Лабораторная робота №13

Обработка изображений в PHP с использованием GD

В этом уроке мы узнаем о библиотеке GD (Graphic Draw) в PHP. Вы увидите, как эту библиотеку можно использовать для управления изображениями путем изменения размера, обрезки, поворота или фильтрации.

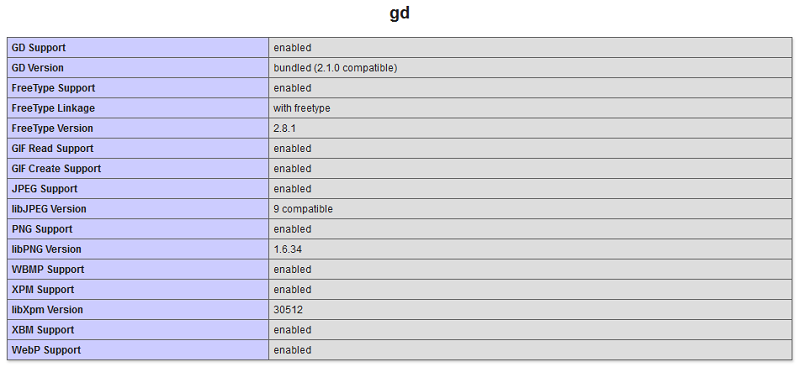
Что такое GD?

PHP может сделать гораздо больше, чем просто отправлять HTML-сообщения посетителям. Например, он имеет возможность манипулировать изображениями. Не только это, но вы также можете создавать свои собственные изображения с нуля, а затем либо сохранять их, либо подавать их пользователям.

PHP может удовлетворить практически все ваши основные потребности в управлении изображениями, используя библиотеку GD - сокращение для Graphic Draw.

Установка

Если вы работаете в Windows, вы можете включить файл php\_gd2.dll в качестве расширения в php.ini. Если вы используете что-то вроде XAMPP, вы найдете файл php\_gd2.dll в каталоге xampp\php\ext. Вы также можете проверить, установлена ли GD в вашей системе, с помощью функции phpinfo();. Если вы просмотрите полученный результат, вы найдете что-то похожее на следующее.



**Функции GD и функции для работы с изображениями**

gd\_info — Вывод информации о текущей установленной GD библиотеке

getimagesize — Получение размера изображения

getimagesizefromstring — Получение размера изображения из строки данных

image\_type\_to\_extension — Получение расширения файла для типа изображения

imagearc — Рисование дуги

imagebmp — Вывести BMP-изображение в браузер или файл

imagechar — Рисование символа по горизонтали

imagecharup — Рисование символа вертикально

imagecolorallocate — Создание цвета для изображения

imagecolorallocatealpha — Создание цвета для изображения

imagecolorat — Получение индекса цвета пиксела

imagecolorexact — Получение индекса заданного цвета

imagecolorexactalpha — Получение индекса заданного цвета и альфа компонента

imagecolorresolve — Получает идентификатор конкретного цвета или его ближайший аналог

Больше функций вы можете найти на <https://www.php.net/manual/ru/book.image.php>

Создание нового изображения

Функция imagecreatetruecolor() окажется полезной, если у вас нет исходного источника изображения, которое вы хотите изменять. Она принимает два целочисленных параметра: ширину и высоту. Она вернет ресурс изображения, если все пойдет по плану. Возвращаемый ресурс изображения в основном представляет собой черное изображение с заданной шириной и высотой.

**Загрузка файла изображения**

Если вы планируете работать с изображениями, которые уже хранятся где-то, вам пригодится использование таких функций, как imagecreatefromjpeg(), imagecreatefrompng() и imagecreatefromgif(). Они создадут ресурс изображения со всеми данными из загруженного файла изображения. Эти функции принимают единственный параметр, который указывает местоположение загружаемого изображения, как URL-адрес или путь к файлу.

**Создание изображения из строки**

Библиотека GD также позволяет создавать изображения из строки, используя функцию imagecreatefromstring() в PHP. Помните, что вам нужно будет использовать base64\_decode() для данной строки перед imagecreatefromstring(). Функция может автоматически определять, является ли тип изображения JPG, PNG, GIF или другим поддерживаемым форматом.

Открытие изображения

Итак, есть исходное изображение PNG 400x400px:

[](https://snipp.ru/uploads/images/donut.png)

С помощью функции getimagesize() получим ширину, высоту и тип, далее откроем его функциями в зависимости от типа:

|  |  |
| --- | --- |
|  | $filename = \_\_DIR\_\_ . '/donut.png'; |
|  |  |
|  | $info = getimagesize($filename); |
|  | $width = $info[0]; |
|  | $height = $info[1]; |
|  | $type = $info[2]; |
|  |  |
|  | switch ($type) { |
|  | case 1: |
|  | $img = imageCreateFromGif($filename); |
|  | imageSaveAlpha($img, true); |
|  | break; |
|  | case 2: |
|  | $img = imageCreateFromJpeg($filename); |
|  | break; |
|  | case 3: |
|  | $img = imageCreateFromPng($filename); |
|  | imageSaveAlpha($img, true); |
|  | break; |
|  | } |

## Изменение размера изображения (resize)

Приведенный код уменьшает или увеличивает изображение не искажая его пропорции.

|  |  |
| --- | --- |
|  | // Размеры новой фотки. |
|  | $w = 200; |
|  | $h = 0; |
|  |  |
|  | if (empty($w)) { |
|  | $w = ceil($h / ($height / $width)); |
|  | } |
|  | if (empty($h)) { |
|  | $h = ceil($w / ($width / $height)); |
|  | } |
|  |  |
|  | $tmp = imageCreateTrueColor($w, $h); |
|  | if ($type == 1 || $type == 3) { |
|  | imagealphablending($tmp, true); |
|  | imageSaveAlpha($tmp, true); |
|  | $transparent = imagecolorallocatealpha($tmp, 0, 0, 0, 127); |
|  | imagefill($tmp, 0, 0, $transparent); |
|  | imagecolortransparent($tmp, $transparent); |
|  | } |
|  |  |
|  | $tw = ceil($h / ($height / $width)); |
|  | $th = ceil($w / ($width / $height)); |
|  | if ($tw < $w) { |
|  | imageCopyResampled($tmp, $img, ceil(($w - $tw) / 2), 0, 0, 0, $tw, $h, $width, $height); |
|  | } else { |
|  | imageCopyResampled($tmp, $img, 0, ceil(($h - $th) / 2), 0, 0, $w, $th, $width, $height); |
|  | } |
|  |  |
|  | $img = $tmp; |

PHP

#### Результат

|  |  |  |
| --- | --- | --- |
| $w = 200; $h = 0; <https://snipp.ru/uploads/images/donut-resize-200x0.png> | $w = 200; $h = 100; <https://snipp.ru/uploads/images/donut-resize-200x100.png> | $w = 100; $h = 200; <https://snipp.ru/uploads/images/donut-resize-100x200.png> |

3

## Обрезать изображение (crop)

Пример вырезает из исходного изображения часть размером $w на $h.  
$x и $y задают начальные координаты в пикселях или процентах.

|  |  |
| --- | --- |
|  | $w = 200; |
|  | $h = 200; |
|  |  |
|  | $x = '100%'; |
|  | $y = '100%'; |
|  |  |
|  | if (strpos($x, '%') !== false) { |
|  | $x = intval($x); |
|  | $x = ceil(($width \* $x / 100) - ($w / 100 \* $x)); |
|  | } |
|  | if (strpos($y, '%') !== false) { |
|  | $y = intval($y); |
|  | $y = ceil(($height \* $y / 100) - ($h / 100 \* $y)); |
|  | } |
|  |  |
|  | $tmp = imageCreateTrueColor($w, $h); |
|  | if ($type == 1 || $type == 3) { |
|  | imagealphablending($tmp, true); |
|  | imageSaveAlpha($tmp, true); |
|  | $transparent = imagecolorallocatealpha($tmp, 0, 0, 0, 127); |
|  | imagefill($tmp, 0, 0, $transparent); |
|  | imagecolortransparent($tmp, $transparent); |
|  | } |
|  |  |
|  | imageCopyResampled($tmp, $img, 0, 0, $x, $y, $width, $height, $width, $height); |
|  | $img = $tmp; |

PHP

#### Результат

|  |  |  |
| --- | --- | --- |
| $x = 0; $y = 0; <https://snipp.ru/uploads/images/donut-crop-0-0.png> | $x = '50%'; $y = '0%'; <https://snipp.ru/uploads/images/donut-crop-50-0.png> | $x = '100%'; $y = '0%'; <https://snipp.ru/uploads/images/donut-crop-100-0.png> |

4

## Поворот изображения

Функция imagerotate() поворачивает изображение на заданный угол против часовой стрелки, отрицательный угол меняет направление поворота.

|  |  |
| --- | --- |
|  | // Поворот против часовой стрелки на 45°. |
|  | $transparent = imagecolorallocatealpha($img, 0, 0, 0, 127); |
|  | $img = imagerotate($img, 45, $transparent); |
|  |  |
|  | // Поворот по часовой стрелки на 90° |
|  | $transparent = imagecolorallocatealpha($img, 0, 0, 0, 127); |
|  | $img = imagerotate($img, -90, $transparent); |

PHP

5

## Зеркальное отражение

imageflip($img, IMG\_FLIP\_HORIZONTAL);

PHP

Imageflip() зеркалит изображение, могут быть следующие параметры:

|  |  |
| --- | --- |
| IMG\_FLIP\_HORIZONTAL | По горизонтали |
| IMG\_FLIP\_VERTICAL | По вертикали |
| IMG\_FLIP\_BOTH | По горизонтали и вертикали |

6

## Наложение водяного знака (watermark)

Для защиты на картинки наносят копирайт, например, данный скрип накладывает картинку watermark.png на основное изображение:



|  |  |
| --- | --- |
|  | $watermark = \_\_DIR\_\_ . '/watermark.png'; |
|  |  |
|  | $x = '50%'; |
|  | $y = '50%'; |
|  |  |
|  | $info = getimagesize($watermark); |
|  | switch ($info[2]) { |
|  | case 1: |
|  | $tmp = imageCreateFromGif($watermark); |
|  | break; |
|  | case 2: |
|  | $tmp = imageCreateFromJpeg($watermark); |
|  | break; |
|  | case 3: |
|  | $tmp = imageCreateFromPng($watermark); |
|  | break; |
|  | } |
|  |  |
|  | if (strpos($x, '%') !== false) { |
|  | $x = intval($x); |
|  | $x = ceil(($width \* $x / 100) - ($info[0] / 100 \* $x)); |
|  | } |
|  | if (strpos($y, '%') !== false) { |
|  | $y = intval($y); |
|  | $y = ceil(($height \* $y / 100) - ($info[1] / 100 \* $y)); |
|  | } |
|  |  |
|  | imagecopy($img, $tmp, $x, $y, 0, 0, $info[0], $info[1]); |
|  | imagedestroy($tmp); |

PHP

#### Результат

|  |  |  |
| --- | --- | --- |
| $x = '50%'; $y = '50%'; <https://snipp.ru/uploads/images/donut-watermark-50-50.png> | $x = '100%'; $y = '0%'; <https://snipp.ru/uploads/images/donut-watermark-100-0.png> | $x = '100%'; $y = '100%'; <https://snipp.ru/uploads/images/donut-watermark-100-100.png> |

7

## Добавление фона

Актуально для PNG с прозрачностью. Скрипт вставит на задний фон картинку с положением $x и $y. Размер основного изображения не изменится.

|  |  |
| --- | --- |
|  | $file = \_\_DIR\_\_ . '/donut\_bg.jpg'; |
|  |  |
|  | // Положение фона. |
|  | $x = '50%'; |
|  | $y = '50%'; |
|  |  |
|  | $info = getimagesize($file); |
|  | switch ($info[2]) { |
|  | case 1: |
|  | $bg = imageCreateFromGif($file); |
|  | break; |
|  | case 2: |
|  | $bg = imageCreateFromJpeg($file); |
|  | break; |
|  | case 3: |
|  | $bg = imageCreateFromPng($file); |
|  | break; |
|  | } |
|  |  |
|  | if (strpos($x, '%') !== false) { |
|  | $x = intval($x); |
|  | $x = ceil(($info[0] \* $x / 100) - ($width / 100 \* $x)); |
|  | } |
|  | if (strpos($y, '%') !== false) { |
|  | $y = intval($y); |
|  | $y = ceil(($info[1] \* $y / 100) - ($height / 100 \* $y)); |
|  | } |
|  |  |
|  | $tmp = imageCreateTrueColor($width, $height); |
|  | imagecopy($tmp, $bg, 0, 0, $x, $y, $width, $height); |
|  | imagedestroy($bg); |
|  |  |
|  | imagecopy($tmp, $img, 0, 0, 0, 0, $width, $height); |
|  | $img = $tmp; |

PHP

|  |  |
| --- | --- |
| Фон <https://snipp.ru/uploads/images/donut-bg.jpg> | Результат <https://snipp.ru/uploads/images/donut-bg-result.png> |

8

## Фильтры

imagefilter($img, $filtertype, $arg1, $arg2);

PHP

Функция imagefilter() применяет фильтр к изображению.  
В параметре $filtertype указывается константа применяемого фильтра, а в следующих его настройки.

## Сохранение

### Вывод изображения в браузер

До вызова функции header() скрипт ничего не должен выводить (echo, ?>...<?php), иначе картинка будет битой.

|  |  |
| --- | --- |
|  | switch ($type) { |
|  | case 1: |
|  | header('Content-Type: image/gif'); |
|  | imageGif($img); |
|  | break; |
|  | case 2: |
|  | header('Content-Type: image/jpeg'); |
|  | imageJpeg($img, null, 100); |
|  | break; |
|  | case 3: |
|  | header('Content-Type: image/x-png'); |
|  | imagePng($img); |
|  | break; |
|  | } |
|  |  |
|  | imagedestroy($img); |
|  | exit(); |

PHP

Чтобы браузер отдал фото на скачивание, в начало кода нужно добавить заголовок:

header('Content-Disposition: Attachment;filename=' . basename($src));

PHP

### Сохранение изображения в файл на сервере

|  |  |
| --- | --- |
|  | switch ($type) { |
|  | case 1: |
|  | imageGif($img, $src); |
|  | break; |
|  | case 2: |
|  | imageJpeg($img, $src, 100); |
|  | break; |
|  | case 3: |
|  | imagePng($img, $src); |
|  | break; |
|  | } |
|  |  |
|  | imagedestroy($img); |

PHP

### Вывод в браузер и сохранение в файл

|  |  |
| --- | --- |
|  | switch ($type) { |
|  | case 1: |
|  | header('Content-Type: image/gif'); |
|  | imageGif($img, $src); |
|  | break; |
|  | case 2: |
|  | header('Content-Type: image/jpeg'); |
|  | imageJpeg($img, $src, 100); |
|  | break; |
|  | case 3: |
|  | header('Content-Type: image/x-png'); |
|  | imagePng($img, $src); |
|  | break; |
|  | } |
|  |  |
|  | imagedestroy($img); |
|  | readfile($src); |
|  | exit(); |

**Задание**

Создать изображение и отредактировать его применив: изменение размера изображения, обрезка изображения, наложение водного знака и добавление фона.