

PHP ve GD Kütüphanesi ile Resim İşleme

LKD Ankara Seminerleri

1 Eylül 2004 - Ankara

Linux Kullanıcıları Derneği

Ekin Meroğlu
ekin@fisek.com.tr

PHP Script Dili

Php (PHP:Hypertext Preprocessor) çoğunlukla dinamik web sayfalarının hazırlanmasında kullanılan açık kaynak kodlu, özgür bir script (betik) dilidir.

Php, C, Perl ve Java ile benzer sözdizimine sahip, kolay öğrenilebilir bir dildir. Genel kullanım amacı web sayfaları oluşturmak olsa da PHP ile yapabileceklerimiz bununla sınırlı değildir..

Php, Rasmus Lerdorf tarafından 1995 yılında PHP/FI olarak geliştirilmeye başlanmıştır. 1997 yılında duyurulan PHP/FI 2.0 ile kullanımı birkaç bin web sitesine ulaşmış, kayıtlı domainlerin % 1'ine karşılık gelen 50.0000 domainde kurulu olduğu tahmin edilmiştir.

Şu anda kullandığımız PHP'nin ilk versiyonu diyebileceğimiz PHP 3.0 Haziran 1998'de uzun bir test sürecinden sonra duyurulmuş ve 1998 yılı sonunda kayıtlı domainlerin % 10'unda kullanılmaya başlanmıştır.

PHP Script Dili

PHP 4.0 sürümü ise PHP kod tabanının tamamen baştan yazılması ve Zend motorunun kullanımı ile Mayıs 1999 'da duyurulmuştur. Php 4 için şu anki güncel sürüm 4.3.8 (kararlı) ve 4.3.9RC2'dir. Yüzbinlerce geliştiricinin PHP üzerinde uygulama geliştirdiği ve kayıtlı domainlerin %20'sinden fazlası tarafından kullanıldığı tahmin edilmektedir.

Yeni sürüm olan Php 5.0, 13 Haziran 2004'de duyurulmuş ve en son olarak 12 Ağustos 2004'de 5.0.1 sürümü çıkarılmıştır.

<http://www.php.net>

GD Resim İşleme Kütüphanesi

GD , resim işleme amaçlı bir ANSI C kütüphanesidir. Boutell.Com tarafından açık kaynak kodlu özgür bir yazılım olarak geliştirilmekte ve dağıtılmaktadır . Son versiyonu 21 Haziran 2004 tarihinde yayınlanan 2.0.28 sürümüdür. Bu sürümün en önemli özelliği, patent sorunları nedeniyle uzun süredir eksik olan GIF yazma-okuma desteğinin tekrar eklenmiş olmasıdır.

PHP içinde bütünleşik olarak geliştirilen bir GD sürümüne sahiptir. İleride bu sürüm ile boutell sürümünün kod tabanlarının birleştirilmesi amaçlanmaktadır.

www.boutell.com

PHP 'de GD Kütüphanesi Desteği

PHP de GD fonksiyonlarının kullanılabilmesi için PHP derlenirken GD fonksiyonlarının da derlenmesi gerekir. PHP 4.3.0 versiyonundan itibaren bütünleşik bir GD kütüphanesi de içermektedir. Bununla birlikte istenirse sisteme kurulu bir GD kütüphanesi de kullanılabilir.

GD 1.6'dan daha eski versiyonlarda bulunan GIF desteği, bu sürümden sonra PNG ile değiştirilmiştir. PHP ile bütünleşik GD kütüphanesinde ise salt-okunur GIF desteği vardır.

Değişik formatlar için şu kütüphaneler kurulabilir :

jpeg-6b <ftp://ftp.uu.net/graphics/jpeg/>

png <http://www.libpng.org/pub/png/libpng.html>

xpm <ftp://metalab.unc.edu/pub/Linux/libs/X/!INDEX.html>

Freetype 1,2 ve Tlib font kütüphaneleri de desteklenmektedir.

PHP 'de GD Kütüphanesi Desteği

PHP'nin GD desteği ile derlenmesi için configure sırasında

`--with-gd[=DIR]`

parametresi kullanılmalıdır. Bütünleşik GD kütüphanesini kullanmak için

`--with-gd`

yeterlidir. Bütünleşik GD kütüphanesinin kullanımı her durumda önerilmektedir.

Windows ortamında ise `php_gd2.dll` kütüphanesinin `php.ini` de tanımlanması yeterlidir.

Bunun dışında ek kütüphaneler için aşağıdaki parametreler ile kullanılabilir :

jpeg-6b `--with-jpeg-dir=DIR`

png `--with-png-dir=DIR`

(Not : libpng, zlib kütüphanesine ihtiyaç duyar, bu nedenle

`--with-zlib-dir[=DIR]` parametresini de kullanmanız gereklidir.)

xpm `--with-xpm-dir=DIR`

PHP 'de GD Kütüphanesi Desteği

Çeşitli font kütüphaneleri için kullanılan `configure` parametreleri ise şunlardır :

```
FreeType 1.x  --with-ttf[=DIR]
FreeType 2    --with-freetype-dir=DIR.
T1lib         --with-t1lib[=DIR].
Native ttf    --enable-gd-native-ttf.
```

Ayrıca PHP resim dosyalarının headerlarında bulunan *exif* verilerini de okuyabilir. *exif* fonksiyonları için GD kütüphanesi kullanılmaz.

exif desteği için `configure` sırasında
 `--enable-exif`
paramatresi kullanılmalıdır.

Resim Ve Resim Formatları

Dijital resim söz konusu olduğunda karşımıza iki ayrı format çıkar : **Vektörel** ve **Bitmap** resimler. GD kütüphanesi, Bitmap resimler üzerinde çalışan bir seri fonksiyon sunar.

Bitmap resimler pixel olarak adlandırılan minik noktalardan oluşur ve resim dosyası içinde bu her pixelin renk bilgisi tutulur. Bitmap resimlerin dezavantajı, büyütüldüklerinde kalite kaybına uğramalarıdır. Bitmap resimler büyüdükçe büyük dosyalar ve yüksek hafıza ihtiyaçları ortaya çıkar. Bu nedenle resmin kullanılacağı medyaya göre optimum bir çözünürlük seçilmeli ve resim üzerinde baştan itibaren bu çözünürlükte çalışılmalıdır.

Resim Ve Resim Formatları

Bitmap Resimlerle ilgili temel özellikler şunlardır :

- Çözünürlük : Resmin her inch uzunluğundaki parçasında kaç pixel (Dot) bulunduğunu belirtir. Sadece web üzerinde kullanılacak dosyalar için 72 DPI genellikle yeterlidir, fiziksel baskı yapılacak dosyalar ise baskı tekniğine göre 200 - 1200 DPI olabilir.
- Renk Derinliği : Resim içinde kullanılabilecek renk sayısı. BitPerPixel (BPP) veya renk sayısı olarak ifade edilir. 8 BPP , her pixel için 8 bit ($2^8 = 256$) renk kullanılmasına izin verirken 24 BPP ($2^{24} = 16.777.216$) renk kullanılmasına izin verir.
- Palet bazlı Bitmap : Resimde kullanılacak renklerin listesinin resim dosyası içinde saklandığı resim dosyaları. Kullanılmayan renkler paletten çıkarılarak dosya boyutunun düşmesi sağlanabilir.
- Gerçek Renkli Bitmap : Resimde her pixelin 24 BPP renk skalasından istediği rengi alabildiği resim dosyalarıdır.

Resim Ve Resim Formatları

Yoğun olarak kullanılan bitmap dosya formatları şunlardır :

- GIF : Graphics Exchange Format : Oldukça eski ve kayıpsız sıkıştırma kullanan bir dosya formatıdır. Sıkıştırma algoritmasındaki patent sorunları nedeniyle giderek daha az kullanılmaktadır. Avantajları, birden fazla kareden oluşan animasyonlar içerebilmesi ve şeffaflık desteğidir. Palet bazlı bir bitmap dosya formatıdır.
- JPEG : Joint Picture Experts Group : Kayıplı sıkıştırma kullanan yaygın bir dosya formatıdır. Sıkıştırma oranı, dolayısıyla kalitesi ayarlanabilir. Fotoğraf gibi yüksek renkli resim dosyalarında daha iyi sonuçlar verir, bilgisayar ile üretilmiş resimlerde (render vb.) tavsiye edilmez. Web'de kullanılan en yaygın formattır. Gerçek renkli bir bitmap formatıdır.
- PNG : Portable Graphics Format : Yeni ve kayıpsız sıkıştırma kullanan bir formattır. Şeffaflık desteği vardır ve GIF yerine yaygın olarak kullanılmaya başlanmıştır. Bazı browserlarda şeffaf PNG'ler tam desteklenmeyebilir. PNG formatı palet bazlı veya gerçek renkli resimlere izin verir.

PHP - GD ile Resim Oluşturma

PHP'de GD kütüphanesi tarafından işlenecek resim dosyaları, *handler*'lar tarafından tutulurlar. Resim oluşturma olarak tanımladığımız işlem, belli bir resim dosyasını temsil edecek bir *handler*'in yaratılmasıdır.

Resimlerin oluşturulup işlenmesi, ardından browser'a gönderilmesi şu beş temel adımda özetlenebilir :

1. Boş veya varolan bir resmi kullanarak bir resim handleri oluşturun.
(`imagecreate` - `imagecreatetruecolor`-`imagecreatefrom...`)
2. Resim üzerinde değişiklik yapın veya yeni elemanlar ekleyin.
3. HTML başlık bilgilerini (header) gönderin.(`header`)
4. Resmi gönderin.(`imagepng` - `imagejpeg` - `imagewbmp...`)
5. Hafızada resim handler'i tarafından tutulan bölgeyi temizleyin.
(`imagedestroy`)

Yeni (Boş) Resim Oluşturma

Palet bazlı resim oluşturma:

```
handler imagecreate (int x, int y)
```

Gerçek Renkli resim oluşturma :

```
handler imagecreatetruecolor (int x, int y)
```

Bu fonksiyonlar, $x * y$ pixel boyutlarında boş bir resim dosyasını temsil eden *handler*lar oluştururlar.

```
$image = imagecreatetruecolor (300,400);
```

Özel bir neden yoksa truecolor kullanımı tavsiye edilir.
(GIF dosya çıktısı alınacaksa truecolor kullanılamaz)

Resim Dosyası Kullanarak Resim Oluşturma

GD kütüphanesi destek verdiği resim formatlarındaki dosyaları okuyarak bunlar üzerinde de işlem yapabilir. Oluşturulan resim dosyasını yine bir *handler* temsil eder.

```
handler imagecreatefrompng (string dosya_adi)
handler imagecreatefromjpeg (string dosya_adi)
handler imagecreatefromgif (string dosya_adi)
handler imagecreatefromwbmp (string dosya_adi)
handler imagecreatefromgd2 (string dosya_adi)
```

Bu fonksiyonların hepsi fopen wrappers izin verdiği sürece `dosya_adi` olarak URL kabul eder.

```
$yeni_resim = imagecreatefrompng ( 'images/logo-buyuk.png' );
```

Resim Dosyası Bilgileri

Kullanılan resim dosyalarının büyüklük ve başlık bilgilerini okumak için kullanılan fonksiyonlardır.

```
dizi getimagesize (string dosya_adi, [&dizi dosya_veri])  
  
$boyut = getimagesize ('images/logo-buyuk.png', &$bilgi);
```

Fonksiyondan dönen dizi aşağıdaki bilgileri içerir :

\$boyut[0] : Resim dosyasının genişliği,
\$boyut[1] : Resim dosyasının yüksekliği,
\$boyut[2] : Resim dosyasının tipi. Alabileceği değerler : 1- GIF, 2- JPG, 3- PNG, 4- SWF, 5- PSD, 6- BMP, 7- TIFF(intel), 8- TIFF(motorola-MAC), 9- JPC, 10- JP2, 11- JPX, 12- JB2, 13- SWC, 14- IFF

Resim Dosyası Bilgileri

`$boyut[3]` : Resim dosyasının boyutlarını `width="xxx" height="yyy"` formatında belirten bir string. Bu string `` HTML taglerinde doğrudan kullanılabilir.

```
<?php
$boyut = getimagesize ( "images/flag.jpg" );
echo "<img src=\"images/flag.jpg\" {$boyut[3]}>";
?>
```

`$boyut['channels']` : RGB formatlı dosyalar için 3, CMYK formatlı dosyalar için 4 değerini verir.

`$boyut['bits']` : Her renk için kullanılan bit değerini verir.

`$boyut['mime']` : Dosyanın geçerli mime bilgisini verir.

```
header ( "Content-type: {$boyut['mime']}" );
```

Resim Dosyası Bilgileri

Opsiyonel olan dosya_veri dizisi, resim dosyasının başlık kısmında saklanmış bazı verileri içerir. En sık kullanılanı IPTC verileridir :

```
<?php
$boyut = getimagesize ( "testimg.jpg" ,&$info );
if ( isset ( $info[ "APP13" ] ) ) {
    $iptc = iptcparse ( $info[ "APP13" ] );
    var_dump ( $iptc );
}
?>
```

Bu fonksiyonlar için GD kütüphanesi desteği gerekli değildir.

GD Altında Renk Kullanımı

Herhangi bir resim fonksiyonunda kullanılacak olan tüm renkler kullanılmadan önce tanımlanmalıdır. Tanımlanan her renk için hafızada bir *handler* tutulur ve resim üzerinde yapılan işlemlerde bu renk *handler*ları kullanılır.

Üzerinde işlem yapılan resim palet bazlı ise tanımlanan renkler palet üzerindeki renkleri, gerçek renkli bir resim ise herhangi bir RGB gerçek rengi temsil eder.

```
handler renk = imagecolorallocate (handler resim, int r,  
int g, int b)
```

Palet bazlı resimlerde ilk olarak tanımlanan renk resmin arkaplan rengi olarak atanır.

Varolan palet bazlı resimler üzerinde çalışılırken tanımlanmak istenen rengin palette bulunmaması ve palette yeni bir renk eklenecek boşluk olmaması halinde renk tanımlanamaz ve fonksiyon -1 değeri döndürür.

GD Altında Renk Kullanımı

```
/*
png_deneme.php
50 x 50 pixel boyutlarında bos bir png dosyası olusturur.
Kodun bazı bolumleri (c) www.nyphp.org
*/

// Hex rengi rgb degerlerine cevirelim
$renk = hex2int($_GET['color']);

// Adim 1. Yeni Bir Bos Resim Olusturalim ..
$im = imageCreate(50,50);

// Adim 2. Arkaplan Rengini Tanımlayalım ..
$background = imageColorAllocate($im,$renk['r'],$renk['g'],$renk['b']);

// Adim 3. Baslik Bilgilerini Gonderelim ..
header('Content-type: image/png');

// Adim 4. Resmi Gonderelim ..
imagePNG($im);
```

GD Altında Renk Kullanımı

```
// Adım 5. Hafızadaki Resim Handler'ini silelim ..
imageDestroy($im);

/*
Hexadecimal Renk Degerlerini r, g, b dizisine cevirir :
*/
function hex2int($hex) {
    return array( 'r' => hexdec(substr($hex, 0, 2)), // Ilk cift
                  'g' => hexdec(substr($hex, 2, 2)), // Ikinci cift
                  'b' => hexdec(substr($hex, 4, 2))  // Ucuncu cift
                );
}
?>
```

Kod;

`png_deneme.php?color=EF09E1`

şeklinde çağrılarak istenen renkte boş bir resim dosyası yaratılabilir.

GD Altında Renk Fonksiyonları

`bool imageColorSet (handler resim,int in,int r,int g,int b)`
Palet bazlı resimlerde `in` ile belirtilen rengi değiştirir..

`int imageColorAt (handler resim, int x, int y)`
(`x`, `y`) pixelinin rengini verir.

`int imageColorClosest (handler resim,int r,int g,int b)`
Resmin paletinde verilen renge en yakın rengi döndürür.

`int imageColorClosestHWB (handler resim,int x,int y,
int renk)`

Resmin paletinde verilen renge Hue - White - Blackness olarak en yakın rengi döndürür.

`int imageColorExact (handler resim,int r,int g,int b)`
Verilen rengin resim paletindeki index değerini döndürür. Eğer renk palette yoksa -1 değeri döner.

GD Altında Renk Fonksiyonları

`int imageColorResolve` (handler resim, int r, int g, int b)
Verilen rengin veya en yakın rengin resim paletindeki index değerini döndürür.

`int imageColorsTotal` (\$handler resim)
Resmin paletinde bulunan renk sayısını verir.

`int imageColorDeallocate` (handler resim, int renk)
Tanımlanmış olan rengi hafızadan siler.

`array imageColorsForIndex` (handler resim, int index)
Index'i verilen rengin red, green, blue ve alpha değerlerini içeren bir dizi döndürür.

`void imageTrueColorToPalette` (handler resim, bool di, int n)
Gerçek renkli bir resmi palet bazlı bir resme dönüştürür. Dithering işlemi isteniyorsa `di` TRUE olmalıdır. `n` değeri palette bulunabilecek en fazla renk sayısıdır.

GD Altında Çizim Fonksiyonları

`int imageSetPixel (handler resim, int x, int y, int renk)`
(x,y) pixelini renk ile boyar.

`int imageLine (handler resim, int x1, int y1, int x2, int y2, int renk)`
(x1,y1) noktasından (x2,y2) noktasına renk renginde bir çizgi çizer.

`int imageRectangle (handler resim, int x1, int y1, int x2, int y2, int renk)`
`int imageFilledRectangle (handler resim, int x1, int y1, int x2, int y2, int renk)`

`imageRectangle` sol üst köşesi (x1,y1), sağ alt köşesi (x2,y2) olan bir dikdörtgeni renk rengini kullanarak çizer.

`imageFilledRectangle` ise bu dikdörtgeni çizer ve içini doldurur.

GD Altında Çizim Fonksiyonları

```
int imagePolygon ( handler resim, array noktalar,  
    int kose_sayisi, int renk)  
int imageFilledPolygon ( handler resim, array noktalar,  
    int kose_sayisi, int renk)
```

`imagePolygon`, noktalar dizisinin elemanlarının köşe noktalarını tanımladığı `kose_sayisi` köşeli poligonu `renk` rengini kullanarak çizer. `ImageFilledPolygon` ise bu poligonu çizer ve içini doldurur.

```
int imageEllipse ( handler resim, int cx, int cy, int w,  
    int h, int renk)  
int imageFilledEllipse ( handler resim, int cx, int cy,  
    int w, int h, int renk)
```

`imageEllipse`, merkezi (cx, cy) olan ve `w` genişliğinde, `h` yüksekliğinde bir elipsi, `renk` rengini kullanarak çizer. `imageFilledEllipse` ise bu elipsi çizerek içini doldurur.

GD Altında Çizim Fonksiyonları

```
int imageArc ( handler resim, int cx, int cy, int w,  
              int h, int s, int e, int color)  
int imageFilledArc ( handler resim, int cx, int cy,  
                   int w, int h, int s, int e, int color, int mod)
```

`imageArc`, merkezi (cx, cy) olan ve w genişliğinde, h yüksekliğinde bir elipsin s açısından başlayıp e açısına kadar olan yayını renk rengini kullanarak çizer. Açılarda 0 derece saat 3 yönüdür ve saatin tersi yönüne doğru artar. `ImageArcFilled`, `mod` parametresine bağlı olarak farklı şekillerde doldurulmuş yaylar çizer. `mod` parametresi şu değerleri alabilir :

`IMG_ARC_CHORD` : Yayın başlangıç noktası ile bitiş noktasını birleştiren bir doğru çizer.

`IMG_ARC_PIE` : Yayın kendisini çizer.

`IMG_ARC_NOFILL` : Yayın içinin doldurulmamasını sağlar.

GD Altında Çizim Fonksiyonları

IMG_ARC_EDGED : Yayın başlangıç ve bitiş noktalarını merkeze bağlayan doğrular da çizilir.

Bu mod değerleri OR mantıksal bağlacı yardımı ile birlikte kullanılabilir.

`int imageFill (handler resim, int x, int y, int renk)`
(x,y) noktasından başlayarak renk ile doldurma işlemi yapar.

`int imageFillToBorder (handler resim, int x, int y, int sınır, int color)`

(x,y) noktasından başlayarak renk ile doldurma işlemi yapar.

Bu sırada sınır olarak sadece `sınır` ile tanımlanan rengi dikkate alır.

GD Altında Text Kullanımı

PHP'de GD altında resme text elemanları eklerken 3 ayrı font tipi kullanılabilir : Bitmap fontlar, Type1 fontlar ve TrueType fontlar. Hangi font tiplerinin kullanılabileceği PHP derlenirken verilen configure parametrelerine bağlıdır.

FreeType 1.x `--with-ttf[=DIR].`

FreeType 2 `--with-freetype-dir=DIR.`

T1lib `--with-t1lib[=DIR].`

Bitmap Fontlar

GD ile birlikte bitmap font olarak 5 ayrı font gelmektedir : `gdFontTiny`, `gdFontSmall`, `gdFontMediumBold`, `gdFontLarge`, ve `gdFontGiant`. (1-5)

```
int imageString ( handler resim, int font, int x,  
                  int y, string text, int renk )
```

`text` karakter dizisini `font` fontu ile sol üst köşesi `(x,y)` olacak şekilde `renk` rengini kullanarak yazar.

```
int imageStringUp ( handler resim, int font, int x,  
                   int y, string text, int renk )
```

`text` karakter dizisini `font` fontu ile yatay olarak sol üst köşesi `(x,y)` olacak şekilde `renk` rengini kullanarak yazar.

```
int imageChar ( handler resim, int font, int x,  
               int y, string t, int renk )
```

`t` karakter dizisinin ilk karakterini `font` fontu ile sol üst köşesi `(x,y)` olacak şekilde `renk` rengini kullanarak yazar.

Bitmap Fontlar

```
int imageCharUp ( handler resim, int font, int x,  
                  int y, string t, int renk)
```

t karakter dizisinin ilk karakterini font fontu ile yatay olarak sol üst köşesi (x,y) olacak şekilde renk rengini kullanarak yazar.

```
int imageFontHeight ( int font )
```

font fontundaki karakterlerin pixel cinsinden yüksekliğini verir.

```
int imageFontWidth ( int font )
```

font fontundaki karakterlerin pixel cinsinden genişliğini verir.

Öntanımlı 5 font dışında yeni bitmap fontlar kullanılmak istendiğinde

```
int imageLoadFont ( string file )
```

ile yeni fontlar yüklenebilir. Kullanılacak bitmap font dosya formatı GD kütüphanesinin üzerinde çalıştığı platforma bağlıdır.

PostScript Fontlar

GD kütüphanesinde PostScript (Type1) fontları T1lib aracılığı ile kullanılır. Bu nedenle sistemde T1lib kurulu olmalı ve PHP uygun parametre kullanılarak derlenmiş olmalıdır.

```
int imagePSLoadFont ( string filename )
```

Kullanılacak fontlar kullanılmadan önce bu fonksiyonla yüklenmelidir. Fonksiyon başarılı olursa bu fontu temsil eden bir index değeri, başarısız olursa FALSE değeri döner.

Bu fontların kullanımları sona erdikten sonra kapladıkları hafıza boşaltılmadır.

PostScript Fontlar

```
array imagePSText ( handler resim, string text, int font,  
    int boyut, int renk, int arkaplan, int x, int y  
    [, int bosluk [, int darlik [, float aci  
    [, int anti_aliasing]]])
```

resim üzerinde font PostScript fontunu kullanan bir text oluşturur. boyut pixel cinsinden font büyüklüğü, aci derece cinsinden yazının açısı, renk yazı rengi, arkaplan anti-aliasing uygulanırken kullanılacak arka plan rengi, bosluk fontun öntanımlı boşluk değerine eklenecek bir boşluk değeri, darlik ise fontun öntanımlı darlık değerine eklenecek bir darlık değeridir.

(x,y) noktası, yazının referans noktasıdır ve yaklaşık olarak ilk karakterin sol-alt köşesidir.

anti-aliasing olarak 4 veya 16 değerleri kullanılabilir ve antialiasing için kullanılacak renk sayısını gösterir.

PostScript Fontlar

Fonksiyondan dönen değer ise şu elemanlara sahip bir dizi olacaktır :

- 0 => sol - alt köşe x koordinatı,
- 1 => sol - alt köşe y-koordinatı,
- 2 => sağ - üst köşe x-koordinatı,
- 3 => sağ - üst köşe y-koordinatı.

Sadece bu koordinat değerlerini elde etmek için

```
array imagePSBox ( string text, int Font, int boyut  
    [, int bosluk [, int darlik [, float angle]])
```

fonksiyonu kullanılabilir.

```
void imagePSFreeFont ( int font )
```

Kullanımı biten fontun hafızada kapladığı alanın boşaltılmasını sağlar.

PostScript Fontlar

```
bool imagePSExtendFont ( int font, float ex)
```

`font` fontu üzerinde extend (genişletme) veya condense (daraltma) işlemi uygular. Pozitif `ex` değerleri genişletme, negatif değerler ise daraltmaya neden olur.

```
bool imagePSSlantFont ( int font, float sl )
```

`font` fontunu `sl` değerine bağlı olarak eğ.

```
int imagePSCopyFont ( int font )
```

Fontun bir kopyasını çıkarır. Kaynak font üzerinde değişiklik yapılmış olabilir fakat daha önce kopyalanarak tanımlanmış bir font değil, `imagePSLoadFont` ile yüklenmiş bir font olmalıdır.

TrueType Fontlar

GD kütüphanesinde TrueType fontlar FreeType (1 ve 2) kütüphanesi aracılığı ile kullanılır. Bu nedenle sistemde bu kütüphaneler kurulu olmalı ve PHP uygun parametreler kullanılarak derlenmiş olmalıdır.

TrueType fontlar, genel olarak Type1 fontlardan daha düşük kalitede olmalarına rağmen yaygınlıkları sebebiyle tercih edilmektedirler. Kaliteleri genel amaçlı grafik uygulamaları için yeterlidir.

TrueType Fontlar

PHP, Freetype 1 desteği ile derlenmişse şu fonksiyonlar kullanılabilir :

```
array imageTTFText (handler resim, int boyut,  
    int aci, int x, int y, int renk,  
    string fontfile, string text)
```

text karakter dizisini `resim` resmi üzerinde `fontfile` fontu ile yazar. Dikkat edilmesi gereken nokta, fontun daha önce yüklenmemiş olduğu ve doğrudan fontun fiziksel dosya isminin kullanıldığıdır. Parametreler arasındaki `boyut` font büyüklüğü, `aci` derece cinsinden yazının açısı, `renk` ise kullanılacak renktir. `(x,y)` noktası ise ilk karakterin referans noktasıdır ve yaklaşık olarak karakterin sol - alt köşesidir.

Fonksiyondan dönen dizi ise yazıyı çevreleyen dikdörtgenin sırasıyla alt sol, alt sağ, üst sağ ve üst sol köşe olmak üzere köşelerinin koordinatlarını içerir.

TrueType Fontlar

Yazıyı yazmaksızın bu değerleri elde etmek için ise ;

```
array imageTTFBox ( int boyut, int aci, string fontfile,  
    string text)
```

fonksiyonu kullanılabilir.

FreeType 2 için de bu fonksiyonların karşılıkları şöyledir :

```
array imageFTText ( handler resim, int boyut, int aci,  
    int x, int y, int renk, string fontfile, string text,  
    array extrainfo )
```

```
array imageFTBBox ( int boyut, int aci, string fontfile,  
    string text, array extrainfo)
```

extrainfo dizi parametresi, PHP manualında opsiyonel olarak belirtilmiş olsa da bazı versiyonlarda mutlaka boş bir dizi olarak belirtilmesi gerekebilir. (array () gibi) . Bu parametre şu anda belgelendirilmemiştir.

Kopyalama, Boyutlandırma, Kesme – Biçme :-)

GD , üzerinde çalıştığımız resimler arasında kopyalama, resimlerin yeniden boyutlandırma gibi işlemler için kullanılabilecek bir seri fonksiyon da sunar :

```
int imageCopy ( handler hedef, handler kaynak,  
    int hed_x, int hed_y, int kay_x, int kay_y, int kay_w,  
    int kay_h)
```

kaynak resminin, kay_x, kay_y noktasından başlayan ve kay_w * kay_h boyutlarındaki parçasını, hedef resmi üzerine, hed_x, hed_y koordinatlarından başlayacak şekilde kopyalar.

Kopyalama, Boyutlandırma, Kesme – Biçme :-)

```
int imageCopyMerge ( handler hedef, handler kaynak,  
    int hed_x, int hed_y, int kay_x, int kay_y,  
    int kay_w, int kay_h, int yuzde)
```

Fonksiyon, `imageCopy` gibi çalışır fakat sonuç, oran değerine göre her iki resmin karışımı olacaktır. Oran 1 - 100 aralığındadır ve 100 değerine doğru yaklaştıkça kaynak resmin karışımındaki ağırlığı artacaktır.

```
int imageCopyResized ( resource hedef, resource kaynak,  
    int hed_x, int hed_y, int kay_x, int kay_y, int hed_w,  
    int hed_h, int kay_w, int kay_h)
```

Kaynak ve hedefin farklı büyüklüklerde olması durumunda, bu fonksiyon ile yeniden boyutlandırma (resizing) yapılarak kopyalama gerçekleştirilebilir.

Kopyalama, Boyutlandırma, Kesme – Biçme :-)

```
int imageCopyResampled ( resource hedef, resource kaynak,  
    int hed_x, int hed_y, int kay_x, int kay_y, int hed_w,  
    int hed_h, int kay_w, int kay_h)
```

imageCopyResized ile aynı şekilde çalışan bu fonksiyon, resizing yerine interpolation kullanarak yeniden boyutlandırma sırasında oluşan renk değişimlerinde daha başarılı sonuçlar verir.

```
handler imageRotate ( handler resim, float aci, int renk )
```

resim resmini aci değeri kadar döndürür. Bu sırada oluşabilecek boş alanlar renk rengi ile doldurulur.

Kaydetme - Görüntüleme

Resimler üzerinde istediğimiz değişiklikleri yaptıktan sonra onları kaydetmek veya çıktı olarak görüntülememiz gerekecektir. Bu işlemler için bir seri fonksiyon kullanacağız. Kaydetme ve çıktı fonksiyonları temelde aynı fonksiyonlar olmalarına rağmen, kullanım şekline göre kaydetmek veya browser'e çıktı göndermek için kullanılabilirler.

Browser'a çıktı gönderileceği durumlarda, header () fonksiyonu kullanılarak uygun image header'lari gönderilmiş olmalıdır. Ayrıca bu kullanımda, header () fonksiyonundan önce browser'a herhangi bir bilgi yollanmamış olmalıdır (echo vb. ile)

Resmin kaydedileceği durumlarda ise, web sunucusunun veya komut satırından php çalıştıran kullanıcının resmin kaydedileceği klasöre yazma izni olduğundan emin olunmalıdır.

Kaydetme - Görüntüleme

```
int imagePNG ( handler resim [, string dosya_adi])
```

```
int imageJPEG ( handler_resim [, string dosya_adi [,  
    int kalite]])
```

Fonksiyonlar, `dosya_adi` parametresi boş ise browser'a çıktı verirler. Aksi durumda ise `remi dosya_adi` adı altında kaydetmeye çalışırlar. `imageJPEG` fonksiyonundaki `kalite` parametresi, JPEG dosyanın sıkıştırma oranını belirler ve 0 - 100 arasında değerler alabilir (0 en düşük, 100 en yüksek kalite. Öntanımlı değer 75 civarındadır.)

```
int imageWBMP ( handler resim [, string dosya_adi [,  
    int renk]])
```

```
int imageGIF ( handler resim [, string dosya_adi])
```

```
int imageGD ( handler resim [, string dosya_adi])
```

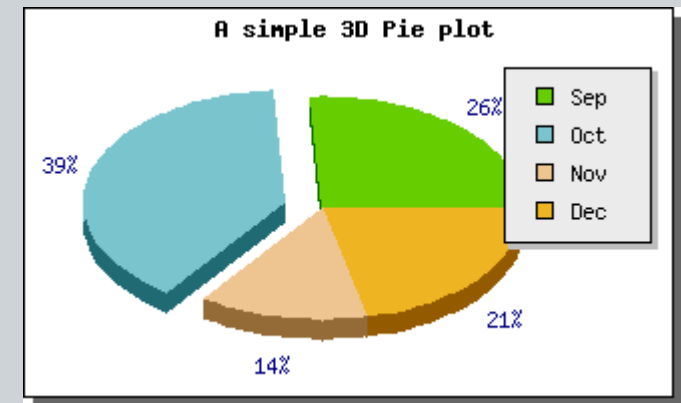
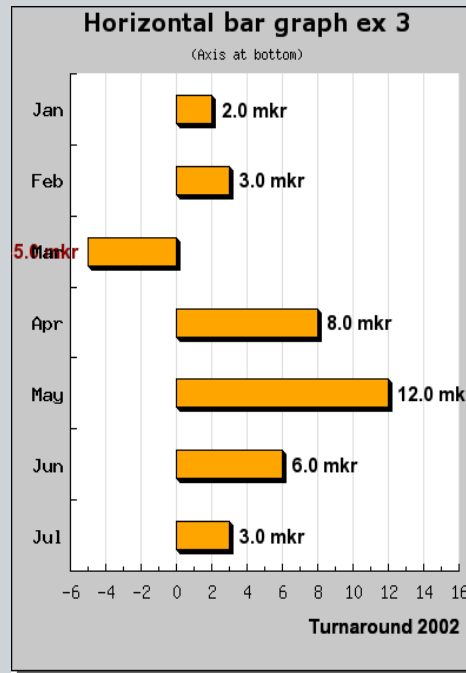
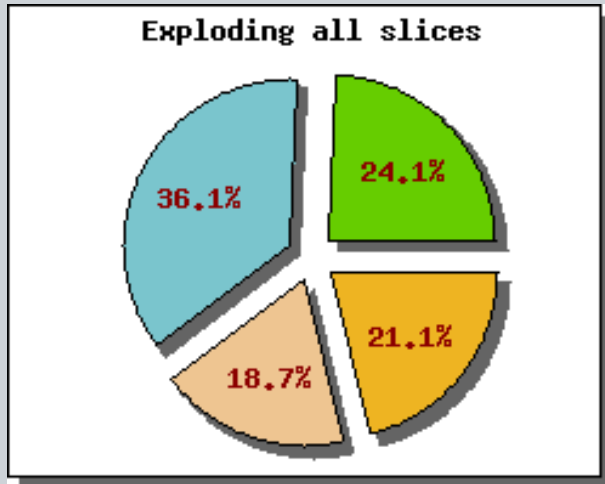
```
int imageGD2 ( handler resim [, string dosya_adi [,  
    int chunk_size [, int type]])
```


Uygulamalar – jpGraph

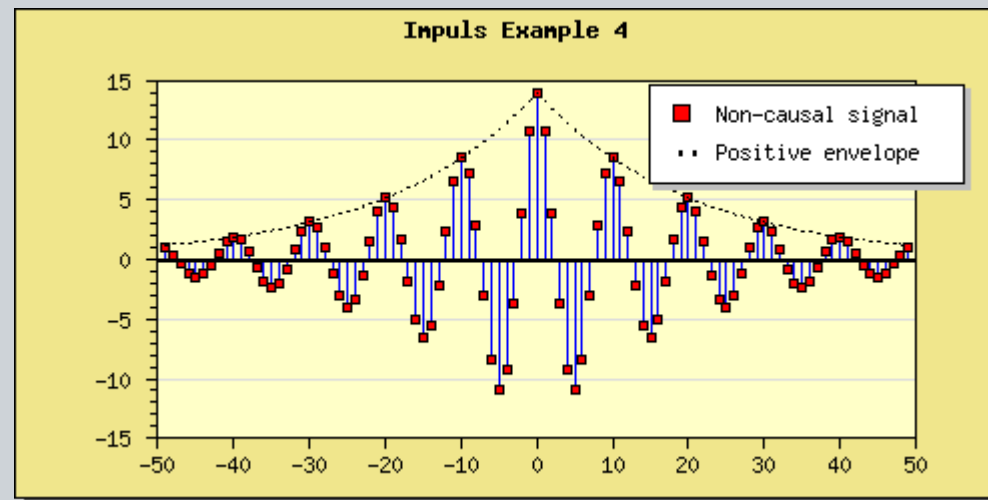
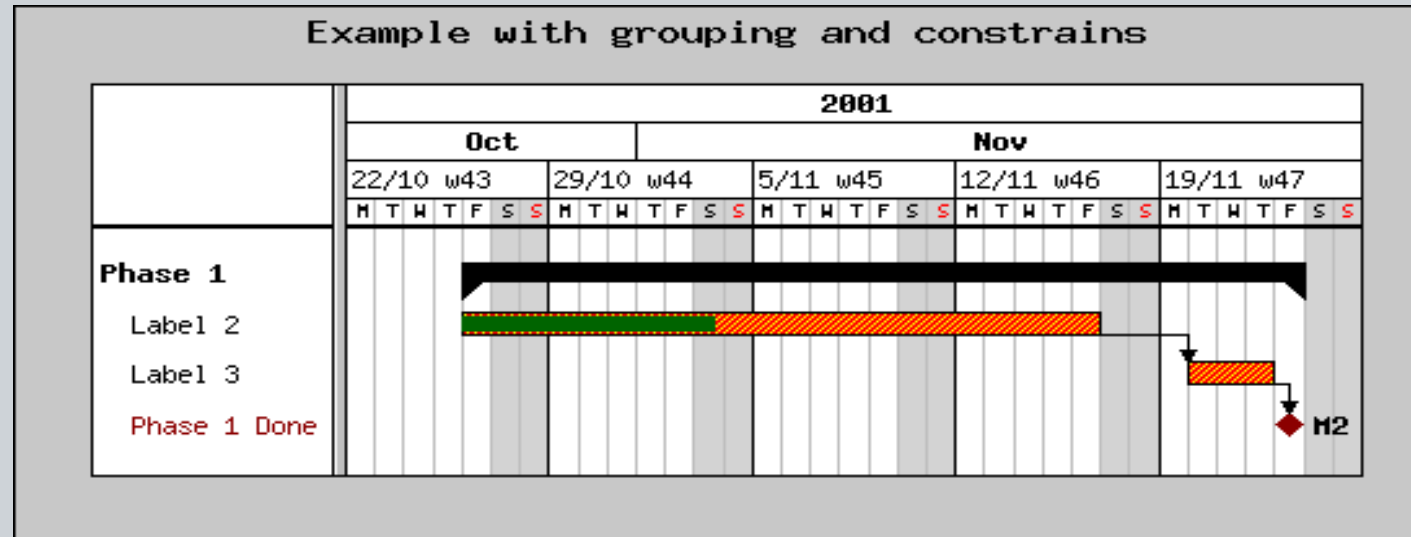
JpGraph

Php için Object Oriented bir grafik sınıf kütüphanesidir ve GD kütüphanesini kullanmaktadır.

JpGraph, her türlü standart bar grafik, dilim grafikleri gibi grafikleri hazırlamak için sınıflar sunar.



Uygulamalar – jpGraph



Birkaç Kaynak ...

PHP Manual :

<http://www.php.net/manual/en/ref.image.php>

GD Graphics Library:

<http://www.boutell.com/gd/>

PHP ve GD'ye giriş :

<http://codewalkers.com/tutorials.php?show=3>

PHP 4.3 altında GD kurulumu :

http://www.onlamp.com/pub/a/php/2003/03/27/php_gd.html

PNG Anasayfası :

<http://www.libpng.org/pub/png/>

NYPHP :

<http://nyphp.org/>

linux-programlama@liste.linux.org.tr

Bu Belge Hakkında.

Bu belgeyi, Free Software Foundation tarafından yayınlanmış bulunan GNU Özgür Belgeleme Lisansının 1.2 ya da daha sonraki sürümünün koşullarına bağlı olarak kopyalayabilir, dağıtabilir ve/veya değiştirebilirsiniz. Lisans'ın bir kopyasını <http://www.gnu.org/copyleft/gfdl.html> adresinde bulabilirsiniz.

Bu belgedeki bilgilerin kullanımından doğacak sorumluluklar ve olası zararlardan belge yazarı sorumlu tutulamaz. Bu belgedeki bilgileri uygulama sorumluluğu uygulayana aittir.

Tüm telif hakları aksi özellikle belirtilmediği sürece sahibine aittir. Belge içinde geçen herhangi bir terim, bir ticari isim ya da kuruma itibar kazandırma olarak algılanmamalıdır. Bir ürün ya da markanın kullanılmış olması ona onay verildiği anlamında görülmemelidir.

Belgenin ilk sürümü : 07 Şubat 2004, *Ekin Meroğlu* <ekin@fisek.com.tr>
Belgenin bu sürümü : 01 Eylül 2004, *Ekin Meroğlu* <ekin@fisek.com.tr>

PHP ve GD ile Resim İşleme

Sorularınız ..

TKD
Linux Kullanıcıları Derneği