GROUND GURUS

"Effective center for teaching, learning, creating and development"

PHP Date and Time

- date() function is used to format a date and/or a time
- formats a timestamp to a more readable date and time

```
date(format,timestamp)
```

- *returns the current date/time of the server*

```
time()
```

- *returns the current unix timestamp*

PHP Include/Require Files

- The include (or require) statement takes all the text/code/markup that exists in the specified file and copies it into the file that uses the include statement.
- require will produce a fatal error (E_COMPILE_ERROR) and stop the script
- include will only produce a warning (E_WARNING) and the script will continue

PHP Include Files (Example)

```
<!DOCTYPE html>

<html>

    <body>

        <h1>Welcome to my home page!</h1>

        <p>Some text.</p>

        <p>Some more text.</p>

        <?php include 'footer.php';?>

    </body>

</html>
```

PHP Cookies

- often used to identify a user
- a small file that the server embeds on the user's computer
- Each time the same computer requests a page with a browser, it will send the cookie too
- With PHP, you can both create and retrieve cookie values.

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

Only the name parameter is required. All other parameters are optional.

PHP Cookies

- **name** - The name of the cookie.
- **value** - The value of the cookie.
- **expires** - The time the cookie expires.
- **path** -The path on the server in which the cookie will be available on.
- **domain** - The (sub)domain that the cookie is available to.
- **secure** - Indicates that the cookie should only be transmitted over a secure HTTPS connection from the client. When set to true, the cookie will only be set if a secure connection exists.
- **httponly** - When true the cookie will be made accessible only through the HTTP protocol. This means that the cookie won't be accessible by scripting languages, such as JavaScript.

PHP Creating a Cookie

```
$cookie_name = "user";

$cookie_value = "John Doe";

setcookie($cookie_name, $cookie_value, time() + ( 86400 * 30), "/");

setcookie("user", "John Doe", time() + ( 86400 * 30), "/");

// 86400 = 1 day
```

Note: The setcookie() function must appear BEFORE the <html> tag.

PHP Retrieving a Cookie

```php
if(!isset($_COOKIE[$cookie_name])) {
  echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
  echo "Cookie '" . $cookie_name . "' is set!<br>";
  echo "Value is: " . $_COOKIE[$cookie_name];
}
```

PHP Modifying a Cookie

```
$cookie_name = "user";

$cookie_value = "Peter Griffin";

setcookie($cookie_name, $cookie_value, time() + ( 86400 * 30), "/");

// 86400 = 1 day
```

Note: The setcookie() function must appear BEFORE the <html> tag.

PHP Deleting a Cookie

```
// set the expiration date to one hour ago

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

Note: The setcookie() function must appear BEFORE the <html> tag.

PHP Sessions

- a way to store information (in variables) to be used across multiple pages
- Unlike a cookie, the information is not stored on the users computer.

PHP Starting Sessions/Setting Session Variables

- A session is started with the session_start() function.
- Session variables are set with the PHP global variable: $_SESSION.

PHP Starting Sessions

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

PHP Setting Session Variables

```
<!DOCTYPE html>

<html>

      <body>

            <?php

                  // Set session variables

                  $_SESSION["favcolor"] = "green";

                  $_SESSION["favanimal"] = "cat";

                  echo "Session variables are set.";

            ?>

      </body>

</html>
```

PHP Getting Session Variables

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

PHP Getting Session Variables

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

PHP Getting Session Variables

```php
<?php
	session_start();
?>

// assuming the block of codes below is inside the <html> tag

<?php
	// to change a session variable, just overwrite it
	$_SESSION["favcolor"] = "yellow";
	echo "Favorite color is " . $_SESSION["favcolor"] . ".<br>";
?>
```

PHP Destroy a Session

```
<?php
     session_start();
?>
// assuming the block of codes below is inside the <html> tag
<?php
     // remove all session variables
     session_unset();

     // destroy the session
     session_destroy();
?>
```

PHP Callback Functions

- functions which is passed as an argument into another function

```php
<?php

function exclaim($str) {

  return $str . "! ";

}

function ask($str) {

  return $str . "? ";

}

function printFormatted($str, $format) {

  // Calling the $format callback function

  echo $format($str);

}
```

PHP Callback Functions

- functions which is passed as an argument into another function

```
// Pass "exclaim" and "ask" as callback functions to printFormatted()

printFormatted("Hello world", "exclaim");

printFormatted("Hello world", "ask");
```

PHP And JSON

- JSON stands for JavaScript Object Notation
- a syntax for storing and exchanging data
- used as a data format by any programming language

PHP json_encode()

- used to encode a value to JSON format

```
<?php
    $age = array("Peter"=>35, "Ben"=>37, "Joe"=>43);
    echo json_encode($age);

    $cars = array("Volvo", "BMW", "Toyota");
    echo json_encode($cars);
?>
```

PHP json_decode()

- used to encode a value to JSON format

```
<?php
      $jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';
      var_dump(json_decode($jsonobj));
      echo $jsonobj->Ben;
?>
<?php
      $jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';
      var_dump(json_decode($jsonobj, true));
      echo $jsobobj['Ben'];
?>
```

PHP File Handling

- an important part of any web application
- You often need to open and process a file for different tasks

PHP File Handling

Be careful when manipulating files!

When you are manipulating files you must be very careful.

You can do a lot of damage if you do something wrong. Common errors are: editing the wrong file, filling a hard-drive with garbage data, and deleting the content of a file by accident.

PHP readfile() Function

- reads a file and writes it to the output buffer.

```
<?php
    echo readfile("webdictionary.txt");
?>
```

PHP Open File - fopen()

- gives you more options than the readfile() function

```
<?php

      $myfile = fopen("webdictionary.txt", "r") or die("Unable to open file!");

      echo fread($myfile,filesize("webdictionary.txt"));

      fclose($myfile);

?>
```

PHP readfile() Function

Modes	Description
r	Open a file for read only. File pointer starts at the beginning of the file
w	Open a file for write only. Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file
a	Open a file for write only. The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist
x	Creates a new file for write only. Returns FALSE and an error if file already exists
r+	Open a file for read/write. File pointer starts at the beginning of the file
w+	Open a file for read/write. Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file
a+	Open a file for read/write. The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist
x+	Creates a new file for read/write. Returns FALSE and an error if file already exists

PHP Read Single Line - fgets()

- used to read a single line from a file

```php
<?php

	$myfile = fopen("webdictionary.txt", "r") or die("Unable to open file!");

	echo fgets($myfile);

	fclose($myfile);

?>
```

PHP Check End-Of-File - feof()

- checks if the "end-of-file" (EOF) has been reached
- useful for looping through data of unknown length.

```php
<?php
	$myfile = fopen("webdictionary.txt", "r") or die("Unable to open file!");
	// Output one line until end-of-file
	while(!feof($myfile)) {
	  echo fgets($myfile) . "<br>";
	}
	fclose($myfile);
?>
```

PHP File Create/Write

- The fwrite() function is used to write to a file.

```php
<?php

	$myfile = fopen("newfile.txt", "w") or die("Unable to open file!");

	$txt = "John Doe\n";

	fwrite($myfile, $txt);

	$txt = "Jane Doe\n";

	fwrite($myfile, $txt);

	fclose($myfile);

?>
```

PHP Form

```
<html>
    <body>
        <form action="welcome.php" method="POST">
            Name: <input type="text" name="name" /><br />
            E-mail: <input type="text" name="email" /><br />
            <input type="submit" />
        </form>
    </body>
</html>
```

PHP GET/POST

```
Welcome <?php echo $_POST["name"]; ?><br>

Your email address is: <?php echo $_POST["email"]; ?>
```

```
Welcome <?php echo $_GET["name"]; ?><br>

Your email address is: <?php echo $_GET["email"]; ?>
```

PHP Exceptions

- an object that describes an error or unexpected behaviour of a PHP script

```php
<?php

    function divide($dividend, $divisor) {

      if($divisor == 0) {

        throw new Exception("Division by zero");

      }

      return $dividend / $divisor;

    }



    echo divide(5, 0);

?>
```

PHP Exceptions

```php
<?php
    function divide($dividend, $divisor) {
      if($divisor == 0) {
        throw new Exception("Division by zero");
      }
      return $dividend / $divisor;
    }
    try {
      echo divide(5, 0);
    } catch(Exception $e) {
      echo "Unable to divide.";
    }
?>
```

PHP OOP

```php
<?php

    function divide($dividend, $divisor) {
      if($divisor == 0) {
        throw new Exception("Division by zero");
      }
      return $dividend / $divisor;
    }
    try {
      echo divide(5, 0);
    } catch(Exception $e) {
      echo "Unable to divide.";
    }

?>
```

Thank You!

Contact us

www.groundgurus.com
09171100312 | 09771673162
Facebook: @groundgurus
Instagram: @ggurus2016

GROUND GURUS