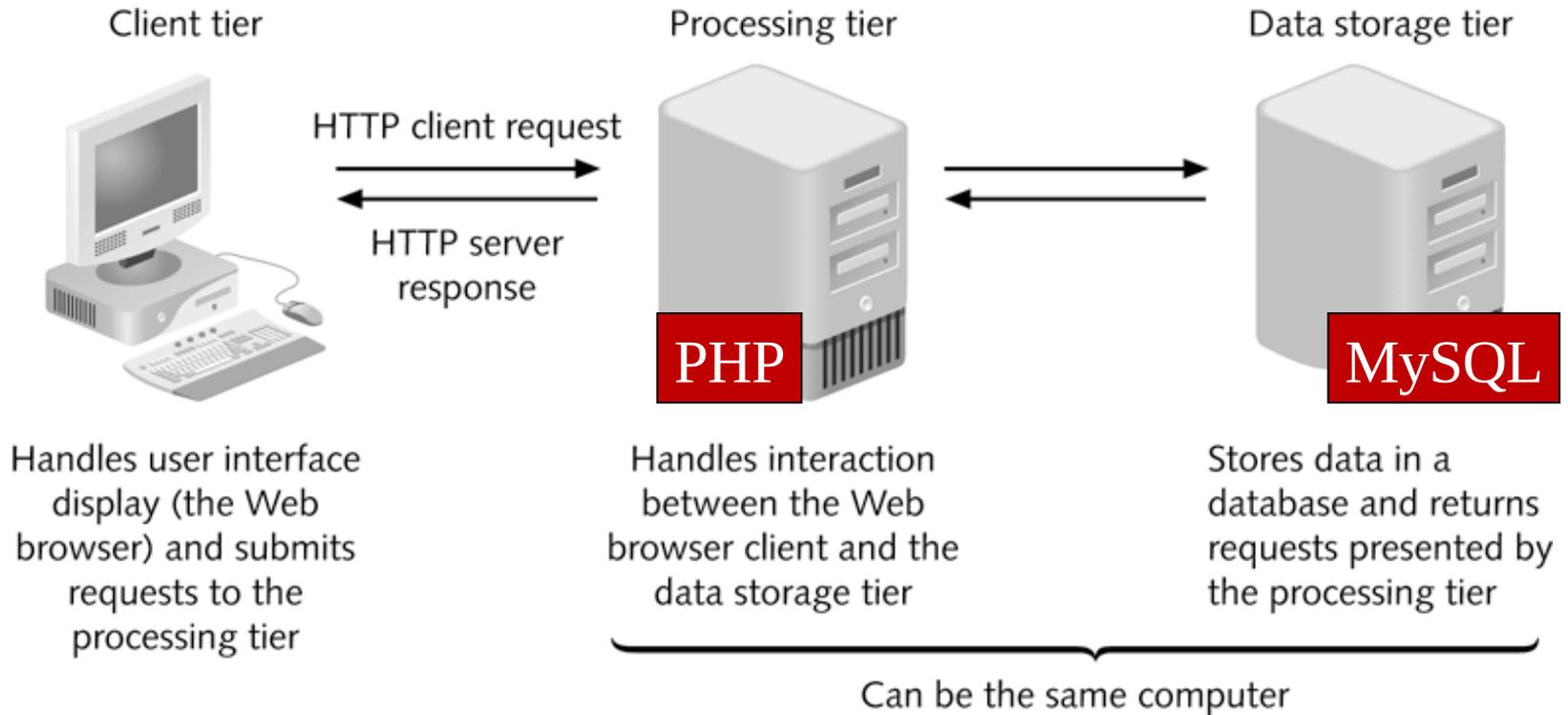
- Web programming architecture
 - PHP introduction
 - Variables
 - Operators
 - Conditional statements
 - Looping
-

Client/Server 2-Tier Architecture



- Web Browser Client (“front end”):
 - ❑ interface to the user
 - ❑ Gathers information from the user, submits it to a server, and presents the results returned from the server
- Web Server (“back end”):
 - ❑ Managing the request or serving
 - ❑ Responsible for data storage and management

Client/Server 3-Tier Architecture



The design of a three-tier client/server system

PHP Introduction

- PHP is a powerful tool for making dynamic and interactive Web pages.
 - PHP is a powerful tool for making dynamic and interactive Web pages.
 - PHP: created by Rasmus Lerdorf (1994)
 - Current version: PHP 5
-

PHP Introduction

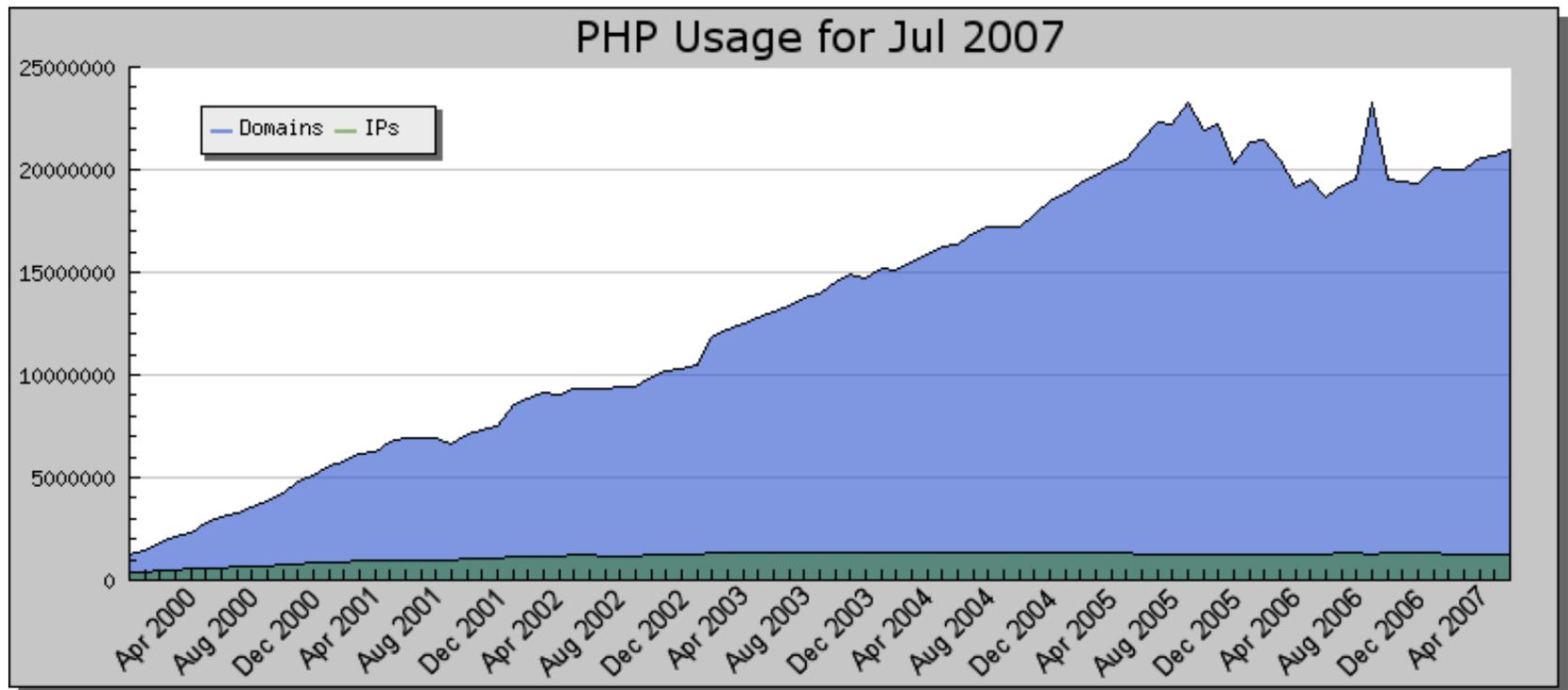
- Cross Platform
 - Servers: Apache, MS IIS,...
 - OS: UNIX, Mac OSX, Windows,...
 - DBs: DB2, MySQL, ORACLE, SQL SERVER,...
 - Platform: FREE
 - Development Tools: FREE (Netbeans, Eclipse, PHPEditor...)
-

Growth of PHP

PHP: 20,917,850 domains, 1,224,183 IP addresses

Source: [Netcraft](#)

Addresses <http://www.php.net/usage.php>

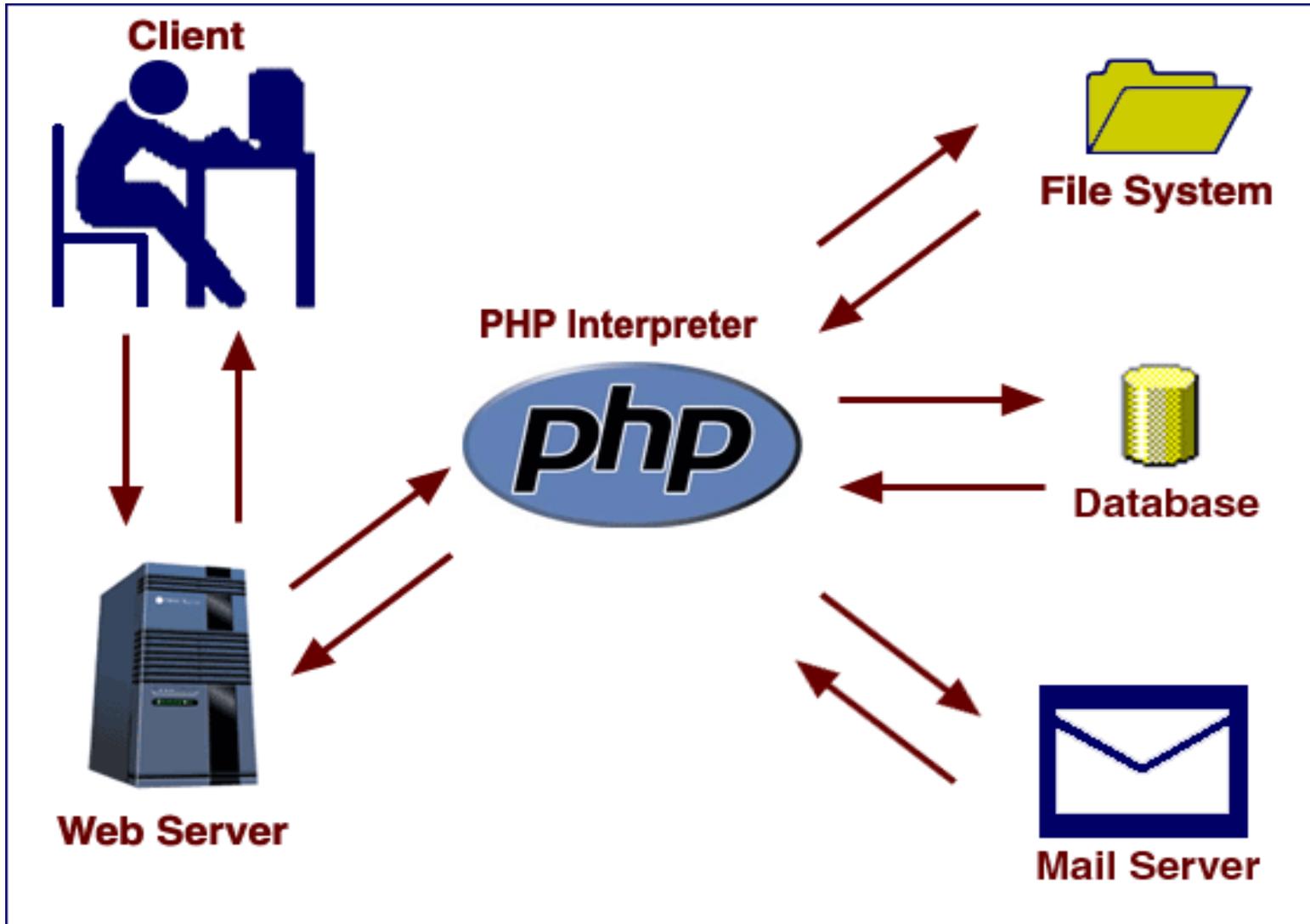


Getting Started

Tag	Ending tag	Notes
<?php	?>	Preferred method as it allows the use of PHP with XHTML
<?	?>	Not recommended. Easier to type, but has to be enabled and may conflict with XML
<script language="php">	?>	Always available, best if used when FrontPage is the HTML editor
<%	%>	Not recommended. ASP tags support was added in 3.0.4

```
<html><head>
<title>My First PHP Page</title>
</head>
<body>
<?php
echo "Hello World!";
?>
</body></html>
```

How PHP Works



phpinfo()

- The `phpinfo()` function shows the php environment
 - Use this to read system and server variables, setting stored in `php.ini`, versions, and modules
 - Notice that many of these data are in arrays
 - This is the first script you should write...
-

PHP Variables

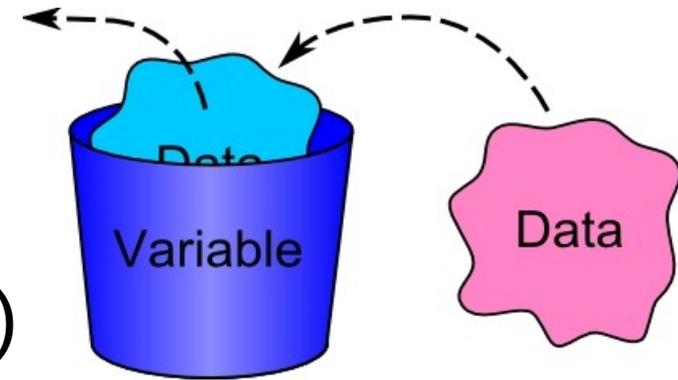
- Variables are used for storing values, such as numbers, strings or function results, so that they can be used many times in a script.
 - All variables in PHP start with a \$ sign symbol.
 - Variables are assigned using the assignment operator "="
-

PHP Variables example

```
<?php
$var1 = 'PHP'; // Assigns a value of 'PHP' to $var1
$var2 = 5; // Assigns a value of 5 to $var2
$var3 = $var2 + 1; // Assigns a value of 6 to $var3
$var2 = $var1; // Assigns a value of 'PHP' to $var2
echo $var1; // Outputs 'PHP'
echo $var2; // Outputs 'PHP'
echo $var3; // Outputs '6'
echo $var1 . ' rules!'; // Outputs 'PHP rules!'
echo "$var1 rules!"; // Outputs 'PHP rules!'
echo '$var1 rules!'; // Outputs '$var1 rules!'
?>
```

Variable Naming Rules

- start with a letter or an underscore "_"
- only contains (a-z, A-Z, 0-9, _)
- in case of more than one word, it should be separated with an underscore (`$my_string`), or with capitalization (`$myString`)
- case sensitive
- Not need to be declared before being set.



PHP Operators

- Mathematical operators **+**, **-**, **/**, *****, **++**, **--**
- Comparison operators **==**, **>**, **<**, **>=**, **<=**, **!=**, **<>**
- Assignment operators
 - **x+=y** **↔** **x=x+y**
- Logical operators **&&**, **||**, **!**, **xor**

+	-	×	÷
±	=	*	:
±	≠	±	/
>	<	≥	≤

PHP Mathematical Operators

Operator	Description	Example	Result
+	Addition	$x=2$ $x+2$	4
-	Subtraction	$x=2$ $5-x$	3
*	Multiplication	$x=4$ $x*5$	20
/	Division	$15/5$ $5/2$	3 2.5
%	Modulus (division remainder)	$5\%2$ $10\%8$ $10\%2$	1 2 0
++	Increment	$x=5$ $x++$	$x=6$
--	Decrement	$x=5$ $x--$	$x=4$

PHP Assignment Operators

Operator	Example	Is The Same As
<code>=</code>	<code>x=y</code>	<code>x=y</code>
<code>+=</code>	<code>x+=y</code>	<code>x=x+y</code>
<code>-=</code>	<code>x-=y</code>	<code>x=x-y</code>
<code>*=</code>	<code>x*=y</code>	<code>x=x*y</code>
<code>/=</code>	<code>x/=y</code>	<code>x=x/y</code>
<code>.=</code>	<code>x.=y</code>	<code>x=x.y</code>
<code>%=</code>	<code>x%=y</code>	<code>x=x%y</code>

PHP Assignment Operators

Operator	Description	Example
<code>==</code>	is equal to	<code>5==8</code> returns false
<code>!=</code>	is not equal	<code>5!=8</code> returns true
<code><></code>	is not equal	<code>5<>8</code> returns true
<code>></code>	is greater than	<code>5>8</code> returns false
<code><</code>	is less than	<code>5<8</code> returns true
<code>>=</code>	is greater than or equal to	<code>5>=8</code> returns false
<code><=</code>	is less than or equal to	<code>5<=8</code> returns true

Conditional Statements

■ If/ else

- After each statement stands (;)
- If more than one command should be executed, use curly braces { }

```
<?php
$today = date("D");
if ($today == "Fri")
    echo "Have a nice weekend!";
elseif ($today == "Sun")
    echo "Have a nice Sunday!";
else
    echo "Have a nice day!";
?>
```

Conditional Statements

■ Switch / break

- Used for choosing one possibility from multiple cases

```
switch ($x)
{
case 1:
    echo "Number 1";
    break;
case 2:
    echo "Number 2";
    break;
default:
    echo "No number between 1 and 3";
}
```

PHP Looping

■ while

- loops repeat until final condition is reached

```
$i = 1;
while ($i<=10) {
    echo $i;
    $i++;
}
```



■ do...while

- kind of reversed while function
- **Do** { code to be executed;}
- **While**(final condition);

PHP Looping

■ for

- Repeats the specific part of code so many times we choose

```
for ($i=1; $i<=10; $i++)
```

Initial condition
description

final condition

running

