Chp:4

The Server Tier

Prepared by: Er. Simanta Kasaju

Web Server Concept

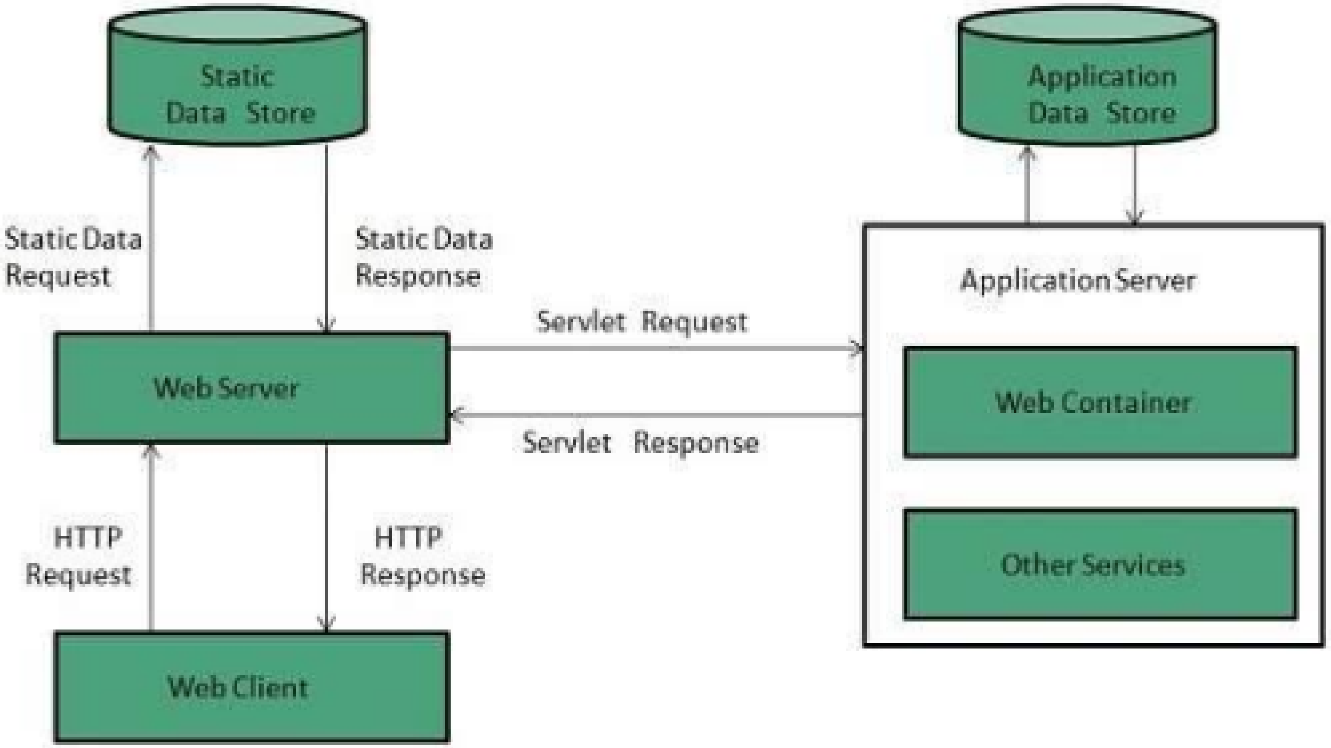
- Web server is a computer where the web content is stored.
- Web server is a computer system which receives the HTTP request via TCP, which are used to distribute information on www.
- Commonly a web browser or web crawler initiates communication by making HTTP request for a specific resource and the server response with the content of that resource or an error message if unable to do so.
- Basically web server is used to host the web sites but there exists other web servers also such as gaming, storage, FTP, email etc.
- Web site is collection of web pages while web server is a software that respond to the request for web resources.

Cont.

- The main function of a web server is to store, process and deliver web pages to clients. The communication between client and server takes place using the HTTP.
- Pages delivered are most frequently HTML documents, which may include images, stylesheets and scripts in addition to text content.
- Web servers are not always used for serving the www. They can also be found embedded in devices such as printers ,routers, webcams and serving only a local network.

Web Server Working

- Web server respond to the client request in either of the following two ways:
- Sending the file to the client associated with the requested URL.
- Generating response by invoking a script and communicating with database



Key Points
•When client sends request for a web page, the web
server search for the requested page if requested page
is found then it will send it to client with an HTTP response.
•If the requested web page is not found, web server will
the send an HTTP response: Error 404 Not found.
•If client has requested for some other resources then the web server will contact to the application server and data store to construct the HTTP response.

Examples: Following table describes the most leading web servers available today:

1 Apache HTTP Server

This is the most popular web server in the world developed by the Apache Software Foundation. Apache web server is an open source software and can be installed on almost all operating systems including Linux, UNIX, Windows, FreeBSD, Mac OS X and more. About 60% of the web server machines run the Apache Web Server.

2. Internet Information Services (IIS)

The Internet Information Server (IIS) is a high performance Web Server from Microsoft. This web server runs on Windows NT/2000 and 2003 platforms (and may be on upcoming new Windows version also). IIS comes bundled with Windows NT/2000 and 2003; Because IIS is tightly integrated with the operating system so it is relatively easy to administer it.

3. Lighttpd

The lighttpd, pronounced lighty is also a free web server that is distributed with the FreeBSD operating system. This open source web server is fast, secure and consumes much less CPU power. Lighttpd can also run on Windows, Mac OS X, Linux and Solaris operating systems.

4. Sun Java System Web Server

This web server from Sun Microsystems is suited for medium and large web sites. Though the server is free it is not open source. It however, runs on Windows, Linux and UNIX platforms. The Sun Java System web server supports various languages, scripts and technologies required for Web 2.0 such as JSP, Java Servlets, PHP, Perl, Python, and Ruby on Rails, ASP and Coldfusion etc.

5. Jigsaw Server

Jigsaw (W3C's Server) comes from the World Wide Web Consortium. It is open source and free and can run on various platforms like Linux, UNIX, Windows, and Mac OS X Free BSD etc. Jigsaw has been written in Java and can run CGI scripts and

PHP rorams.

Load limits

- A web server has defined load limits. It can handle only a limited number of concurrent client connections and it can serve only a certain maximum number of requests per second depending on.
- Its own setting
- Hardware and software limitation of OS
- HTTP request type
- Content origin(static or dynamic) etc.

When a web server is near to or over its limits ,it becomes unresponsive.

Overload cause

At any time web servers can be overloaded because of

- Too much web traffic: when millions of clients connecting to the web site in a short interval overloaded can occur.
- Distributed Denial of service attacks
- Computers worms that sometimes cause abnormal traffic because of millions of infected computers
- Web servers partial unavailability. This can happen because of required or urgent maintenance or upgrade, hardware or software failures, back end(eg database) failures etc, In these cases the remaining web servers get too much traffic and become overloaded.
- Internet connection slowdowns, so that client requests are served more slowly and the number of connections increases so much that server limits are reached

Anti-overloaded technique

To partially overcome load limits and to prevent overload we can use techniques like:

- Managing network traffic by using:

• Firewalls to Block unwanted traffic coming from bad IP sources, • HTTP traffic managers to drop , redirect or rewrite requests having bad HTTP patterns

• Bandwidth management and traffic shaping

- Deploying web cache techniques

• Use different domains to serve different content (static and dynamic) by separate Web servers

• Add more hardware resources

• Use many web servers

Using control flow to control dynamic content generation

- Dynamic content in the context of HTML and the World Wide Web refers to website content that constantly or regularly changes based on user interactions, timing and other parameters that determine what content is delivered to the user.
- This means that the content of the site may differ for every user because of different parameters.
- Facebook is an excellent example of a site that delivers dynamic content, as every user gets different content based on their friends and social interactions, although the layout stays generally the same.
- This could be different text, video, advertisements or even an entirely different layout and color.
- Any element in a page which contains movement and can change over time may be considered as dynamic content.

There are two ways to provide dynamic content:

1. Using client-side scripting and frameworks such as JavaScript, AJAX and Bootstrap to change the UI behavior within a specific Web page in response to specific user actions and timings. This gives dynamic

behavior to the UI presentation. This is normally used in Web applications and interactive

websites.

Cont..

- 2. Using server-side scripting and processing to dynamically form both the layout and content to be delivered to the user based on parameters such as the user’s location, time of day, browser being used or user preferences.
- Some good examples of this are social networking sites and content delivery sites.
- Social networking sites such as Facebook and Twitter provide entirely different content per user because of the difference of their connections and subscribed services, while sites like YouTube and Amazon provide dynamic content based on user-specific preferences based on past purchases or views; the server gives suggested items or content that the user may like based on historical data.

What is PHP?

- PHP is an acronym for "PHP: Hypertext Preprocessor"
- PHP is a widely-used, open source scripting language
- PHP scripts are executed on the server
- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code is executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension ".php"

What Can PHP Do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can be used to control user-access
- PHP can encrypt data
- With PHP you are not limited to output HTML. You can output images, PDF files,

and even Flash movies. You can also output any text, such as XHTML and XML.

Basic PHP Syntax

- A PHP script can be placed anywhere in the document.
- A PHP script starts with <?php and ends with ?>:
-
```
<?php
// PHP code goes here
?>
```
- The default file extension for PHP files is ".php".
- A PHP file normally contains HTML tags, and some PHP scripting code.
- Below, we have an example of a simple PHP file, with a PHP script that uses a built-in PHP function "echo" to output the text "Hello World!" on a web page:

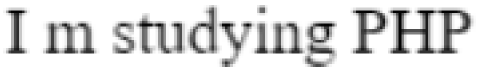
- Example

```
<!DOCTYPE html> <html>
<body>

<h1>My first PHP  page</h1>

<?php
echo "Hello World!"; ?>

</body>
</html>
```

sample

```
<html>
<body>
<?php
echo "I m studying php"; ?>
</body>
</html>
```

```
<?php
$txt = "PHP";
echo "I m studying $txt";
?>
```



```
I m studying PHP
```

Comments in PHP

- A comment in PHP code is a line that is not executed as a part of the program. Its only purpose is to be read by someone who is looking at the code.
- Comments can be used to:
- Let others understand your code
- Remind yourself of what you did - Most programmers have experienced coming back to their own work a year or two later and having to re-figure out what they did. Comments can remind you of what you were thinking when you wrote the code
- PHP supports several ways of commenting:

```
// This is a single-line comment

# This is also a single-line comment
```

```
/*
This is a multiple-lines comment block that
spans over multiple
lines
*/
```

PHP Variables

- Variables are "containers" for storing information
- In PHP, a variable starts with the $ sign, followed by the name of the

variable: <?php

```
$txt = "Hello  world!";
$x = 5;
$y = 10.5;
?>
```

PHP Variables

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume).

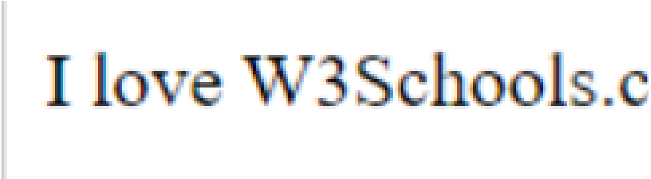
Rules for PHP variables:

- A variable starts with the $ sign, followed by the name of the variable • A variable name must start with a letter or the underscore character • A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive ($age and $AGE are two different variables)

Output Variables

- The PHP echo statement is often used to output data to the screen. • The following example will show how to output text and a variable:

```
<?php
$txt = "W3Schools.com";
echo "I love $txt!";
?>
```



```
I love W3Schools.c
```

```
<?php
$txt = "W3Schools.com";
echo "I love " . $txt .
"!"; ?>
```

```
<?php
$x = 5;
$y = 4;
echo $x + $y; ?>
9
```

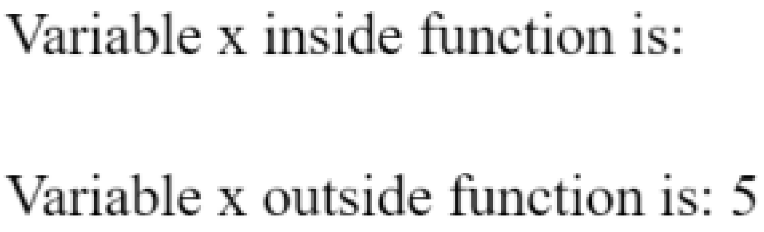
PHP Variables Scope

- In PHP, variables can be declared anywhere in the script.
- The scope of a variable is the part of the script where the variable can be referenced/used.
- PHP has three different variable scopes:
- local
- global
- static

Global and Local Scope

- A variable declared **outside** a function has a GLOBAL SCOPE and can only be accessed outside a function:

```
<?php
$x = 5; // global scope

function myTest() {
  // using x inside this function will
generate an error
  echo "<p>Variable x inside function
  is:
$x</p>";
}
myTest();

echo "<p>Variable x outside function is:
$x</p>";
?>
```

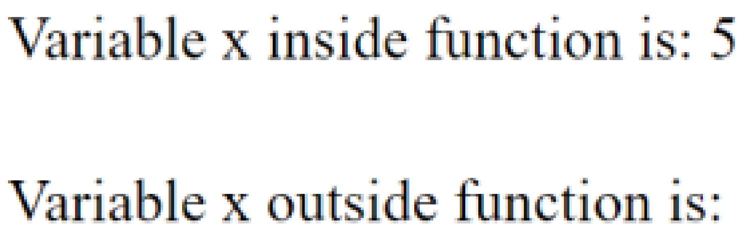


```
Variable x inside function is:

Variable x outside function is: 5
```

LOCAL SCOPE

- A variable declared **within** a function has a LOCAL SCOPE and can only be accessed within that function:

```
<?php
function myTest() {
  $x = 5; // local scope
  echo "<p>Variable x inside
function is: $x</p>";
}
myTest();

// using x outside the function will
generate an error
echo "<p>Variable x outside function
is: $x</p>";
?>
```



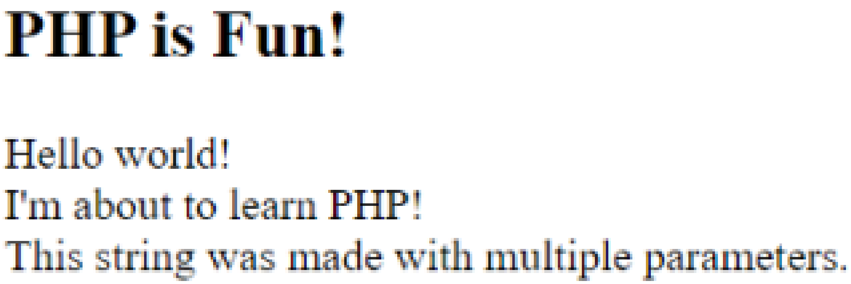
```
Variable x inside function is: 5

Variable x outside function is:
```

PHP echo and print Statements

- `echo` and `print` are more or less the same. They are both used to output data to the screen.
- The differences are small: `echo` has no return value while `print` has a return value of 1 so it can be

used in expressions. echo can take multiple parameters (although such usage is rare) while print can take one argument. echo is marginally faster than print.

```
<?php
echo "<h2>PHP is
Fun!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn
PHP!<br>";
echo "This ", "string ",
"was
", "made ", "with multiple
parameters.";
?>
```



```
PHP is Fun!

Hello world!
I'm about to learn PHP!
This string was made with multiple parameters.
```

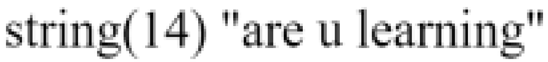
PHP Data Types

- Variables can store data of different types, and different data types can do different things.

- PHP supports the following data types:
- String
- Integer
- Float (floating point numbers - also called double)
- Boolean
- Array
- Object
- NULL
- Resource

PHP String

- A string is a sequence of characters, like "Hello world!".
- A string can be any text inside quotes. You can use single or double quotes:



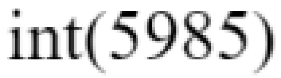
```
string(14) "are u learning"
```

```php
<?php
$A="are u learning";
echo var_dump($A);
?>
```

PHP Integer

- An integer data type is a non-decimal number between -2,147,483,648 and 2,147,483,647.
- Rules for integers:
- An integer must have at least one digit
- An integer must not have a decimal point
- An integer can be either positive or negative

- Integers can be specified in: decimal (base 10), hexadecimal (base 16), octal (base 8), or binary (base 2) notation

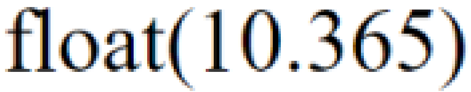


```
int(5985)
```

```php
<?php
$x = 5985;
var_dump($x);
?>
```

PHP Float

- A float (floating point number) is a number with a decimal point or a number in exponential form.
- In the following example $x is a float. The PHP var_dump() function returns the data type and value:



```
float(10.365)
```

```
<?php
$x = 10.365;
var_dump($x);
?>
```

PHP Boolean

- A Boolean represents two possible states: TRUE or FALSE.
- $x = true;
  $y = false;

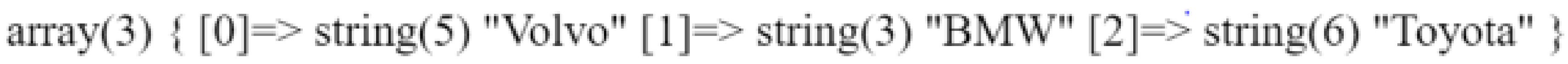
- Booleans are often used in conditional testing.

```
$A=true;
echo var_dump($A);
```

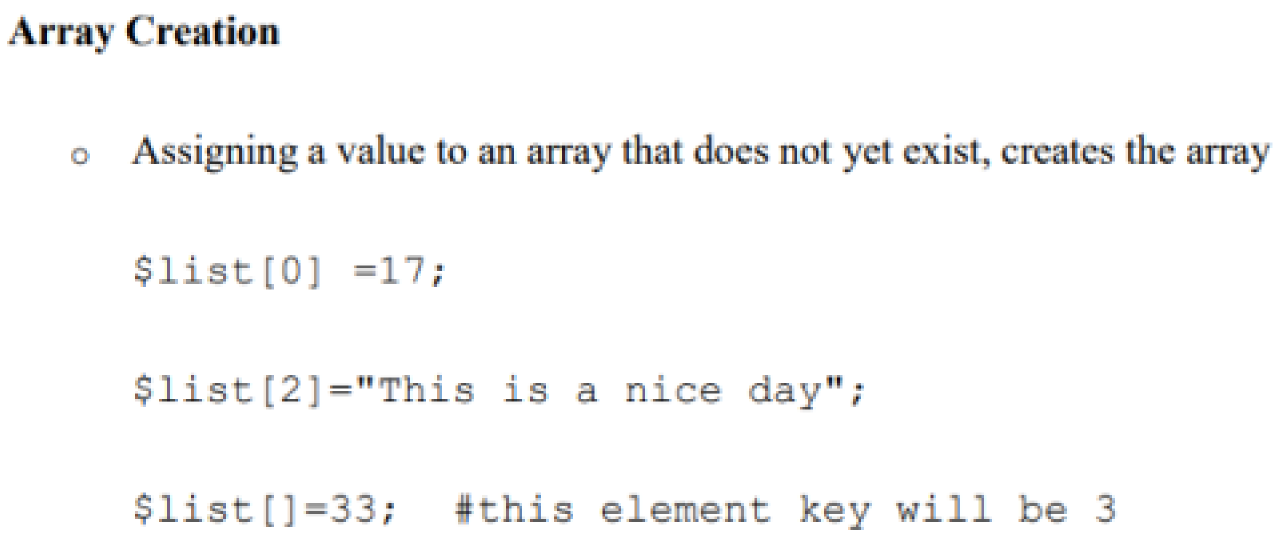
PHP Array

- An array stores multiple values in one single variable.
- In the following example $cars is an array. The PHP var_dump() function returns the data type and value:

```
<?php
$cars =
array("Volvo","BMW","Toy
ota");
var_dump($cars);
?>
```



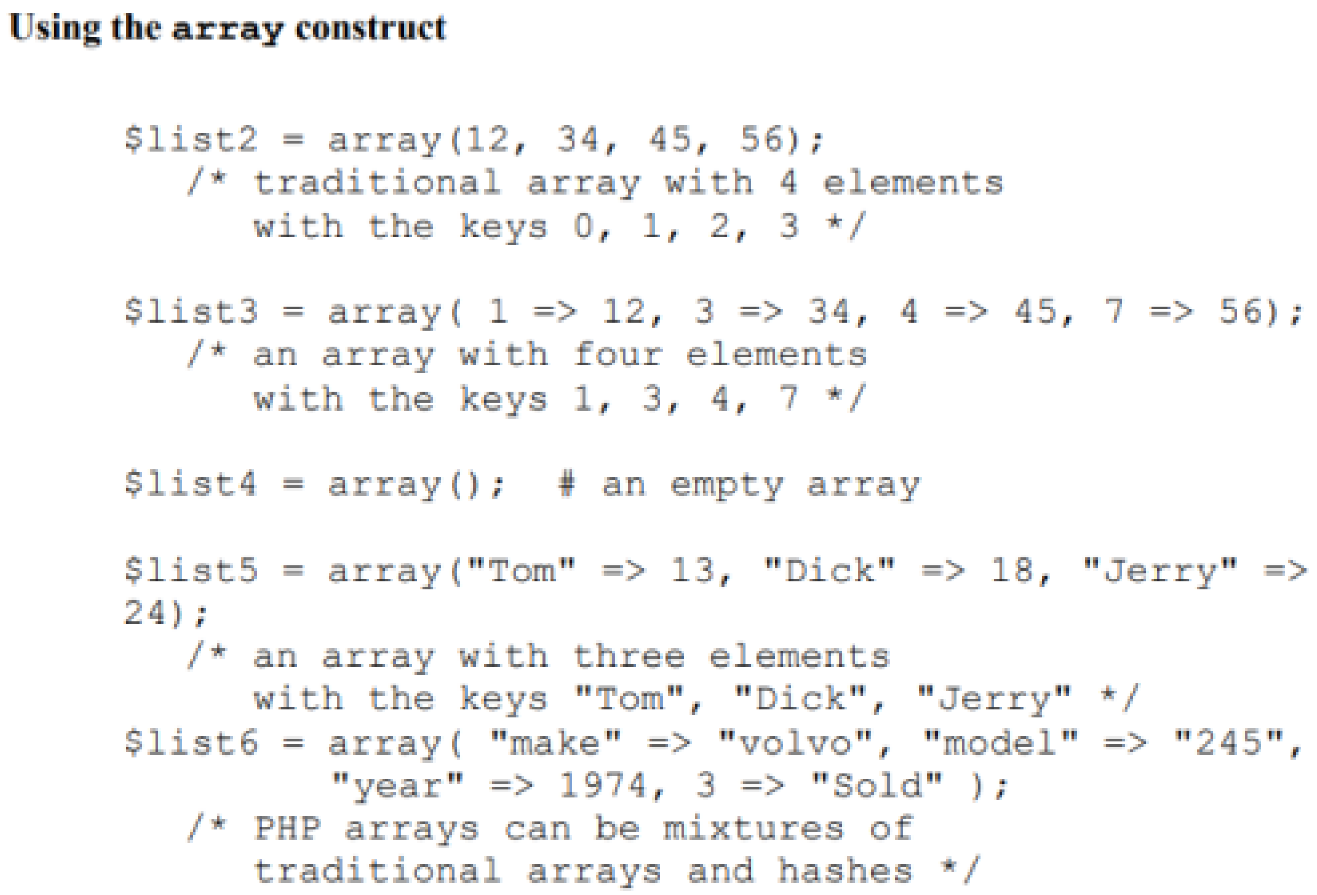
```
array(3) { [0]=> string(5) "Volvo" [1]=> string(3) "BMW" [2]=> string(6) "Toyota" }
```



Array Creation

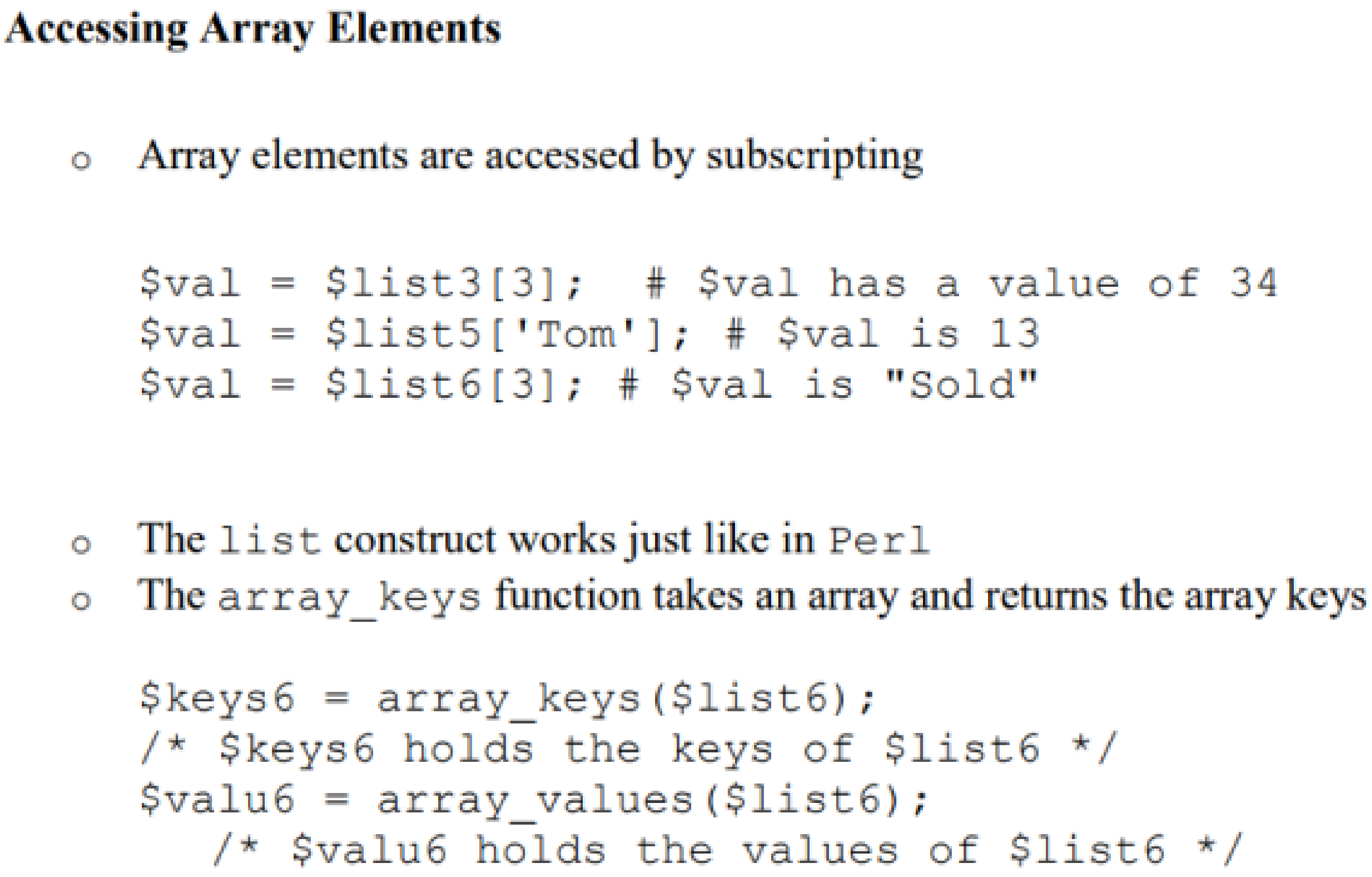
- Assigning a value to an array that does not yet exist, creates the array

```
$list[0] =17;

$list[2]="This is a nice day";

$list[]=33;  #this element key will be 3
```



Using the `array` construct

```
$list2 = array(12, 34, 45, 56);
   /* traditional array with 4 elements
      with the keys 0, 1, 2, 3 */

$list3 = array( 1 => 12, 3 => 34, 4 => 45, 7 => 56);
   /* an array with four elements
      with the keys 1, 3, 4, 7 */

$list4 = array();  # an empty array

$list5 = array("Tom" => 13, "Dick" => 18, "Jerry" =>
24);
   /* an array with three elements
      with the keys "Tom", "Dick", "Jerry" */
$list6 = array( "make" => "volvo", "model" => "245",
         "year" => 1974, 3 => "Sold" );
   /* PHP arrays can be mixtures of
      traditional arrays and hashes */
```



Accessing Array Elements

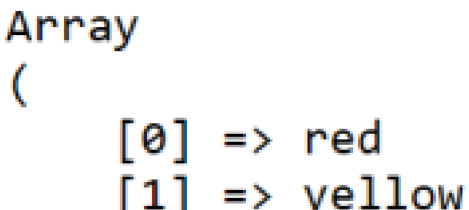
- Array elements are accessed by subscripting

```
$val = $list3[3];  # $val has a value of 34
$val = $list5['Tom']; # $val is 13
$val = $list6[3]; # $val is "Sold"
```

- The `list` construct works just like in `Perl`
- The `array_keys` function takes an array and returns the array keys

```
$keys6 = array_keys($list6);
/* $keys6 holds the keys of $list6 */
$valu6 = array_values($list6);
   /* $valu6 holds the values of $list6 */
```

```
 <?php
//with array keyword
   $color = array("red", "yellow",
   "green"); echo "<pre>";
```



```
Array
(
    [0] => red
    [1] => yellow
    print_r($color);
   // echo "<br>";
   echo "</pre>";

   //without array keyword
   echo "<pre>";
   $vehicle = ["car", "truck", "bus"];
    print_r($vehicle);
   echo "</pre>";

   //with loop
   $hobby=array("music","cricket","dance","travel")
   ; for($i=0;$i<4;$i++)
   {
      echo $hobby[$i] . "<br>";
   }
?>
```

Type of Array in PHP:

In PHP, there are three types of arrays:

i. **Indexed arrays** - Arrays with a numeric index
ii. **Associative arrays** - Arrays with named keys
iii. **Multidimensional arrays** - Arrays containing one or more arrays

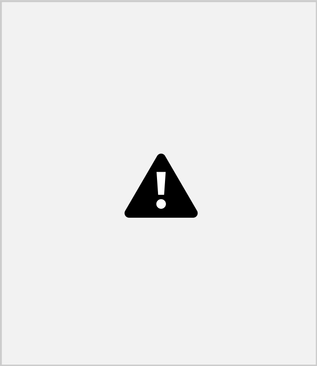
```
foreach ($array as $value)                      { code to be executed;
```

```
     }

<?php
$colors = array("red", "green", "blue", "yellow");

foreach ($colors as $value) {
  echo "$value <br>";
}
?>
```

```
$sub = array("dbms", "Wt", "LC", "NM", "Java");
  foreach($sub as $value)
  {
      echo $value ."<br>";
  }
```

```
echo "<ul>";
```

```
    $sub = array("A"=>"dbms",
"B"=>"Wt","C"=> "LC","D"=> "NM","E"=>
"Java");
   foreach($sub as $x=> $value)
   {
       echo "<li>$x= $value </li>";
   }
```



```
<?php
   $Family = [
[1,"simanta","kasaju",28],
[2,"Prashant","kasaju",32],
[3,"shashi","kasaju",45],
[4,"salina","kasaju",34]
```

```
    ];
    echo "<table border='2px'>";

 for($i=0;$i<4;$i++){
       echo "<tr>";
    for($j=0;$j<4;$j++)
    {
           echo "<td>" .$Family[$i][$j].
    "</td>"; }
       echo "</tr>";
 }
?>
```

Using foreach

```
   foreach($Family as
 $value1){ echo "<tr>";
foreach($value1 as
$value2) {
   echo "<td>" .$value2.
"</td>"; }
echo "</tr>";
}
```

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
sort($cars);

$clength = count($cars);
for($x = 0; $x < $clength; $x++) {
  echo $cars[$x];
  echo "<br>";
}
?>
```

```
$original = array("Simanta"=> 29, "Zunet"
```

```
<?php
```

```
=>  20, "Prince" =>
"Charley","Apple" => 40, "Elaphent" =>
"hobby" );
?>
<h4>Original Array</h4>
<?php
foreach ($original as $key => $value
) print("[$key] => $value <br />");
$new = $original;
sort($new);
?>
<h4>Array Sorted by
<tt>sort</tt></h4> <?php
foreach ($new as $key => $value )
print("[$key] => $value <br />");
$new = $original;
sort($new, SORT_NUMERIC);
?>
<h4>Array Sorted by <tt>sort,
SORT_NUMERIC</tt></h4>

    foreach ($new as $key => $value )
```

```
print("[$key] => $value <br />");
$new = $original;
rsort($new);
?>
<h4>Array Sorted by
<tt>rsort</tt></h4> <?php
foreach ($new as $key => $value )
print("[$key] => $value <br />");
$new = $original;
asort($new);
?>
<h4>Array Sorted by
<tt>asort</tt></h4> <?php
foreach ($new as $key => $value )
print("[$key] => $value <br />");
$new = $original;
arsort($new);
?>
<h4>Array Sorted by
<tt>arsort</tt></h4> <?php
foreach ($new as $key => $value
) print("[$key] => $value <br
/>"); ?>
```

PHP Object

- Classes and objects are the two main aspects of object-oriented programming.
- A class is a template for objects, and an object is an instance of a class.
- When the individual objects are created, they inherit all the properties and behaviors from the class, but each object will have different values for the properties.

```
<?php
class Car {
  public $color;
  public $model;
  public function __construct($color, $model) {
    $this->color = $color;
    $this->model = $model;
  }
  public function message() {
    return "My car is a " . $this->color . " " . $this->model .
    "!";
  }
}

$myCar = new Car("black", "Volvo");
echo $myCar -> message();
echo "<br>";
$myCar = new Car("red", "Toyota");
echo $myCar -> message();
?>
```

PHP NULL Value

- Null is a special data type which can have only one value: NULL.
- A variable of data type NULL is a variable that has no value assigned to it. • **Tip:** If a variable is created without a value, it is automatically assigned a value of NULL. • Variables can also be emptied by setting the value to NULL:

```
<!DOCTYPE html>
<html>
<body>
<?php
$x = null;
var_dump($x); ?>
</body>
</html>
NULL
```

PHP Resource

- The special resource type is not an actual data type. It is the storing of a reference to functions and resources external to PHP.
- A common example of using the resource data type is a database call.
- We will not talk about the resource type here, since it is an advanced topic.

```
<?php
function marks($score)
{
   $sum = 0;
   foreach($score as $value)
   {
      $sum += $value;
   }
   return $sum;
}
$sim=["Nepali"=>40,"English"=>70,"math"=>100,"Opt"=>100]
; $total = marks($sim);
echo "you have scored $total marks out of
400"; ?>
```

PHP type cast

PHP Operators

Operators are used to perform operations on variables and values.

PHP divides the operators in the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Increment/Decrement operators
- Logical operators
- String operators
- Array operators
- Conditional assignment operators

PHP Arithmetic Operators

PHP Assignment Operators

PHP Comparison Operators

PHP Increment / Decrement Operators

PHP Logical Operators

PHP String Operators

PHP Array Operators(The PHP array operators are used to compare arrays.)

PHP Conditional Assignment Operators

PHP

Constants

- PHP

Constants

```
Case insensitive
define("GREETING", "Welcome to
W3Schools.com!", true);
```

- A constant is an identifier (name) for a simple value. The value cannot be changed during the script.
- A valid constant name starts with a letter or underscore (no $ sign before the constant name). •

Note: Unlike variables, constants are automatically global across the entire script.

Create a PHP Constant

- To create a constant, use the `define()` function. • Syntax
- define(*name*, *value*, *case-insensitive*) • Parameters:
- *name*: Specifies the name of the constant • *value*: Specifies the value of the constant

Case sensitive

```
<?php
define("GREETING", " Welcome to W3Schools.com!"); echo GREETING;
?>
```

```
Welcome to W3Schools.com!
```

- *case-insensitive*: Specifies whether the constant name should be case-insensitive. Default is false Er.

PHP Constant Arrays

- In PHP7, you can create an Array constant using the `define()` function.

```
<?php
define("cars", [ "Alfa
Romeo", "BMW",
"Toyota"
]);
```

```
echo cars[2]; ?>
```

```
Toyota
```

PHP Conditional Statements

- Very often when you write code, you want to perform different actions for different conditions. You can use conditional statements in your code to do this.
- In PHP we have the following conditional statements:
- `if` statement - executes some code if one condition is true
- `if...else` statement - executes some code if a condition is true and another code if that condition is false
- `if...elseif...else` statement - executes different codes for more than two

conditions • `switch` statement - selects one of many blocks of code to be executed

PHP - The if Statement

- The `if` statement executes some code if one condition is

true. • Syntax

- if (*condition*) {
    code to be executed if condition is true;
  }

```
<?php
$t = 40;

if ($t <"45") {
echo
"congratulation"; ?>
```

congratulation

PHP - The if...else Statement

- The `if...else` statement executes some code if a condition is true and another code if that condition is false.
- Syntax
- if (*condition*) {

code to be executed if condition is true; } else { code to be executed if condition is false; }

```
<?php
$t = 40;

if ($t >"45") {
echo "congratulation"; }
else
echo "sorry ";
?>
```

```
sorry
```

PHP - The if...elseif...else Statement

- The `if...elseif...else` statement executes different codes for more than two conditions.
- Syntax
- if (*condition*) {
code to be executed if this condition is true;
} elseif (condition) {
code to be executed if first condition is false and this condition is true; } else {
code to be executed if all conditions are false;
}

```
<?php
$t = 40;

if ($t >"90") { echo "A+"; }
else if($t <"85" && $t>"80"){
echo "B+ ";}
else
echo "fail";
?>
fail
```

The PHP switch Statement

- Use the `switch` statement to **select one of many blocks of code to be executed**.
- Syntax
- switch (*n*) {

```
  case label1:
    code to be executed if n=label1;
    break;
  case label2:
    code to be executed if n=label2;
    break;
  case label3:
    code to be executed if n=label3;
    break;
    ...
  default:
    code to be executed if n is different from all labels;
}
```

- This is how it works: First we have a single expression *n* (most often a variable), that is evaluated once. The value of the expression is then compared with the values for each case in the structure. If there is a match, the block of code associated with that case is executed. Use `break` to prevent the code from running into the next case automatically. The `default` statement is used if no match is found.

```
<?php
$favcolor = "red";

switch ($favcolor) {
case "red":
echo "Your favorite color is red!";
break;
case "blue":
echo "Your favorite color is blue!";
break;
case "green":
echo "Your favorite color is green!";
break;
default:
echo "Your favorite color is neither
red,  blue, nor green!";
}
?>
```

```
Your favorite color is red!
```

PHP Loops

- Often when you write code, you want the same block of code to run over and over again a certain number of times. So, instead of adding several almost equal code-lines in a script, we can use loops.
- Loops are used to execute the same block of code again and again, as long as a certain condition is true.

- In PHP, we have the following loop types:
- `while` - loops through a block of code as long as the specified condition is true
- `do...while` - loops through a block of code once, and then repeats the loop as long as the specified condition is true
- `for` - loops through a block of code a specified number of times
- `foreach` - loops through a block of code for each element in an array
- The following chapters will explain and give examples of each loop type.

The PHP while Loop

- The `while` loop executes a block of code as long as the specified condition is true. • Syntax
- while (*condition is true*) {

code to be executed;

}

```
<?php
$x = 1;
while($x <= 5) {
echo "The number is: $x
<br>"; $x++;
}
?>
```

```
The number is: 1 The number is: 2 The number is: 3 The number is: 4 The number is: 5
```

```
<?php
$x = 0;

while($x <= 100) {
echo "The number is: $x <br>";
$x+=10;
}
?>
```

```
The number is: 0 The number is: 10 The number is: 20 The number is: 30 The number is: 40 The number is: 50 The number is: 60 The number is: 70 The number is: 80 The number is: 90 The number is: 100
```

The PHP do...while Loop

- The `do...while` loop will always execute the block of code once, it will then check the condition, and repeat the loop while the specified condition is true.
- Syntax
- do {
  code to be executed;

} while (*condition is true*);

```
<?php
$x = 1;

do {
echo "The number
is: $x <br>";
$x++;
} while ($x <= 5);
?>
```

```
The number is: 1 The number is: 2 The number is: 3 The number is: 4 The number is: 5
```

```
<?php
$x = 6;

do {
echo "The number is: $x
<br>";
$x++;
} while ($x <= 5);
?>
```

```
The number is: 6
```

The PHP for Loop

- The `for` loop is used when you know in advance how many times the script should run. • Syntax

- for (*init counter; test counter; increment counter*) {
  code to be executed for each iteration;
  }
- Parameters:
- *init counter*: Initialize the loop counter value
- *test counter*: Evaluated for each loop iteration. If it evaluates to TRUE, the loop continues. If it evaluates to FALSE, the loop ends.
- *increment counter*: Increases the loop counter value

```
<?php
for ($x = 0; $x <= 10; $x++) {
echo "The number is: $x
<br>";
}
?>
```

The number is: 0 The number is: 1 The number is: 2 The number is: 3 The number is: 4 The number is: 5 The number is: 6 The number is: 7 The number is: 8 The number is: 9 The number is: 10

```
<?php
for ($x = 0; $x <= 100;
$x+=10) {
echo "The number is: $x
<br>";
}
?>
```

```
number is: 10 The number
is: 20 The number is: 30 The
number is: 40 The number
is: 50 The number is: 60 The
number is: 70 The number
is: 80 The number is: 90 The
number is: 100
```

```
The number is: 0 The
```

Session & State

- Web is stateless that means the web browser does not know who you are and what you do,because the HTTp address doesn’t maintain state.If user inserts some information and move to the next page that data will be lost and user would not able to retrive the information.so we need to store the information about user.

- Session provides that facility to store information on server memory. • A session is a way to store information(in variables) to be used across multiple pages.
- Session variables hold information about one single user and are available to all pages in one application.
- By default session variables last until the user closes the browser.

Cont..

- Unlike a cookie, the information is not stored on the users computer.
- We all know that the web uses the HTTP protocol and the HTTP protocol is a stateless protocol; in other words, when a client sends a request to the server, an instance of the page is created and the page

is converted to HTML format and then the server provides the response and then the instance of the page and the value of the control are destroyed.

- So if we have a requirement to store the values of the controls and pass them into another web form then a State Management Technique is used.

Cont..

- Session is a State Management Technique. A Session can store the value on the Server.
- It can support any type of object to be stored along with our own custom objects.

- A session is one of the best techniques for State Management because it stores the data as client-based, in other words the data is stored for every user separately and the data is secured also because it is on the server.

Cont..

Cont..

- State management using session is one of the best features, because it is secure, transparent from users, and we can store any kind of object in it.

- Along with these advantages, some times session can cause performance issues in high traffic sites because it is stored in server memory and clients read data from the server.

Advantages

- It helps maintain user state and data all over the application.

- It is easy to implement and we can store any kind of object eg. Database, dataset etc.
- Stores client data separately.
- Session is secure and transparent from the user.

Disadvantages

- As session is stored on the server memory, it is not advisable to use session state when there is a large volume of data.
- With the use of session state, it will affect the performance of memory,

because it is stored in server memory until you destroy the state

- The problem with sessions is that when you close your browser you also lose the session so, if you had a site requiring a login, this couldn't be saved as a session like it could as a cookie, and the user would be forced to re-login every time they visit.

Creating a session

- Simply by calling the session_start() function you can create a new session.Session variables are set with the PHP global: $_SESSION.

Syntax:

Session_start();

This is the initialization of session.Each and every page at the top script,we must initiate the session,if we need anhy session value.

Syntax:

$_SESSION['session_name]="value";

Example

```
• <?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>
```

```
<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat"; echo "Session
variables are set."; ?>
</body>
</html>
```

Note: The session_start() function must be the very first thing in your document.
Before any HTML tags.

Session variables are set.

Retrieving the value from session

- <?php
  session_start();
  ?><!DOCTYPE html> <html>
  <body>

<?php
We can retrive the value in any page if we store the value in session.Session

carries the values drom one page to   another

```
// Echo session variables that were set on previous page
echo "Favorite color is " . $_SESSION["favcolor"] .
".<br>"; echo "Favorite animal is " .
$_SESSION["favanimal"] . ".";
?>
</body> </html>
```

```
Favorite color is . Favorite animal is .
```

To view variables

- <?php

```
session_start();
?>
<!DOCTYPE html>
<html>
```

```
<body>
<?php
print_r($_SESSION); ?>
</body>
</html>
```

```
// to change a session variable, just
overwrite it
$_SESSION["favcolor"] = "yellow";
print_r($_SESSION);
?>
```

Modify a PHP Session Variable

```
<?php
```

Destroy a PHP Session

- <?php
  session_start();

```
?>
<!DOCTYPE html>
<html>
<body>
<?php
// remove all session variables
session_unset();
// destroy the session
session_destroy();
?>
</body>
</html>
```

To remove all global session variables and destroy the session,
use `session_unset()` and `session_destroy()`: