

**T.C.
MİLLÎ EĞİTİM BAKANLIĞI**

BİLİŞİM TEKNOLOJİLERİ

**AÇIK KAYNAK KODLA GRAFİK
İŞLEMLERİ VE NESNEYE YÖNELİK
PROGRAMLAMA
482BK0134**

Ankara, 2012

- Bu modül, mesleki ve teknik eğitim okul/kurumlarında uygulanan Çerçeve Öğretim Programlarında yer alan yeterlikleri kazandırmaya yönelik olarak öğrencilere rehberlik etmek amacıyla hazırlanmış bireysel öğrenme materyalidir.
- Millî Eğitim Bakanlığınca ücretsiz olarak verilmiştir.
- **PARA İLE SATILMAZ.**

İÇİNDEKİLER

AÇIKLAMALAR	iii
GİRİŞ	1
ÖĞRENME FAALİYETİ-1	3
1. GRAFİK İŞLEMLERİ	3
1.1. Basit Grafik Oluşturma	4
1.1.1. header() Deyimi	4
1.1.2. imagecreatetruecolor Deyimi	5
1.1.3. imagedestroy Deyimi	5
1.1.4. imagecreate Deyimi	6
1.2. Grafiklere Renk Verme	7
1.2.1. imagecolorallocate Deyimi	7
1.2.2. imagefill Deyimi	8
1.2.3. imagefilledrectangle Deyimi	9
1.3. Grafik Türleri İle Çalışma	11
1.4. Grafikleri Kaydetme	14
1.5. Grafik Üzerine Yazı Yazma	16
1.5.1. imagestring Deyimi	16
1.5.2. imagefttext Deyimi	17
1.6. Güvenlik Kodu İçin Grafik Üretme	19
1.7. Grafik Boyutunu Ayarlama	20
1.7.1. Getimagesize Deyimi	20
1.7.2. imagecreatefrompng Deyimi	20
1.7.3. imagecopyresampled Deyimi	21
UYGULAMA FAALİYETİ	23
ÖLÇME VE DEĞERLENDİRME	24
ÖĞRENME FAALİYETİ-2	25
2. NESNE YÖNELİMLİ PROGRAMLAMA	25
2.1. Sınıflar	26
2.1.1. Sınıf Bildirimi	27
2.2. Nesnelere ve Erişim Belirleyiciler	28
2.2.1. Public Erişim Belirleyicisi	29
2.2.2. Private Erişim Belirleyicisi	30
2.2.3. Protect Erişim Belirleyicisi	30
2.2.4. Extends Erişim Belirleyicisi	31
2.2.5. \$this Anahtar Sözcüğü	31
2.3. Metotlar	31
2.3.1. Tek Yönlü Çalışan Metotlar	31
2.3.2. Çift Yönlü Çalışan Metotlar	32
2.4. Yapıcı Metotlar (Constructors)	34
2.5. Kalıtım	35
2.6. Çok Biçimlilik (Polymorphism)	37
UYGULAMA FAALİYETİ	38
ÖLÇME VE DEĞERLENDİRME	39
MODÜL DEĞERLENDİRME	40
CEVAP ANAHTARLARI	43
KAYNAKÇA	45

AÇIKLAMALAR

KOD	482BK0134
ALAN	Bilişim Teknolojileri
DAL/MESLEK	Web Programcılığı
MODÜLÜN ADI	Açık Kaynak Kodla Grafik İşlemleri ve Nesne Yönelimli Programlama
MODÜLÜN TANIMI	Grafik ve nesne yönelimli programlama işlemlerini yapabilmeye yönelik öğrenme materyalidir.
SÜRE	40/32
ÖN KOŞUL	“Açık Kaynak Kodla Form İşlemleri ve Oturum Yönetimi” modülünü tamamlamış olmak
YETERLİK	Grafik ve nesne yönelimli programlama işlemlerini yapmak
MODÜLÜN AMACI	Genel Amaç Grafik ve nesne yönelimli programlama işlemlerini yapabileceksiniz. Amaçlar 1. Grafik işlemlerini yapabileceksiniz. 2. Nesne yönelimli programlama işlemlerini yapabileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Uygulamaların yapılabileceği uygun laboratuvar ortamı
ÖLÇME VE DEĞERLENDİRME	Her faaliyet sonrasında, o faaliyetle ilgili değerlendirme soruları ile kendi kendinizi değerlendireceksiniz. Modül içinde ve sonunda verilen öğretici sorularla edindiğiniz bilgileri pekiştirecek, uygulama örneklerini ve testleri gerekli süre içinde tamamlayarak etkili öğrenmeyi gerçekleştireceksiniz. Sırasıyla araştırma yaparak, grup çalışmalarına katılarak ve en son aşamada alan öğretmenlerine danışarak ölçme ve değerlendirme uygulamalarını gerçekleştireceksiniz.

GİRİŞ

Sevgili Öğrenci,

Her modül bir önceki modülün tamamlayıcısı, sonraki modülün temelini oluşturur. Her modülün kendine özgü bilgi düzeyi olsa da önceki ve sonraki modüllerle birlikte kullanıldığında ancak bir bütünü oluşturur.

Bu modülde grafikler üzerinde çalışmayı ve bilgi güvenliğini sağlamaya yönelik birtakım bilgiler öğreneceksiniz. Dinamik grafikler üretebileceksiniz. Açık kaynak kodun güzelliğini keşfedecek, sınıflara ve nesnelere can vereceksiniz. Programlarınızı parçalara ayırmak ve bu parçaları mantıklı birleştirmenin size ne kadar hız kazandırdığını ilerleyen zamanlarda çok daha iyi anlayacaksınız.

Bu modülü başarı ile tamamladığınızda, bir sonraki modülü daha etkili kullanabileceksiniz.

ÖĞRENME FAALİYETİ-1

AMAÇ

Açık kaynak kod kullanarak grafik oluşturabilecek, grafik üzerine yazı yazabilecek, grafik boyutlandırma, kaydetme ve güvenlik kodu için grafik oluşturabileceksiniz.

ARAŞTIRMA

- Kütüphane nedir?
- Kütüphaneler nasıl kullanılır?
- Grafik işlemleri için hangi kütüphane kullanılır?
- Grafik formatları hangileridir?

1. GRAFİK İŞLEMLERİ

Grafik kelimesinin iki farklı anlamı vardır. Sayısal verilerin çizgilerle ifade edilmesine grafik denir. Bu tanım genelde matematiksel veya istatistiksel işlemlerin sonuçlarını görsel olarak değerlendirmek için kullanılır. Diğer anlamı ise, bir olayı veya nesneyi göstermek için çizilen şekle grafik denir. Grafik işleme ise, bir şekil, fotoğraf veya resim üzerinde düzenleme yapmak anlamına gelir. Bu bölümde daha çok grafik oluşturma, boyutlandırma gibi grafik işlemeye yönelik çalışmalar yapacağız.

PHP de grafik işlemleri gerçekleştirmek için GD Kütüphanesi kullanılır. GD resim işleme amaçlı bir ANSI C kütüphanesidir. Boutell.Com tarafından açık kaynak kodlu özgür bir yazılım olarak geliştirilmekte ve dağıtılmaktadır. GD kütüphanesi, JPEG, PNG, BMP, GIF ve XPM gibi grafik formatlarını destekler. GD kütüphanesi PHP ile birlikte gelmektedir ancak kullanabilmek için etkinleştirilmelidir.

➤ GD kütüphanesinin etkinleştirilmesi

GD eklentisinin etkin olup olmadığını test etmek için test.php dosyasını,

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

şeklinde oluşturarak çalıştırınız. Sayfada GD eklentisinin aktif olduğu görülmelidir.

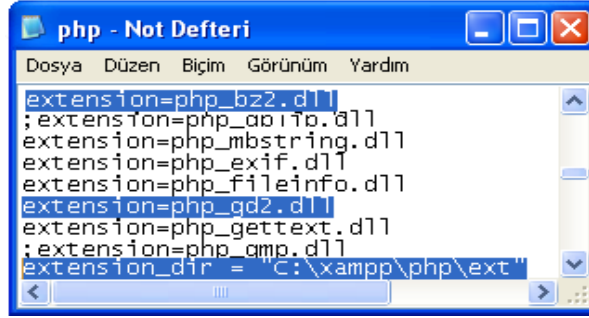
gd

GD Support	enabled
GD Version	bundled (2.0.34 compatible)
FreeType Support	enabled

Resim Hata! Belgede belirtilen stilde metne rastlanmadı..1: GD kütüphanesi

Eğer GD eklentisi etkin değil ise aşağıdaki işlem basamaklarını uygulayınız.

- Gd eklentisi yok ise bu eklenti indirilerek php dizinine kopyalayınız.
- Php dizininde yer alan php.ini dosyasını herhangi bir metin editörüyle açınız.
- extension_dir satırını bulunuz. (Kullanılan php sürümüne göre farklılıklar gösterebilir. Bu satır kütüphanenin hangi dizinde olduğunu gösterir.)
- Varsa önündeki açıklama satırı anlamına gelen ; işareti siliniz.
- ;extension=php_bz2.dll ile başlayan satırı bulup önündeki ';' işaretini kaldırınız.
- ;extension=php_gd2.dll ile başlayan satırı bulup önündeki ';' işaretini kaldırınız.
- Web sunucunuzu (apache) yeniden başlatınız.



```
php - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
extension=php_bz2.dll
;extension=php_dioip.dll
extension=php_mbstring.dll
extension=php_exif.dll
extension=php_fileinfo.dll
extension=php_gd2.dll
extension=php_gettext.dll
;extension=php_omp.dll
extension_dir = 'C:\xampp\php\ext'
```

Resim Hata! Belgede belirtilen stilde metne rastlanmadı..2: php.ini dosyası

1.1. Basit Grafik Oluşturma

Şimdi 200x200 boyutlarında siyah renkli grafiği oluşturan php kodlarını yazalım.



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..3: Default resim

1.1.1. header() Deyimi



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..4: Tabla

Header tabelaya benzer. Nasıl ki bir tabela bir yerin ne amaçla kullanıldığını ifade ediyorsa, header da sayfanın ne sayfası (html, grafik, pdf) olduğunu tarayıcıya bildirir. Tarayıcı header'a bakarak başlık bilgilerine göre sayfa içeriğini işler. İçerik text olabildiği gibi image da olabilir. header ("Content-type: image/png"); Content-type içeriğin ne olacağını tarayıcıya bildirildiği kısımdır. Grafik oluşturulacağı için, içerik image ve içerik formatı olarak da png belirliyoruz.

Dikkat: header komutunu php kodlarının en başına yazmak daha doğru bir uygulama olacaktır.

```
<?php
header ("Content-type: image/png"); //Sayfanın grafik olduğunu tarayıcıya
bildiriyoruz.
?>
```

1.1.2. imagecreatetruecolor Deyimi

Default (varsayılan) renkte (siyah) bir resim oluşturur. İki parametresi vardır. Bu parametreler ile resmin boyutunu ayarlanır.

```
imagecreatetruecolor($en,$boy);
```

Örnek: imagecreatetruecolor komutunun kullanımı.

```
<?php
header ("Content-type: image/png"); //Sayfanın grafik olduğunu tarayıcıya
bildiriyoruz.
$resim = imagecreatetruecolor(200,200); //Resmimizi oluşturuyoruz
imagepng ($resim); //Resmimizi png formatına çeviriyoruz ve tarayıcıda
görüntülüyoruz
?>
```

Kullanıcı tercihi arkaplan ayarlamak istiyorsanız imagecreate komutu kullanmalısınız.

1.1.3. imagedestroy Deyimi

Oluşturulan her resim hafızada yer kaplar. Bu resimler oluşturulduktan sonra hafızadan silinmez ise sunucu gereksiz yere meşgul edilmiş olur. Bu durum, bir süre sonra sunucumuzun yavaşlamasına hatta hizmet veremeyecek duruma gelmesine sebep olacaktır. Bu tür problemlerin önüne geçmek için oluşturulan resimler hafızadan silinmelidir. imagedestroy deyimi resim için ayrılan belleği serbest bırakır.

```
imagedestroy($resim);
```

Örnek: imagedestroy komutunun kullanımı.

```
<?php
header("Content-type: image/png"); //Sayfanın grafik olduğunu tarayıcıya bildiriyoruz.
$resim = imagecreatetruecolor(200,200); //Resmimizi oluşturuyoruz
imagepng($resim); //Resmimizi png formatına çeviriyoruz ve tarayıcıda görüntülüyoruz
imagedestroy($resim); //Oluşturduğumuz resmi hafızadan siliyoruz ?>
```

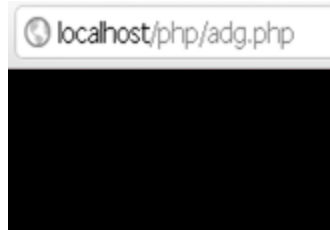
1.1.4. imagecreate Deyimi

Kullanıcının istediği renkte ve boyutta resim oluşturmasını sağlar. Bu komut ile resim oluştururken resmin renginin de belirlenmesi gerekir. Aksi halde varsayılan renk olarak siyah kullanılır. Aşağıdaki örnekte yine varsayılan renkte bir resim oluşturalım ve diğer konumuzla birlikte renkli arkaplanın nasıl oluşturulacağını öğrenelim.

Kullanımı: imagecreate (int \$genişlik , int \$yükseklik)

Örnek : imagecreate komutu ile varsayılan resim oluşturma.

```
<?php
header("Content-type: image/png"); //Sayfanın grafik olduğunu tarayıcıya bildiriyoruz.
$resim = imagecreate(250,250); //Resmimizi oluşturuyoruz.
imagepng($resim); //png formatında grafiği oluşturuyoruz ve tarayıcıda görüntülüyoruz.
imagedestroy($resim); //Oluşturduğumuz resmi hafızadan siliyoruz. ?>
```



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..5: Ekran Çıktısı

Örnek :

	➤ Genişlik ve Uzunluk değerleri girilen imajı sonuc.php sayfasında gösteren programı yapınız?
--	---

1.2. Grafiklere Renk Verme




1.2.1. imagecolorallocate Deyimi

imagecreate komutunu açıklarken renkli arka plan oluşturabileceğimizi ve resim oluşturma esnasında arkaplan renginin ayarlanması gerektiğini aksi halde default (varsayılan) değer olan siyah arkaplan rengi oluşturulacağından bahsetmiştik. Şimdi istenilen renkte nasıl arkaplan oluşturulacağını öğrenelim. Resim oluşturma esnasında arkaplanı renklendirmek için **imagecolorallocate** komutu kullanılır. Renklendirmek için RGB (R: Kırmızı, G: Yeşil, B: Mavi) olarak nitelendirilen 3 temel rengin değerleri kullanılır. 3 rengin değeri ayrı ayrı değiştirilerek tonlama yapılabilir. Renk değerleri 0–255 arasında olabilir. Ayrıca renkler 0xff gibi hexadecimal olarak da ayarlanabilir.

`imagecolorallocate ($resim int $kırmızı , int $yeşil , int $mavi);`

Mavi renk oluşturmak için kullanılan renk kodunu R=0, G=0; B=205 olarak belirledik. Siz de bu değerlerle oynayarak istediniz rengi oluşturabilirsiniz. Ancak RGB değerlerinin herbiri 0-255 arasında olmalıdır.

Bazı renk kodları:

R, G, B	Sonuç
138,43,226	
0,206,209	
139,69,19	

Tablo 1.1: RGB renk kodları

Örnek : Kırmızı renkli 250x250 boyutlarında bir resim oluşturma

```
<?php
header("Content-type: image/png"); //Sayfanın grafik olduğunu tarayıcıya bildiriyoruz.
$resim = imagecreate(250,250); //Resmimizi oluşturuyoruz.
imagecolorallocate( $resim, 255,0,0 ); //Resmin rengini kırmızı olarak belirliyoruz.
imagepng( $resim); //png formatında grafiği oluşturuyoruz ve tarayıcıda görüntülüyoruz.
imagedestroy($resim); //Oluşturduğumuz resmi hafızadan siliyoruz.      ?>
```



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..6: Kırmızı

1.2.2. imagefill Deyimi

Resim boyamak için kullanılır. 4 parametre alır. Resmi `imagecreatetruecolor` komutuyla veya `imagecreate` komutuyla oluşturmuş olsanız bile bu komut ile istediğiniz renge boyayabilirsiniz.

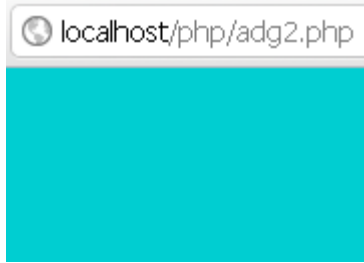
```
imagefill($image, x, y, $color);
```

Parametrelerin görevleri:

- \$image değişkeni: **Boyanacak resmi,**
- x ve y değişkenleri: **Boyamaya başlanacak bölümün X ve Y konumları,**
- \$color değişkeni: **Dolgu rengi, [imagecolorallocate\(\)](#) ile oluşturulan dolgu rengini ifade eder.**

Örnek: Varsayılan renkte oluşturulan grafiği renklendirme

```
<?php
header("Content-type: image/png"); //Sayfanın grafik olduğunu tarayıcıya bildiriyoruz.
$resim = imagecreatetruecolor (250,250);//Resmimizi varsayılan siyah olarak oluşturuyoruz.
$renk=imagecolorallocate( $resim, 0,206,209); //Resmin rengini belirliyoruz.
imagefill ( $resim,0,0,$renk); // Resmimizi boyuyoruz.
imagepng($resim); // png formatında grafiği oluşturuyoruz ve tarayıcıda görüntülüyoruz.
imagedestroy($resim); //Oluşturduğumuz resmi hafızadan siliyoruz.
?>
```



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..7: Ekran Çıktısı

Örnek :

- Yukarıdaki uygulamada `imagefill ($resim,0,0,$renk);` satırını pasif yapıp tekrar çalıştırıp `imagecolorallocate` ve `imagecreatetruecolor` komutları arasındaki farkı gözlemleyiniz.
- Aşağıdaki uygulamayı yapınız.

localhost/php/index.php

Genişlik:

Uzunluk:

R:

G:

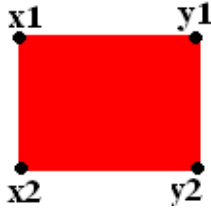
B:

➤ Genişlik, uzunluk ve RGB değerleri girilen grafiği sonuc.php sayfasında gösteren programı yapınız?

1.2.3. imagefilledrectangle Deyimi

Belirtiren *resim* üzerinde başlangıç ve bitiş koordinatları verilen dikdörtgeni *renk* ile boyayarak çizer. Resmin sol üst köşesinin koordinatları 0,0'dır.. 6 parametresi vardır.

`imagefilledrectangle($image , int $x1 , int $y1 , int $x2 , int $y2 , int $renk);`



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..8: Dörtgen parametreleri

Bu parametreler:

- \$image: oluşturulacak resmin adını,
- \$x1,\$y1: 1. Noktanın X ve Y konumu,
- \$x2,\$y2 : 2. Noktanın X ve Y konumu,
- \$renk: Dolgu rengi. imagecolorallocate() ile oluşturulan bir renk. Dörtgenin rengini belirler.

Şimdi 250x250 boyutlarında kırmızı renkli resim oluşturalım. Sonrasında üzerine beyaz ve kırmızı renkte birkaç dörtgen yerleştirelim.



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..9: İç İç Renkler

```
<?php
header("Content-type: image/png"); //Sayfanın grafik olduğunu tarayıcıya bildiriyoruz.
$resim = imagecreate(250,250); //Resmimizi oluşturuyoruz.
```

```

$Kirmizi= imagecolorallocate( $resim, 255,0,0 ); //Resmin rengini kırmızı olarak
belirliyoruz.
$beyaz = imagecolorallocate( $resim, 255,255,255 ); // beyaz rengimizi oluşturuyoruz.
imagefilledrectangle($resim,30,30,210,210,$beyaz); // dörtgenin alanlarını belirliyoruz ve
içini beyaza boyuyoruz.
imagefilledrectangle($resim,50,50,190,190,$kirmizi); // dörtgenin alanlarını belirliyoruz ve
içini kırmızıya boyuyoruz.
imagefilledrectangle($resim,70,70,170,170,$beyaz);
imagefilledrectangle($resim,90,90,150,150,$kirmizi);
imagefilledrectangle($resim,110,110,130,130,$beyaz);
imagepng ( $resim); //png formatında grafiği oluşturuyoruz ve tarayıcıda görüntülüyoruz.
imagedestroy($resim); //Oluşturduğumuz resmi hafızadan siliyoruz.    ?>

```

Şimdi Resim 1.10'da görülen uygulamayı döngü kullanarak nasıl yaparız ona bakalım:



Resim Hata! Belgede belirtilen stilde metne rastlanmadı.**10: İç içe dörtgenler**

```

<?php
header("Content-type: image/png"); //Sayfanın grafik olduğunu tarayıcıya bildiriyoruz.
$resim = imagecreate(250,250); //Resmimizi oluşturuyoruz.
$Kirmizi= imagecolorallocate( $resim, 255,0,0 );//Resmin rengini kırmızı olarak belirliyoruz.
$beyaz = imagecolorallocate( $resim, 255,255,255 ); // beyaz rengimizi oluşturuyoruz.
$a=0;
for($x=10,$y=240;$x<130;$x+=10,$y-=10)//x ve y koordinatlarını ayarlıyoruz.
{
    $a++; // renk değişimini gerçekleştirmek için kullanıyoruz
    if($a %2 == 0)
        $color=$beyaz; // bu kısım bir kırmızı bir beyaz olması için gerekli
    else
        $color=$kirmizi;
    imagefilledrectangle($resim,$x,$x,$y,$y,$color);//rengi göre boyama yapılıyor
}
imagepng ( $resim); //png formatında grafiği oluşturuyoruz ve tarayıcıda görüntülüyoruz.
imagedestroy($resim); //Oluşturduğumuz resmi hafızadan siliyoruz.
?>

```

1.3. Grafik Türleri İle Çalışma

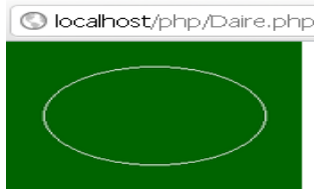
Konumuzun başlangıcında php ile birlikte gelen GD kütüphanesinin hangi grafik türlerini desteklediğinden bahsetmiştik. Şimdiye kadar oluşturmuş olduğumuz tüm resimlerin png türünde olduğu dikkatinizi çekmiştir. Şimdi png dışında hangi grafik türlerinde resim oluşturabileceğimizi öğreneceğiz.

➤ Png türünde resim oluşturmak için:

```
header("Content-type: image/png");  
imagepng ( $resim [, string $dosyaismi [, int $kalite [, int $süzgeçler ]]] );
```

komutları birlikte kullanılır. Buradaki ifadeler;

- \$resim : Resim oluşturma işlevinde dönen resim verisi
- \$dosyaismi : Resmin kaydedileceği dosyanın yolu. Belirtilmez veya NULL verilirse doğrudan ham resim akımı çıktılanır.
- \$kalite : Sıkıştırma düzeyi: 0'dan (sıkıştırma yok), 9'a kadar.
- \$süzgeçler : PNG dosyasının boyutunu küçültme imkanı sağlar.



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..11: Elips

Resim 1.11'deki görüntüyü oluşturan programı yapalım. Elips oluşturmak için `imagefilledellipse` komutu kullanılır. Bu komutun kullanımını aşağıda görmektedir.

```
imagefilledellipse($resim, $x, $y, $genislik, $yukseklk, $renk )
```

- **\$x**: elips merkezinin x koordinatı
- **\$y**: elips merkezinin y koordinatı
- **\$genislik**: elipsin genişliği
- **\$yukseklk**: elipsin yüksekliği
- **\$renk**: elipsin rengi

```
<?php  
header("Content-type: image/png"); // png türünde grafik oluşturacağımızı bildiriyoruz  
$resim = imagecreate(200, 150); // resmin boyutlarını belirliyoruz.  
imagecolorallocate($resim, 0, 100, 0); // Rengi yeşil olarak belirliyoruz.  
$elips = imagecolorallocate($resim, 255, 255, 255); // elipsimizin rengini belirliyoruz.  
imageellipse($resim, 100, 75, 150, 100, $elips); // elipsimizi çiziyoruz.  
imagepng($resim); // resmimizi görüntülüyoruz.  
imagedestroy($resim); // resmimizi hafızadan siliyoruz. ?>
```

➤ Jpeg türünde resim oluşturmak için:

`header("Content-type: image/jpeg");`
`imagejpeg($resim [, string $dosyaismi [, int $kalite]]);` komutları birlikte kullanılır. Buradaki ifadeler;

- `$resim` : Resim oluşturma işleminde dönen resim verisi
- `$dosyaismi`: Resmin kaydedileceği dosyanın yolu. Belirtilmez veya NULL verilirse doğrudan ham resim akımı çıktılır. Kalite değiştirgesini kullanmak için bu değıştirgeye NULL değeri vererek atlayabilirsiniz.
- `$kalite` : kalite isteğe bağı olup 0'dan (en kötü kalite, en küçük dosya) 100'e (en yüksek kalite, en büyük dosya) kadar bir değeri belirtilebilir. Öntanımlı değeri, öntanımlı IJG kalite değeri (yaklaşık 75).



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..12: **Daire**

Resim 1.12'deki görüntüyü oluşturan programı yapalım. Bu görüntüyü hazırlarken sadece elips kullanacağız.

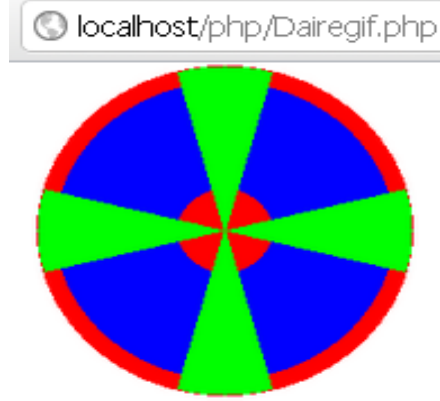
```
<?php
header("Content-type: image/jpeg"); // png türünde grafik oluşturacağımızı bildiriyoruz
$resim = imagecreate(200, 150); // resmin boyutlarını belirliyoruz.
imagecolorallocate($resim, 0, 100, 0); // Rengi yeşil olarak belirliyoruz.
$kafa = imagecolorallocate($resim, 255, 255, 255); // kafanın rengini belirliyoruz.
$goz=imagecolorallocate($resim, 255, 127, 80); // gözlerin rengini belirliyoruz.
imagefilledellipse($resim, 100, 75, 150, 100, $kafa); // (kafa) çiziyoruz.
imagefilledellipse($resim, 65, 65, 30, 20, $goz); // sol gözü çiziyoruz.
imagefilledellipse($resim, 135, 65, 30, 20, $goz); // sağ gözü çiziyoruz.
imagefilledellipse($resim, 100, 70, 10, 50, $goz); // burunu çiziyoruz.
imagefilledellipse($resim, 100, 105, 100, 10, $goz); // ağızı çiziyoruz.
imagepng($resim); // resmimizi görüntülüyoruz.
imagedestroy($resim); // resmimizi hafızadan siliyoruz. ?>
```

➤ **Gif türünde resim oluşturmak için:**

`header("Content-type: image/gif");`
`imagegif($image, $filename);` komutları birlikte kullanılır.

Kullanım amacına göre parametreler deęiştirilebilir. Tarayıcıda görüntülemek için tek parametre yeterli iken, dosya olarak kaydetmek istediğimizde parametre sayısı artırılabilir.

Imagepng (\$image); Tarayıcıda görüntülemek için yeterlidir.



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..13: Pasta dilimi

Resim 1.13'teki pasta dilimini oluşturan kodları yazalım. Bu resmi oluştururken `imagefilledarc` komutunu kullanıyoruz. Bu komutun kullanım şekli aşağıdaki gibidir.

`imagefilledarc($resim , $x,$y,$genislik,$yukseklık,$bas,$son,$renk,$tarz)` komutu belirtilen resim üzerinde merkez koordinatlarına göre yay parçası çizmek için kullanılır.

- **\$x,\$y:** resmin merkezinin x ve y konumunu
- **\$genislik:** Çizilecek yayın genişliğini
- **\$yukseklık:** Çizilecek yayın yüksekliğini
- **\$bas:** Derece cinsinden yayın başlangıç açısını
- **\$son:** Derece cinsinden yayın bitiş açısını
- **\$renk:** `imagecolorallocate()` ile oluşturulan bir renk tanıtıcısı
- **\$tarz:** `IMG_ARC_PIE`, `IMG_ARC_CHORD`, `IMG_ARC_NOFILL`, `IMG_ARC_EDGED` değerlerini alır. Yayın biçimini belirler. Pasta dilimi için `pie` kullanılır.

```
<?php
header("content-type: image/gif"); // gif türünde grafik oluşturacağımızı bildiriyoruz
$resim = imagecreate(400,300); // resmin boyutlarını belirliyoruz
imagecolorallocate($resim, 255, 255, 255); // arkaplan rengini beyaz yapıyoruz
$mavi = imagecolorallocate($resim, 0, 0, 255); // mavi rengi oluşturuyoruz
$yesil = imagecolorallocate($resim, 0, 255, 0); // yeşil rengi oluşturuyoruz
$kirmizi = imagecolorallocate($resim, 255, 0, 0); // kırmızı rengi oluşturuyoruz
imagefilledellipse($resim, 200, 150, 200, 200, $kirmizi); // kırmızı daire oluşturuyoruz
imagefilledellipse($resim, 200, 150, 180, 180, $mavi); // mavi daire oluşturuyoruz
imagefilledellipse($resim, 200, 150, 50, 50, $kirmizi); // kırmızı daire oluşturuyoruz
imagefilledarc($resim, 200, 150, 200, 200, 345, 15, $yesil, IMG_ARC_PIE); // yeşil daire
dilimi oluşturuyoruz
```

```
imagefilledarc($resim, 200, 150, 200, 200, 255, 285, $yesil, IMG_ARC_PIE);
imagefilledarc($resim, 200, 150, 200, 200, 165, 195, $yesil, IMG_ARC_PIE);
imagefilledarc($resim, 200, 150, 200, 200, 75, 105, $yesil, IMG_ARC_PIE);
imagegif($resim); // resmi görüntülüyoruz
imagedestroy($resim); // resmin hafızadan siliyoruz
?>
```

Örnek: Resim 1.14'te gösterilen grafiği hazırlayınız.



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..14: Pati

1.4. Grafikleri Kaydetme

Grafik Türleri İle Çalışma konusunda [imagepng](#) , [imagejpeg](#) , [imagegif](#) ve [imagebmp](#) komutları ile ilgili açıklamalarda bulunur iken, oluşturulan grafiğin kaydedilebileceğinden bahsetmiştik. Bu kısımda oluşturmuş olduğumuz bir resmi nasıl kaydedebileceğimizi öğreneceğiz.

Oluşturulan resmin türüne göre resmin kayıt özellikleri de değişmektedir.

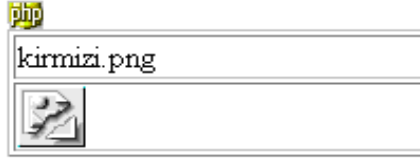
```
imagepng ( $image, $filename, $quality, $filters);
imagejpeg ( $image, $filename, $quality);
imagebmp ( $image, $filename,);
imagegif ( $image, $filename,);
```

Komutları oluşturulan grafiği dosya olarak kaydetmek için kullanılır. Bu kullanımda grafiğin kalite ve filtre gibi özellikleri de ayarlanmak istenirse `$quality` ve `$filter` parametreleride kullanılabilir. Bu parametreler sayısal değerler alırlar. `$quality` parametresi 0-9 arası bir değer almaktadır.

```
<?php
header("Content-type: image/png"); //Sayfanın grafik olduğunu tarayıcıya bildiriyoruz
$resim = imagecreate(250,250); //Resmimizi oluşturuyoruz
imagecolorallocate( $resim, 255,0,0 ); //Resmin rengini kırmızı olarak belirliyoruz
imagepng ( $resim, "kirmizi.png", 8, 10); //png formatında grafiği oluşturuyoruz ve
kirmizi.png ismiyle kaydediyoruz. Kaliteyi 8 ve filtreyi de 10 olarak belirliyoruz
imagedestroy($resim); //Oluşturduğumuz resmi hafızadan siliyoruz
?>
```

Oluşturulan resim dosya olarak websitesi dizinine kaydedilir. Resim tarayıcıda görüntülenmez. Tarayıcıda görüntülemek için `` gibi resim görüntüleme kodları kullanılır. Aşağıda örneği inceleyelim.

➤ Tasarımı resim 1.15'teki gibi hazırlıyoruz.



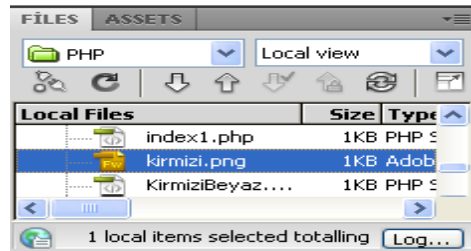
Resim Hata! Belgede belirtilen stilde metne rastlanmadı..15: Tablo

- Sonrasında aynı sayfaya php kodlarını yazıyoruz.

```
<html>
<body> <?php
header("Content-type: image/png");
$resim = imagecreate(250,250); //Resmimizi oluřturuyoruz
imagecolorallocate( $resim, 255,0,0 ); //Resmin rengini belirliyoruz
imagepng( $resim,"kirmizi.png",8,10); // grafięi oluřturuyoruz ve kaydediyoruz
imagedestroy($resim); //Oluruřturduęumuz resmi hafızadan siliyoruz
?>
```

```
<table width="200" border="1">
<tr>
<td>kirmizi.png</td>
</tr>
<tr>
<td></td>
</tr>
</table>
</body>
</html>
```

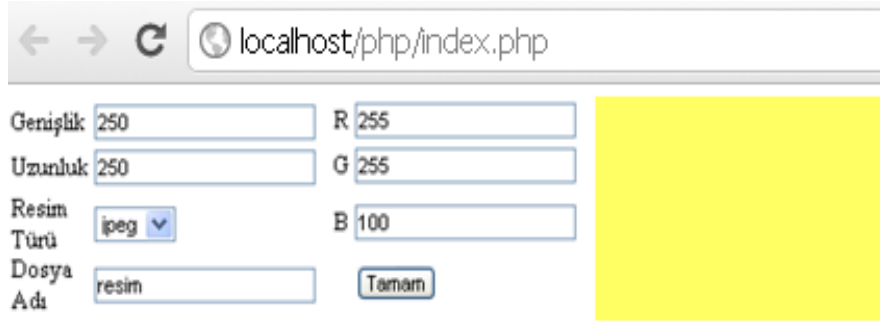
- Sayfayı tarayıcıda açtıęımızda ařaęıdaki gibi bir sonu elde ediyoruz.



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..16: Ekran Çıktısı

Resim Hata! Belgede belirtilen stilde metne rastlanmadı..17: Local görünüm

Örnek : Ařaęıdaki tasarımı gerekleřtiriniz. Girilen bilgilere gre oluřturulan resmi aynı sayfada grntileyen programı yapınız. Kalite ve filtre deęerlerini deęiřtirerek dosya boyutu deęiřikliklerini gzlemleyiniz.



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..18: İsteğe bağlı resim oluşturma

1.5. Grafik Üzerine Yazı Yazma

Her resmin üzerine yazı yazılabildiği gibi, php ile oluşturduğumuz resimlerin de üzerlerine yazı yazabiliriz. Resim üzerine yazı yazmak için ilk yöntem [imagestring](#) metodudur. Bir diğer yöntem ise [imagefttext](#) metodudur.

1.5.1. imagestring Deyimi

[imagestring](#) (\$resim, \$yazıtipi, \$x, \$y, \$metin, \$renk) komutuna ait parametrelerin görevleri şunlardır:

- **\$yazıtipi:** Yazıda kullanılacak yazı tipini ifade eder sayısal değer alır.
- **\$x ve \$y:** Yazının x ve y koordinatlarını belirler.
- **\$metin:** Resim üzerine yazılacak yazıdır.
- **\$renk:** Yazının rengini belirler.



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..19: Resim üzerine yazı

Örnek: Resim Hata! Belgede belirtilen stilde metne rastlanmadı..19'da görülen grafiği oluşturan kodu yazalım.

```
<?php
header('Content-type: image/png');
$resim= imagecreate(200, 100);
imagecolorallocate($resim, 135,206,235);
```

```
$yazirengi = imagecolorallocate($resim, 139,35,35);
imagestring($resim, 5, 20, 20, "CORUM ", $yazirengi); // yazıların konumlarını
imagestring($resim, 7, 180, 20, "19 ", $yazirengi); // ayarlıyoruz ve yazıları
imagestring($resim, 6, 40, 40, "AMASYA ", $yazirengi); // istediğimizi koordinatlara
imagestring($resim, 6, 160, 40, "05 ", $yazirengi); // yazdırıyoruz.
imagestring($resim, 7, 60, 60, "SAMSUN ", $yazirengi);
imagestring($resim, 5, 130, 60, "55", $yazirengi);
imagepng($resim);
imagedestroy($resim);
?>
```

1.5.2. imagettftext Deyimi

Resim üzerine true type yazı yazdırmak için kullanılır. TrueType yazı tipleri her boyuta göre ölçeklenebilir, tüm boyutlarda nettir. Ayrıca bu yazı tipleri, işletim sistemi tarafından desteklenen herhangi bir yazıcı veya başka bir çıkış aygıtına gönderilebilir. Komutun kullanımı aşağıdaki gibidir.

İmagettftext (\$resim, \$boyut, \$saci, \$x, \$y, \$renk, \$yazıtipi, \$metin) komutuna ait parametrelerin görevleri şunlardır.

- **\$boyut:** yazıda boyutunu belirler.
- **\$saci:** Derece cinsinden yazının hangi açı ile yazılacağını belirler.
- **\$x ve \$y:** Yazının x ve y koordinatlarını belirler.
- **\$yazıtipi:** Yazıtipini belirler. ttf uzantılı yazıtipi dosyaları kullanılır.
- **\$renk:** Yazının rengini belirler.
- **\$metin:** Resim üzerine yazılacak yazıdır.



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..20: Açılı yazı

Örnek: Resim **Hata! Belgede belirtilen stilde metne rastlanmadı..20** resmindeki çıktıyı veren programı yazalım.

```
<?php
header("content-type: image/png");
$resim = imagecreate(400,300);
```

```

$Kirmizi = imagecolorallocate($resim, 255, 0, 0);
$beyaz = imagecolorallocate($resim, 255,255,255);
$metin= "BAFRA EML"; //metnimizi belirliyoruz
$boyut=44; // yazı boyutunu belirliyoruz
$aci=340; // yazının açısını belirliyoruz
$x=10; // x koordinatını belirliyoruz
$y=90; // y koordinatını belirliyoruz
imagefttext($resim, $boyut, $aci, $x, $y, $beyaz, "arial.ttf", $metin);
imagepng($resim);
imagedestroy($resim);
?>

```

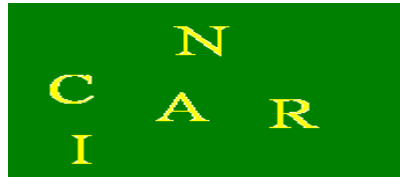
Tru type yazı tipi kullanabilmek için ttf uzantılı yazı tiplerini barındıran dosyayı dizinimize kopyalamamız gerekir.

Örnek : Aşağıdaki görüntüyü oluşturan programı döngü kullanarak yapınız.



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..21: **Açılı yazı -1**

Örnek : Oluşturmuş olduğunuz bir resmin üzerine rastgele harfleri yazan programı yapınız.



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..22: **Harfler**

Örnek: En fazla 6 karakter uzunluğunda girilen bir kelimeyi harflerine ayıran ve bu harfleri resim üzerine rasgele dağıtan programı yapınız.



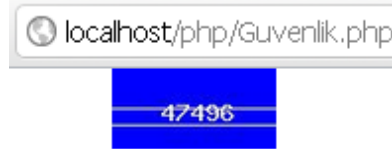
Resim Hata! Belgede belirtilen stilde metne rastlanmadı..23: Harfler-1

1.6. Güvenlik Kodu İçin Grafik Üretme



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..24: Güvenlik

Bir web sitesine üye olurken, unuttuğunuz şifreniz için hatırlatma formu doldururken, bir anket uygulamasına katılırken hatta üye olduğunuz bir web sitesine giriş yaparken bile güvenlik kodları ile karşılaşmışsınızdır. Güvenlik kodunu girmediğinizde veya yanlış girdiğinizde, diğer bilgileriniz doğru olsa bile anket dolduramıyor, kullanıcı girişi yapamıyorsunuz. Peki neden güvenlik kodu kullanılıyor? Bunun sebebi internet haydutları. Bu haydutlar web sitelerini işlemez hale getirmek, kullanıcı şifreleri çalmak için uğraşıyorlar. Bu uğraşlara karşı durabilmek için de site yöneticileri güvenlik önlemi almak zorunda kalıyorlar. Programlar vasıtası ile otomatik kullanıcı girişinin ve şifre denemelerinin önüne geçmek için geliştirilmiş basit ama çok faydalı bir önlemdir güvenlik kodu. Çünkü programların güvenlik kodunu okumaları çok güçtür. Güvenlik kodu sayesinde şifre denemeleri ve otomatik kullanıcı girişi veya kayıt formu doldurmanın önüne geçilmiş oluyor.



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..25: Güvenlik kodu

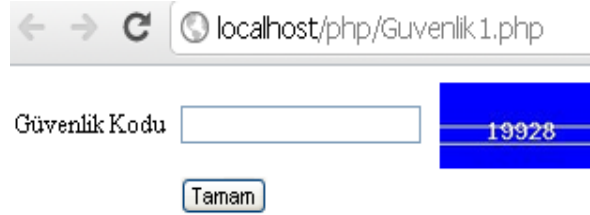
Örnek: Resim 1.25'te görülen güvenlik kodunu oluşturan programı yapalım.

```
<?php
//doğrulama kodunun türünü belirtiyoruz
header("content-Type: image/jpeg");
//doğrulama kodunun boyutunu giriyoruz
$en = 100;
$boy = 50;
```



```
//resimi oluřturuyoruz
$resim = imagecreate($en, $boy);
//kullanacađımız renkleri oluřturuyoruz
$mavi = imagecolorallocate($resim, 0, 0, 255);
$beyaz = imagecolorallocate($resim, 255, 255, 255);
//resmin uzerine yazacađımız kodu rastgele oluřturuyoruz
$GüvenlikKodu=rand(10000,99999);
//rastgele deđerimizi resmin uzerine yazıyoruz
imagestring($resim, 9, 30, 20, $GüvenlikKodu, $beyaz);
//resme çizgi atarak kırılmasını zorlařtırıyoruz
imageline($resim, 0, 25,100, 25, $beyaz);
imageline($resim, 0, 35,100, 35, $beyaz);
//resmi oluřturuyoruz ve tarayıcıda görüntüleyoruz.
imagejpeg($resim);
imagedestroy($resim); //kaynađımızı temizliyoruz
?>
```

Örnek: Ařađıdaki sayfayı hazırlayınız. Kullanıcı kodu dođru girdiđinde sonuc.phpsayfasında “Hoř Geldiniz.” řeklinde uyarı veren programı yapınız.



The screenshot shows a web browser window with the address bar displaying 'localhost/php/Güvenlik.1.php'. Below the address bar, there is a text input field labeled 'Güvenlik Kodu' containing the number '19928'. To the right of the input field is a blue button with the text 'Tamam'.

Resim Hata! Belgede belirtilen stilde metne rastlanmadı..26: Güvenlik kodu -1

1.7. Grafik Boyutunu Ayarlama

Öđrenme faaliyetimiz boyunca hep kendi grafiklerimizi nasıl oluřturabileceđimize, bu grafikler uzerinde nasıl deđişiklik yapacađımıza dair bilgiler öđrendik. řimdi var olan bir grafiđin boyutunu nasıl ayarlayabiliriz onu öđrenelim. Grafik boyutlandırma işlemini gerçekteřtirebilmemiz için birkaç komutu bilmemiz gerekiyor. Bu komutların neler olduđunu teker teker inceleyelim.

1.7.1. Getimagesize Deyimi

Bir resmin boyutlarını öđrenmemizi sađlar. Geriye en ve boy olmak üzere iki deđer gönderir. Gelen deđerler dizi tipindedir.

`$Boyut = getimagesize($resim);` // Boyut[0] deđerine en, Boyut[1] deđerine ise boy deđerleri yüklenir.

1.7.2. imagecreatefrompng Deyimi

Bu komut var olan bir resim dosyasından yeni bir resim dosyası oluřturur.

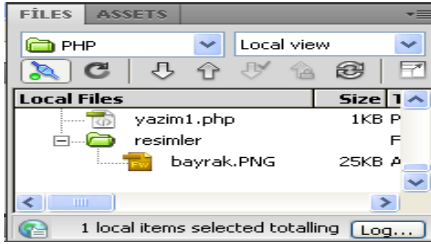
```
$KaynakResim = imagecreatefrompng($resim);
```

1.7.3. imagecopyresampled Deyimi

Bu komutun görevi, bir resimden bir başka resme kopyalama yapmaktır. Boyutlandırma amacıyla kullanılır ve boyutlandırma yapılırken resim kalitesinde bozulma olmaz.

```
imagecopyresampled ($HedefResim, $KaynakResim , $HedefX , $HedefY ,  
$KaynakX, $KaynakY , $HedefEn, $HedefBoy , $KaynakEn , $KaynakBoy )
```

Hedef resmin \$HedefX, \$HedefY noktasından \$HedefEn, \$HedefBoy ile belirlenen parçasını alır, kaynak resmin \$KaynakX, \$KaynakY noktasında, \$KaynakEn ve \$KaynakBoy ile belirlenen parçasına yerleştirerek kopyalama işlemini gerçekleştirir. Resim ölçeklendirme ile ilgili bu komutları öğrendikten sonra bir uygulama ile devam edelim. Öncelikle websitesi dizinimize resimler adında bir klasör oluşturuyoruz ve içerisinde 1024x768 boyutlarında bayrağımızı çizeriz ve bayrak.png olarak kaydediyoruz.



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..27: Local görünüm
Resim Hata! Belgede belirtilen stilde metne rastlanmadı..28: Bayrak

Örnek: Resim boyutlandırma.

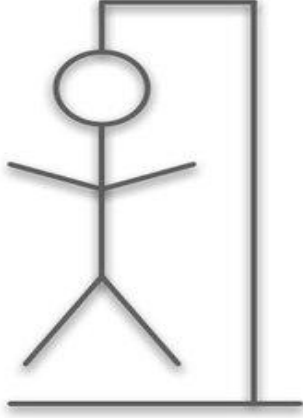
```
<?php  
//Hedef resmimizi alıyoruz.  
$resim = "resimler/bayrak.png";  
//Resmimiz ölçeklendirmek için maksimum ebatları ayarlıyoruz  
$max_Boy =300; //Boy 300 pikselden fazla olamayacak  
$max_En = 300; //En 300 pikselden fazla olamayacak  
//Kaynak resmin boyutlarını alıyoruz.  
$Boyut= getimagesize($resim);  
//Yeni resmin boyutları için oranlama işlemi yapıyoruz.  
$BoyOrani = ($Boyut[0] / $max_En); // hangi oranda küçülteceğimizi ayarlıyoruz  
$EnOrani = ($Boyut[1] / $max_Boy); // Boy oranı 768/300 = 2,56. En oranı 1024/300=3,4  
// Büyük oranı alıyoruz.  
  
if($BoyOrani >=$EnOrani)  
    $Oran = $BoyOrani; // Küçültme oranımız 3,4 olarak belirlendi  
else  
    $Oran = $EnOrani;  
//oluşturacağımız resmin boyutlarını ayarlıyoruz.
```

```
$YeniEn = ($Boyut[0] / $Oran); // YeniEn, 1024/3,4= 401 300 den büyük olduğu için maksimum boyuta ayarlanıyor en 300 olarak belirleniyor.  
$YeniBoy = ($Boyut[1] / $Oran); //YeniBoy 768/4= 225 piksel olarak ayarlandı.  
//Header tanımlamamızı yapıyoruz.  
header("Content-Type: image/png");  
//Resmimizi oluşturuyoruz.  
$KaynakResim = imagecreatefrompng($resim);  
$YeniResim = imagecreatetruecolor($YeniEn,$YeniBoy);  
  
imagecopyresampled($YeniResim,$KaynakResim,0,0,0,$YeniEn,$YeniBoy,$Boyut[0],$Boyut[1]);  
  
imagepng($YeniResim,"bayrak1.png");  
imagedestroy($KaynakResim);  
imagedestroy($YeniResim);  
?>
```



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..**29: Boyutlandırma**

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<ul style="list-style-type: none">➤ Uygulama.php adında bir sayfa oluşturunuz.➤ Sayfanızı grafik olarak belirleyiniz.	<ul style="list-style-type: none">➤ Dreamweaver ya da text editör kullanabilirsiniz.➤ Hangi grafik türünü seçeceğinize dikkat ediniz.
<ul style="list-style-type: none">➤ Resim boyutunu 300x600 olarak belirleyiniz.➤ Arkaplan rengini mor olarak belirleyiniz.	<ul style="list-style-type: none">➤ Grafik oluşturma komutlarının kullanımını hatırlayın.
<ul style="list-style-type: none">➤ Çizgi rengini beyaz olarak belirleyiniz.➤ Aşağıdaki resmi oluşturunuz. 	<ul style="list-style-type: none">➤ Çizgi, elips ve dörtgen çizme komutlarını kullanınız.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. PHP de grafik işlemleri gerçekleştirmek için hangi kütüphane kullanılır?
A) GD B) GP C) DG D) GL
2. Hangisi grafik dosyası değildir?
A) jpeg B) png C) ttf D) bmp
3. Hangisi grafik kütüphanesini aktif yapmak için kullanılan dosyadır?
A) php.cgi B) php.ini C) xampp D) apache.ini
4. imagecreatetruecolor ile oluşturulan resmin arka plan rengi hangisidir?
A) Beyaz B) Kırmızı C) Siyah D) Hiçbiri
5. Hangisi yeni oluşturulan grafiği dosya olarak kaydetmek için kullanılmaz?
A) imagedestroy B) imagepng C) imagejpeg D) imagegif

Aşağıdaki cümlelerde boş bırakılan yerlere doğru sözcükleri yazınız.

6. Tarayıcı bakarak başlık bilgilerine göre sayfa içeriğini işler.
7. Grafikleri renklendirmek için olarak nitelendirilen 3 temel rengin değerleri kullanılır.

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

8. () imagestring grafikler üzerine çizgi çizmek için kullanılır.
9. () imagepng komutu ile jpeg grafik oluşturulabilir.
10. () imagecreate komutu ile oluşturulacak grafiğin boyutu ayarlanabilir.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

ÖĞRENME FAALİYETİ-2

AMAÇ

Açık kaynak kod kullanarak sınıf oluşturmanın, sınıflara nesne ekleyebilmenin temel taşlarını öğreneceksiniz. Sınıf elemanları ekleyebilecek, metotlar tanımlayabilecek ve açık kaynak kodla nesneye yönelik programlamanın temellerini kavrayabileceksiniz.

ARAŞTIRMA

- Programlama türleri hangileridir?
- Nesne tabanlı programlama dilleri hangileridir?
- Sınıf nedir?

2. NESNE YÖNELİMLİ PROGRAMLAMA

Öncelikle nesne nedir? Türk Dil Kurumu nesneyi şöyle açıklıyor: *Belli bir ağırlığı ve hacmi, rengi olan her türlü cansız varlık, şey, obje. Örneğin, ev, araba vb.* Evet gerçek yaşamda nesnenin anlamını öğrendik. Bilgisayar dünyasında da aslında buna benzer bir anlamıyla kullanılır. Her nesnenin kendine ait bir özelliği olduğu gibi bazı ortak özellikleri de bulunabilir. Örneğin lego



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..30: Lego

Legoların ortak özellikleri olduğu gibi farklı özellikleri de vardır. Kimisi kısa iken kimisi uzun, kare, dikdörtgen, oval olabildikleri gibi farklı renklerde de olabilirler. Legoların her biri ayrı ayrı bir nesnedir. Bir araya geldikleri zaman da yeni nesnelere ortaya çıkartırlar. Lego üreticisi, legoları üretirken rengini, ham maddesini veya kalıbını değiştirerek legonun özelliğini de değiştirmiş olur. Bir nesne tek başına tam anlamıyla bir işe yaramaz. Ancak diğer nesnelere bir araya gelerek veya diğer nesnelere etki ederek buldukları sisteme katkı sağlar ki nesnelere görevi de budur aslında. Sistemler nesnelere oluşur ve bir sistemin yürütmesi ancak nesnelere sağlıklı birlikteliği ile olur. Her nesne var olduğu sürece bir

eylemde bulunur. Buradaki kilit nokta ise bir nesne hangi eylem ya da eylemlerde bulunacak. Hemen aklınıza şu soru gelebilir: Hani nesnelere cansız varlıkları? Cansız varlıklar bir eylemde bulunabilir mi? Araba da cansızdır ve bir nesnedir ancak hareket edebilmektedir. Öyleyse eylem yalnızca canlı varlıklar için kullanılan bir sözcük değildir. Dışarıdan bir müdahale ile nesnelere bir eylem gerçekleştirilebilir. Bir nesne oluşturulmadan önce modelleme yapılır. Oluşturulacak nesnenin:

- Özelliği
- Eylemi
- Sistemdeki görevi ve diğer nesnelere ilişkisi

Modelleme esnasında ele alınması gereken konulardır. Legoların üretilmeden önce modelleme ile hangi parça hangi renkte, hangi boyutta ve şekilde olacağı belirlenir. Kullanıcı da legoları kullanarak yeni nesnelere üretirler. Ama her lego tek bir amaca hizmet eder.

Nesneye yönelik programlamada da legolara benzer bir yapı vardır. Birçok nesne oluşturulur ve bu nesnelere bir araya gelerek yeni bir nesne oluştururlar. Yeni oluşturulan nesne, başka bir nesnenin oluşturulması için kullanılabilir. Oluşturulan nesne yeni özelliklere sahip olmakla birlikte oluşturduğu diğer nesnelere özelliklerini de taşır. Bu şekilde programlar parçalara ayrılır ve her bir parçayı bir başka kişi oluşturabilir. Oluşturulan bu nesnelere bir araya gelerek bir program oluşturur.

Nesnelere oluşturulurken öncelikle o nesnenin sınıfı belirlenir. Her nesne bir sınıfa ait olmalıdır. Nesne oluşturulmadan önce sınıflar oluşturulmalıdır. Nesnelere ancak sınıfların belirttiği özelliklere sahip olabilir. Dünyada birçok isimless nesne vardır. Nesnelere ait oldukları sınıflarla isimlendirilir. Örneğin otomobil de bir arabadır, seyyar satıcının tezgâhı da bir arabadır. Peki, sınıf nedir?

2.1. Sınıflar

Nesneye yönelik programlamada sınıf, bir nesnenin tüm özelliklerini ve eylemlerini tanımlayan şablondur. Bir nesne oluşturulacağı zaman sınıfın verdiği bilgilere bakılır ve bu bilgiler doğrultusunda nesne oluşturulur. Sınıflar nesnelere ilk oluşumu sırasında gerekli olan başlangıç durumlarını da belirtir.

Örneğin, bir araba üretmek istediniz. Bu arabaya ait bütün özellikler bir sınıfta toplanmalıdır. Nedir bu özellikler;

- Vites türü
- Modeli
- Yakıt türü
- Kasa tipi

Bu özelliklerin tümü sınıf içerisinde yer almalıdır. Bir sınıf oluşturulurken benzer amaca hizmet eden öğeler aynı sınıf içerisinde toplanmalıdır. Sınıf konusunu daha iyi anlamak için okul örneğini verebiliriz. Okul sınıflardan oluşur ve her sınıfta farklı öğrenme seviyelerinde öğrenciler bulunur. Tabii bu tanıma birleştirilmiş sınıfları katmamamız gerekir.

Nasıl ki bir okulda birden fazla sınıf olabilir ise, bir projede de birden fazla sınıf oluşturma ihtiyacı duyulabilir. Peki sınıf nasıl oluşturulur?

2.1.1. Sınıf Bildirimi

Sınıf oluşturmak için **class** anahtar kelimesi kullanılır. Şimdi otomobil adında bir sınıf tanımlayalım ve bu otomobil sınıfının nasıl kullanıldığını aşama aşama öğrenelim. Bunun için öncelikle **Otomobil.php** adında bir sayfa oluşturalım ve aşağıdaki kodları yazalım.

```
<?php
class Otomobil
{
    /
    / sınıfın üyeleri
}
?>
```

Otomobil adında bir sınıf oluşturduğumuza göre, bu sınıfta yer alacak elemanların hangi özelliklere sahip olacağını, hangi eylemlerde bulunabileceklerini belirleyebiliriz. Sınıfımıza nasıl eleman (veya nesne) ekliyoruz öğrenelim.

2.1.1.1. Sınıf Nesneleri Tanımlama

Sınıf içerisinde tanımlanan nesnelere sınıf nesnesi diyoruz. Sınıf nesneleri değişken, sabit veya metot olabilir. Şimdi sırası ile hepsini öğreneceğiz. Yukarıda bir sınıf nasıl oluşturulur öğrenmiştik. Bu sınıfa nesne nasıl eklenir öğrenelim. Sınıf içerisinde değişken tanımlamak için **var** deyimini kullanıyoruz.

```
<?php class Otomobil
{
    var $Renk;
}
?>
```

Biz de otomobil sınıfı içerisine yeni değişken ekledik. Artık Renk nesnesi Otomobil sınıfının bir elemanı oldu. Bir sınıfa birden fazla nesne ekleyebiliriz.

2.1.1.2. Birden Fazla Sınıf Nesnesi Tanımlama

Sınıf tanımını yaparken bir nesneyi tanımlayan şablondur demiştik. Şablonlar elbette ki birçok özelliği bir arada barındırır. Öyleyse sınıflar da birçok nesneyi barındırabilir. O zaman sınıfımızın elemanlarını artıralım.

```
<?php
class Otomobil
{
    var $Renk;
    var $Marka;
}
?>
```


Yukarıda sınıf içerisinde değişkenler, sabitler ve metotlar gibi nesnelere eklenebildiğinden bahsetmiştik. Değişkenlerin nasıl oluşturulduğunu öğrendik. Şimdi bir de sabit ekleyelim.

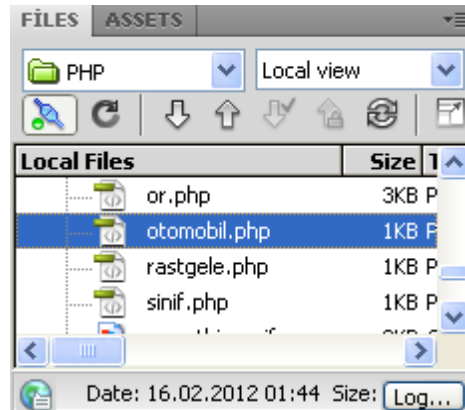
```
<?php
class Otomobil
{
    var $Renk="Kırmızı";
    var $Model="Albina";
    const Marka="Rüzgar; //sabit oluşturma

} ?>
```

Sınıflara değişken ve sabit eklemek bu kadar kolay. Ancak tek başına hiçbir işe yaramazlar. Değişkenler ve sabitler ancak ve ancak bir metot ile kullanıldıklarında değer kazanırlar. Metotlar sınıfların temel taşıdır. Metotlar eylem gerçekleştiren nesnelere ve çok önemlidir. İlerleyen konumuzda metotlar nasıl oluşturulur ve kullanılır öğreneceğiz. Ancak şimdi bir sınıf oluşturduğumuza göre bu sınıftan nasıl yeni bir nesne oluşturabiliriz onu öğrenelim.

2.2. Nesnelere ve Erişim Belirleyiciler

Sınıfın içerisindeki her şey bir nesnedir. Peki bir sınıftan yeni bir nesne nasıl türetilir? Aslında şimdiye kadar değişken veya sabit tanımlamakla bir nesne oluşturmuş oluyoruz. Çünkü değişkenler ve sabitler de bir sınıfın üyeleridir ve biz bu sınıfı kullanarak yeni değişkenler (nesnelere) türetebiliyoruz. Bir sınıftan yeni bir nesne türetmek için **new** anahtar kelimesi kullanılır. Yukarıda Otomobil adında bir sınıf tanımlamıştık. Şimdi bu sınıftan yeni bir nesne türetilim. Öncelikle yukarıda hazırlamış olduğumuz Otomobil sınıfını otomobil.php olarak kaydedelim.



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..31: Otomobil sınıfı

Bir php sayfasını çalışmamıza eklemek için **require** veya **include** komutu kullanılır. Bizde `require("Otomobil.php")` veya `include("Otomobil.php")` komutunu kullanarak

oluşturmuş olduğumuz sınıf dosyamızı yeni oluşturmuş olduğumuz uygulama.php dosyasına çağırıyoruz. Bu şekilde otomobil sınıfındaki tüm nesnelere uygulama.php dosyamızdan ulaşabileceğiz.

Örnek: Sınıf dosyasını çağırma

```
<?php
require "otomobil.php";
?>
```

Otomobil sınıfımızı çağırdığımızı göre bu sınıftan yeni nesneyi `$Arabam = new Otomobil();` komutu ile otomobil sınıfından **Arabam** nesnesini türetiyoruz. **\$Arabam** nesnesinin özelliklerine ulaşmak için `->` işareti kullanılıyor.

Örnek: Nesne türetme

Uygulama.php	Otomobil.php
<pre><?php require "otomobil.php"; \$Arabam = new Otomobil(); \$AracRengi = \$Arabam-> Renk; echo "\$AracRengi
"; \$AracModel=\$Arabam->Model; echo "\$AracModel"; ?></pre>	<pre><?php class Otomobil { var \$Renk="Kırmızı"; var \$Model="Albina"; const Marka="Rüzgar"; }</pre>

```
<?php
require "otomobil.php";
$Arabam = new Otomobil();
$AracRengi = $Arabam->
echo "$AracRengi <br>";
$AracModel=$Arabam->Model;
echo "$AracModel";
?>
```

Model	Otomobil	otomobil.php
Renk	Otomobil	otomobil.php

Resim Hata! Belgede belirtilen stilde metne rastlanmadı..32: Nesne tanımlama

Bir sınıftan türetilen nesnelere türetildiği sınıfın tüm özelliklerine erişemezler. Nesnelerin hangi sınıf elemanlarına erişip erişemeyeceğini belirlemek için **“Erişim Belirleyici”** ler kullanılır. PHP 5 ile gelen bir özellik olan erişim belirleyicilerin neler olduğunu öğrenelim.

PHP de erişim belirleyici olarak public, private, extends ve protected deyimleri kullanılmaktadır.

2.2.1. Public Erişim Belirleyicisi

Public kelimesinin Türkçe anlamı kamu, halka açık, genel gibi anlamları vardır. Bu anlamlardan da anlaşılacağı gibi **public** sözcüğü bir nesneyi herkesin kullanımına açar. Yani bu nesneye her yerden ulaşılabilir. Bir değişkeni veya fonksiyonu **public** olarak tanımladığınızda o nesneye sınıf dışından ulaşabilirsiniz. Bu şekilde tanımlanan sınıfların özellikleri kalıtımla diğer nesnelere aktarılırlar.

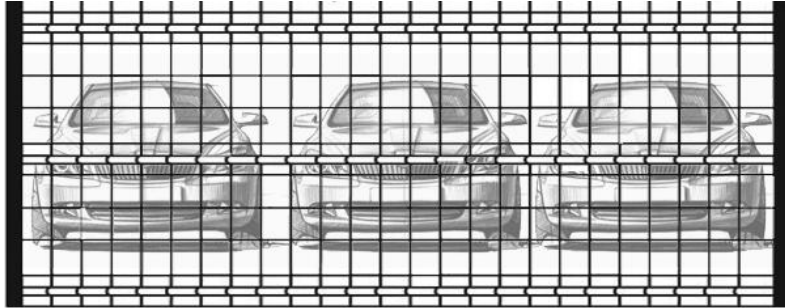
Örneğin; **public \$otobus**; otobus değişkenine artık sınıf dışından da erişebiliriz.



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..33: **Public**

2.2.2. Private Erişim Belirleyicisi

Private kelimesi, özel, gizli, şahsi anlamına gelir. Private olarak tanımladığınız bir nesneye sadece tanımladığınız sınıf içerisinde ulaşabilirsiniz.



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..34: **Private**

2.2.3. Protect Erişim Belirleyicisi

Protect ise korumalı anlamına gelir. Protect olarak tanımladığınız nesneye bulunduğu sınıftan veya alt sınıftan ulaşabilirsiniz.

Örneğin: `protect $otomobil: otomobil` değişkenine artık sınıf dışından erişemeyiz ancak alt sınıflardan erişebiliriz.



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..35: **Protect**

2.2.4. Extends Erişim Belirleyicisi

Oluşturulacak bir sınıf bir başka sınıfın alt sınıfı olacaksa **extends** kullanılır.

2.2.5. \$this Anahtar Sözcüğü

\$this anahtar sözcüğü sınıf içerisindeki bütün nesnelere sınıf içerisinde (metot, değişken) ulaşmak için kullanılır.

2.3. Metotlar

Metotlar sınıflar içerisinde tanımlandığı gibi sınıf dışında da tanımlanabilir. Metotlar, tek taraflı ve çift taraflı olmak üzere iki türdür. Metotları kullanarak sınıf elemanlarının özelliklerini tanımlayabilir, onlara eylem kazandırabiliriz. Metot oluşturmak için **function** deyimini kullanılır.

2.3.1. Tek Yönlü Çalışan Metotlar

Bu tür metotlar işlem yaptıktan sonra geriye herhangi bir değer döndürmez. Sınıf dışından da ulaşabileceğimiz (public), otomobil markası oluşturan bir metot yazalım.

Örnek: Marka metodunun oluşturulması.

Uygulama.php	Otomobil.php
<pre><?php require "otomobil.php"; \$Arabam = new Otomobil(); \$Arabam-> Marka; ?></pre>	<pre><?php class Otomobil { public function Marka() //metot tanımlandı { echo "Hitit Markası Oluşturuldu"; // metodun yapacağı işlemleri belirledik. } } ?></pre>

localhost/php/uygulama.php

Hitit Markası Oluşturuldu

Resim Hata! Belgede belirtilen stilde metne rastlanmadı..36: Marka metodu

Örnek: Albina modelini oluşturan ve kaç adet Albina model araç üretilceğini belirleyen metodu yazalım.

Uygulama.php	Otomobil.php
<pre><?php require "otomobil.php"; \$Arabam = new Otomobil(); \$Arabam-> Albina(5); ?></pre>	<pre><?php class Otomobil { public function Albina(\$Adet) { echo "\$Adet Albina Üretildi"; } } ?></pre>

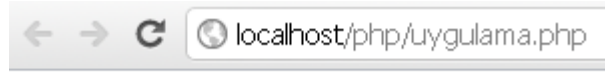
Örnek: Sizde Azemet, Bafcar ve Toprak modellerini ve kaç adet üretilceğini belirleyen metotları yazınız.

2.3.2. Çift Yönlü Çalışan Metotlar

Bu metotlar da kendisine gelen veriye göre işlem gerçekleştirir ve işlem sonucunu çağrıldığı yere kendi ismi ile geri gönderir. Biz bir bilgi gönderdiğimizde , metot da bu bilgiyi işler ve bize geri gönderir. Geriye değer döndürmek için **return** deyimi kullanılır.

Örnek: Albina modelinin hangi renk seçeneklerinin olduğunu ekrana yazan programı yapalım.

Uygulama.php	Otomobil.php
<pre><?php require "otomobil.php"; \$Arabam = new Otomobil(); \$renk = \$Arabam->Renkler(); echo "Otomobil Renk Seçenekleri
"; for(\$a=0;\$a<sizeof(\$renk);\$a++) { echo "\$renk[\$a]
"; } ?></pre>	<pre><?php class Otomobil { //renk seçeneklerini belirliyoruz. //Renk seçeneğinin private olduğuna dikkat edin. private \$Renk=array("Kırmızı","Beyaz","Siyah"); public function Renkler() { //this ile yukarıda tanımladığımız //renk dizisini alıyoruz ve bu diziyi //foknsiyonun çağrıldığı yere gönderiyoruz. return \$this->Renk; } } ?></pre>



Otomobil Renk Seçenekleri
Kırmızı
Beyaz
Siyah

Resim Hata! Belgede belirtilen stilde metne rastlanmadı..37: Ekran çıktısı

```
<?php
$Araban->
$renk = $
echo "Oto
for ($a=0;
(
    echo "$renk[$a] <br>";
```

Marka	Otomobil	otomobil.php
Marka()	Otomobil	otomobil.php
Albina (\$Adet)	Otomobil	otomobil.php
Renkler()	Otomobil	otomobil.php

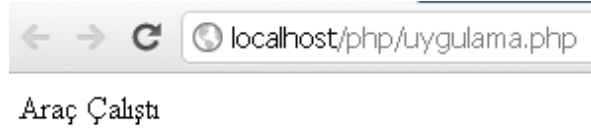
Resim Hata! Belgede belirtilen stilde metne rastlanmadı..38: Public ve Private erişim belirleyici

Nesnelerimizi oluşturduğumuza göre şimdi bu nesnelere eylem kazandıralım. Öncelikle aracımızı çalıştıracak bir metod yazalım.

Örnek: Aracımızı bir anahtar ile çalıştıran metodu yazalım.

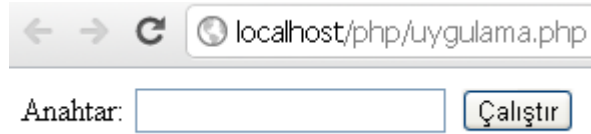
Uygulama.php	Otomobil.php
<pre><?php require "otomobil.php"; \$Arabam = new Otomobil(); \$Sonuc = \$Arabam-> Calistir ("12345"); if(\$Sonuc) echo "Araç Çalıştı ,"; else echo "Yanlış Anahtar"; ?></pre>	<pre><?php class Otomobil { public function Calistir(\$Anahtar) { if(\$Anahtar=="12345") return true; else return false; } } ?></pre>

Örneğimizde Calistir metoduna anahtarı gönderiyoruz. Calistir metodu kendisine gelen bilginin doğruluğuna göre karar veriyor. Anahtar doğru ise Calistir metodu geriye true (doğru), yanlış ise false (yanlış) bilgisini gönderiyor. Gelen sonuca göre aracın çalışıp çalışmadığını biliyoruz ve sonucu ekrana yazıyoruz.



Resim Hata! Belgede belirtilen stilde metne rastlanmadı..39: Ekran çıktısı

Örnek: Kullanıcının girdiği anahtara göre aracın çalışıp çalışmayacağını ekrana yazan programı yapınız.



Resim 2.11: Ekran Çıktısı

- Aracı durduran metodu yazınız.

2.4. Yapıcı Metotlar (Constructors)

Bir nesne oluşturulduğu anda otomatik olarak çalıştırılan metotlara yapıcı metot diyoruz.. Bu metotların kullanım amacı, sınıf elemanlarına ilk değer atamak veya sınıf için birtakım düzenlemeler yapmaktır. Örneğin, otomobil sınıfı için varsayılan renkte, modelde bir araç oluşturma işlemini yapan metotlar gibi.

Yapıcı metotların geri dönüş değeri yoktur ve public olarak tanımlanmalıdırlar. Yapıcı metot oluşturmak için __construct deyimi kullanılır. Şimdi otomobil adlı bir sınıf için yapıcı bir metot oluşturalım.

Örnek: Yapıcı metot

Uygulama.php	Otomobil.php
<pre><?php require "otomobil.php"; \$Arabam = new Otomobil(); ?></pre>	<pre><?php class Otomobil { public function __construct() { echo "Otomobil Sınıfı Başlatıldı"; } } ?></pre>

Uygulama.php sayfası yüklendiğinde ekranda Otomobil Sınıfı Başlatıldı uyarısını göreceksiniz. Dikkat ederseniz, şimdiye kadar tanımladığımız metotların çalışması için o metodu çağırmanız gerekmekte idi. Ancak yapıcı metotlar yalnızca nesne oluşturulduğu anda çalışırlar ve her ne kadar public olarak tanımlı olsalar da dışarıdan bu metoda \$Arabam->__construct şeklinde erişim yoktur. Bir başka örnek ile komumuza devam edelim ve yapıcı metotların bize kazandırdığı avantajları birlikte görelim. Aşağıda yazacağımız

örnek ile yukarıdaki örnekleri çok dikkatli inceleyiniz.. Çünkü yapıcı metotlar bir sınıfı daha geniş kullanma imkânı sunar. Otomobil sınıfı için konuşacak olursak şimdi yapacağımız örnekte farklı markalar ve modeller belirleyebileceğiz. Bir sınıfımız yalnızca bir markaya ait olmayacak. Programcıya sınıfı daha geniş yelpazede kullanabilme olanağı sağlayacaktır.

Örnek: Otomobil sınıfı içerisinde marka, model ve renk bilgisini tutan verileri ekleyelim. Otomobil sınıfının bu 3 üye değişkenini public olarak tanımlayalım.

Uygulama.php	Otomobil.php
<pre><?php require "otomobil.php"; \$Araba = new Otomobil("Rüzgar","Albina","kirmizi"); \$Oto= new Otomobil("Bafcar","Toprak","Beyaz"); echo \$Oto->Marka; echo "
"; echo \$Araba->Model; ?></pre>	<pre><?php class Otomobil { public \$Marka,\$Model,\$Renk; public function __construct(\$Mmarka,\$Mmodel, \$Rrenk;) { \$this->Marka=\$Mmarka; \$this->Model=\$Mmodel; \$this->Renk=\$Rrenk; } } ?></pre>

uygulama.php sayfası yüklendiğinde aşağıdaki çıktıyı elde etmelisiniz. Daha önceki metot uygulamalarında markaları sınıf içerisinde belirliyorduk. Bu uygulamamızda ise aynı sınıftan farklı markalar türetebiliyoruz.



Albina
Bafcar

Resim Hata! Belgede belirtilen stilde metne rastlanmadı..40: Yapıcı metot

Örnek: Sizde otomobil sınıfı hazırlayınız. Otomobilin markasını ve varsayılan rengini belirleyen bir yapıcı metot yazınız. İki farklı marka belirleyerek bu markaları oluşturunuz. Çalıştır ve Durdur metotlarını yazınız. Her iki marka otomobili çalıştıran ve durduran metotları kullanınız. (Anahtar değişkeni private olmalı ve her model için nesne oluşturma esnasında değer aktarılmalıdır.)



Albina çalıştırıldı
Bafcar çalıştırıldı

Resim Hata! Belgede belirtilen stilde metne rastlanmadı..413: Ekran çıktısı

2.5. Kalıtım

Kalıtım aslında biyoloji bilimi ile ilgilidir ve kalıtımın biyolojik anlamda canlının genetik şifresinin kendisinden sonra gelen nesle/yavrulara aktarılmasıdır. Nesneye yönelik programlamada da bir sınıfın özelliklerinin bir başka sınıfa aktarılmasına kalıtım denir.

Kalıtım (inheritance), yeni sınıflar yazarken elimizdeki eski sınıfları kullanmamızı sağlar. Kalıtım, yazılımın yeniden kullanılmasına büyük ölçüde katkıda bulunur. Zaten kalıtım kavramının ortaya çıkmasının sebebi, özellikle büyük çaplı yazılım geliştirme çalışmalarında aynı kod parçalarının tekrar tekrar kullanılmasını önlemek ve yeni sınıfları var olanların üstüne inşa etmektir. Kalıtım yoluyla yeni bir sınıf oluşturmak için **extends** anahtar sözcüğü kullanılır.

Kalıtım konusunu bir örnek ile açıklamaya çalışalım. Geçtiğimiz konularda otomobil isimli bir sınıf oluşturmuştuk. Bu sınıfta marka, model, renk gibi özellikler tanımlamıştık. Şimdi otomobil sınıfını kullanarak yeni bir sınıf daha üretelim. Bu sınıfımızın adı da YarisOtomobili olsun. Buraya dikkat! Yarış otomobili ile binek otomobillerin birçok özelliği aynıdır. Marka, renk gibi. Bunun yanında da motor hacmi, motor gücü gibi birçok özelliği de farklıdır. Kalıtım yoluyla yeni bir sınıf tanımlayarak konumuzu anlaşılır hale getirelim.

Örnek: Kalıtım yoluyla yeni bir sınıf türetmek.

Uygulama.php	Otomobil.php
<pre><?php require "otomobil.php"; \$Yaris=newYarisOtomobili("Hitit","Gunes","Kırmızı"); \$AracimMarka = \$Gunes->Marka; \$AracimModel = \$Gunes->Model; \$AracMotor=\$Gunes->MotorSec(); echo \$AracimMarka,\$AracimModel,\$AracMotor; ?></pre>	<pre><?php class Otomobil { public \$Marka,\$Model,\$Renk; public function __construct(\$Mmarka,\$Mmodel, \$Rrenk;) { \$this->Marka=\$Mmarka; \$this->Model=\$Mmodel; \$this->Renk=\$Rrenk; } } class YarisOtomobili extends Otomobil { public function MotorSec() { return "6.00 Lt"; } } ?></pre>



Hitit Guneş 6.00 Lt

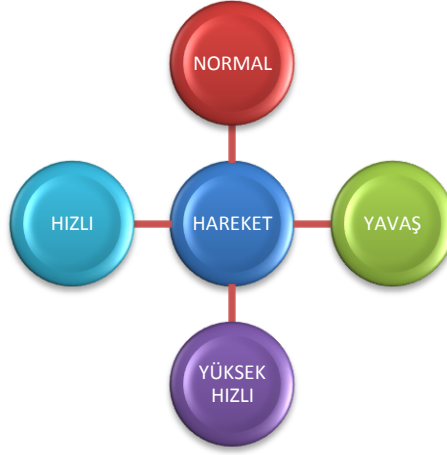
Resim Hata! Belgede belirtilen stilde metne rastlanmadı..14: Kalıtım

Örneğimizi açıklayalım. İlk olarak otomobil sınıfından kalıtım yoluyla **YarisOtomobili** sınıfını oluşturduk. Yeni sınıfımız olan **YarisOtomobili** sınıfı **Otomobil** sınıfındaki tüm nesnelere miras aldı. **YarisOtomobili** sınıfı miras aldığı tüm nesnelere

kullanabildiği gibi kendisine özel yeni nesnelere de sahip olabilmekte. (örneğin MotorSec metodu YarisOtomobili sınıfının elemanıdır). Dikkat ederseniz marka, model ve renk için Otomobil sınıfını, motor seçeneği için **YarisOtomobili** sınıfını kullandık. Bu şekilde kalıtım yoluyla yeni sınıflar türetilir.

2.6. Çok Biçimlilik (Polymorphism)

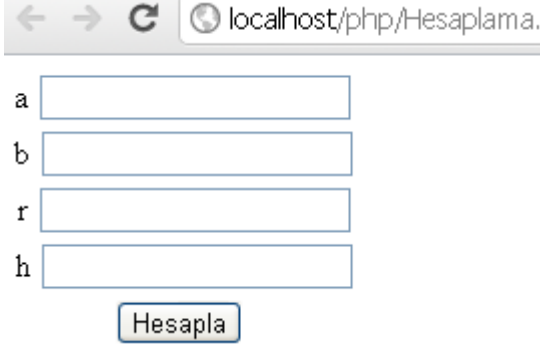
Bir metodun farklı nesnelere farklı sonuçlar üretmesidir. Örneğin hareket etmek bütün vasıtalar için geçerli bir özelliktir.



Resim Hata! Belgede belirtilen stilde metne rastlanmadı.**42: Çok Biçimlilik**

Fakat hareket metodu ait olduğu sınıfa göre farklı sonuçlar üretir. Örneğin otomobil hızlı gidebilirken, otobüs normal hızla gider. Metot aynı ancak sonuçları farklı olabilir.

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<ul style="list-style-type: none">➤ Hesap adında bir sınıf oluşturunuz.➤ Sınıfınıza a,b,r,h değişkenlerini ve pi sabitini tanımlayınız.	<ul style="list-style-type: none">➤ Sınıf oluşturma konusuna göz atın.
<ul style="list-style-type: none">➤ Hesap sınıfına tanımladığınız değişkenlere ilk değer atayan yapıcı metodu yazınız.	<ul style="list-style-type: none">➤ __construct deyimini hatırlayın.
<ul style="list-style-type: none">➤ Hesap sınıfına karenin alanını hesaplayan bir metot yazınız.➤ Hesap sınıfına üçgenin alanını hesaplayan bir metot yazınız.➤ Hesap sınıfına dairenin alanını hesaplayan bir metot yazınız.	<ul style="list-style-type: none">➤ Metodun dışardan değer alacağına dikkat edin.
 <p>➤ Tasarımını gerçekleştiriniz.</p>	<ul style="list-style-type: none">➤ Formlar modülüne göz atınız.
<ul style="list-style-type: none">➤ Yeni bir php dosyası açarak Hesap sınıfınızı bu sayfaya çağırınız.➤ Kullanıcının girdiği bilgilere göre hesaplamaları metotları kullanarak yapan programı yapınız.	<ul style="list-style-type: none">➤ Nesnelere konusuna göz atın.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Bir nesnenin tüm özelliklerini ve eylemlerini tanımlayan şablona ne ad verilir?
A) Sınıf B) Metot C) Function D) Obje
2. Bir sınıf içerisinde değişken tanımlamak için hangisi kullanılır?
A) rar B) var C) dim D) int
3. Aşağıdaki hangi erişim belirleyici bize o nesneye her yerden ulaşmamızı sağlar?
A) private B) extends C) protect D) public
4. Bir php dosyasını çağırmak için hangisi kullanılır?
A) require B) download C) install D) add
5. Sınıf oluşturmak için hangi keyword (anahtar kelime) kullanılır?
A) Close B) New C) Class D) Create

Aşağıdaki cümlelerde boş bırakılan yerlere doğru sözcükleri yazınız.

6. Metot oluşturmak için deyimi kullanılır.
7. anahtar sözcüğü sınıf içerisindeki bütün nesnelere sınıf içerisinde (metot, değişken) ulaşmak için kullanılır.
8. Eylem gerçekleştiren nesneleredenir.

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

9. () Belli bir ağırlığı ve hacmi, rengi olan her türlü canlı, cansız varlığa nesne denir.
10. () Her nesne bir sınıfa aittir.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise “Modül Değerlendirme”ye geçiniz.

MODÜL DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. GD kütüphanesinin görevi nedir?
A) Grafik işlemlerini gerçekleştirmek
B) Nesne oluşturmak
C) Nesneye yönelik programlama yapmak
D) Hiçbiri
2. Sayısal kameralardan bilgi okumak için hangi eklenti kullanılır?
A) GD
B) exif
C) Class
D) read
3. Aşağıdakilerden hangisi gd eklentisinin çalıştığını test etmeye yarar?
A) php
B) info
C) phpinfo
D) php.ini
4. Aşağıdakilerden hangisi php.ini dosyasında açıklama satırı oluşturur?
A) //
B) ;
C) :
D) ”
5. Aşağıdakilerden hangisi resmi hafızadan siler?
A) image
B) imagepng
C) deleteimage
D) imagedestroy
6. Aşağıdakilerden hangisi grafik oluşturmak için kullanılır?
A) imagecreate
B) imagebuilt
C) image
D) create
7. Aşağıdakilerden hangisi imagecreatetruecolor komutu ile oluşturulan resmi renklendirir?
A) imagecolorallocate
B) imagefill
C) imagedestroy
D) Hiçbiri

8. Aşağıdakilerden hangisi daire oluşturmak için kullanılır?
A) imagecolorallocate
B) imagefilledrectangle
C) imagecreate
D) imageellipse
9. Aşağıdakilerden hangisi yay çizmek için kullanılır?
A) imagefill
B) imagepng
C) imagefilledarc
D) imagefillrectangle
10. Aşağıdakilerden hangisi grafik üzerinde pasta dilimi oluşturmak için kullanılan bir parametredir?
A) IMG_ARC_PIE
B) IMG_ARC_EDGED
C) IMG_ARC_CHORD
D) IMG_ARC_NOFILL
11. Grafik üzerinde açılı yazı yazmak için aşağıdakilerden hangisi kullanılır?
A) imagestring
B) imagecreate
C) imagewrite
D) imagettftext
12. Grafik üzerinde çizgi çizmek için aşağıdakilerden hangisi kullanılır?
A) imagepaint
B) imageelips
C) imageline
D) imasetline
13. \$a=rand(3,15) komutu çalıştığında “a”nın değeri hangisi olamaz?
A) 3
B) 13
C) 15
D) 18
14. Aşağıdakilerden hangisi erişim belirleyicisi değildir?
A) var
B) public
C) private
D) extends
15. Bir metodun farklı nesnelere farklı sonuçlar üretmesine ne denir?
A) Kalıtım
B) Çok Biçimlilik
C) Metot
D) Yapıcı

Aşağıdaki cümlelerin sonunda boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

16. () Kalıtım yoluyla bir sınıf oluşturmak için extends erişim belirleyici kullanılır.
17. () Bir nesne elemanı olduğu sınıfın bütün elemanlarına dışarıdan ulaşabilir.
18. () Bir php sayfasını bir başka sayfaya yüklemek için install komutu kullanılır.
19. () Function deyimi sınıf oluşturmak için kullanılır.
20. () -> işareti sınıf nesnelere ulaşmak için kullanılır.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki modüle geçmek için öğretmeninize başvurunuz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ-1'İN CEVAP ANAHTARI

1	A
2	C
3	B
4	C
5	A
6	header
7	RGB
8	Yanlış
9	Yanlış
10	Doğru

ÖĞRENME FAALİYETİ-2'NİN CEVAP ANAHTARI

1	A
2	B
3	D
4	A
5	C
6	function
7	this
8	metot
9	Doğru
10	Doğru

MODÜL DEĞERLENDİRMENİN CEVAP ANAHTARI

1	A
2	B
3	C
4	B
5	D
6	A
7	B
8	D
9	C
10	A
11	D
12	C
13	D
14	A
15	B
16	Doğru
17	Yanlış
18	Yanlış
19	Yanlış
20	Doğru

KAYNAKÇA

- ARTYMIÁK Jacek, **Dynamic Bitmap Graphics With PHP and gd**
- SKLAR David, **Learning PHP 5**
- POWERS David, **PHP Object Oriented Solutions**
- ocw.metu.edu.tr/