PHP. Библиотека GD.

- **Графика средствами PHP**
- **Обработка изображений**

Графика средствами PHP

- В PHP нет ограничений на создание простого HTML-вывода. Он может также использоваться для создания изображений и манипуляций с файлами изображений различных форматов, включая gif, png, jpg, wbmp и xpm. Что особенно удобно: PHP может выводить поток изображения непосредственно в браузер. Вам для этого необходимо скомпилировать PHP с библиотекой GD функций изображений.

Графика средствами PHP

- Использование средств работы с графикой позволяет создавать новые и изменять изображения, загруженные из файлов.
- Данные возможности применяются довольно часто для оптимизации изображений, построения графиков, генерации кодов подтверждения.

Графика средствами PHP

- Библиотека GD предоставляет широкий набор функций для работы. Описание полного набора функций доступен по ссылке http://docs.php.net/manual/ru/book.image.php
- Рассмотрим общие принципы создания новых и обработки загруженных из файла изображений.

Графика средствами PHP

- PHP может выводить поток изображения непосредственно в браузер. Однако, смешивать формат вывода (изображение и текст) невозможно, поскольку он задается в заголовке ответа HTTP:
- `header ("Content-type: image/png");`
- Само же формирование изображения начинается вызовом функции
- `$im=@imagecreate($size_x, $size_y) or die ("Cant Initialize GD");`

Графика средствами PHP

- Аргументы этой функции – размер создаваемого изображения в пикселях. Функция возвращает ссылку на созданное пустое изображение, а в случае неудачи создания работа сценария прерывается. Символ @ служит для подавления выдачи стандартного сообщения об ошибке в браузер.
- Далее необходимо выделить цвета, используемые в изображении:
- `$color=imagecolorallocate ($im, 255, 0, 255);`

Графика средствами PHP

- **`int imagecolorallocate (resource image, int red, int green, int blue)`** - выделение цвета для изображения. Возвращает идентификатор цвета, представляющий цвет, составленный из данных RGB-компонентов.
- Аргумент *im* возвращается от **`imagecreate()`**.
- *red*, *green* и *blue* это значения красного, зелёного и синего компонентов цвета. Эти параметры являются целыми числами в диапазоне от 0 до 255. imagecolorallocate() обязана вызываться для создания каждого цвета, используемого в изображении *image*.
- Возвращает -1, если выделение не удалось.

Графика средствами PHP

- `$tcolor=imagecolortransparent($im, $color);`
- Вызов этой функции обеспечивает прозрачность цвета для его использования в качестве фонового.
- В созданном изображении выделенными цветами рисуются графические примитивы – линии, сектора, прямоугольники, эллипсы.
- Вывод линии на изображении с координатами начала и окончания указанным цветом:
- `int imageline (image, int x1, int y1, int x2, int y2, int col)`

Графика средствами PHP

- Вывод дуги на изображении с координатами центра, заданной ширины и высоты, углом начала и окончания указанным цветом и указанным стилем:
- **`int imagearc (image, int cx, int cy, int w, int h, int s, int e, int col)`**
- Вывод прямоугольника на изображении с координатами левого верхнего и правого нижнего углов указанным цветом:
- **`int imagerectangle (image, int x1, int y1, int x2, int y2, int col)`**

Графика средствами PHP

- Вывод эллипса на изображении с координатами центра, заданной ширины и высоты указанным цветом :
- `int imageellipse (image, int cx, int cy, int w, int h, int col)`
- Замкнутые фигуры, в том числе все изображение или его часть, могут иметь заливку:
- `int imagefilledrectangle (image, int x1, int y1, int x2, int y2, int col)`
- `bool imagefill (image , int $x , int $y , int $col)`

Графика средствами PHP

- В изображении может присутствовать текст:
- **`bool imagestring (image, int $font , int $x , int $y , string $string , int $color )`**
- **`array imagettftext (image, float $size , float $angle , int $x , int $y , int $color , string $fontfile , string $text )`**
- Шрифт текста может быть встроенным (для функции **`imagestring()`** это 1-5 в зависимости от размера или экземпляр класса GdFont, который возвращает функция imageloadfont().).

Графика средствами PHP

- В функции **imagettftext()** можно задать размер и угол наклона текста (**float $size , float $angle)**.
- Для функции **imagettftext()** шрифт может быть типа TrueType или предварительно загруженным специальным типа .gd, :
- **int imageloadfont (string $file)**
- Когда шрифт ttf находится в том же каталоге, что и скрипт PHP, поможет следующий трюк.
- **putenv('GDFONTPATH='.realpath('.'));**
  // расширение .ttf не указывается)
  $font = 'SomeFont';

Графика средствами PHP

- Даже при указании шрифта функции **imagettftext()** кириллица может выводиться в неверной кодировке.
- Есть несколько способов решения этой проблемы. Самый простой – задать кодировку явно:
- **setlocale(LC_ALL,"ru_RU.WIN1251");**
- или конвертировать строку с символами кириллицы в кодировку UTF-8:
- **$text = iconv("windows-1251", "UTF-8", "наш текст");**

Графика средствами PHP

- Осталось вывести полученное изображение в нужном графическом формате в браузер или файл:
- **`bool imagepng ($im [, string $filename ]);`**
- **`bool imagegif ( resource $im [, string $filename])`**
- **`imagewbmp ($im);`**
- **`int imagejpeg (resource image [, string filename [, int quality]])`**
- Здесь **`quality`** – качество изображения.

Графика средствами PHP

- Если в этих функциях не указывать второй параметр (имя файла), то изображение будет выводиться непосредственно в браузер. Перед этим необходимо установить верный тип выводимого содержимого (MIME-type), например, image/png:
- `header('Content-Type: image/png; charset=UTF-8');`
- `imagepng($im);`

Графика средствами PHP

- Качество может иметь диапазон значений от 0 (наихудшее качество, наименьший файл) до 100 (наилучшее качество, наибольший файл). По умолчанию используется значение по умолчанию IJG quality (около 75).
- Некоторые форматы могут не поддерживаться библиотекой GD, поэтому надо проверять их доступность:
- `if (function_exists("imagegif")) { header("Content-type: image/gif");`
- `imagegif ($im); }`

Графика средствами PHP

- При прочих равных условиях рекомендуется использовать формат png в связи с его лицензионной чистотой.
- Для отображения динамически сформированного (сценарием **`graph.php`**) изображения в тексте HTML-страницы применяется известный тег HTML:
- **`<img src=graph.php>`**
- Библиотека GD позволяет создавать не только новые изображения, но и обрабатывать существующие, например, загруженные пользователем на сервер.

Обработка изображений

- Вначале обсудим загрузку файлов изображений на сервер. Для этого необходима форма следующего вида:
- `<form action="foto.php" method="POST" enctype="multipart/form-data">`
- `<input type="file" name="f">`
- `<input type="submit" value="Загрузить">`
- `</form>`
- Обратите внимание на тип формы и поле типа file.

Обработка изображений

- Перед обработкой файла необходимо убедиться, что он был выбран для загрузки и загружен без ошибок:
- `if (isset($_FILES["f"]) and isset($_FILES["f"]["name"])and $_FILES["f"]["error"] == 0)`
- Выбрать имя для хранения и папку:
- `$fnam=basename($_FILES["f"]["name"],. "jpg");`
- `$fPath = "../lab10/".$ fnam;`
- И переместит файл из временной папки в выбранную:

Обработка изображений

- `move_uploaded_file($_FILES["f"]["tmp_name"], $fPath);`
- Теперь можно загрузить изображение из файла и определить его размеры:
- `$src = imagecreatefromjpeg ($fPath);`
- `$sx = imagesx($src);`
- `$sy = imagesy($src);`
- Для обеспечения защиты от вирусов типа «Анна Курникова» рекомендуется не обрабатывать открытый файл, а скопировать его графическое содержимое в новое изображение:

Обработка изображений

- `if (function_exists("imagecreatetruecolor")) {`
- `$img = imagecreatetruecolor($imW, $imH); }`
- `else {`
- `$img = imagecreate($imW, $imH); }`
- Далее можно скопировать изображение из старого в новое с изменением размера либо качества:
- `imagecopyresampled($img, $src, 0, 0, 0, 0, $imW, $imH, $sx, $sy);`

Обработка изображений

- **`imagecopyresampled()`** копирует прямоугольную часть изображения src начиная с int srcX, int srcY в другое изображение (dst), начиная с int dstX, int dstY, плавно интерполируя пикселные значения таким образом, что изменение размера изображения сохранит его чёткость и яркость.
- **`imagecopyresized($img, $src, 0, 0, 0, 0, $imW, $imH, $sx, $sy);`**
- Эта функция копирует указанную часть изображения без интерполяции.

Обработка изображений

- Если новое изображение планируется сохранить на диск:
- `imagepng($img, $fnam);`
- рекомендуется его подписать полупрозрачной надписью для сохранения авторских прав:
- `$font= imageloadfont("Arial.gdf");`
- `$tcolor=imagecolorallocate($img, 192, 192,192);`
- `imagecolortransparent($img, $tcolor);`
- `imagestring($img, $font, 50, 50, "Это фото украдено с сайта . . .", $tcolor);`

Обработка изображений

- Пример первый. Изображение с текстом. Код.
- Пример второй. Изображение с динамическим текстом. Код.
- Пример третий. График линейный. Код.
- Пример четвертый. Гистограмма. Код.
- Пример пятый. Круговая диаграмма. Код
- Пример шестой. Работа с изображениями. Код.