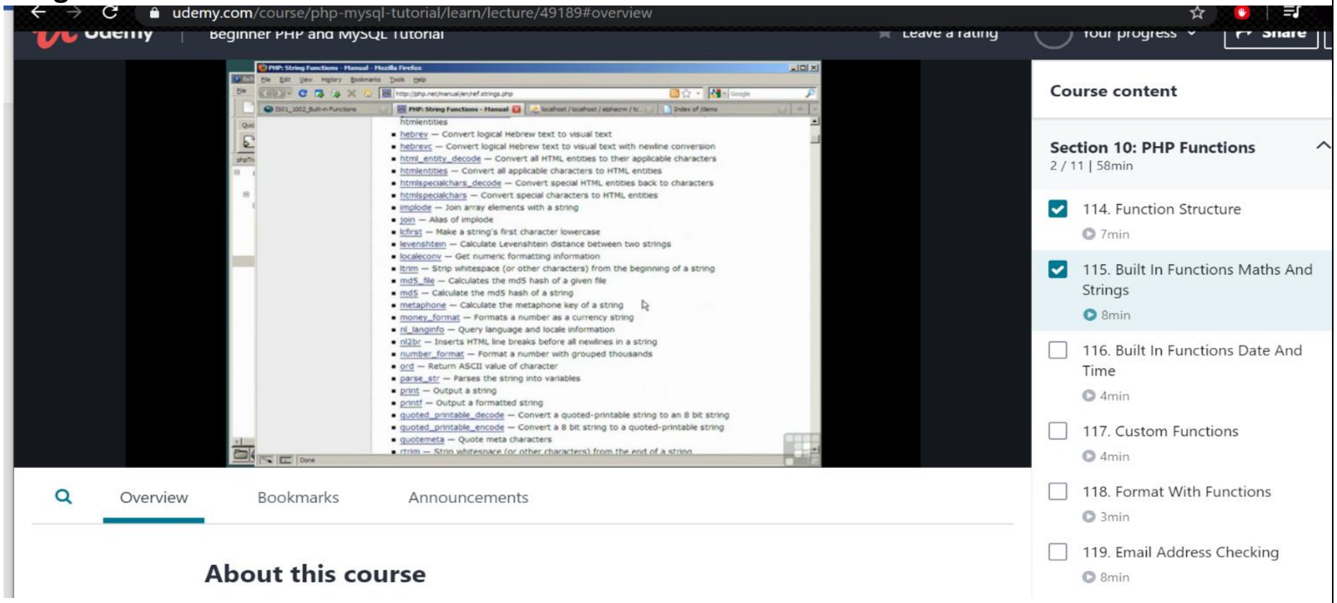
DAILY ASSESSMENT FORMAT

Date:	12th June 2020	Name:	Sushmitha R Naik
Course:	MYSQL	USN:	4AL17EC090
Topic:	• PHP Functions. • Using external files and images	Semester & Section:	6th sem 'B' sec
GitHub Repository:	Sushmitha_naik		

FORENOON SESSION DETAILS

Image of session



udemy.com/course/php-mysql-tutorial/learn/lecture/49189#overview

Beginner PHP and MySQL Tutorial

PHP: String Functions - Manual - Mozilla Firefox

http://php.net/manual/en/ref.strings.php

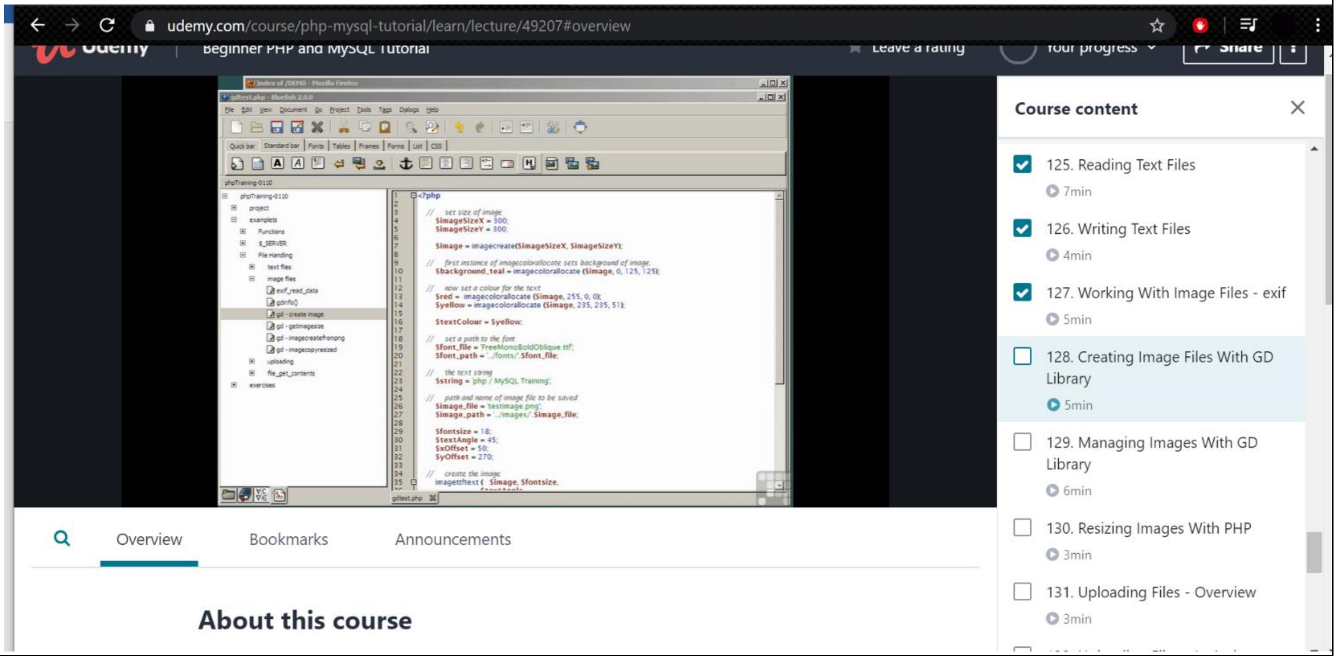
htmlentities
- hebrev — Convert logical Hebrew text to visual text
- hebrevc — Convert logical Hebrew text to visual text with newline conversion
- html_entity_decode — Convert all HTML entities to their applicable characters
- htmlentities — Convert all applicable characters to HTML entities
- htmlspecialchars_decode — Convert special HTML entities back to characters
- htmlspecialchars — Convert special characters to HTML entities
- implode — Join array elements with a string
- join — Alias of implode
- lcfirst — Make a string's first character lowercase
- levenshtein — Calculate Levenshtein distance between two strings
- localeconv — Get numeric formatting information
- ltrim — Strip whitespace (or other characters) from the beginning of a string
- md5_file — Calculates the md5 hash of a given file
- md5 — Calculate the md5 hash of a string
- metaphone — Calculate the metaphone key of a string
- money_format — Formats a number as a currency string
- nl_langinfo — Query language and locale information
- nl2br — Inserts HTML line breaks before all newlines in a string
- number_format — Format a number with grouped thousands
- ord — Return ASCII value of character
- parse_str — Parses the string into variables
- print — Output a string
- printf — Output a formatted string
- quoted_printable_decode — Convert a quoted-printable string to an 8 bit string
- quoted_printable_encode — Convert a 8 bit string to a quoted-printable string
- quotemeta — Quote meta characters
- rtrim — Strip whitespace (or other characters) from the end of a string

Course content

Section 10: PHP Functions
2 / 11 | 58min

- ☑ 114. Function Structure — 7min
- ☑ 115. Built In Functions Maths And Strings — 8min
- □ 116. Built In Functions Date And Time — 4min
- □ 117. Custom Functions — 4min
- □ 118. Format With Functions — 3min
- □ 119. Email Address Checking — 8min

About this course



udemy.com/course/php-mysql-tutorial/learn/lecture/49207#overview

Beginner PHP and MySQL Tutorial

```
<?php

    //  set size of image
    $imageSizeX = 300;
    $imageSizeY = 300;

    $image = imagecreate($imageSizeX, $imageSizeY);

    //  first instance of imagecolorallocate sets background of image.
    $background_teal = imagecolorallocate ($image, 0, 125, 125);

    //  now set a colour for the text
    $red =  imagecolorallocate ($image, 255, 0, 0);
    $yellow = imagecolorallocate ($image, 235, 235, 51);

    $textColour = $yellow;

    //  set a path to the font
    $font_file = 'FreeMonoBoldOblique.ttf';
    $font_path = '../fonts/'.$font_file;

    //  the text string
    $string = 'php / MySQL Training';

    //  path and name of image file to be saved
    $image_file = 'testimage.png';
    $image_path = '../images/'.$image_file;

    $fontsize = 18;
    $textAngle = 45;
    $xOffset = 50;
    $yOffset = 270;

    //  create the image
    imagettftext (  $image, $fontsize,
```

Course content

- ☑ 125. Reading Text Files — 7min
- ☑ 126. Writing Text Files — 4min
- ☑ 127. Working With Image Files - exif — 5min
- □ 128. Creating Image Files With GD Library — 5min
- □ 129. Managing Images With GD Library — 6min
- □ 130. Resizing Images With PHP — 3min
- □ 131. Uploading Files - Overview — 3min

About this course

Report:

PHP Functions:

PHP functions are similar to other programming languages. A function is a piece of code which takes one more input in the form of parameter and does some processing and returns a value.

You already have seen many functions like fopen () and fread () etc. They are built-in functions but PHP gives you option to create your own functions as well.

There are two parts which should be clear to you −

- **Creating a PHP Function**
- **Calling a PHP Function**

In fact, you hardly need to create your own PHP function because there are already more than 1000 of built-in library functions created for different area and you just need to call them according to your requirement.

Please refer to PHP Function Reference for a complete set of useful functions.

Creating PHP Function

Its very easy to create your own PHP function. Suppose you want to create a PHP function which will simply write a simple message on your browser when you will call it. Following example creates a function called write Message () and then calls it just after creating it.

PHP Functions with Parameters

PHP gives you option to pass your parameters inside a function. You can pass as many as parameters your like. These parameters work like variables inside your function. Following example takes two integer parameters and add them together

```
<html>
  <head>
    <title>Writing PHP Function with Parameters</title>
  </head>
  <body>
    <?php
      function addFunction ($num1, $num2) {
        $sum = $num1 + $num2;
        echo "Sum of the two numbers is: $sum";
      }

      addFunction (10, 20);
    ?>

  </body>
</html>
```

Using external files and images:

You can save your uploading images in the database table for later use e.g. display user profile or product image, create the image gallery, etc.

There are two ways of doing this –

- **Save the path or name of an image**
- **Encode image into a base64 format**

Table structure:

- **name – This field is used to store the image file name.**
- **image – This field is used to store the image base64 generated value.**

Configuration

- **Create a new config.php file for database configuration.**

- **Save path or name, Youcan either save the full path or name of an image in your MySQL database table. Retrieve the image name or path from the MySQL database and use it to make an image source.**

- **Here, I am storing the file name in the MySQL database.**

Retrieve

Select the name or path of the image which you have stored in the database table and use it in the image source.

base64_encode ()

- **You can store the full image in the Database table by converting it into the base64 format. You don't need to store image reference in the Database table e.g. name, path, and not require to store the image on your server.**

- **In PHP base64_encode () method is been used for base64 conversion. Before storing it in the database I append data: image/'. $ imageFileType.';base64, text with base64 value.**

- **Now when you need to display the image just fetch the value and use it as an image source.**

Conclusion

In my opinion, instead of storing an image in the MySQL database in the base64 format, it’s better to store in the server and save the reference in the database table to keep track of the location.

It is fast and consumes less space in the database table compare to base64.