

**T.C.  
MİLLÎ EĞİTİM BAKANLIĞI**

## **BİLİŞİM TEKNOLOJİLERİ**

**AÇIK KAYNAK KODLA TEMEL  
UYGULAMALAR  
482BK0135**

**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. KURULUM.....	3
1.1. Web Sunucusu .....	3
1.2. ISS ve PHP.....	4
1.2.1. Apache Kurulumu.....	4
1.2.2. Php Kurulumu.....	11
1.3. Veri Tabanı Sunucusunun Kurulması .....	14
1.3.1 MySQL Kurulumu.....	15
1.4. Hazır Paketler.....	23
UYGULAMA FAALİYETİ .....	28
ÖLÇME VE DEĞERLENDİRME .....	31
ÖĞRENME FAALİYETİ-2 .....	32
2. TEMEL İŞLEMLER .....	32
2.1. Programların Kaydedilmesi .....	32
2.2. Kodların Yazılışı .....	33
2.3. Yorum Satırı.....	34
2.4. Editör Kullanarak Hızlı Kod Yazma ve Çalıştırma .....	35
UYGULAMA FAALİYETİ .....	40
ÖLÇME VE DEĞERLENDİRME .....	41
ÖĞRENME FAALİYETİ-3 .....	42
3. TEMEL PROGRAMLAMA UYGULAMALARI.....	42
3.1. Değişkenler .....	42
3.2. Sabitler ve Kullanımı .....	44
3.3. Veri Türleri .....	44
3.4. Operatörler .....	45
3.5. Program Denetimi .....	48
3.5.1. If – Else Kullanımı.....	49
3.5.2. Else if .....	51
3.5.3. if Deyimi İle Kullanılan Komutlar .....	53
3.5.4. Switch Kullanımı .....	56
3.5.5. Döngüler .....	59
3.6. Diziler (Array) .....	67
3.7. Fonksiyonlar .....	72
3.7.1. Zaman Fonksiyonları.....	75
3.7.2. Mail ( ) Fonksiyonu .....	80
3.8. Dosya ve Dizin İşlemleri.....	82
3.8.1. file_exists ( ) .....	82
3.8.2. is_file ( ) / is_dir ( ).....	82
3.8.3. mkdir ( ) .....	83
3.8.4. Touch ( ) .....	84
3.8.5. Fwrite ( ) .....	84
3.8.6. Fget ( ) .....	85

---

UYGULAMA FAALİYETİ .....	86
ÖLÇME VE DEĞERLENDİRME .....	89
MODÜL DEĞERLENDİRME .....	91
CEVAP ANAHTARLARI.....	93
KAYNAKÇA .....	94

# AÇIKLAMALAR

<b>KOD</b>	<b>482BK0135</b>
<b>ALAN</b>	<b>Bilişim Teknolojileri</b>
<b>DAL/MESLEK</b>	<b>Web Programcılığı</b>
<b>MODÜLÜN ADI</b>	<b>Açık Kaynak Kodla Temel Uygulamalar</b>
<b>MODÜLÜN TANIMI</b>	Web programlamada gerekli olan sunucu ve derleyicilerin kurulumu ve temel programların çalıştırılması
<b>SÜRE</b>	40/32
<b>ÖN KOŞUL</b>	“Programlama Temelleri” dersi modüllerini tamamlamış olmak.
<b>YETERLİK</b>	Açık kaynak kodlu programlama için gerekli ortamı oluşturarak, temel programlama işlemlerini yapmak.
<b>MODÜLÜN AMACI</b>	<b>Genel Amaç</b> Bu modül ile gerekli ortam sağlandığında web sunucusu kurulumu yapıp temel düzeyde programlar yazabileceksiniz. <b>Amaçlar</b> 1. Web programlama için gerekli yazılımları kurabileceksiniz. 2. Temel programlama işlemlerini yapabileceksiniz. 3. Temel Programlama uygulamalarını yapabileceksiniz.
<b>EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI</b>	Bilgisayar, açık kaynak kod tabanlı web yazılımları, İnternet Bağlantısı
<b>ÖLÇME VE DEĞERLENDİRME</b>	Modül içinde yer alan her öğrenme faaliyetinden sonra verilen ölçme araçları ile kendinizi değerlendireceksiniz. Öğretmen modül sonunda ölçme aracı (çoktan seçmeli test, doğru-yanlış testi, boşluk doldurma, eşleştirme vb.) kullanarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek sizi değerlendirecektir.

# GİRİŞ

## **Sevgili Öğrenci,**

Programlama temelleri dersinde gördüğümüz algoritma ve akış şemasının bir adım ötesi programcılığa giriş olmaktadır.

İnternetin günlük hayatın bir parçası olduğu bu zaman diliminde bankacılıktan alış-verişe, haberleşmeye, eğitime, sağlığa vb. birçok konuda dinamik yapıya sahip web sitelerini ziyaret etmekteyiz.

Günümüz insanının vaktini geçirdiği sosyal paylaşım sitelerinden, forum sitelerinden, haber sitelerinden birinin sahibi de sizler olabilirsiniz.

Sıkça kullandığımız bu tür sitelerin yapılması ve yayınlanması işlemini siz de gerçekleştirebilirsiniz. Bu modülde size bunların yapılması hakkında çeşitli temel bilgiler verilecektir.

Bu konuda sizlere, ihtiyaca uygun web sunucusunu kurma ve programcılığın temelleri hakkında bilgiler verilecektir.



# ÖĞRENME FAALİYETİ-1

## AMAÇ

Açık kaynak kod tabanlı programları çalıştırmak için sunucu ve veri tabanı kurulumlarını gerçekleştirebileceksiniz.

## ARAŞTIRMA

- Host firmalarının verdiği sunucu hizmetlerini araştırınız.
- Açık kaynak kodla hangi tür web sayfaları yapıldığını öğreniniz.

## 1. KURULUM

### 1.1. Web Sunucusu

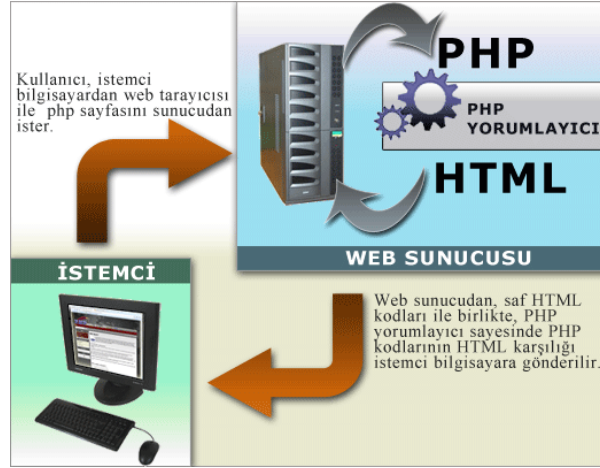
Programcılık temel olarak bir işi daha kolay ve kısa bir biçimde çözme işlemidir. Örneğin bir ismi 200 defa alt alta yazdırma işlemi sizden istendiğinde,” Bunu kolay yoldan nasıl yapabilirim?” düşüncesi aklımıza gelir. Normal yazma işlemine girdiğinizde 200 satır yazmanız gerekecektir. Ama bu işlemi program yardımıyla 3-4 satırda halledebilirsiniz.

Çalışma şekli bakımından **iki tür web sayfası vardır. Birincisi sadece istemci tarafından yorumlanan sayfalar, ikincisi ise sunucu tarafından yorumlanan sayfalardır. İstemciden gelen taleplere cevap veren bilgisayarlara sunucu denir. Web Server(Web Sunucu) istemciden gelen bu talebe bağlı olarak sayfanın sunucu tarafından yorumlanıp yorumlanmayacağına karar vererek gerekli işlemleri yürütür.**

**Günlük hayatta sürekli karşılaştığımız etkileşimli web sitelerinin arka planına bakıldığında sunucu ve derleyicilerden faydalandığı görülmektedir.**

**Alışveriş sitesinde sepete eklediğiniz ürünün tutarını hesaplayan veya paylaşım sitelerinde paylaştığınız bir resmi arkadaşlarınıza da gösteren internet programlarını çalıştıran, onları kullanıcının anlayacağı şekle getiren sunucu ve derleyicidir. Yazılan program, sunucu ve derleyici olmadan hiçbir anlam ifade etmez.**





Resim 1.1: **PHP'nin çalışma prensibi**

PHP yorumlayıcısı ilgili dosya içindeki scriptleri çalıştırarak geriye döndürdüğü sonucu web sunucuya tekrar gönderir. Web sunucuya ulaştırılan sonuç, istemci tarafına HTML dosyası olarak yollarır.

## 1.2. ISS ve PHP

PHP ile etkileşimli programlar oluşturmak istiyorsak web sunucu ve yorumlayıcıya ihtiyaç vardır. Birinin olmaması durumunda program çalışmaz.

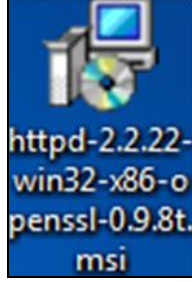
Üye kaydı, ürün bilgisi gibi verilerin kullanılacağı bir web sayfasında ise veri tabanı sunucu gerekir. Veri tabanı sunucusunu daha kolay kontrol etmek ve verileri düzenlemek için de yardımcı veri tabanı programı yönetimi programları ( PhpMyAdmin vb.) kullanılabilir.

### 1.2.1. Apache Kurulumu

Apache bir web sunucu programdır. Bu sunucuyu kurduğunuzda **80. Portu** kullanarak çalışır. Apache güçlü, sağlam, yetenekli ve esnek bir web sunucusudur. **Açık kaynak kodlu bir yazılım olup lisansı ücretsizdir.** Yazılım firmalarına, kurumlara verdikleri hizmetten (kurulum, teknik destek, vb.) kazanç sağlarlar. Apache web sunucu programını <http://www.apache.org> sitesinden ücretsiz indirebilirsiniz.

Biz de bilgisayarımıza bir web sunucu programı yükleyeceğiz. Modülün hazırlandığı sırada sitede ([www.apache.org](http://www.apache.org)) bulunan apache sunucusunun son sürümü **Apache Http Server 2.2.22** olması nedeniyle modül çalışmamızda bu sürümü kullanacağız.

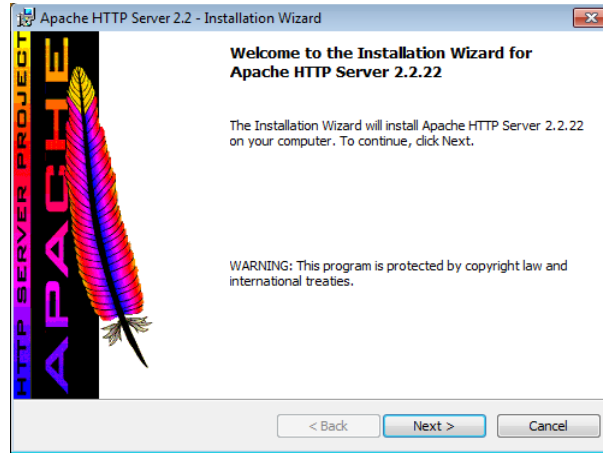
Apache Http Server 2.2.22 programını indiriyoruz.



**Resim 1.2: Kurulum programı**

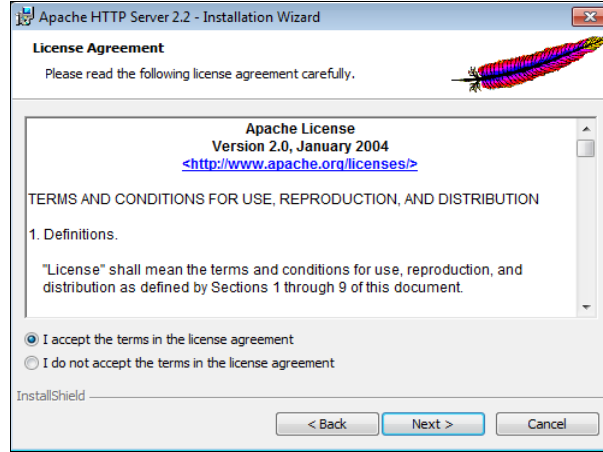


Kurulum esnasında uyarı verebilecek olası güvenlik yazılımları için izin onay vermemiz gerekmektedir.



**Resim 1.3: Kurulum başlama**

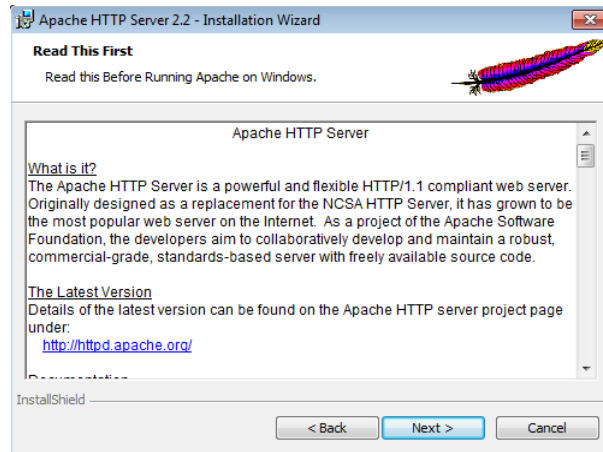
Kurulumun ilk penceresi sürüm ve program hakkında bilgileri içerir. Kurulum devam etmek için karşımıza gelen “Next” butonuna tıklıyoruz.



**Resim 1.4: Lisans sözleşmesi onay penceresi**

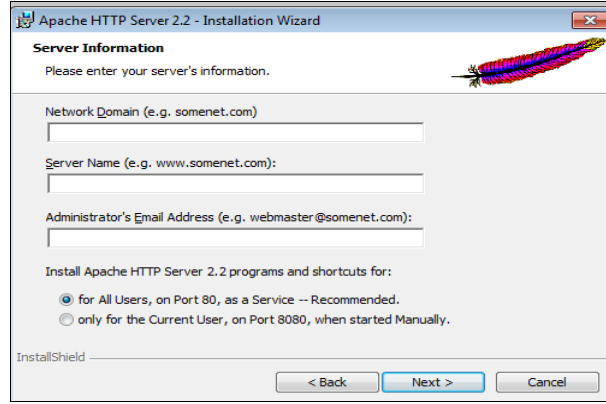
Bir sonraki pencere lisans sözleşmesi hakkında bilgilerin verildiği penceredir. Bir- çok program kurulumundan da aşına olduğunuz bu pencerede “Sözleşmeyi kabul ediyorum. (I accept the terms in the license agreement.)” seçeneğini seçmeniz gerekmektedir. Aksi takdirde **Next** butonu aktif olmayacaktır.

Diğer penceremiz Apache nin kendini tanıttığı ve son versiyona ulaşım adresi bilgisinin verildiği penceredir. Bu pencere bilgilendirme amaçlı olduğundan okuyup “**Next**” butonuna tıklıyoruz.



**Resim 1.5: Apache bilgilendirme penceresi**

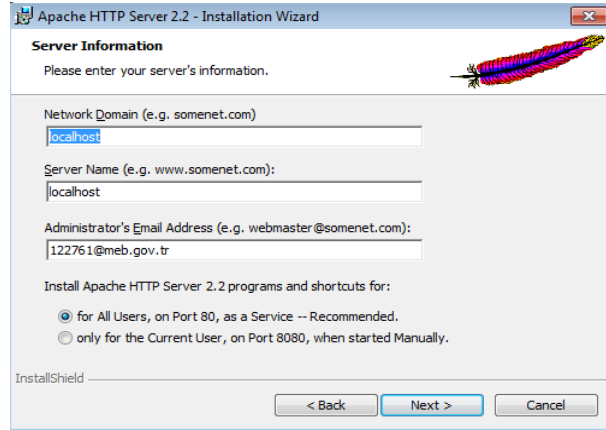
Daha sonraki kurulum sihirbazında ise sunucu bilgileri istenmektedir. Daha önceki sürümü kullanmış arkadaşlar bu penceredeki alanları boş bıraktıkları zaman kurulumu devam edebilmektedirler.



**Resim 1.6: Sunucu için gerekli bilgilerin girileceği pencere**

Önceki sürümlerde boş geçilen sunucu bilgileri default değeri olarak localhost kabul etmekteydi.

Bu sürümde bütün alanların doldurulması gerekmektedir. Aksi takdirde hata mesajı ile karşılaşılmaktadır.



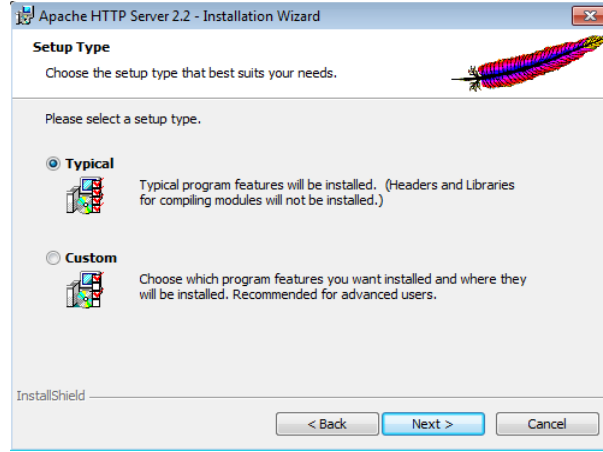
**Resim 1.7: Sunucu bilgilerinin girilmiş hâli**

Bu pencerede adı geçen alanlar şu manalara gelmektedir:

- **Network Domain: İnternet adresi**
- **Server Name: Sunucu ismi**
- **Administrator' Email Adress : Yönetici e-mail adresi**

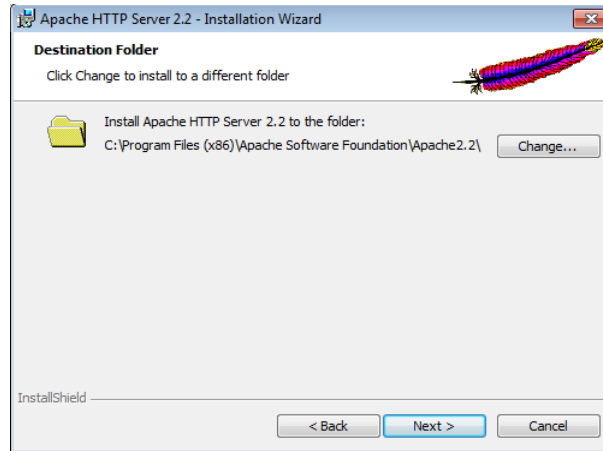
Radio butonlarından ise “for all users, on port 80 as a service Recommend” seçeneği seçilerek programın 80 port üzerinden çalışması istenir.

Bir sonraki aşamada, diğer program kurulumlarında olduğu gibi kurulum tipi ekranında seçim yapmamız istenir. Bilindiği gibi Typical(tipik), Custom(özel) kurulum manalarına gelir. Tipik kurulumda sunucumuz standart program özelliklerini içerecek şekilde yüklenir. Özel kurulumda ise sunucuyu kuran kişi istediği program özelliklerini seçerek yükleme yapabilir.



**Resim 1.8: Kurulum tipinin seçilmesi**

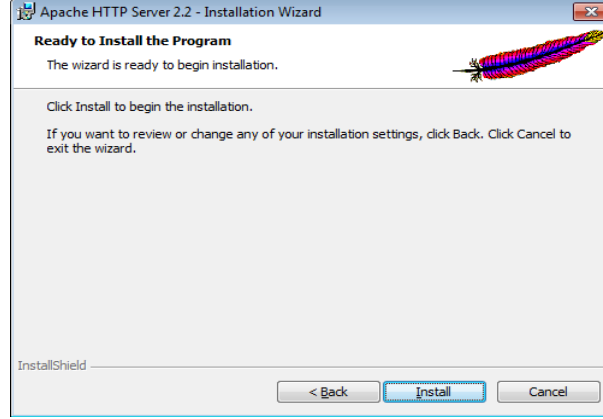
Bir sonraki aşama programın kurulacağı yerin seçilmesi işlemidir.”**Change..**” butonu ile dizin değişikliği yapılabilir.



**Resim 1.9: Kurulum dizininin seçilmesi**

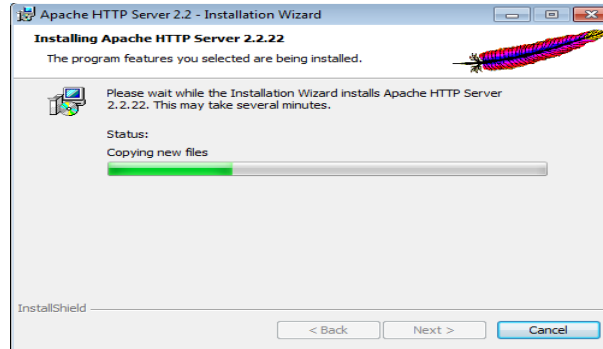
Dizin değişikliği işlemine gitmeden ” **Next**” butonuna tıklıyoruz.

Kurulacak olan dizin belirlendikten sonra **“Install”** butonu ile kurulum işlemini başlatıyoruz.



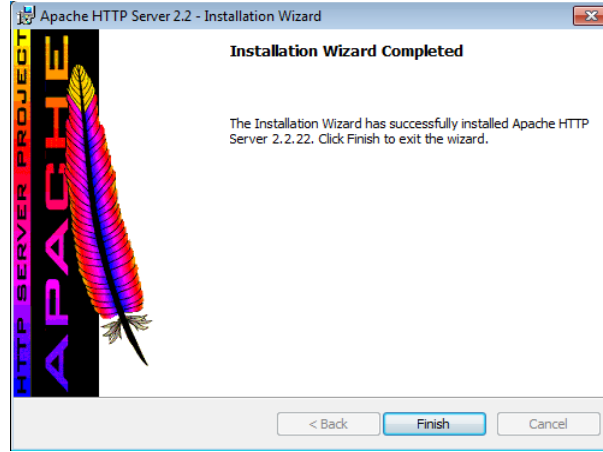
**Resim 1.10: Kurulumun başlatılması**

Kurulumun hangi aşamada olduğunu, hangi dosyaların yüklendiği gibi bilgileri kurulum ekranından takip edebiliriz.



**Resim 1.11: Kurulumun devam etmesi**

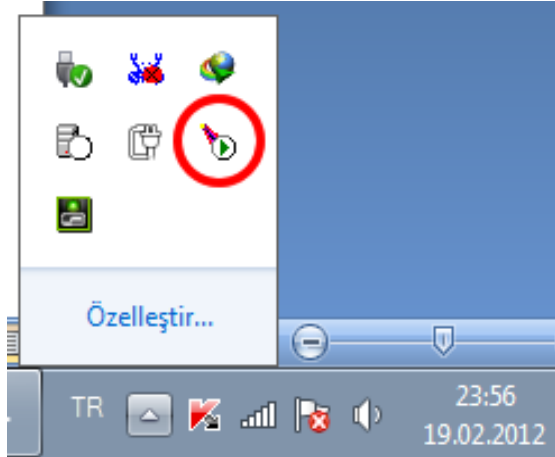
Kurulumun bitmesini bekliyoruz.



**Resim 1.12: Kurulum işleminin tamamlanması**

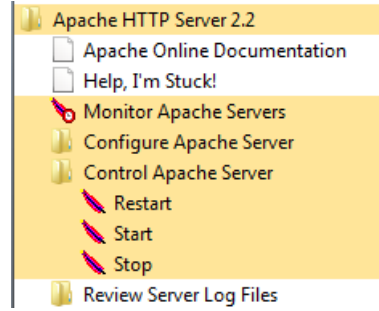
Son pencerede yükleme sihirbazı kurulumu tamamlamış olacaktır. “**Finish**” butonu ile işlemimizi bitiriyoruz.

Kurulum tamamlandıktan sonra System Tray bölümüne Apache web sunucusunu temsil eden bir ikon yerleşecektir.



**Resim 1.13: System tray apace ikonu**

Ayrıca başlat menüsünden programlar bölümünde **Apache-Http Server** kurulum dizinini seçerek de sunucuyu çalıştırabilirsiniz.



**Resim 1.14: Program files ile apache çalıştırma**

Sunucu kontrolünde aşağıdaki hususların bilinmesinde fayda vardır.

- **Monitor Apache Servers** :Apache web sunucusu ile ilgili pencereyi açar.
- **Restart**: Çalışmakta olan web sunucusunu durdurur ve yeniden başlatır.
- **Start**: Web sunucusunu başlatır.
- **Stop**: Çalışmakta olan web sunucusunu durdurur.

Apache sunucunun çalışmasının kontrol etmek için internet tarayıcımızın adres çubuğuna **http://localhost** veya **http://127.0.0.1** yazıldığında aşağıdaki gibi bir ekran görüntüsü aldığımız sunucu çalışıyor demektir. Burada yeri gelmişken şunu da belirtmekte fayda var. Aşağıdaki ekran görüntüsü Apache Http Server 2.2.22 sunucusu için geçerlidir. Diğer sunucularda farklı görüntüler alınabilir. Önemli olan "Sayfa görüntülenemedi." mesajı almamaktır.



**Resim 1.15:Apache kontrol edilmesi**

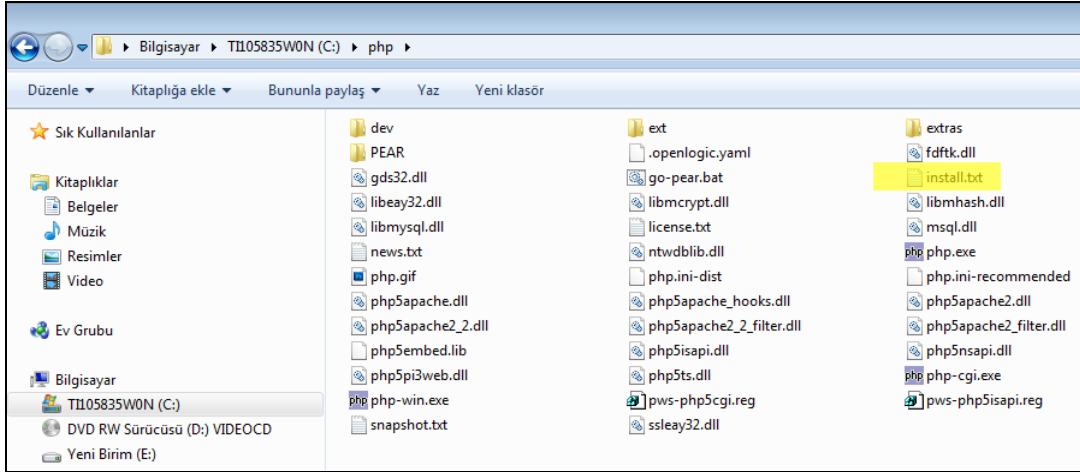
## 1.2.2. Php Kurulumu

Php derleyicimiz olup apache ile uyumlu çalışması gerekmektedir. Apache'nin son çıkan versiyonları ile bazen senkronize bir çalışma gerçekleştirilmeyebilir. Bir önceki konuda bahsettiğimiz gibi bazen son versiyonlar arasında uyum problemi ortaya çıkmaktadır.

Bu nedenle bir alt sürümler tercih edilebilir. Kurulumu iki yolla yapabiliriz: Paket programı kurarak veya sıkıştırılmış dosya halinde indirdiğimiz dosyaları PHP adında bir dizin içine açarak.

Biz, burada, farklılık olması açısından ikinci yolu tercih edeceğiz. Dosyayı edinmek için **www.php.net** sitesini ziyaret edelim. Buradan indirdiğimiz PHP derleyici dosyalarını C:/>PHP dizininin içine açalım.





Resim 1.16: Php dizin içeriği

Resimde de işaretli bir şekilde gözüken `install.txt` dosyasını not defteri yardımıyla açıyoruz. Bu dosya ayar ve kurulum bilgilerini içermektedir.

Bu dosyada ilk önce Apache'nin hangi sürümünün kurulumunu yapmışsak onun **Windows** için olan ayar bölümünü buluyoruz. Örneğin Apache 2.0. kurulmuşsa;

```
highlight_file('some_php_script.php'); ?>.
```

Apache 2.0.x on Microsoft windows

This section contains notes and hints speci  
of PHP on Microsoft windows systems. We als

Daha sonra bu satırın biraz altında işaretlenmiş üç satırı seçip kurduğumuz Apache programının kurulu olduğu dizinin altında bulunan **conf** klasörünün altındaki **httpd.conf** dosyasını not defteri ile açıp en altına kaydediyoruz.



Httpd.conf dosyası kaydetme esnasında Windows 7 'de hata mesajı verirse yönetici olarak çalıştırabilir veya masaüstüne aynı isimde farklı kaydedip sonra conf klasörünün içindeki dosyanın üzerine yazdırabilirsiniz.

#### Installing as a CGI binary

You need to insert these three lines to your Apache configuration file to set up the CGI binary:

Example 2-5. PHP and Apache 2.0 as CGI

```
ScriptAlias /php/ "c:/php/"
AddType application/x-httpd-php .php

# For PHP 4
Action application/x-httpd-php "/php/php.exe"

# For PHP 5
Action application/x-httpd-php "/php/php-cgi.exe"
```

**For PHP 5**'in altındaki satırı seçmemizin sebebi sıkıştırılmış olarak indirdiğimiz kurulum dosyalarının PHP5 versiyonu olması.

Yukarıda üstü ilk çizilen satırda bulunan **"C:/php/"** kısmını PHP klasörlerini açtığınız dizine göre değiştirmemiz gerekmektedir.

Derleyici kurulumu işlemimiz bitmiştir.

**Apache>conf> httpd.conf** dosyasında bilinmesi gereken bazı özelliklerin olduğunu söylemeden geçemeyeceğiz.

- **DocumentRoot:** Dosyaların sunulacağı dizini belirtir.
- **ServerRoot:** Apache'nin ana dizinini belirtir.
- **Port:** Tek başına (standalone) sunucunun dinlediği 'port'. Servis verilecek port/portlar belirtir.

Ayrıca Apache'nin bazı özelliklerini aktif hale getirmek için satırın başında bulunan noktalı virgül(;) kaldırılmalıdır.

Özellik Pasif	Özellik Aktif
;extension=php_imap.dll	extension=php_imap.dll
;extension=php_sockets.dll	extension=php_sockets.dll
;extension=php_zip.dll	extension=php_zip.dll

### 1.3. Veri Tabanı Sunucusunun Kurulması

Veri tabanı, içinde bilgiler saklayan, birbiriyle ilişkili tablolar topluluğudur. Bilgiler belli gruplara göre saklanır ve istenildiğinde belli kriterlere göre çağrılır.



**Resim 1.17:Veri tabanı kavramı**

Günümüzde kullanılan etkileşimli web sayfalarının çoğunda veri tabanları kullanılmıştır.

Üyelerine satış yapan bir alışveriş sitesinde üye adı, üye adresi, üye telefon numarası vb. bir tabloda, ürünlere ait bilgiler, fiyatı, rengi, boyutu vb. gibi özellikler başka bir tabloda tutulup hepsine birden veri tabanı denilmektedir.

Veri tabanı kullanmadan etkileşimli sayfalar olmazdı.Forum sayfaları, sosyal paylaşım siteleri, alışveriş vb.. gibi sitelerin alt yapısı veri tabanlarında saklıdır.



**Resim 1.18:Veri tabanı sunucuları**

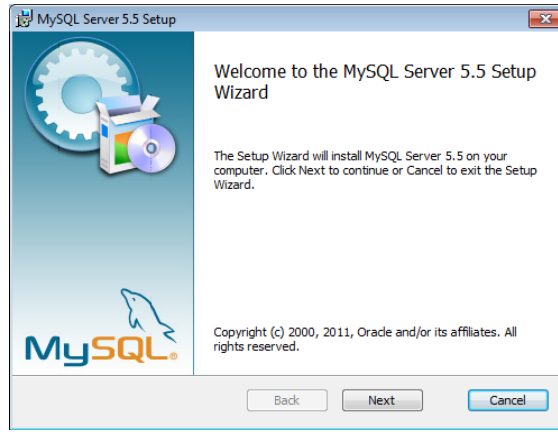
Günümüzde kullanılan birçok veri tabanı sunucu yazılımları vardır. Biz bunlardan PHP ile uyumlu çalışan MySQL sunucusunun kurulumundan bahsedeceğiz.

### 1.3.1 MySQL Kurulumu

Mysql sunucusu kurulumu için bilgisayarımıza [www.mysql.com](http://www.mysql.com) adresinden kurulum dosyasını indiriyoruz.



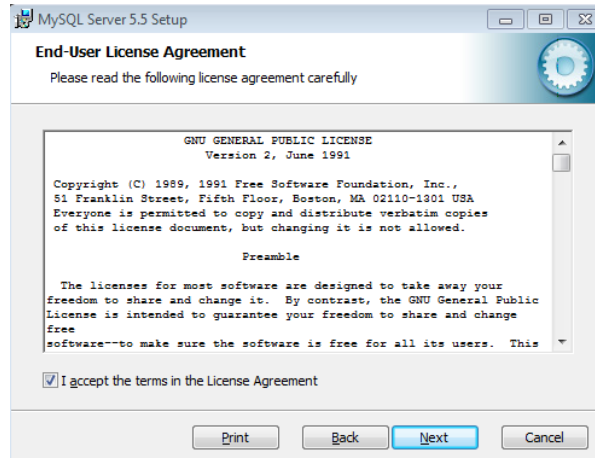
**Resim 1.19: Mysql kurulum programı**



**Resim1.20. Kurulumu başlama**

Siz güncel sürümleri takip edebilirsiniz. Kurulum programı çalıştırdığımızda karşımıza gelen ilk ekran, kurulum ekranıdır. "Next" butonu ile kurulumu devam ediyoruz.

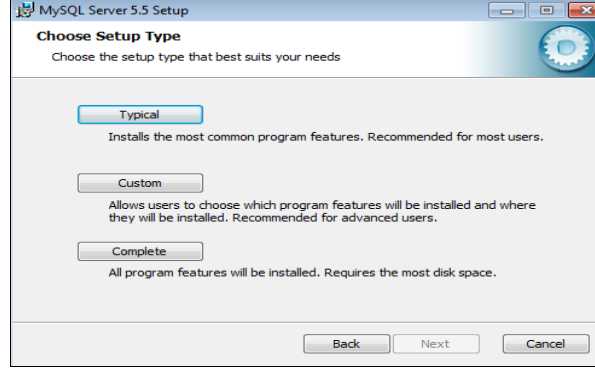
İkinci kurulum sihirbazı penceremiz, sözleşmeyi kabul ve onay penceresidir.



**Resim1.21: Lisans sözleşmesi onay penceresi**

Bildiğiniz gibi sözleşmeyi onaylamadan devam edemezsiniz. Biz de sözleşmeyi onaylayıp “Next” butonuna tıklıyoruz.

Sözleşme onayından sonra bizden kurulum tipini belirlememiz istenecektir.

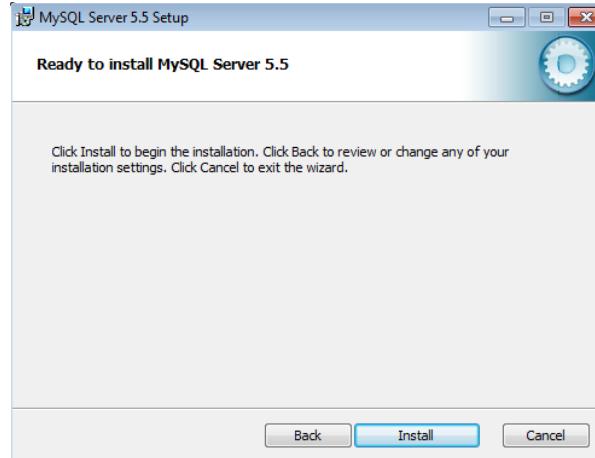


**Resim 1.22: Kurulum tipinin seçilmesi penceresi**

Kurulumu yapmak için 3 seçenek bulunmaktadır. Bunlar:

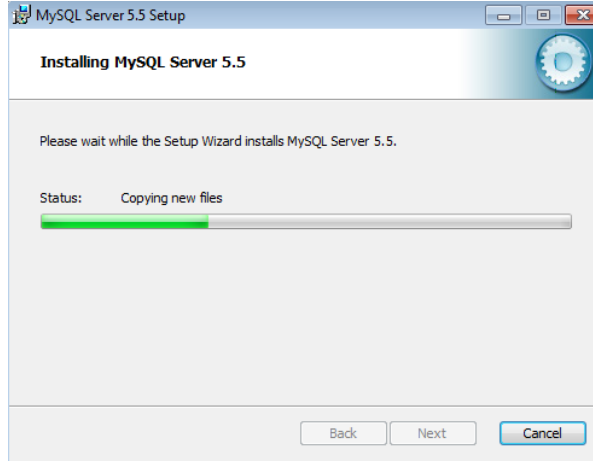
- **Typical(tipik):** Genel kullanım için önerilen yükleme türüdür. Bu seçenekle ortak program özellikleri yüklenir.
- **Custom(isteğe bağlı):** İstenilen belli özellikler kurulumda dahil edilip çıkarılabilir.
- **Complete(tamamen):** Bu seçenekte programın tüm özellikleri yüklenir. Bilgisayarımızda daha fazla disk alanı kaplar.

Biz bu kurulumda “Typical” olanı seçiyor ve devam ediyoruz.



**Resim1.23: Kurulum İşleminin Başlatılması**

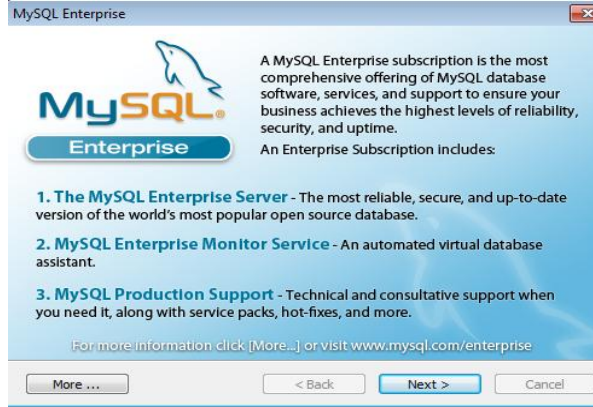
“Install “ butonuna tıklayıp kurulum işlemini başlatıyoruz.



**Resim1.24:Kurulumun devam etmesi**

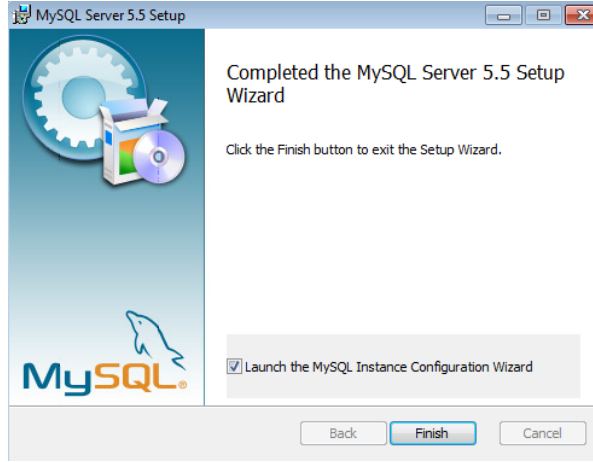
Kurulumun tamamlanmasını bekliyoruz.

Bundan sonra gelen pencere tamamen bilgilendirme penceresi olup MySQL hakkında bilgi vermektedir.



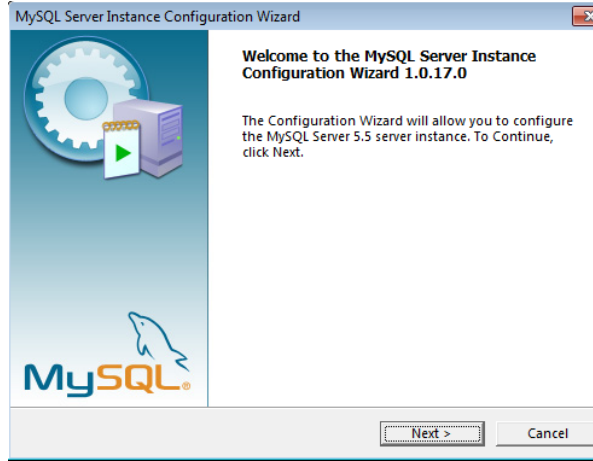
**Resim1.25: Bilgilendirme penceresi**

Özetle “MySQL Enterprise; Sunucu, Monitör ve 7x24 Ürün Desteğinden oluşur.” vurgusu yapılmaktadır. Bu aşamadan sonrada “Next” butonuyla bir sonraki pencereye geçiyoruz.



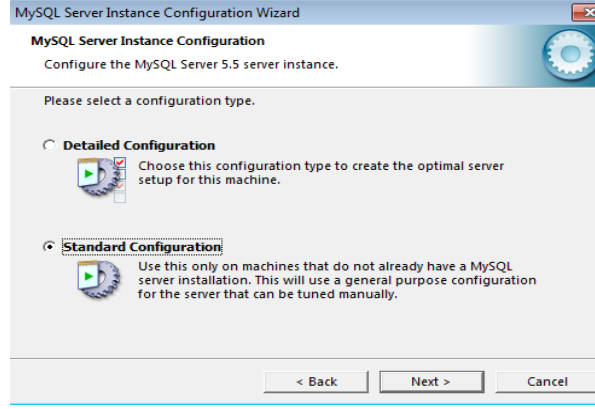
**Resim1.26:Kurulum işleminin tamamlanması**

“**Finish**” butonu ile kurulumu başarılı bir şekilde tamamlıyoruz. İşlemimiz henüz bitmiş değil. Konfigürasyon (yapılandırma) ayarları bölümüne sihirbazımız otomatik geçiş yapacaktır.



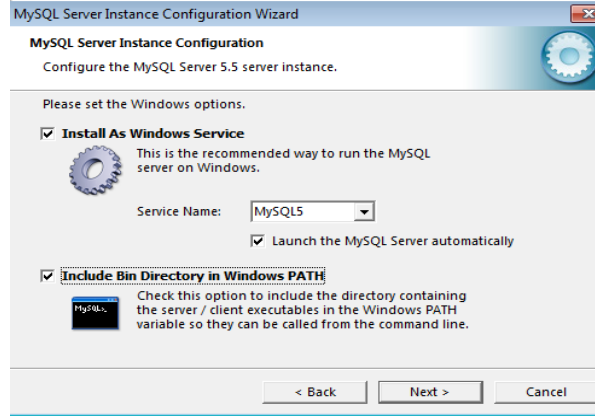
**Resim1.27:Yapılandırma penceresine geçiş**

Yapılacak olan yapılandırma işleminde iki seçeneğimiz bulunmaktadır. Bunlardan ilki detaylı yapılandırma, diğeri standart yapılandırmadır.



Resim1.28:Yapılandırma tercihleri seçim penceresi

Detaylı yapılandırma isteğe bağlı özelliklerin eklenmesi veya çıkarılması ve bazı ayarlamaların yapıldığı bölümdür. Biz burada **Standart Configuration**(Yapılandırma)'u tercih edip devam ediyoruz.



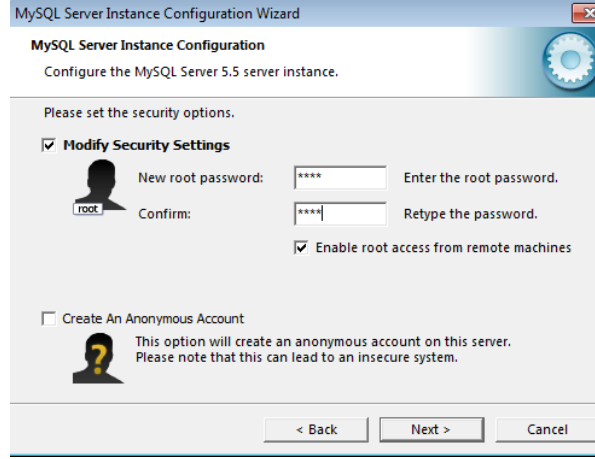
Resim1.29:Servis ayarları yapılması

Burada bulunan üç tane check box kutusundan:

- **(Install As Windows Service)**, Windows'un altında çalışacak service ismini seçmek için
- **(Launch the Mysql Server automatically)**, sunucunun Windows açılır açılmaz başlamasını onaylamak için kullanılır.
- **(Include Bin Directory in Windows PATH)** ise komut penceresini kurulumla dahil etme işlemi için kullanılır.

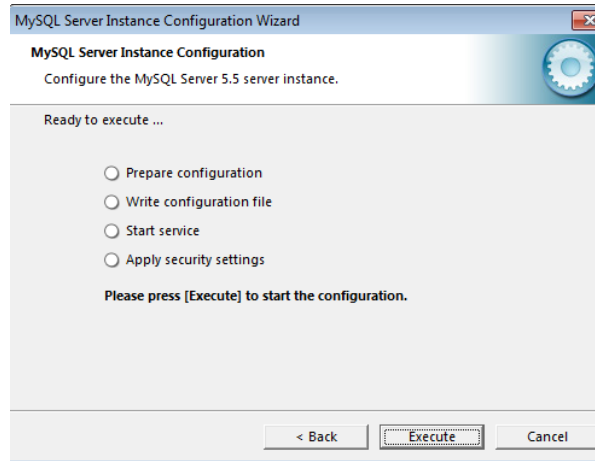


Bütün checkboxları onayladıktan sonra servis ismini(**service name**) MySQL5 olarak belirleyip “Next” butonu ile kurulumu devam ediyoruz.



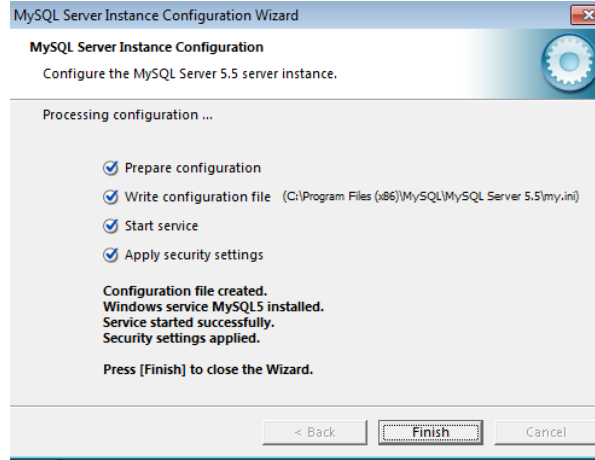
**Resim1.30: Kullanıcı giriş şifre ayarları**

Daha önceki sürümlerde default olarak “**root**” şifre olarak tanımlanmaktaydı. Biz burada klavyeden şifremizi kendimiz belirleyip “**Next**” butonuna tıklıyoruz.



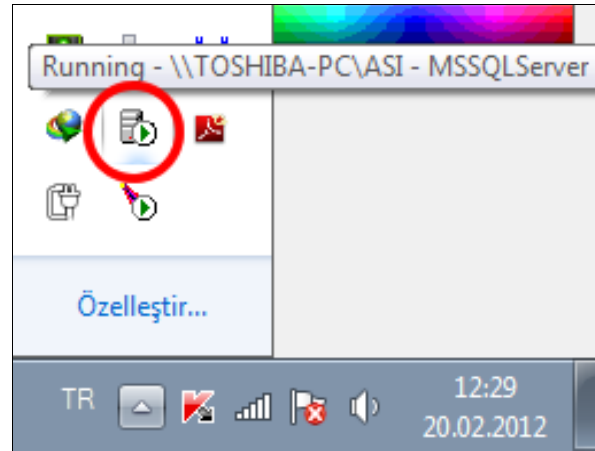
**Resim1.31:Kurulumun çalıştırılması**

Gerekli yapılandırmayı gerçekleştirmek için Execute (çalıştır) butonuna tıklıyoruz. Bir sonraki sihirbaz gerekli yapılandırmaların yapılacağı pencere olacak.



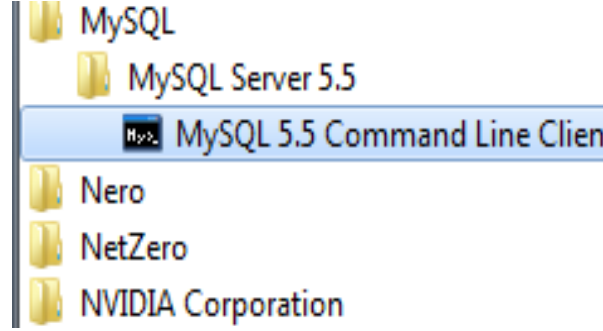
**Resim1.32: Kurulumun devam etmesi**

Yapılandırma hazırlandı, yapılandırma dosyası oluşturuldu ve MySQL Windows servisi başlatıldı. Finish butonuna bastıktan sonra **System Tray** bölümünde MsSQL Server'in çalışmasını görürüz.



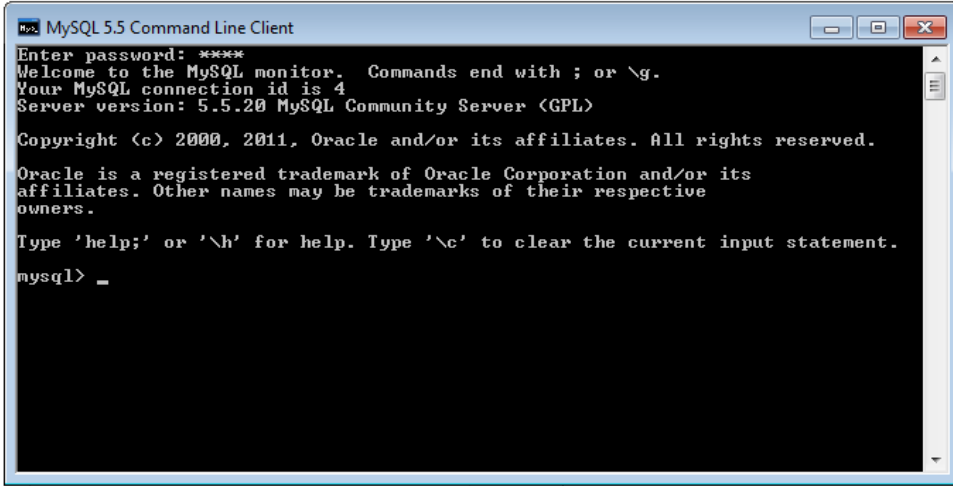
**Resim1.33: System Tray sunucu çalışması**

MySQL server kullanım ekranına geçmek için programı kurduğumuz dizini program files altından bulup çalıştırıyoruz.



**Resim1.34: Program files ile sunucuyu çalıştırma**

Veri tabanındaki verilere erişmek ve veri tabanını yönetmek için SQL komutlarını bilmek gerekir. Aradığımız veriyi listelemek, silmek, tekrar veri oluşturmak hep komutlar yardımıyla gerçekleşir. Bunun için DOS ekranına benzer MySQL'ın bir kullanım ekranı mevcuttur.



**Resim1.35: Mysql sunucunun çalıştırılması**

Bu pencerenin açılabilmesi için yukarıda “Enter password:” yazan alana kurulumda kullandığımız şifreyi yazmamız gerekmektedir.

MySQL kullanımı Veri Tabanı Yönetimi dersinin konusu olup biz bu dersin ilerleyen modüllerinde MySQL sunucuyu dinamik site yapımı için kullanacağız. Örneğin üyelik sistemi, ziyaretçi defteri vb...

MySQL kullanmak ve bu sunucunun ara yüzünü daha anlaşılır bir şekilde kullanmak için yardımcı programlardan da faydalanılabilir. Bunlardan en bilineni ve kullanılanı PhpMyAdmin programıdır.

PhpMyAdmin kurulumu için herhangi bir kurulum (setup) dosyasına ihtiyaç yoktur.

Sıkıştırılmış klasör içinde bulunan dosyaları açarak web sunucusu klasörüne (apache web sunucusu için htdocs klasörü) kaydetmek yeterlidir.



Sunucu localhost dosya klasörü, bazı sunucular için “htdocs”, bazıları içinse “www” dizini olarak değişmektedir.

## 1.4. Hazır Paketler

Sevgili öğrenciler belki de buraya kadar anlatılanlar size biraz karışık gelmiş olabilir. **“Bu işi daha kolay yapmanın bir yolu yok mu?”** sorusunu duyar gibiyim. Aslına bakarsanız yukarıda anlattıklarımızın hepsini tek bir program yardımıyla yapabilir, bütün programları tek bir paket programa yaptırabiliriz.

Daha önceki konularda da gördüğümüz gibi **php yorumlayıcı, server ve veri tabanı kurmak zahmetli bir iş**. Bu zahmeti ortadan kaldırmayı düşünen ve **tek bir pakette sunucu, derleyici ve veri tabanını içinde barındıran paketler piyasaya sunulmuştur**.

Hazır paketleri kullanmak istiyorsanız elinizde birkaç seçenek var. Bunlardan en çok bilinenleri XAMPP, WAMP ve EasyPHP. Hepsi aynı işi görüyor olmasına rağmen paketin içeriklerindeki ekstralar değişebiliyor.

Örnek olarak aşağıdaki tabloda bu programların isimleri ve içeriklerinden bir kesit verilmiştir.

EasyPHP	WAMP	XAMPP
<a href="http://www.easyphp.org">www.easyphp.org</a>	<a href="http://www.wampserver.com">www.wampserver.com</a>	<a href="http://www.apachefriends.org/en/xampp.html">www.apachefriends.org/en/xampp.html</a>
PHP Apache VC MySQL PhpMyAdmin Xdebug	Apache Php Mysql XDebug XDC PhpMyAdmin SQLBuddy webGrind	Apache MySQL PHP phpMyAdmin FileZilla FTP Server Tomcat

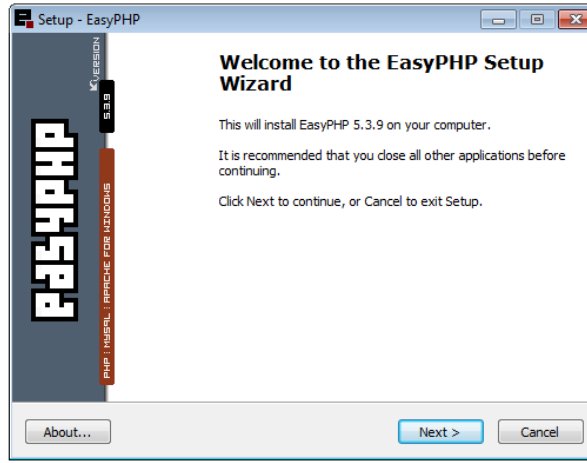
Bu paket yazılımlardan en sık kullanılan EasyPHP paket yazılımının kurulumu anlatılacaktır.

İşe EasyPHP paketinin [www.easyphp.org](http://www.easyphp.org) sitesinden indirilmesi ile başlayalım. Burada indirilen sürüm EasyPHP-5.3.9 ‘dur.



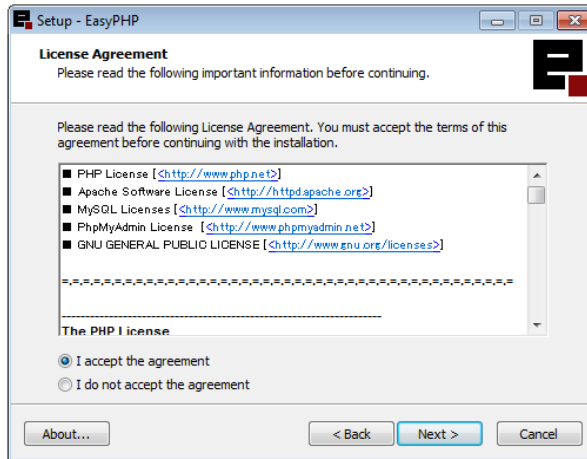
**Resim1.36: Kurulum paketi**

Kurulum paket programını çalıştırdığımızda karşımıza kurulum ekranı gelecektir. Next butonu ile bir sonraki adıma geçiyoruz.



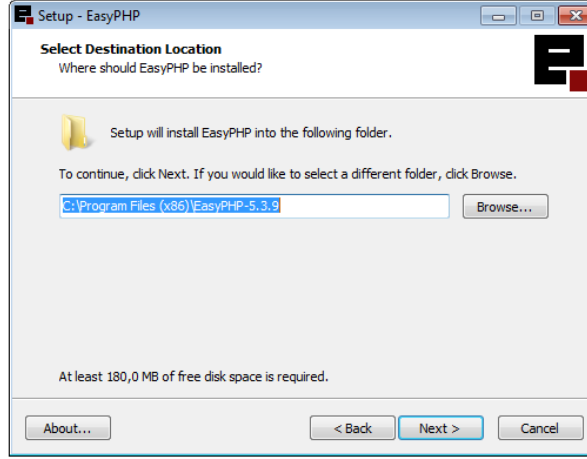
**Resim1.37: Kurulum başlama**

Bir sonraki pencere lisans onay ekranı olup kabulden sonra Next butonu ile devam ediyoruz.



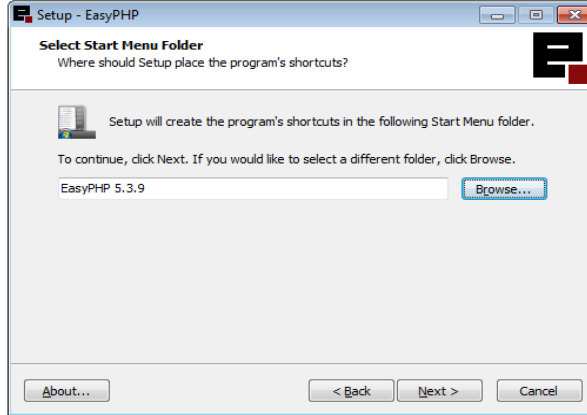
**Resim1.38: Kullanım sözleşmesinin onaylanması**

Bir sonraki pencere easyphp hakkında kısa bir bilgi verir.Bu pencereyi Next butonu ile geçiyoruz. Karşımıza kurulumun nereye yapılacağını belirten diğer pencere gelecektir.



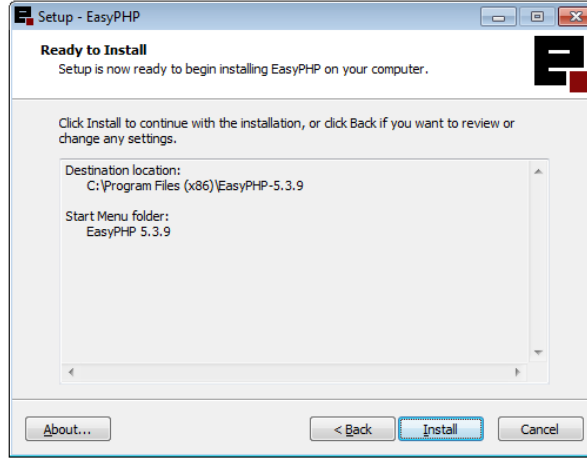
**Resim1.39: Kurulum dizini seçme**

Browse butonu ile istenirse kurulum dizinimiz değiştirilebilir. Fakat biz burada değiştirmeden Next butonu ile bir sonraki adıma geçiyoruz.



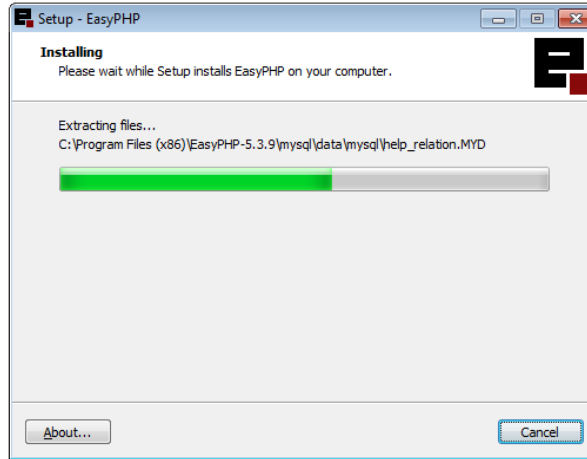
**Resim1.40: Başlat menü ismini verme**

Bu pencerede de programlar altındaki isim istenirse değiştirilebilir. Fakat biz burada değişiklik yapmadan “Next” butonu ile bir sonraki adıma geçiyoruz.



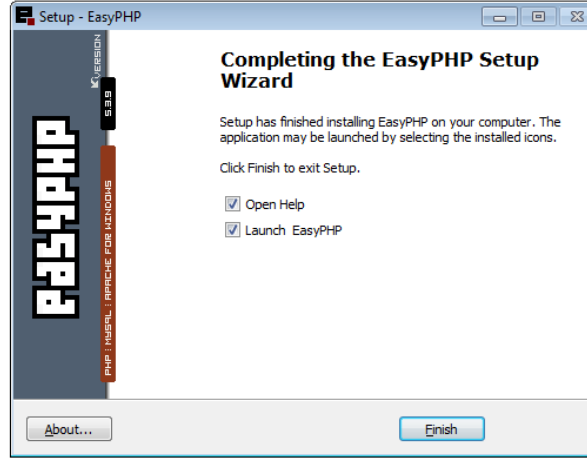
**Resim1.41: Kurulum işlemini başlatma**

Kurulum yerini ve ismini onaylıyorsak “Install” butonu ile kurulum işlemini ilgili klasöre gerçekleştiriyoruz. Aksi takdirde “Back” butonu ile kurulum yerini ve ismini değiştirebiliriz.



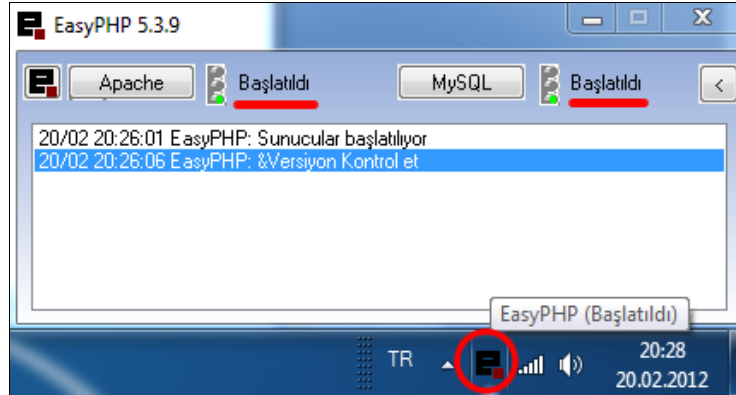
**Resim1.42:Kurulumun devam etmesi**

Gerekli dosyalar yüklenerek program kurulumu devam etmektedir. Dosya yükleme işlemi bittiğinde ise kurulum tamamlanmış olacaktır.



**Resim1.43:Kurulumun tamamlanması**

Kurulumu “Finish” butonu ile başarılı bir şekilde tamamlamış oluyoruz.



**Resim1.44: System Tray’da EasyPHP simgesi**

EasyPHP programımızı kurduğumuz dizinde bulup çalıştırdığımızda ekrandaki görüntüyü elde ederiz. EasyPHP’nin baş harfı olan “e” simgesi programımızın simgesidir. Üzerine çift tıkladığımızda ise paket içinde bulunan “Apache” ve “MySQL” programlarına erişebilir ve kontrol edebiliriz. Ayrıca dil değişikliği de yine buradan yapılabilir.



## UYGULAMA FAALİYETİ

Aşağıda verilen işlem basamaklarını takip ederek konuyu daha da pekiştirelim.

İşlem Basamakları	Öneriler
➤ Sunucular hakkında kısa bir araştırma yapınız.	➤ Host veren firmaların sunucu özelliklerini inceleyebilirsiniz.
➤ Apache web sunucunun kurulum dosyasını elde ediniz.	➤ <a href="http://www.apache.org">http://www.apache.org</a> internet adresinden bilgisayarınıza indirebilirsiniz. Güncel sürümü tercih ediniz.
➤ Kur (setup) dosyasını açarak kurulumu başlatınız.	➤ Kurulum yaparken dikkatinizi toplayınız.
➤ Gerekli sunucu bilgilerini giriniz.	➤ Alanları doğru olarak doldurunuz.
➤ Dosyaları yükleyerek kurulumu tamamlayınız.	➤ Dosyaların yüklendiğini kurulum sihirbazında kolayca görebilirsiniz. Finish butonunu görüyorsanız kurulum tamamlanmıştır.
➤ Web sunucunuzu test ediniz.	➤ İnternet tarayıcınızın adres çubuğuna <a href="http://localhost">http://localhost</a> yazmanız yeterli. Test sayfası görüntüleniyorsa sunucunuz hatasız çalışıyor demektir.
➤ Web sunucunuzun gerekli kontrollerini yapınız (Başlat, Durdur, Yeniden Başlat).	➤ Görev çubuğunun sağındaki apache simgesinden veya Başlat / programlar/ apache yolunu kullanabilirsiniz.
➤ PHP yorumlayıcısının kurulum dosyasını elde ediniz.	➤ <a href="http://www.php.net">http://www.php.net</a> İnternet adresinden bilgisayarınıza indirebilirsiniz.
➤ Yapılandırmak için web sunucunuzu seçiniz.	➤ Kullandığınız sunucunun versiyonuna dikkat ediniz.
➤ Yüklenmesini istediğiniz bileşenleri seçiniz.	➤ Daha sonra ihtiyacınız olabilecek bileşenleri de seçmeye özen gösteriniz.
➤ Veri tabanı kavramını sınıfınızda tartışınız.	➤ ÖSYM-Nüfus İdaresi-e-okul sistemlerini örneklerinize ekleyebilirsiniz.
➤ MySQL veri tabanı sunucunu elde ediniz.	➤ Bunun için <a href="http://www.mysql.com">www.mysql.com</a> sitesinden faydalanabilirsiniz.
➤ Gerekli yapılandırmaları yaparak programı kurunuz.	➤ Kurulum bölümünü dikkatlice inceleyiniz.
➤ Kurulum tamamlandıktan sonra çalışıp çalışmadığını kontrol ediniz.	➤ Durum çubuğunda ilgili simgeye bakınız. ➤ Başlat>Programlar>Mysql yolu
➤ Hazır paketlerin avantajlarını ve dezavantajlarını sınıfınızda tartışınız.	➤ İlgili hazır paket sitelerini ziyaret edebilirsiniz. ➤ <a href="http://www.easyphp.org">www.easyphp.org</a> - <a href="http://www.wampserver.com">www.wampserver.com</a> - <a href="http://www.apachefriends.org">www.apachefriends.org</a>

➤ Easyphp hazır paketini indiriniz.	➤ <a href="http://www.easyphp.org">www.easyphp.org</a> adresinden faydalanabilirsiniz.
➤ Gerekli yapılandırmaları yaparak programı kurunuz..	➤ Modüldeki kurulum konusunu dikkatlice inceleyiniz.
➤ Kurulum tamamlandıktan sonra çalışıp çalışmadığını kontrol ediniz.	➤ Durum çubuğunda ilgili simgeye bakınız. ➤ Başat>Programlar>Easyphp yolunu kullanınız.

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.


Değerlendirme Ölçütleri	Evet	Hayır
1. Sunucu ve yorumlayıcı kavramı hakkında bilgi sahibi oldunuz mu?		
2. Apache ve PHP paketlerini bilgisayarınıza kurup çalıştırdınız mı?		
3. Apache ve PHP paketlerini bilgisayarınızda çalıştırdınız mı?		
4. Veri tabanını hakkında bilgi sahibi oldunuz mu?		
5. Mysql veri tabanı sunucusunu bilgisayarınıza kurdunuz mu?		
6. Veri tabanı sunucu programını çalıştırabildiniz mi?		
7. Hazır paketlerin neler olduğu hakkında bilgi sahibi oldunuz mu?		
8. Easy PHP programını bilgisayarınıza kurabilidiniz mi?		

## DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise “Ölçme ve Değerlendirme”ye geçiniz.

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Aşağıdakilerden hangisi bir web sunucu programıdır?  
A) Apache  
B) Nero  
C) Mysql  
D) Excel
2. Hangisi bir web veritabanı sunucusudur?  
A) PHP  
B) Mysql  
C) Apache  
D) Phpmyadmin
3. PHP ile hazırlanmış web programlarının veya web sayfalarının çalışabilmesi için en az hangi yazılımların bulunması gerekmektedir?  
A) Web sunucu ve PHP yorumlayıcı  
B) Sadece PHP yorumlayıcı  
C) Sadece web sunucu  
D) Veri tabanı ve web sunucu
4. Hazır kurulum paketlerinden olan EasyPHP'nin içinde aşağıdakilerden hangisi yoktur  
A) Mysql  
B) PHP  
C) Dreamweaver  
D) Phpmyadmin
5.  simgesi hangi programın ikonudur?  
A) Phpmyadmin  
B) WebGrind  
C) PHP  
D) Apache

## 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

Temel programlama işlemlerini yapabileceksiniz.

## ARAŞTIRMA

- Web yazılım editörleri nelerdir? Araştırınız.
- Derleyicinin ne olduğunu araştırıp öğreniniz.

## 2. TEMEL İŞLEMLER

### 2.1. Programların Kaydedilmesi

**PHP'yi** kısaca, HTML kodları arasına gömülmüş server tabanlı script dili olarak tanımlayabiliriz. Bu tanımda dikkat etmemiz gereken üç önemli nokta vardır:

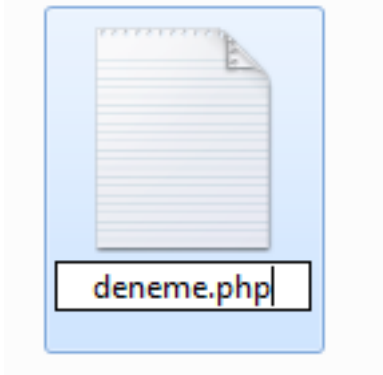
- **HTML kodları arasına gömülmüş olması.**
- **Server tabanlı olması**
- **Script dili olması.** Şimdi bu üç unsurun ne anlama geldiğine bakalım:

Daha önceden basic veya daha farklı bir dilde program yazmış arkadaşlar bilirler. Yazdığımız programın çalıştırılabilmesi için o program diline özgü derleyici tarafından derlenerek exe veya com uzantılı dosya oluşturulması gerekmektedir. Derleme sonucu oluşan exe veya com uzantılı bu dosyaları bir metin editörü ile açtığımızda birçok anlamsız karakterden oluştuğunu görürüz. Bunun sebebi yazdığımız kodların derlenerek makine diline çevrilmiş olmasıdır. Programcılar arasında bu tarz programlama dillerine gerçek program adı verilmektedir.

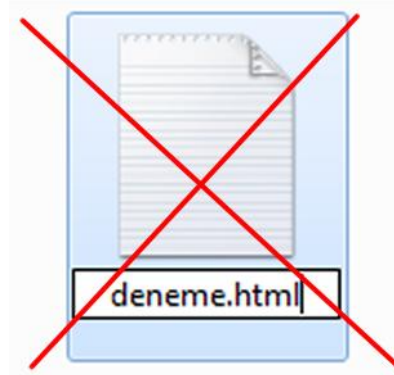
**Gerçek program dillerinde derleme işlemi bir kez yapılır ve bundan sonra derlenen program derleyiciden bağımsız olarak çalışır. Oysa script dili tarzında yazılmış bir program açık koda sahip olmakla birlikte(encode edilmiş php dosyaları hariç, metin editörü ile baktığımızda kodu her zaman görebiliriz ) derleme işlemi programın her çalıştırılmasında yeniden yapılır.**

**Bu nedenle script dili ile yazılmış bir program derleyicisi olmadan hiçbir işe yaramayacaktır.**

**Bu da bize şunu gösteriyor ki uygulamalarımıza başlamadan önce PHP derleyicisine sahip olmalıyız.**



Resim 2.1: Dosya doğru isimlendirme



Resim 2.2: Dosya yanlış isimlendirme



PHP derleyici ve sunucunun bulunmadığı ortamlarda php uzantılı dosyalar çalışmaz. Php kodları kullanılarak yazılan dosyalara php uzantısı verilmelidir.

## 2.2. Kodların Yazılışı

PHP' nin bu kadar kolay kullanılabilir bir dil olmasındaki en önemli unsurlardan biri de HTML kodları arasına gömülebilir olmasıdır. PHP kodları `<? PHP?>` ayraçları arasında yer alır. Frontpage veya başka bir editör kullanarak oluşturduğumuz web sayfamızın HTML kodları arasına PHP kodlarını kolaylıkla ekleyebiliriz. Bir browser php uzantılı dosyamızı görüntülemek istediğinde server bunun PHP derleyicisi tarafından derlenmesi gerektiğini anlayarak bu dosyayı PHP derleyicisine gönderir.

```
1 <html>
2 <head>
3 <title></title>
4 </head>
5 <body>
6 <?php
7 echo "PHP ile ekrana yazı yaz";
8 ?>
9 </body>
10 </html>
```

```
1 <html>
2 <head>
3 <title></title>
4 </head>
5 <body>
6
7 HTML ile ekrana yazı yaz;
8
9 </body>
10 </html>
```

Yukarıdaki örnekte, soldaki sunucu ve derleyici olmadan çalışmaz. Sağdaki program ise direk internet tarayıcı ile çağrıldığı zaman çalışır.

Php uzantılı dosya yorumlayıcı tarafından okunarak. `<? PHP?>` ayraçları arasındaki kodlar değerlendirilir ve bir sonuca bağlanır. Ayraçlar dışında kalan kısımlar hiçbir şekilde işlem görmeden browser'a yollanır.

1	<html>	1	<html>
2	<head>	2	<head>
3	<title></title>	3	<title></title>
4	</head>	4	</head>
5	<body>	5	<body>
6	<?PHP	6	
7	for (\$i=1;\$i<=15;\$i++)	7	123456789101112131415
8	printf(\$i);	8	
9	?>	9	
10	</body>	10	</body>
11	</html>	11	</html>

Yukarıdaki örnekte, sağdaki program yorumlayıcı tarafından okunarak sağdaki gibi değerlendirilir.

PHP kodu, HTML kodlarının içine 4 farklı şekilde yazılabilir.

<b>Kısa Stil</b>	<? Buraya PHP kodu yazılacak ?>
<b>XML Stil</b>	<?php Buraya PHP kodu yazılacak?>
<b>Script Stil</b>	<script language='php'> Buraya PHP kodu yazılacak </script>
<b>Asp Stil</b>	<%Buraya PHP kodu yazılacak%>

Biz çalışmalarımızda çoğunlukla XML stilini kullanacağız.

## 2.3. Yorum Satırı

Uzun programlar yazarken belli yerlerde ne yaptığımızı sonradan hatırlamak için kısa açıklamalar yazmak isteyebiliriz. Sunucu tarafından yorumlanmayan bu kodlara yorum satırı denir.PHP’de iki çeşit yorum satırı vardır. Bunlar:

- “//” : Tek satır kullanılarak yapılan açıklamalar için kullanılır.
- “/\* yorum alana \*/” : Birden çok satır kullanılarak yapılan açıklamalarda kullanılır.

Aşağıdaki programda php kullanımını yeni öğrenen birisi için kod sonlarına “//” işareti ile açıklamada bulunduk.

1	<?php
2	\$a=20; //a değişkenine 20 değerini atadık
3	\$b=10; //b değişkenine 20 değerini atadık
4	\$topla=\$a+\$b; //İki sayıyı topladık
5	echo \$topla; //Sonucu ekrana yazdırdık
6	?>

Aynı anda birden fazla yorum satırı bırakmak istiyorsak yorum satırını "/\*" ve "\*/" kodları arasına almamız yeterli olacaktır.

```
1 <?php
2 /*Aşağıdaki programda
3 iki değişken toplanıp
4 sonuç ekrana yazdırılmıştır
5 */
6 $a=20;
7 $b=10;
8 $topla=$a+$b;
9 echo $topla;
10 ?>
```

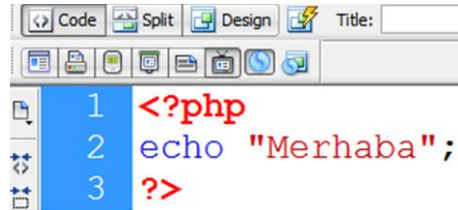
## 2.4. Editör Kullanarak Hızlı Kod Yazma ve Çalıştırma

Php programı yazmak için editör programlarından herhangi birini kullanabiliriz. Bu, not defteri, wordpad veya kelime işlemci yazılımı olabilir. Ama kullanım kolaylığı bakımından Web Tasarım Editörü kullanmak daha avantajlıdır.

Özel amaçlı hazırlanmış bu editör programları kodları, değişkenleri, fonksiyonları farklı renkte verip size kullanım kolaylığı sağlar. Ayrıca kod kütüphanesi sayesinde kodların ilk harfini yazar yazmaz geri kalanını size tamamlamanızda yardımcı olur.



Resim 2.3: Not defteri ile kod yazma



Resim 2.4: Yazılım editörü ile kod yazma

Yukarıdaki örnekte de görüldüğü gibi program yazılım editörlerinden faydalanmak yazım kolaylığı açısından daha avantajlıdır.

Buraya kadar öğrendiklerimizin ışığında biz de ilk programımızı yazalım. Program yazmaktaki amacın bir sorunu çözmek olduğunu da unutmayalım.

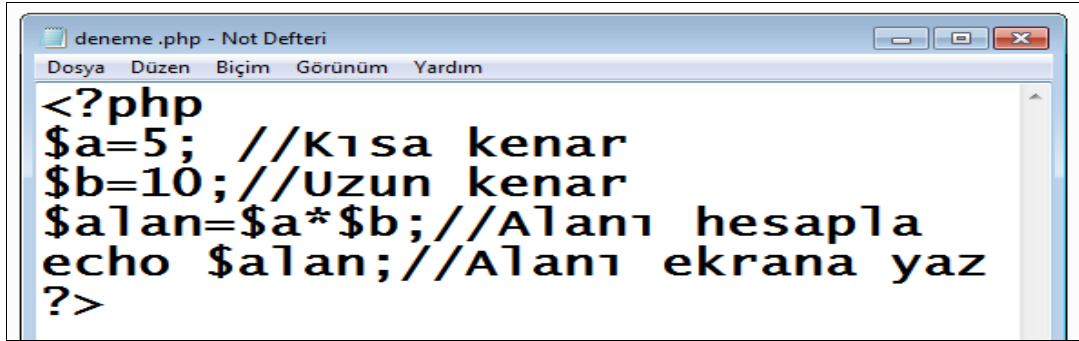


**Örnek:** Kısa ve uzun kenarı verilen dikdörtgenin alanını bulan bir program yazalım:

a=5  
b=10

Alan=a\*b  
Alan=5\*10  
Alan=50

Bu programı not defteri yardımıyla yazalım:



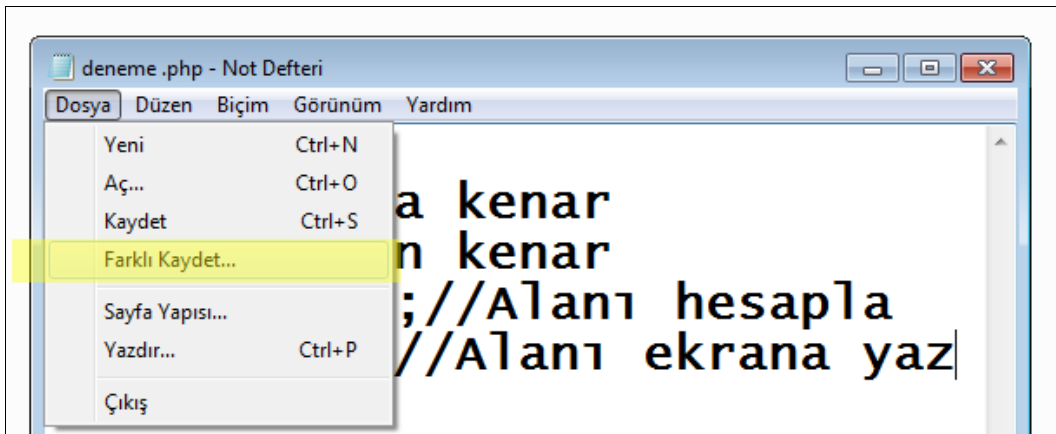
```
<?php
$a=5; //Kısa kenar
$b=10; //Uzun kenar
$alan=$a*$b; //Alanı hesapla
echo $alan; //Alanı ekrana yaz
?>
```

**Resim 2.5: Not defteri ile program çalışması**

- Yazdığımız programı deneme.php uzantısı ile kaydediyoruz.

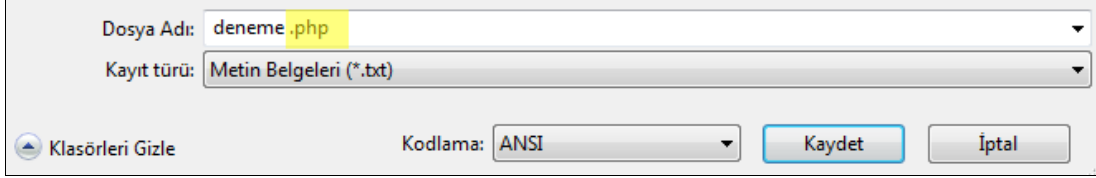


Program yazarken atama ve komutlardan sonraki satır sonlarına noktalı virgül (;) konulmaktadır. Ayrıca çarpma işlemi php dilinde yıldız(\*) işareti ile yapılmaktadır.



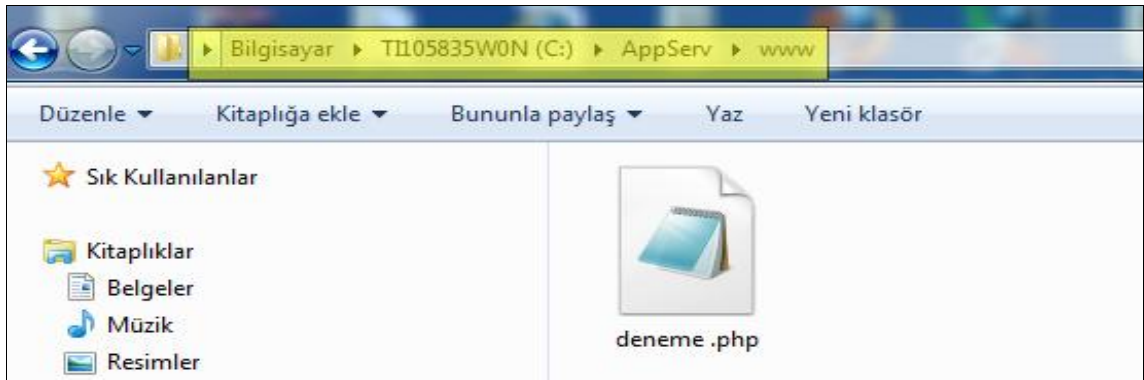
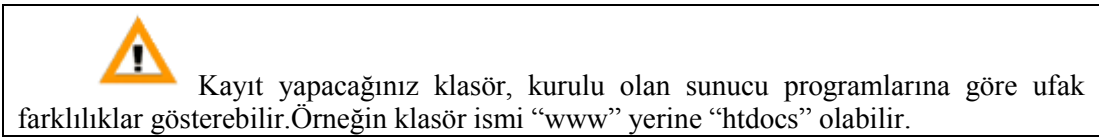
**Resim 2.6: Not defteri ile programın kaydedilmesi**

- Dosya uzantımızın **php** olmasına dikkat ediyoruz.



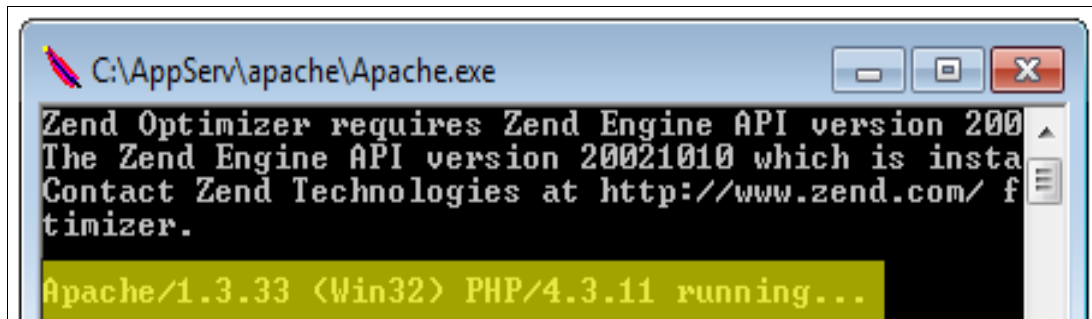
**Resim 2.7: Program isminin ve uzantısının yazılması**

Şu an bu bilgisayarda hazır paket olduğu için **C>Appserv>www** klasörü altına atılan **php** programları çalıştırılacaktır. Farklı yerlere kaydettiğiniz programlarımızı çalıştıramayacağız.



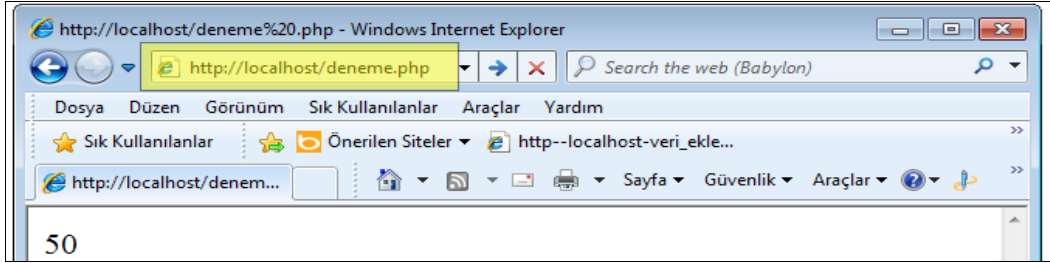
**Resim 2.8: Kayıt dizininin seçilmesi**

- Kaydettiğimiz programı çalıştırmadan önce Apache Server'i çalıştırmamız gerekmektedir.



**Resim 2.9: Apache sunucusunun programının çalıştırılması**

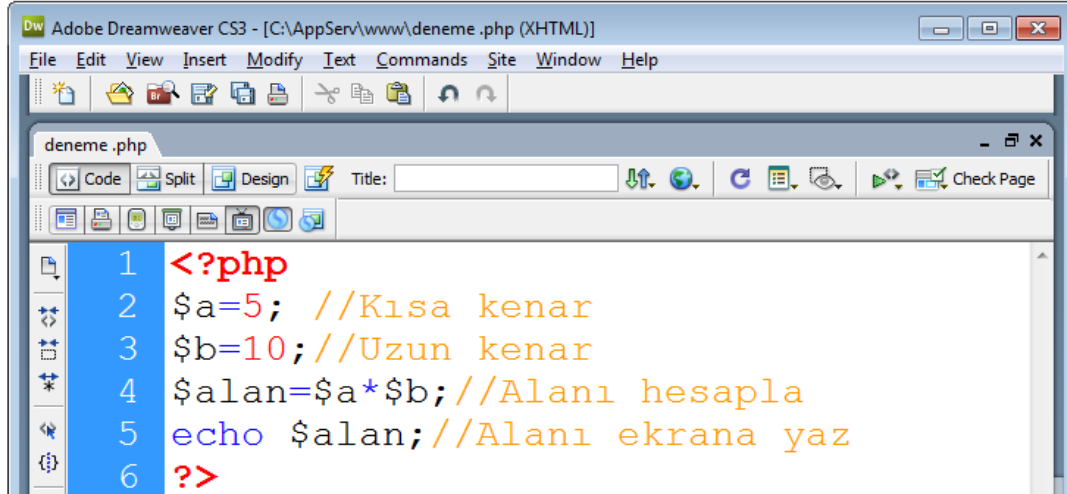
- Programınızı çalıştırmak için tarayıcınızın adres çubuğuna `http://localhost/dosyaadi.php` yazmanız gerekmektedir. Ya da local hostunuzu temsil eden ip numarasını (**127.0.0.1**) yazabilirsiniz.



**Resim 2.10: Tarayıcıdan programın adresini yazıp çalıştırılması**

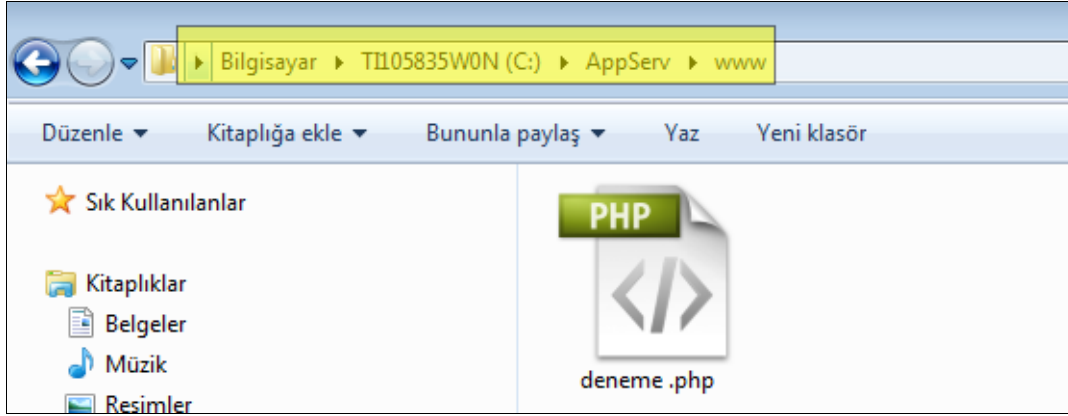
Bu işlemleri yaptığınız takdirde programınızın çalıştığını göreceksiniz. Apache server kapatılıp program çalıştırılmak istenirse bağlantı hatası verdiğini göreceksiniz. Dolayısıyla programınız çalışmayacaktır.

Aynı programı Web Tasarım Editörü kullanarak yazdığımızda aşağıdaki gibi renkli bir görüntü elde ederiz. Bu tür editörlerle çalışmak program yazımı ve takibi açısından kolaylık sağlar.



**Resim 2.11: Yazılım editörü ile program yazma**

- Yazdığımız programı **File>Save** menüsü yardımıyla localhost' un çalıştığı dizine kaydedebiliriz.



**Resim 2.12: Kayıt dizininin seçilmesi**

- Bu program da apache server açıkken tarayıcının adres çubuğuna <http://localhost/dosyaadi.php> adresine girerek çalıştırılabilir.

## UYGULAMA FAALİYETİ

Aşağıda verilen işlem basamaklarını takip ederek konuyu daha da pekiştirelim.

İşlem Basamakları	Öneriler
➤ Basit bir dikdörtgen alan bulma programı yazınız.	➤ Çarpma işlemi için yıldız (*) kullanınız.
➤ Not defteri ile yazdığınız programı php uzantılı olarak kaydediniz.	➤ Dosya>Farkı kaydet
➤ Program içerisine yorum satırı ekleyiniz.	➤ “//” ve “/* */” seçeneklerinden tercih yapınız.
➤ Aynı programı yazılım editörü kullanarak yazmayı deneyiniz	➤ Dreamweaver –phped-frontpage vb...
➤ Html etiketleri arasına programınızı yazmaya çalışınız.	➤ <? php Programınız ?>
➤ Yazdığınız programı kaydediniz.	➤ “www” veya “htdocs” dizini
➤ Apache programını çalıştırınız.	➤ Programlar>Apache veya System Tray kullanınız.
➤ Tarayıcınızı açarak yazdığınız programı çalıştırınız.	➤ http://localhost/dosyaadi.php
➤ Apache serverinizi durdurup programın çalışmasını tekrar kontrol ediniz.	➤ Apache Monitör kullanabilirsiniz.

## ÖLÇME VE DEĞERLENDİRME

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

1. ( ) Yazılan programımızın uzantısı php olmazsa derleyici dosyayı görmez.
2. ( ) Derleyici php dilinde “//” bulunduğu satırdan önceki kodları dikkate almaz.
3. ( ) Php programımız sunucunun kurulu olduğu dizinlerden htdocs içine kaydedilmezse çalışmaz.
4. ( ) Php kodları html etiketleri ile birlikte kullanılmaz.
5. ( ) PHP ile hazırlanmış web programlarının veya web sayfalarının çalışabilmesi için web sunucu ve PHP yorumlayıcıya ihtiyaç vardır.
6. ( ) Php kodları her yazı editörü ile yazılmaz.

### 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-3

## AMAÇ

- Temel basit komutlar ile basit program yazabileceksiniz.
- Yazılan temel programları uygun sunucu ve derleyici ortamında çalıştırabileceksiniz.

## ARAŞTIRMA

- Web yazılım editörleri nelerdir? Araştırınız.
- Derleyicinin ne olduğunu araştırıp öğreniniz.

## 3. TEMEL PROGRAMLAMA UYGULAMALARI

PHP ve MySQL kullanmayı öğrenebilmek için öncelikle, bahsedilen ürünlerin kaynak kodlarının ve Apache Web Server'ın bilgisayarınızda olması gerekmektedir. Bu kaynak kodlar sayesinde.php uzantılı dosyalarınızın çalıştırılabildiğini daha önceki konularımızdan öğrenmiştik. Bu bölümde program temelleri hakkında bilgiler edinip ufak programlar yazacağız. Öncelikle bilinmesi gereken bazı kavramalara değinelim.

### 3.1. Değişkenler

Değişken, kabaca içi boş bir bardak gibi düşünülebilir. Bardak, içine su koyarsan suyu; cıva koyarsan cıvayı tutmuş olur. Bardakta ne olduğu sorusuna içindekine göre farklı cevaplar verilir. Değişkenler, program içerisinde değer atanmak için kullanılan ifadelerdir. Bir karenin alanını hesaplatmada kullanılan ALAN, bir öğrencinin aldığı notları gösteren NOT, kişinin yaşadığı MEMLEKET birer değişken olarak tanımlanabilir. Her seferinde alan, not, memleket değişebilir.

Değişken tanımlarken aşağıdaki hususlara dikkat edilmelidir. Aksi takdirde programın çalışması esnasında hata oluşur.

- Değişkenler **dolar** işareti (\$) ile başlar.

YANLIŞ	DOĞRU
SAYI = 20 PASSWORD=1453	\$SAYI=20 \$PASSWORD=1453

- Değişken ataması yapılırken **eşittir(=)** operatörü kullanılmalıdır.

YANLIŞ	DOĞRU
\$SAYI : 20 \$AD - ERDAL	\$SAYI=20 \$AD="ERDAL"

- Değişkene atanan değer metin ise **tırnak** içinde yazılmalıdır.

YANLIŞ	DOĞRU
\$SEHIR =ANKARA \$MESLEK=DOKTOR	\$SEHIR ="ANKARA" \$MESLEK="DOKTOR"

- Değişkenler **rakamla** başlamaz.

YANLIŞ	DOĞRU
\$1NOT=80 \$5A=15	\$NOT1=80 \$A5=15

- Değişkenler iki kelimedenden oluşuyorsa kelimeler arasında **boşluk** olmamalıdır.

YANLIŞ	DOĞRU
\$BUYUK SAYI=20 \$KUCUK SAYI=10	\$BUYUK_SAYI=20 \$KUCUK_SAYI=10

- Değişkenlerde **Türkçe karakter** (ç, ğ, ü, ö, ş, ı, İ) ve **özel semboller** (£ @ / \* > < { } & % ...) kullanılmaz.

YANLIŞ	DOĞRU
\$YAŞ=35 \$DOĞUM YILI=1980 \$I@D/=12	\$YAS=35 \$DOGUM_YILI=1980 \$ID=12



## 3.2. Sabitler ve Kullanımı

Sabit tanımlarken **define** komutu kullanılır. Bu komut değişken tanımlamadan farklı olup adı üstünde, değişmeyen, yani sabit, içeriği değişmeyen bir bilgi için kullanılır.

Örneğin geometride kullanılan pi sayısı bir sabittir.

Sabit tanımlaması aşağıdaki şekilde yapılır:

```
define ("Sabit_Adi","Sabitin_Değeri");
```

Sabit tanımlarken değişkenlerde olduğu gibi ön ek olarak dolar(\$) işareti kullanılmaz.



Sabitler programda çalıştırıldığında her zaman ilk değerini korur, ikinci bir değer almaz.

```
<?php
define ("metin", "Merhaba Sabit");
define ("metin", "Merhaba Osman");
echo metin;
?>
```

Burada “metin” değişkeni ilkinde “Merhaba Sabit” ikincisinde ise "Merhaba Osman" olarak tanımlanmıştır. Ekrana yazdırılacağı zaman ikinciye değil ilk aldığı değeri “Merhaba Sabit” değerini yazdırır.

Aynı işlemi değişken kullanarak yapalım:

```
<?php
$metin="Merhaba Sabit";
$metin="Merhaba Osman";
echo $metin;
?>
```

Bu örnekte ise "Merhaba Osman" yazar. Çünkü değişkenin değeri ikinci satırda değişmiştir.

## 3.3. Veri Türleri

PHP’de değişkenler bizim için programın akış sırasına göre veri taşır. Bütün programlama dillerinde veri türleri sınıflandırılmıştır. Bir değişkene atanan bir verinin türü önemlidir. Yine bütün programlama dillerinde olduğu gibi PHP, tuttuğu veri türüne göre bilgisayarın belleğinde bunlara yer açar ve orada kullanılmak üzere hazır tutar.

PHP’de deęişken tipleri, oluřturduęunuz deęişkenlere uyguladıęınız deęiřikliklere gre otomatik olarak belirlenir. Yani bir deęiřkene ilk tanımlamasında bir sayısal deęer verdiyseniz tipi otomatik olarak tamsayı ya da duruma gre ift olur.



Deęer, deęiřkene tırnak iřareti kullanılarak verildiyse deęiřken alfa nmerik bir deęer alır.

Veri Tr	Aıklama	rnek
Integer	Tam sayı	23,002, 5315 gibi bir sayı
Double	Ondalık	6,9999 ondalık bir sayı gibi
String	Alfa nmerik	“MEGEP” gibi bir cmle. İinde sayı ve semboller olabilir.
Boolean	Mantıksal	doęru (true) / yanlış (false) gibi bir karřılařtırma
Object	Nesne	Nesne ve sınıfların tanımlandıęı deęiřkenler. \$web->(); gibi
Array	Dizi	\$dizi=array(1,2,3,4); gibi

### 3.4. Operatrler

Deęiřkenin ne olduęuna daha nceki konuda deęinmiřtik. Operatrler ise deęer atamak, karřılařtırma yapmak ve mantıksal sınama yapmak iin kullanılır. PHP dilinde kullanılan operatrleri ařaęıdaki řekilde gruplandırabiliriz:

#### ➤ Matematiksel operatrler

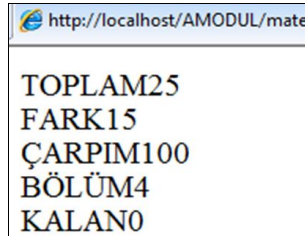
Matematikte kullanılan drt iřlem ve mod iřlemleri iin kullanılan operatrlerin kullanım řekli ařaęıdaki gibidir:

Aıklama	Operatr	rnek Kullanımı
Toplama	+	\$toplama = \$a + 5
ıkarma	-	\$fark=\$a - 10
Blme	/	\$bolum = \$a / 3
arpma	*	\$carpim = \$a * 2
Mod	%	\$ kalan = \$a % 2

Örnek üzerinde operatörün nasıl kullanıldığını inceleyelim:

```
1 <?php
2 $a=20; // A değişkenine 20 değerini atadık.
3 $b=5; // B değişkenine 5 değerini atadık.
4 $toplam=$a+$b; // A ve B toplatıp sonucu $toplam'a atandı.
5 $fark=$a-$b; // A, B den çıkartıp sonucu $fark değişkenine atandı.
6 $carpim=$a*$b; // A ve B çarpılıp $carpim'a atandı.
7 $bolum=$a/$b; //A, B'ye bölünüp sonuç bölüm değişkenine atandı.
8 $kalan=$a%$b; // A nın B ye bölümünden kalanı $kalan atandı.
9 echo "TOPLAM".$toplam; //Toplam değişkenini ekrana yazdırdık
10 echo "<br>"; //Bir satır boşluk bıraktık.
11 echo "FARK".$fark; //Fark değişkenini ekrana yazdırdık
12 echo "<br>";
13 echo "ÇARPIM".$carpim; //Carpim değişkenini ekrana yazdırdık
14 echo "<br>";
15 echo "BÖLÜM".$bolum; //Bolum değişkenini ekrana yazdırdık
16 echo "<br>";
17 echo "KALAN".$kalan; //Kalan değişkenini ekrana yazdırdık
18 ?>
```

Program çalıştırıldığı zaman ekran çıktısı aşağıdaki gibi olacaktır:



```
http://localhost/AMODUL/mate
TOPLAM25
FARK15
ÇARPIM100
BÖLÜM4
KALAN0
```

#### ➤ Kıyas ve koşul operatörleri

Şartlı ifadeler ve döngüler içinde koşul belirtmek ya da kıyaslamak amaçlı kullanılır.

Açıklama	Operatör	Kullanım Şekli
Eşittir	==	\$sayi1 == \$sayi2
Eşit Değil	!=	\$sayi1 != \$sayi2
Büyük	>	\$sayi1 > \$sayi2
Küçük	<	\$sayi1 < \$sayi2
Büyük Eşit	>=	\$sayi1 >= \$sayi2
Küçük Eşit	<=	\$sayi1 <= \$sayi2

➤ **Mantıksal işlem operatörleri:**

Php dilinde kullanılan mantıksal iki ifade vardır. Bunlar ve(AND), veya(OR) olarak kullanılır.

Açıklama	Operatör	Kullanım Şekli
Ve	And	(\$A AND \$B)
Veya	Or	(\$A OR \$B)
Ve	&&	(\$A && \$B)
Veya		(\$A    \$B)
Değil	!	(!\$sayi1)

➤ **Birleştirme operatörleri**

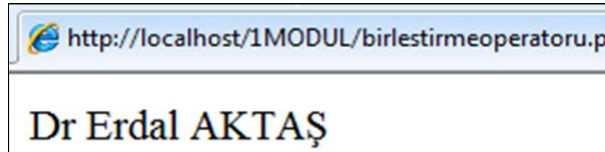
Metin işlemlerinde kullanılan bu operatör birden fazla metni veya içine metin atanmış değişkeni birleştirmek için kullanılır.

Açıklama	Operatör	Kullanım Şekli	Sonuç
Birleştirme	.	"Ankara"."Başkentimiz"	AnkaraBaşkentimiz

Örnek üzerinde operatörün nasıl kullanıldığını inceleyelim:

```
1 <?php
2 $Ad="Erdal"; //Ad değişkenine Erdal ismini atadık.
3 $Soyad="AKTAŞ"; //Soyad değişkenine AKTAŞ ismini atadık.
4 $Unvan="Dr"; //Unvan değişkenine Erdal ismini atadık.
5 $kisi=$Unvan." ".$Ad." ".$Soyad; //Üç değişkeni birleştirdik
6 echo $kisi; //Kişi değişkenini yazdırdık.
7 ?>
```

Program çalıştırıldığı zaman ekran çıktısı aşağıdaki gibi olacaktır.



İki metni birleştirirken arada boşluk olması için iki çift tırnak (" ") kullanılır.

### ➤ Atama operatörü

Değişkenlere değer atamak için kullanılır.

Açıklama	Operatör	Kullanım Şekli
Atama	=	\$sayi=10

### ➤ Azaltma ve artırma operatörleri

Bu operatör değişkenin değerini bir artırmak veya azaltmak için kullanır.

Açıklama	Operatör	Örnek Kullanımı
Sayı değişkeninin değerini 1 artırır	++	\$sayi++ veya ++\$a
Sayı değişkeninin değerini 1 azaltır	--	\$a-- veya --\$a

- Artırma ve azaltma operatörleri değişkenden önce kullanılıyorsa:
  - Önce değer artırılır veya azaltılır,
  - Sonra işlem yapılır.

#### Örnek:

```
$a=5;  
Printf $a++;
```

**Ekran Çıktısı: 5**

- Artırma ve azaltma operatörleri değişkenden sonra kullanılıyorsa:
  - Önce işlem yapılır,
  - Sonra değer artırılır veya azaltılır.

#### Örnek

```
$a=5;  
Printf ++$a;
```

**Ekran çıktısı: 6**

## 3.5. Program Denetimi

Belli bir koşula göre programın işleyişinin değiştirilmesi işlemini denetim komutlarıyla yaparız.

Programın çalışması esnasında birkaç farklı alternatif durum arasından seçim yapmamız, bazı komutları birden fazla tekrarlatmamız, mantıksal kararlar vermemiz ya da istenmeyen hataları önlemek amacıyla programımıza müdahale etmemiz gerekebilir. Bu gibi durumları kontrol altına almak amacıyla kontrol yapılarını ve döngüleri kullanıyoruz. Şarta bağlı olarak işlem yapmak için if ve switch ifadelerini, döngüler için ise for ve while ifadelerini kullanıyoruz. Şimdi bunları sırayla ve örneklerle inceleyelim.

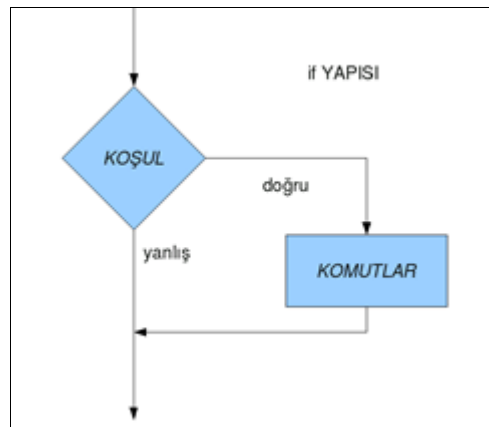
### 3.5.1. If – Else Kullanımı

Bu deyimimin kullanımı aşağıdaki şekildedir:



**if ( koşullar )**

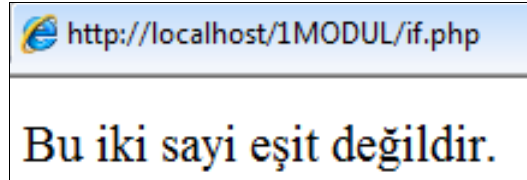
```
{  
koşullar doğru ise yapılacak işlere ilişkin komutlar  
}  
else  
{  
koşul sağlanmamışsa yapılacak işlere ilişkin komutlar  
}
```



Şimdi yukarıda anlatılanları bir programda uygulayalım:

```
1 <?php  
2 $sayi1 = 8;  
3 $sayi2 = 4;  
4 if ($sayi1 == $sayi2) { // Eğer sayi1 sayi2'ye eşitse  
5     echo "Bu iki sayi eşit"; //Ekрана bu iki sayı eşit yazdır  
6 }else { // Değilse  
7     echo "Bu iki sayi eşit değildir."; //Ekрана eşit değil yazdır  
8 }  
9 ?>
```

Bu program çalıştırıldığında sayı1 ile sayı2 eşit olmadığı için sizin de tahmin edeceğiniz gibi ekran çıktısı aşağıdaki gibi olacaktır:



Else, kelime manası olarak değilse manasına gelmektedir. If komutu ile else komutu birlikte kullanılır. Eğer şart oluşmıyorsa else'den sonraki işlem yürütülür.

Şimdi if else kullanarak daha önce öğrendiğimiz mantıksal AND(&&) parametresini de içeren bir örnek yapalım.

```
1 <?php
2 $sifre = 1453;
3 $uye = "Fatih";
4 if ($sifre==1453 AND $uye == "Fatih") {
5     //sifre 1453 ve uye Fatih ise
6     echo "Doğru giriş yaptınız"; //Doğru giriş yaptınız yaz
7     }else { // Değilse
8     echo "Giriş Hatalı"; //Ekrana Giriş Hatalı yazdır
9     }
10 ?>
```

Yukarıdaki programda sifre=1453 , uye =Fatih değerini almıştır. 4. satırda sifre ve uye değerleri kontrol edilip her ikisinin de doğru olması durumunda ekrana **“Doğru Giriş Yaptınız”** yazmaktadır.

Programda her iki değer de doğru olduğu için giriş doğru olacaktır. Fakat siz iki ve üçüncü satırdaki **sifreyi** veya **uye** değerini değiştirerek **“Giriş Hatalı”** mesajı verdirebilirsiniz.

### 3.5.2. Else if

İkiden fazla koşulun olduğu durumlarda Else if kullanılır. “Eğer değilse” manasına gelen bu ikili bütün şartları kontrol ettikten sonra kalan son şartı “Else” deyip icra eder.

```
1 <?php
2 $Ali_yas=35;
3 $Osman_yas=40;
4
5 if($Ali_yas>$Osman_yas)//Ali Osman'dan büyükse
6 {
7     echo "Ali Büyük";
8 }
9 Else if ($Osman_yas>Ali_yas)//Osman Ali'den büyükse
10 {
11     echo "Osman Büyük";
12 }
13 else // Değilse
14 {
15     echo "Yaşlar eşit";
16 }
17 ?>
```

Bu programda üç ihtimâl vardır: Ali büyük, Osman büyük veya yaşlar eşit. Birinci if deyimi ile (5.satır) Ali'nin büyük olma durumunu, ikinci if deyimiyle (9.satır) Osman'ın büyük olma durumunu sorgulattık. İkisi de değilse (13. Satır) son ihtimâli yazdırdık. Yukarıdaki program çalıştırıldığı zaman birinci şart değil, ikinci şart gerçekleştiği için ekrana “**Osman Büyük**” ifadesi yazdırılır.

Sizler de 2. satırda bulunan **\$Ali\_yas** değişkenine farklı değerler vererek diğer iki ihtimale göre ekran çıktısının değişimini uygulayınız.

Şimdi örneğimizdeki ihtimâl sayısını artırarak farklı bir örnek daha çözelim:

Bu örnekte öğrencinin aldığı puana göre ekrana notunu yazdıran programı yapalım. Puan-Not tablosu aşağıda verilmiştir.

PUAN	NOT
0-24	SIFIR
25-44	BİR
45-54	İKİ
55-69	ÜÇ
70-84	DÖRT
85-100	BEŞ

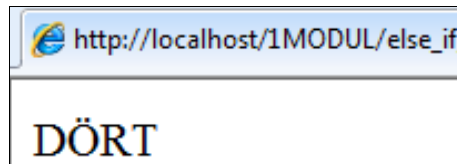


```

1 <?php
2 $PUAN=70; //Not değişkenine 70 değerini atadık.
3 if($PUAN>84) //Puan 84'den büyükse
4     { echo("BES"); } //Ekрана BEŞ yaz
5 elseif($PUAN>69) //Puan 69'den büyükse
6     { echo("DÖRT"); } //Ekрана DÖRT yaz
7 elseif($PUAN>54) //Puan 54'den büyükse
8     { echo("ÜÇ"); } //Ekрана ÜÇ yaz
9 elseif($PUAN>44) //Puan 44'den büyükse
10    { echo("İKİ"); } //Ekрана İKİ yaz
11 elseif($PUAN>24) //Puan 24'den büyükse
12    { echo("BİR"); } //Ekрана BİR yaz
13 else //Değilse
14    { echo("SIFIR"); } //Ekрана SIFIR yaz
15 ?>

```

Yukarıdaki programımız çalıştığı zaman ekran çıktısı aşağıdaki şekilde olmaktadır:



Sizler programımızın 2. satırında bulunan \$PUAN değişkenine puan-not tablosundan faydalanarak değişik değerler atayarak puana göre notun değişip değişmediğini kontrol edin.



\$PUAN>85 ibaresi ile \$PUAN>=84 ibaresi aynı koşulu ifade etmektedir.

```

if($PUAN>84) //Puan 84'den büyükse
//if($PUAN>=85) Puan 85'eşit ve büyükse
    { echo("BES"); } //Ekрана BEŞ yaz

elseif($PUAN>69) //Puan 69'den büyükse
//if($PUAN>=70) Puan 70'eşit ve büyükse
    { echo("DÖRT"); } //Ekрана DÖRT yaz

```

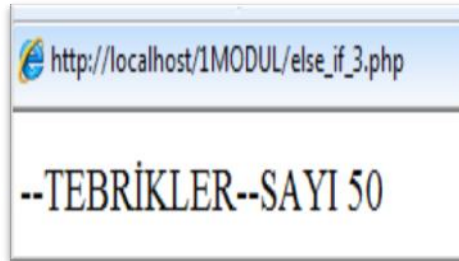
Else if komutunun pekişmesi açısından son bir örnek olarak sayı bulma programı yazalım.

Kişinin tuttuğu sayıyı “Daha Büyük Sayı” veya “Daha Küçük Sayı” gibi yönlendirme mesajlarıyla bulmaya çalışan ve tutulan sayı tahmin edilen sayıya eşit olduğunda da “TEBRİKLER” mesajıyla tahmin edilen sayıyı ekrana yazdıran programı yazalım.

```
1 <?php
2 $sayi=50; //Tutulan sayı değeri
3 $tahmin=50; //Tahmin edilen sayı
4
5 if($sayi>$tahmin) // Tutulan sayı tahminden büyükse
6 {echo "Daha Büyük Bir Sayı"; } //Tahmini artır
7
8 elseif($tahmin>$sayi) // Tutulan sayı tahminden küçükse
9 {echo "Daha Küçük Bir Sayı"; } //Tahmini azalt
10
11 else //Değilse
12 {echo "--TEBRİKLER--SAYI"." ".$sayi; }
13 //Ekrana tebrik mesajı ve sayıyı yazdır
14 ?>
```

Yukarıdaki örneği, bir sonraki modülün konusu olan form yardımıyla yaparsanız daha kullanışlı bir uygulama olacaktır. Bu örnekte 3.satırdaki tahmin değerine sayı değerinden düşük ve yüksek değerler atayarak programın çalışmasını kontrol edebilirsiniz.

Bu haliyle program çalıştığında ekrana aşağıdaki gibi bir görüntü gelecektir.



Ekran çıktısını incelediğimizde --TEBRİKLER--SAYI ibaresinden sonra birleştirme operatörü (.) kullanılıp boşluk bırakılıp sonra tekrar birleştirme operatörü (.) kullanılıp tutulan sayı yazılmıştır.

```
12 {echo "--TEBRİKLER--SAYI"." ".$sayi; }
```

Kare içine alınmış iki noktanın arasında bulunan çift tırnaklar boşluk bırakmak amaçlı kullanılmıştır.

### 3.5.3. if Deyimi İle Kullanılan Komutlar

Bazı zamanlar **if** deyimiyle bazı komutlar kullanırız. Bu komutlar yeri geldiğinde formdan gelen bilgilerin derlenmesinde işe yarayabileceği gibi değişkeni kontrol etmek için de kullanılırlar.

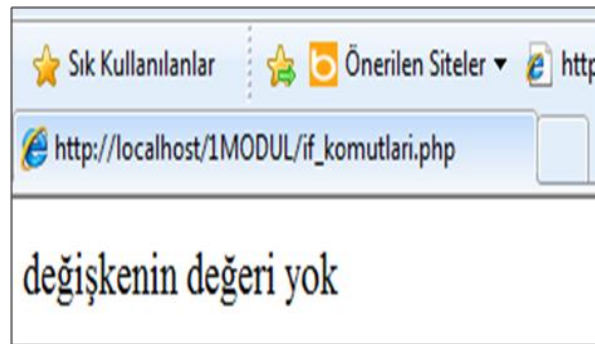
Bu komutların görevlerinin anlaşılması belki tanımdan biraz zor olabilir. Ancak örnek uygulamalarla ne söylemek istenildiği anlaşılacaktır.

Konuyu fazla uzatmamak için bu komutları görevleri ile birlikte aşağıda yazıp sonra örnek uygulamaya geçelim.

- **isset()** : Bu komut bir değişken tanımının içerisinde değer olup olmadığını rapor eder.
- **empty()** : Buda isset ile aynı işi görür esasında. Ancak sınımayı tersten yapar. isset()te dönen değer true ise empty() de aynı değer false olarak döner.

```
1 <?php
2
3 // $sifre değişkeni tanımlamadık
4
5 if(isset($sifre)) //Eğer değişken değeri varsa
6 { echo "Değişkenin değeri var"; }
7 else //Yoksa
8 { echo "Değişkenin değeri yok"; } //Değişken yok yaz
9
10 ?>
```

Yukarıdaki örnekte \$sifre değişkeni tanımlanmamış. Dolayısıyla böyle bir değişken ve içeriği de yok. If komutuyla 5. satırda bu kontrolü yaptığımızda değer olmadığı için ekrana “Değişkenin değeri yok.” yazdı.



Esasen burada yapılan işlem değişkenin olup olmadığı değil, değişkenin içinde değer olup olmadığıdır.

Şimdi de aynı örneği değişken tanımlaması yaparak yazalım.

```
1 <?php
2
3     $sifre=1071; //$sifre değişkeni tanımladık
4
5     if(isset($sifre)) //Eğer değişken değeri varsa
6     { echo "Değişkenin değeri var"; }
7 else //Yoksa
8     { echo "Değişkenin değeri yok"; } //Değişken yok yaz
9
10 ?>
```

Program çalıştırıldığında ekran görüntümüz, değişken tanımlaması yaptığımız için “Değişken değeri var.” şeklinde değişmiştir.



Php dilinde ünlem(!) işareti tersi manasına gelir. Aşağıda gösterildiği gibi biri diğerinin yerine kullanılabilir.

<code>if(!isset(\$sifre))==if(empty(\$sifre))</code>
--

<code>if(!empty(\$sifre))==if(isset(\$sifre))</code>
--

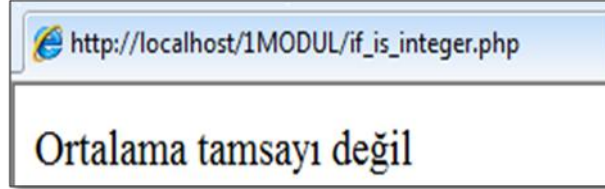
Siz yukarıdaki örnekleri empty komutu ve ünlem(!) işaretini kullanarak farklı kombinasyonlarda deneyebilirsiniz.

- **is\_integer()** : Bu komut değişkenin tamsayı olup olmadığını sınar ve “ True” veya “False” değeri döndürür.

#### Örnek uygulama:

```
1 <?php
2 $sayi1=11;
3 $sayi2=6;
4
5 $ortalama=($sayi1+$sayi2)/2;
6 //İki sayının ortalaması
7
8 if(is_integer($ortalama)) //Ortalama tamsayı ise
9 {echo "Ortalama tamsayı";} // Ekran yaz
10 else //Değilse
11 {echo "Ortalama tamsayı değil";}
12
13 ?>
```

Yukarıdaki örnekte iki sayıyı toplayıp ortalamasını aldırdıktan sonra ortalamanın tamsayı olup olmadığını kontrol ettiriyoruz. Örneğimizde ortalama  $17/2=8,5$  çıkıyor. Dolayısıyla tamsayı değil. Bu nedenle program çalıştığında ekran görüntüsü aşağıdaki gibi olacaktır.



#### 3.5.4. Switch Kullanımı

Türçe karşılığı “anahtar” olan switch deyimi if deyimi gibi görev yapar. Switch, seçilmiş bir değişkenin belli şartlardan hangisine uygun olduğuna bakıp o şart için belirlenmiş işlemleri gerçekleştirir.

Switch ile yaptıklarımızın hepsini if ile yaparız ama if ile yaptığımız her şeyi switch deyimi ile yapamayız.

Örnek bir uygulama ile switch deyiminin kullanımını inceleyelim:

Seçim	1	2	3	4	5	6	7
Günler	Pazartesi	Salı	Çarşamba	Perşembe	Cuma	Cumartesi	Pazar

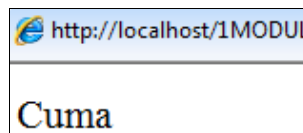
Yukarıdaki tabloda hangi sıra numarasını seçersek o numaraya karşılık gelen günü ekrana yazdıran programı yazalım.

```

1  <?
2  $secim=5;
3  switch($secim)  {
4  case 1;
5  echo "Pazartesi";
6  break;
7  case 2;
8  echo "Salı";
9  break;
10 case 3;
11 echo "Çarşamba";
12 break;
13 case 4;
14 echo "Perşembe";
15 break;
16 case 5;
17 echo "Cuma";
18 break;
19 case 6;
20 echo "Cumartesi";
21 break;
22 case 7;
23 echo "Pazar";
24 break;
25 default;
26 echo "YANLIŞ SEÇİM"; }
27 ?>

```

Secim=5 olduğu için ekrana “Cuma” yazacaktır.



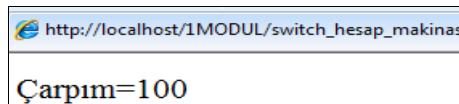
Siz \$secim değişkenine atanan numarayı değiştirerek) ekranda (2. Satırda) farklı günlerin çıkmasını sağlayabilirsiniz. Seçim aralığı [1-7] dışında bir sayı seçildiğinde böyle bir gün olmadığı için default değer olarak “YANLIŞ SEÇİM” gözükecektir.

**Örnek:** İki farklı değişkene atılmış sayıları seçime göre toplama, çıkarma, çarpma veya bölme işlemine tabi tutan sonucu ekrana yazdıran programı yazalım.

```
1  <?
2  $sayi1=20;
3  $sayi2=5;
4  // İŞLEMLER>>>toplama-çıkarma-çarpma-bölme
5  $islem="çarpma";//Çarpma işlemini seçtik
6  switch ($islem)
7  {
8  case "toplama"; //işlem toplama olursa
9  $sonuc=$sayi1+$sayi2; //iki sayıyı topla
10 Echo "Toplam=".$sonuc; //Sonucu yaz
11 Break;
12
13 case "çıkarma";//işlem çıkarma olursa
14 $sonuc=$sayi1-$sayi2; //sayı1'den sayı2'yi çıkar
15 Echo "Fark=".$sonuc; //Sonucu yaz
16 Break;
17
18 case "çarpma";//işlem çarpma olursa
19 $sonuc=$sayi1*$sayi2; //iki sayıyı çarp
20 Echo "Çarpım=".$sonuc; //Sonucu yaz
21 Break;
22
23 Case "bölme"; //işlem bölme olursa
24 $sonuc=$sayi1/$sayi2; //sayı1'i sayı2'ye böl
25 Echo "Bölüm=".$sonuc; //Sonucu yaz
26 Break;
27
28 Default: //İşlem dört işlem dışında ise
29 Echo "Yazdığınız işlem geçersiz";
30 }
31 ?>
```

Bu işlem olarak çarpmayı seçtiğimiz (5.satır) için program 18.satır ile 21. Satır arası çalışıp sonlanmıştır.21.satırda bulunan break komutu switch ten çıkmayı sağlar.

Programın çalışmasından sonraki ekran görüntüsü aşağıdaki gibidir.



### 3.5.5. Döngüler

Bir işlemin birden fazla tekrarlanması durumunda döngüler kullanılır. Döngüdeki mantık başlangıç ve bitiş değeri belli olan işlemin istenilen aralıklarda tekrarlanmasıdır.

İlköğretimde yaptığımız gibi defterimize adımızı 100 kez yazmak, 1'den başlayarak ikişer ikişer 200'e kadar yazmak veya 100'den geriye doğru beşer beşer yazmak birer basit döngü örnekleri olarak verilebilir.



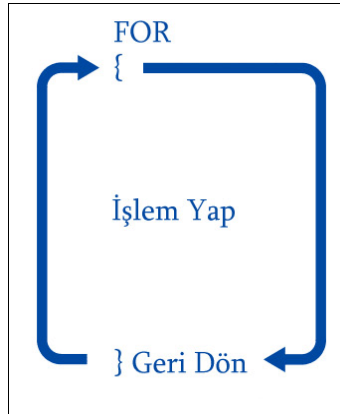
Döngü sayısı için başlangıç-bitiş ve artış değerleri önemlidir. 1 den 100'e kadar olan bir işlemde artış miktarı 2 olarak ayarlanırsa döngü içindeki komutlar **50** defa çalıştırılacaktır.

Döngü işlemleri **for** ve **while** komutlarıyla yapılır.

#### 3.5.5.1. for Döngüsü

For döngüsünün kalıbı aşağıdaki gibidir:

```
for (döngü değişkeni=başlangıç değeri; şart; değişim miktarı)  
{  
    yapılacak işlemler  
}
```





**Örnek:** Ekran 8 defa “MERHABA PHP” yazan programı yazalım.

```
1 <?
2 for($i=1;$i<=8;$i++)
3 //1'den başlayarak 8'e kadar 1'er 1'er artır
4 { //Döngü başlıyor
5     echo "MERHABA PHP"; //Metni ekrana yazdırıyoruz.
6     echo "<br>"; //Bir alt satıra geç.
7 } //Döngü bitiyor
8 ?>
```

Bu programda \$i değişkeni sayaç değişkenidir. Programın 2.satırında gördüğünüz gibi \$i değeri 1'den başlamış, küçük eşit 8 oluncaya kadar değerini birer birer artırmıştır.

```
2 for($i=1;$i<=8;$i++)//1'den başlayarak 8'e kadar 1'er 1'er
```

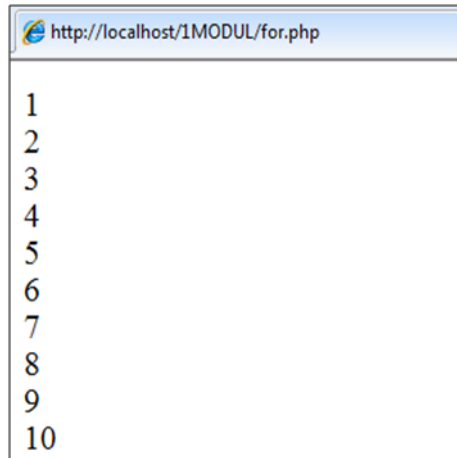
Her artırma işleminde ise süslü parantez arasındaki komutu çalıştırmıştır. Bu işlemi toplam 8 defa yapmıştır. Dolayısıyla programın çalışmasından sonraki ekran görüntüsü aşağıdaki gibi olmuştur.



**Örnek:** 1’den başlayarak 10’ a kadar ekrana alt alta yazdırmak istiyoruz.

```
1 <?
2 for($i=1;$i<=10;$i++)
3 //i'nin değerini 1'den başlayarak 10'a kadar 1'er artır
4 { //Döngü başlıyor
5     echo $i; // i değişkenin değerini ekrana yaz
6     echo "<br>"; //Bir alt satıra geç.
7 } //Döngü bitiyor
8 ?>
```

Döngü işlemlerinde bir tane sayaç olmak zorundadır. Bu sayacın görevi her seferinde döngünün kaçınıcı defa işlendiğini tutmaktır. Bu örnekte kullanılan \$i sayacı aynı zamanda 4. Satırda ekrana yazılmaktadır.



Şimdi yukarıdaki iki örneği birleştirip bir örnekte gösterelim.

**Örnek:** Ekrana 1’den başlayıp sıra numarası vererek ekrana 10 defa “ MERHABA PHP ” yazdıralım.

1.MERAHABA PHP  
2.MERAHABA PHP  
.  
.  
.  
.  
.  
10.MERAHABA PHP gibi

```

1 <?
2 for($i=1;$i<=10;$i++)
3 //i'nin değerini 1'den başlayarak 10'a kadar 1'er artır
4 { //Döngü başlıyor
5     echo $i.". ". "MERHABA PHP"; //Sayacı ve metni yaz
6     echo "<br>"; //Bir alt satıra geç.
7 } //Döngü bitiyor
8 ?>

```

Programın 5. satırında hem sayacı hem de metin birleştirme operatörü kullanılarak döngüye girmiş ve ekrana yazılmıştır.

Programın çalışması aşağıdaki şekilde olmaktadır:



Biz döngülerin içine aynı zamanda html etiketlerini de ekleyebilir, onları döngülerle beraber kullanarak çok değişik örnekler yapabiliriz.

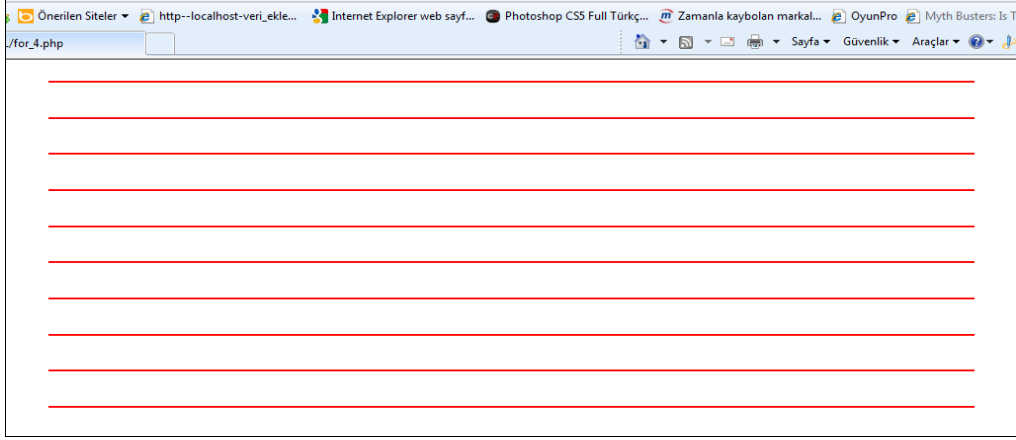
**Örnek:** Ekrana 900 pix uzunluğunda alt alta 10 tane çizgiyi döngü yardımıyla çizdirelim.

```

1 <?
2 for($i=1;$i<=10;$i++) // i'nin değerini 10'a kadar 1'er artır
3
4 { //Döngü başlıyor
5     echo "<hr color=red width=900>";
6     //900 pixel genişliğinde kırmızı çizgi çizer
7     echo "<br>"; //Bir alt satıra geç.
8 }
9 ?>

```

Yukarıdaki programı çalıştırdığımız zaman ekrana alt alta 10 tane 900 pixel genişliğinde kırmızı renkte çizgi çizer. Programın çalışma görüntüsü aşağıdaki gibi olmaktadır.



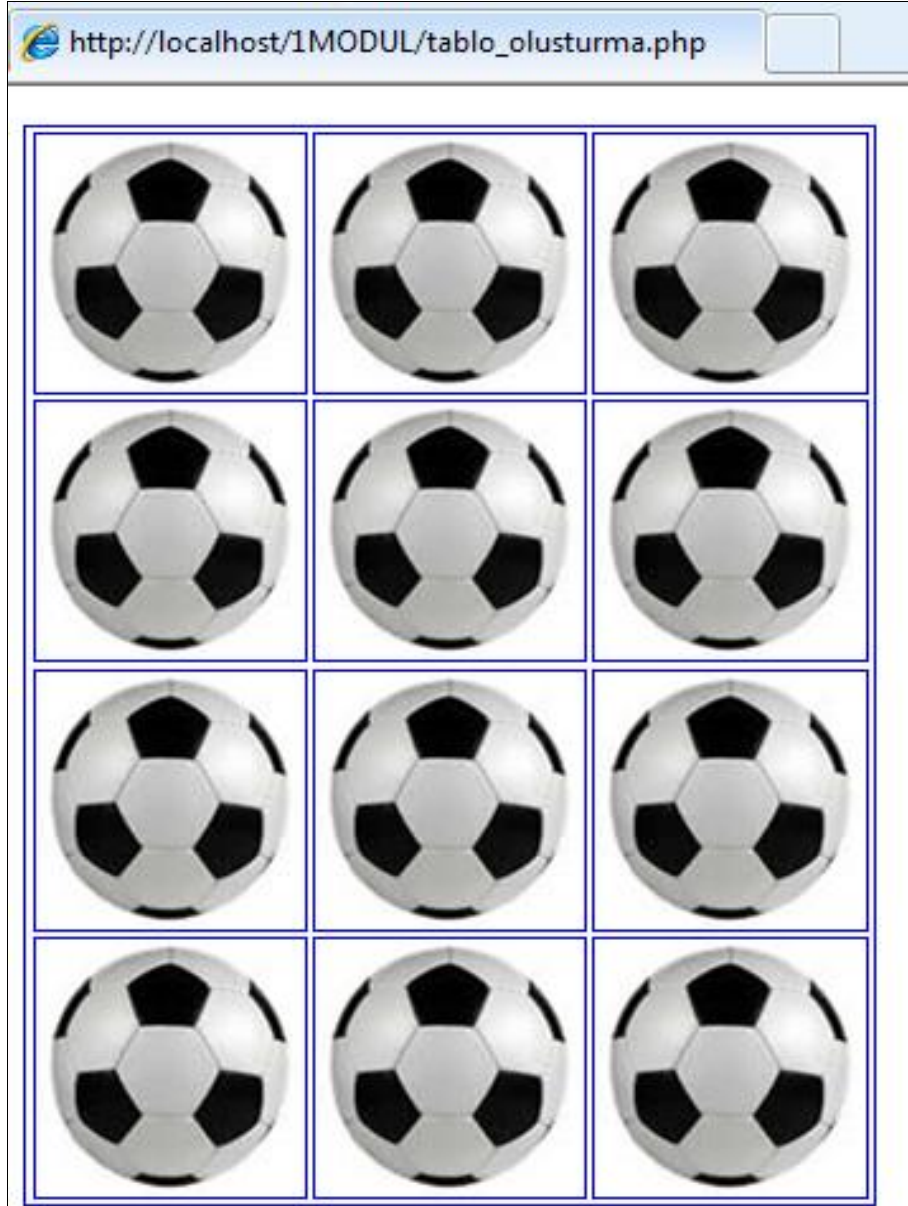
**Örnek: İstenilen** satır ve sütun sayısı kadar tablo oluşturan ve her bir hücreye top resmi ekleyen programı yazınız.(Top.jpg resmini programı çalıştırdığınız dizine ekleyiniz, resmin boyutlarını 100x100 pixel olarak ayarlayınız.)

```
1 <?php
2 $satirsayisi=4; //Satır sayısı
3 $sutunsayisi=3; //Sutun sayısı
4 echo "<table border=1 bordercolor=blue>";//tablo olusturduk
5
6 for($i=1; $i<=$satirsayisi;$i++) //Satır için döngü oluşturduk
7 {echo "<tr>"; // Satırı oluşturunuz
8   for($k=1; $k<=$sutunsayisi;$k++) //Sutun için döngü
9   {echo "<td><img src=top.jpg></td>";} //Sutunlar ve resim ekleniyor
10 echo "</tr>" ;//Açılan satır kapatılıyor
11 }
12 echo "</table>" //Tabloyu kapat
13 ?>
```

Yukarıdaki örnekte satır sayı değerini 4, sütun sayı değerini 3 verelim. Tek tablo oluşacağı için 4. Satırdaki tablo başlangıç etiketi döngü dışında kalıyor.

Burada iç içe iki tane döngü kullandık. Dıştaki döngü satır, içteki döngü sütun oluşturmak için kullanılacaktır. Önce, dıştaki döngü ile satır oluşturulmakta, daha sonra her satırda kaç tane sütun varsa içteki döngü yardımıyla o kadar sütun oluşturulmaktadır. Her satırdaki sütun oluşturma işlemi bitince dıştaki döngü devreye girip bir sonraki satırı oluşturmaktadır. İşlem bu şekilde satır sayısı bitinceye kadar devam etmektedir. Döngüler bittiğinde tablo kapatılmakta ve program tamamlanmaktadır.

Program çalıştırıldığında aşağıdaki şekilde bir görüntü elde edilmektedir:



### 3.5.5.2. While Döngüsü

While döngüsü for döngüsü ile aynı işlemi yapmaktadır. While Döngüsü, tanımlanan koşul sağlandığı sürece döngü içerisine yazılan komut ifadesinin veya komut ifadelerinin çalıştırılmasını sağlar.



Şart ifadesinin doğruluğu her döngünün başında kontrol edilir, buna göre döngünün içerisindeki komut ifadeleri çalıştırılmaya devam eder ya da döngü sona erer. Eğer şart sağlanmıyorsa döngü hiç çalışmaz.

While döngüsünün kalıbı aşağıdaki gibidir:

```
while(şart)
{
    Yapılacak işlemler
}
```

Örnek bir program üzerinde while döngüsünü inceleyelim. Daha önce for döngüsü ile yaptığımız 1' den 10' a kadar olan sayıları yazdıran programı while döngüsü ile yazalım. Böylece hem while döngüsünün kullanımını hem de iki döngü arasındaki farkları görmüş olacağız.

```
1 <?
2 $i=1; //başlangıç değerini belirledik.
3 while ($i<=10) //10 a kadar olma şartımızı belirledik.
4 {
5     echo $i; //değişken değerini yazdırdık.
6     echo "<br>"; //bir alt satıra geçtik.
7     $i++; //değişken değerimizi 1 artırdık.
8 }
9 ?>
```

Yukarıdaki programda ilk olarak tanımlanan bir "i" değişkenine 1 değeri atanmaktadır. While döngüsü, "i", 10' dan küçük olduğu sürece çalışacak şekilde kurulmuştur.

Programın 5. ve 6. satırlarında i'nin değeri ekrana yazdırılmakta ve bir satır alta inilmektedir. \$i++; komutu önemlidir, çünkü "i" değeri bu komutla değiştirilmektedir.

Döngümüz  $i$  değerine bağlı olarak kurulmuştur.  $i$  değerinin sabit kalması durumunda ( $\$i \leq 10$ ) ifadesi ( $1 \leq 10$ ) olarak kalacağından sürekli true sonucunu döndürecektir.

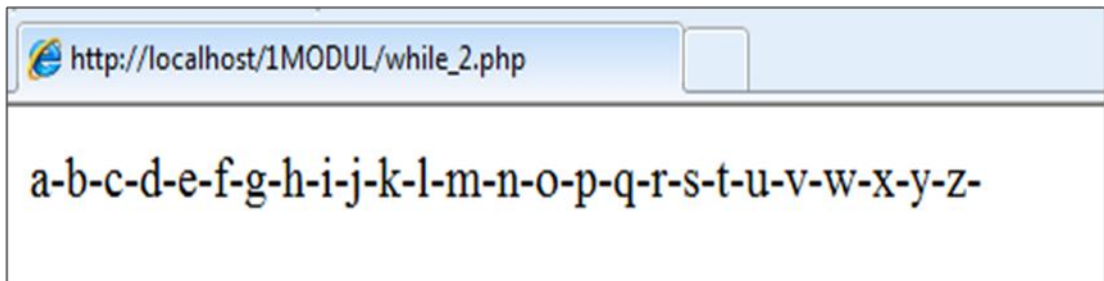
- While (True): Döngüye gir,
- While (False): Döngüye girme, anlamına gelmektedir.

Dikkat edilmesi gereken bir başka nokta ise  $\$i++$  ifadesi (Bu değer değişikliği komutunu farklı artış ve azalmalar sağlayacak şekilde de yazabilirsiniz.) döngümüzün sonunda olduğu için döngümüzün son çalıştığı seferde ( $i=10$  durumunda) son bir kez daha bir artırılarak 11'e eşit olacaktır.  $11 \leq 10$  ifadesi false sonucu döndürdüğünden döngü çalışmayacaktır.

**Örnek:** Bu uygulamada alfa numerik karakterleri (Türkçe karactersiz) yan yana aralarına tire(-) işareti koyarak while döngüsü yardımıyla yazdıralım.

```
1 <?php
2 $alfabe="a"; //Alfabe sayacımız
3 $sayac=1; //Sayma sayacımız
4 while ($sayac<=26) //Sayma sayacı 26 oluncaya kadar
5 { //İşleme başla
6 echo $alfabe."-"; //İlk alfabeyi yaz, "-" işareti koy
7 $sayac++; //Sayma sayacını bir artır
8 $alfabe++; //Alfabenin sonraki harfine geç
9 }
10 ?>
```

Bu program çalıştırıldığı zaman ekranda aşağıdaki gibi bir görüntü verir:



Bu programın, diğer uygulamalardan farklı olarak sadece numerik(sayısal) karakterde değil alfa numerik(alfabetik) karakterlerde de artırma ve azaltma işlemi yapabildiğini gördük.

### 3.6. Diziler (Array)

Diziler grup halindeki değişkenler olarak tanımlanabilir. Dizi içerisinde yüzlerce değişken olabilir. Dizi elemanları indis adı verilen sıra numarası ile çağrılır. Sıra numarası 0 (sıfır) ile başlar.

Dizilerin, tek boyutlu ve çok boyutlu olmak üzere iki çeşidi vardır.

#### ➤ Tek Boyutlu Dizi

Tek boyutlu diziler, sadece aynı özellikteki bir değer grubunu barındırmak için kullanılırlar. Aşağıdaki örnek dizi haftanın günlerini göstermektedir.

Bu örnekte günler, tek boyutlu bir dizi olup içerisinde 7 tane gün vardır.

İndis	0	1	2	3	4	5	6
Gunler	Pazartesi	Salı	Çarşamba	Perşembe	Cuma	Cumartesi	Pazar

Günler dizisi

➤ Dizi tanımlarken **array** deyimi kullanılır.

```
$gunler=array("Pazartesi","Salı","Çarşamba","Perşembe","Cuma","Cumartesi","Pazar");
```

Yukarıdaki örnekte \$gunler dizi değişkeni içine haftanın günlerini atadık. Haftanın günleri \$gunler dizi değişkeninin elemanlarıdır.

Her elemanın dizi içerisinde bir index numarası vardır. Dizi tanımlarken index numarası vermediğimizde otomatik olarak index numarası 0'dan başlar. Dizi içindeki herhangi bir elemana ulaşmak için index numarası kullanılır.

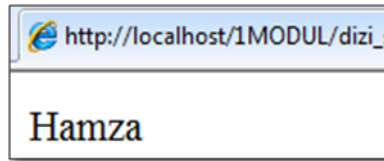
**Örnek:** Üye isimlerini üyeler adındaki bir dizi değişkeni içine atıp 2 numaralı dizi elemanını çağıralım.

```
1 <?
2 $uyeler=array("Yusuf","Ahmet","Hamza","Mehmet","Ömer");
3 echo $uyeler[2]; //index numarası 2 olan uyeyi yaz
4 ?>
```



```
$uyeler [0]= Yusuf  
$uyeler [1]= Ahmet  
$uyeler [2]= Hamza  
$uyeler [3]= Mehmet  
$uyeler [4]= Ömer
```

Örneğimizin 2. satırında uyeler adında kişilerin isimlerini tutan bir diziye array deyimi yardımıyla oluşturduk. 3. satırındaki echo \$uyeler[2] satırı, 2 numaralı dizi elemanını ekrana yazdırır.



Aynı örnekte bütün dizi elemanlarını ekrana yazdırmak istersek:

```
echo $uyeler[0];  
echo $uyeler[1];  
echo $uyeler[2];  
echo $uyeler[3];  
echo $uyeler[4];
```

şeklinde yazmamız gerekecektir. Dizi eleman sayısı çoğaldığında kodları uzunca yazmak yerine bir önceki konuda öğrendiğimiz döngü deyimlerini kullanabiliriz.

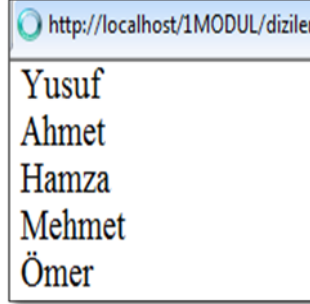
**Örnek:** Yukarıdaki dizi elemanlarının hepsini döngü deyimi kullanarak ekrana yazdıralım.

```
1 <?php  
2 $uyeler=array("Yusuf","Ahmet","Hamza","Mehmet","Ömer");  
3 for($i=0;count($uyeler);$i++) //Dizi elemanı kadar döngü  
4 { // Dönü başlangıcı  
5 echo $uyeler[$i]."<br>"; //$i. uyeler elamanını ekrana yaz  
6 }  
7 ?>
```

Bu örneğimizde:

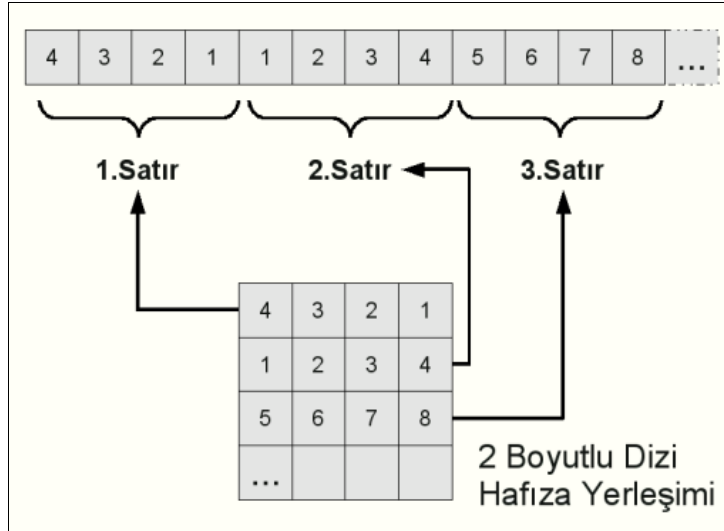
- 3.satırında kullanılan **count** deyimi dizideki eleman sayısını bulur.
- \$i döngü değişkeni 0'dan başlayarak dizi eleman sayısı kadar 1'er artmaktadır.
- 5.satırda ise dizinin 0. elemanından başlayarak dizinin n. elemanına kadar ekrana yazılmaktadır.

Bu programın çalışması sonucu ekran görüntüsü aşağıdaki şekilde olmaktadır:



### ➤ Çok Boyutlu Dizi

Çok boyutlu dizilerin kullanım amacı, farklı özellikteki değer gruplarını tek dizide, yani tek kapta barındırmaktır. Dizilerin çok boyutlu olması sizi yanıltmasın, bilgisayar hafızası tek boyutludur. İster tek boyutlu bir dizi, ister iki boyutlu ya da isterseniz 10 boyutlu bir dizi içerisinde bulunan elemanlar, birbiri peşi sıra gelen bellek hücrelerinde tutulur. İki boyutlu bir dizide bulunan elemanların hafızada nasıl yerleştirildiğini aşağıdaki grafikte görebilirsiniz:



Genellikle çok boyutlu dizilerde elemanlara değer vermek amacıyla ise “=>” operatörü kullanılır.

**Örnek1:** Tek boyutlu dizide kullandığımız elemanlara aşağıdaki tabloyu kullanarak yaş ve memleket bilgilerini ekleyelim. Bu sayede dizimiz çok boyutlu (isim, yaş, memleket) hâle gelmiş olacak.

Oluşturduğumuz bu dizinin elemanlarını daha sonra ekrana döngü deyimi kullanarak yazdıralım.

İndis	İsim	Yaş	Şehir
0	Yusuf	25	Çankırı
1	Ahmet	20	Van
2	Hamza	46	Erzurum
3	Mehmet	26	Bursa
4	Ömer	36	Mersin

“Uyeler” dizisi

```
1 <?php
2 $uyeler= array ( //Çok boyutlu dizisi tanımlanıyor
3 array(adi=> "Yusuf", yasi=>25, memleket=>"Çankırı" ),
4 array(adi=> "Ahmet", yasi=>20, memleket=>"Van" ),
5 array(adi=> "Hamza", yasi=>46, memleket=>"Erzurum" ),
6 array(adi=> "Mehmet", yasi=>26, memleket=>"Bursa" ),
7 array(adi=> "Ömer", yasi=>36, memleket=>"Mersin" ),
8 );
9
10 //Dizi elemanı kadar döngü oluşturuluyor
11 for( $i=0; $i<count($uyeler); $i++)
12 {
13 //Her dizi elemanı $i numarası ile çağrılıp ekrana yazılıyor
14 echo $uyeler[ $i ][adi]." - ".$uyeler[ $i ][yasi].
15 " - ".$uyeler[ $i ][memleket]."<br>";
16 }
17 }
18 ?>
```

http://localhost/1MODUL/dizi\_cok\_boyut.php

Yusuf - 25 - Çankırı  
Ahmet - 20 - Van  
Hamza - 46 - Erzurum  
Mehmet - 26 - Bursa  
Ömer - 36 - Mersin

**Örnek2:**Aşağıdaki tabloyu inceleyelim:

BASKANLAR	SINIF-10	ATL10-A	Sinan
		ATL10-B	Sıddık
		ATL10-C	Mesut
	SINIF-11	ATL11-A	Deniz
		ATL11-B	Cemalettin
		ATL11-C	Erdal

Bu tablonun elemanlarını bir dizi şeklinde tanımlayacak olursak bu dizinin kodları aşağıdaki şekilde olacaktır:

```
1 <?php
2 $baskanlar=array(
3
4 "SINIF-10">array("ATL10-A">"Sinan","ATL10-B">"Sıddık","ATL10-C">"Mesut"),
5 "SINIF-11">array("ATL11-A">"Deniz","ATL11-B">"Cemalettin","ATL11-C">"Erdal")
6
7 );
8 echo $baskanlar["SINIF-11"]["ATL11-B"]; //Ekрана Cemalettin yazacaktır
9 ?>
```



En son tanımlanan dizinin sonunda (5.satırın sonu) virgül (,) yoktur. Çünkü virgül dizi elemanlarını ayırır. En son elemandan sonra eleman olmadığından virgül konmaz.

Yukarıdaki diziyi aşağıdaki şekilde de tanımlayabiliriz:

```
1 <?php
2 $baskanlar["SINIF-10"]["ATL10-A"]="Sinan";
3 $baskanlar["SINIF-10"]["ATL10-B"]="Sıddık";
4 $baskanlar["SINIF-10"]["ATL10-C"]="Mesut";
5
6 $baskanlar["SINIF-11"]["ATL11-A"]="Deniz";
7 $baskanlar["SINIF-11"]["ATL11-B"]="Cemalettin";
8 $baskanlar["SINIF-11"]["ATL11-C"]="Erdal";
9
10 echo $baskanlar["SINIF-11"]["ATL11-B"];
11 // ekrana Cemalettin yazar
12 ?>
```

### 3.7. Fonksiyonlar

Bir programda işlemleri kolaylaştıran ifadeler fonksiyonlardır. Fonksiyonlar, bir kere tanımlanır ve defalarca, ihtiyaç olduğu noktada hizmete çağrılır. Fonksiyonlar genellikle bir iş veya hesaplama yapar ve bunun sonucu olarak ortaya yeni bir değer çıkarır. Bu değere "fonksiyondan dönen değer" denir.



İnternet üzerinde çalışan hesap makinesi yapmak istiyoruz. Topla, çıkar, çarp ve böl isimli 4 tane fonksiyonumuz olduğunu varsayalım. Kullanıcı tek tıklama ile istediği sayıların aritmetiksel işlemlerini yapabilmektedir.

Her işlemin kendine ait fonksiyonu bulunmakta ve kullanıcı hangi işlemi seçtiyse o fonksiyon çağrılmaktadır. Fonksiyon bir kere çağrıldıktan sonra geriye değer döndürür ve bir sonraki çağrılmasına kadar program içinde görevi biter.

PHP'nin kullanıma hazır birçok fonksiyonu bulunmaktadır. echo() , print() veya printf bunların başlıcalarıdır. Bu fonksiyonların aldıkları parametre ile ekrana string yazdırmaktadır. Fonksiyonları yaptıkları işe göre iki gruba ayırabiliriz:

#### ➤ Geriye değer döndürmeyen fonksiyonlar

Örnek üzerinde inceleyelim. Geriye değer döndürmeyen fonksiyonları bu örnekle hemen anlayacaksınız. Fonksiyon çağrıldığında program içine herhangi bir değer döndürmeyecek, ekrana bir mesaj yazdırdıktan sonra işlemi tamamlayacaktır. Ekrana yazdırma işlemi için **"print"** kullanıyoruz. Fonksiyon adını **"yaz"** olarak belirliyoruz.

```
<?php
function yaz($kelime)
{
    print $kelime;
}
?>
```

Bu fonksiyonu kullanırken istediğimiz yazıyı parantez içinde yazmamız yeterlidir.

Örnek:

```
<?php  
yaz ( "Yaz komutu print komutunun yerini aldı" );  
?>
```



Fonksiyon kullanılırken sayfanın başında bir kez tanımlanması yeterlidir. Daha sonra sayfanın altında tekrar tekrar onu kullanmak mümkündür.

Bu şekilde bir fonksiyon pratikte pek iş görmez ama anlamanız açısından faydalı olacaktır. Fonksiyon çalıştığında sonucu direkt olarak ekrana yazdırmaktadır.

#### ➤ Geriye değer döndüren fonksiyonlar

Bu tip fonksiyonlarda, bulunan sonuç başka bir işlemde kullanılmak üzere çağrıldığı program içine değer gönderir.

```
<?php  
function karebul($sayi)  
{  
    return $sayi * $sayi  
}  
?>
```

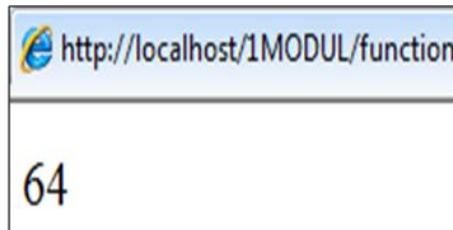
Fonksiyon içinde herhangi bir yazdırma işlemi yapmıyoruz. Evet, bir sonuç bulunuyor ama bu sonuç nereye gidiyor? Kullanımı için aynı sayfada aşağıdaki şekilde bir kod yazıyoruz:

```
<?php  
$sonuc= karebul(5);  
?>
```

Bu satır çalıştığında kare bulma işlemi bulduğu sonucu \$sonuc isimli değişkene aktaracaktır. Daha sonra biz bu \$sonuc değişkenini ister yazdırır ister başka bir işlemde kullanabiliriz.

**Örnek:** Verilen sayının karesini bulan fonksiyonu yazın.

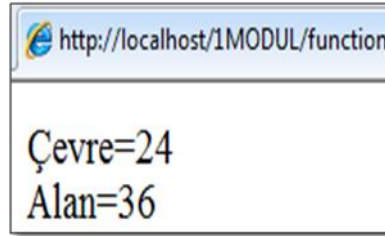
```
1 <?php
2     function hesapla($sayi) //Fonksiyon tanımlandı
3     {
4         return $sayi * $sayi; //Fonksiyonun yapacağı iş
5     }
6     $sonuc=hesapla(8); //8 rakamına fonksiyonu uygula
7     echo $sonuc; //Sonucu ekrana yaz
8 ?>
```



**Örnek:** Bir kenarı verilen karenin çevresini ve alanını hesaplatan fonksiyon yazalım:

```
1 <?php
2     function cevre($kenar) //Çevre fonksiyonunu tanımladık
3     {
4         return $kenar*4; //Çevre fonksiyonunun görevini yazdık
5     }
6     function alan($kenar) //Alan fonksiyonunu tanımladık
7     {
8         return $kenar*$kenar; //Alan fonksiyonunun görevini yazdık
9     }
10    //_____UST KISIM FONKSİYONUN TANIMLANDIĞI KISIM_____//
11
12    $kenar=6; //Bir kenar değerini kenar değişkenine atadık
13    $cevre=cevre($kenar); // Fonksiyon yardımıyla alanı bulduk
14    $alan=alan($kenar); // Fonksiyon yardımıyla çevreyi bulduk
15    echo "Çevre=".$cevre."<br>"; // Çevreyi ekrana yazdırdık
16    echo "Alan=".$alan."<br>"; // Alanı ekrana yazdırdık
17 ?>
```

Tanımladığımız fonksiyonun çalışmasını görmek için fonksiyonun altına bir kenarı 6 verilen karenin çevre ve alanını doğru bir şekilde hesaplattık.



### 3.7.1. Zaman Fonksiyonları

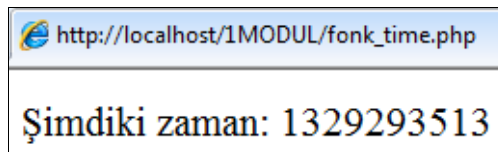
Zamanı gösteren fonksiyonlardır. Kendi içlerinde gruplara ayrılır:

➤ **time()**

Şu anki zaman bilgisini (tarih, saat, dakika ve saniye) unix epoch biçiminde **bir tamsayı** değerine döndürür.

```
1 <?php
2 $simdi=time();
3 echo "Şimdiki zaman: ", $simdi;
4 ?>
```

Bu program ekrana aşağıdaki gibi bir değer yazdırır. Bu değer unix epoch biçiminde bir tamsayı değeridir.



Yukarıda da bahsettiğimiz gibi bu fonksiyon tek başına kullanılmaz. Kullanıldığı takdirde her çalıştırılmasında değişen bir tamsayı değeri ortaya çıkar.

time() fonksiyonuna saniye cinsinden değer ekleyip çıkartarak geçmiş ve gelecek tarihlere ait Unix Epoch biçiminde değer elde edebilir. Bu değeri date(), getdate() gibi fonksiyonlarda kullanabilir. Burada 24\*60\*60 'ın esprisi tahmin ettiğiniz gibi 24 saat, 60 dakika, 60 saniye.



Yarına gitmek için	Düne Gitmek için
<pre>&lt;?php \$simdi=time(); \$yarın=\$simdi+24*60*60; echo "Yarın: ",\$yarın; ?&gt;</pre>	<pre>&lt;?php \$simdi=time(); \$dun=\$simdi-24*60*60; echo "Dün: ",\$dun; ?&gt;</pre>

➤ **mktime (\$saat, \$dakika, \$saniye, \$ay, \$gun, \$yil)**

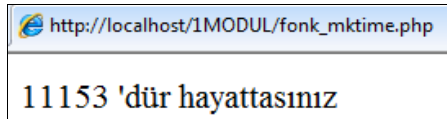
Tam olarak belirtilen zamana gitmeyi sağlar. Başka bir ifadeyle belirtilen zamanı unix epoch biçimine dönüştürür. Eğer zaman belirtilmezse şimdiki zaman bilgisini unix epoch biçiminde verir. Bir başka deyişle zaman belirtilmemişse time() fonksiyonu gibi davranır. Buradaki sıralamaya dikkat ediniz. (saat, dakika, saniye, ay, gün, yıl)

**Örnek:** 20 Mart 1980 doğumlu birisi kaç gündür yaşamaktadır?

```
1 <?php
2 $simdi=time(); //Zamanı $simdi değişkenine ata
3 $dogum=mktime(0,0,0,20,3,1980); //Doğum tarihini $dogum' ata
4 $fark=$simdi-$dogum; //Bu güne kadar geçen süreyi $fark'a ata
5 $gun=$fark/(24*60*60); //Fark değerini güne çevir
6 $gun=floor($gun); // Çıkan gün değerini yuvarla
7 echo $gun." 'dür hayattasınız"; //Ekрана yaz
8 ?>
```

3.satırda kullandığımız mktime fonksiyonunda 20-03-1980 tarihini unix epoch biçiminde bize verir.Biz bu değeri time() fonksiyonundan çıkardığımız zaman aradaki gün farkının unix epoch biçiminde karşılığını buluruz.

Program çalıştırıldığında aşağıdaki şekilde bir çıktı elde ederiz:



➤ **getdate f(\$zaman)**

Unix epoch biçiminde verilen zamanın bilgisini bir dizi şeklinde gönderir. Zaman verilmezse şimdiki zaman bilgisini bir dizi olarak gönderir. Dönen dizinin indisleri ve değerleri şöyledir:

- seconds => saniye
- minutes => dakika
- hours => saat
- mday => ayın kaçınıcı günü olduğu bilgisi (1,2,3, ... 31)
- wday => haftanın kaçınıcı günü olduğu bilgisi (Pazar: 0, Pazartesi: 1,Cumartesi: 6 )
- mon => ay (1,2,...12)
- year => yıl (2001,2002,...)
- yday => yılın kaçınıcı günü olduğu bilgisi (1,2,3, ...365)
- weekday => haftanın hangi günü olduğu bilgisi (Sunday, Monday, ...,Saturday)
- month => ay (January,February,...,December)

**Örnek:** Şimdiki zamanın tüm bilgilerini getdate ( ) ile ekrana yazdıralım:

```
1 <?php
2 $simdi=getdate(); // Dönen bir dizidir
3 print_r($simdi); //Dizideki bütün değerleri yaz
4 echo "<hr>";
5 echo "Saniye: ",$simdi["seconds"],"<br>";
6 echo "Dakika: ",$simdi["minutes"],"<br>";
7 echo "Saat: ",$simdi["hours"],"<br>";
8 echo "Gün: ",$simdi["mday"],"<br>";
9 echo "Haftanın kaçınıcı günü: ",$simdi["wday"],"<br>";
10 echo "Ay: ",$simdi["mon"],"<br>";
11 echo "Yıl: ",$simdi["year"],"<br>";
12 echo "Yılın kaçınıcı günü: ",$simdi["yday"],"<br>";
13 echo "Haftanın hangi günü: ",$simdi["weekday"],"<br>";
14 echo "Yılın hangi ayı: ",$simdi["month"],"<br>";
15 ?>
```

Programın çalışmasından sonra aşağıdaki görüntüyü elde ederiz. Dizi kavramının anlaşılması bakımından önce dizideki bütün değerler **print\_r** deyimiyle yazılmıştır. Daha sonra ise teker teker indis ve değerleri alt alta yazılmıştır.

```
http://localhost/1MOD...
Array ( [seconds] => 21 [minutes] => 56 [hours] => 0 [mday] => 22
[yday] => 3 [mon] => 2 [year] => 2012 [yday] => 52 [weekday] =>
Wednesday [month] => February [0] => 1329864981 )

Saniye: 21
Dakika: 56
Saat: 0
Gün: 22
Haftanın kaçınıcı günü: 3
Ay: 2
Yıl: 2012
Yılın kaçınıcı günü: 52
Haftanın hangi günü: Wednesday
Yılın hangi ayı: February
```

➤ **date (\$bicim, \$zaman)**

Unix epoch biçimindeki zaman bilgisini ekranda istenilen biçimde göstermek için kullanılır. Zaman belirtilmezse şimdiki zaman ile çalışır. Kullanılan parametreler şunlardır:

- a - 12 saat esasına dayanan Anglo-Sakson sistemlerinde öğleden önce ("am") veya öğleden sonra ("pm") işaretini sağlar.
- A - Aynı işaretlerinin büyük harfle yazılmasını sağlar.
- d - İki haneli gün sayısı, tek haneli günlerin önüne sıfır konur: "01"- "31"
- D - Üç haneli gün adı kısaltması: "Sun", "Mon", ..
- F - Uzun ay adı: "January", "February", ..
- h - 12 saatlik sistemde saat: "01"- "12"
- H - 24 saatlik sistemde saat: "00"- "23"
- g - 12 saatlik sistemde tek haneli saatlerin önüne sıfır konmadan saat: "1"- "12"
- G - 24 saatlik sistemde tek haneli saatlerin önüne sıfır konmadan saat: "0"- "23"
- i - Dakika: "00"- "59"
- j - Tek haneli sayıların önüne sıfır konmadan gün sayısı: "1"- "31"
- l - (Küçük L harfi) Uzun gün adı: "Sunday", "Monday", ..
- L - Artık yıl olup olmadığına ilişkin Boolean (doğru/yanlış) değişken. Artık yıl ise 1, değilse 0 döndürür.
- m - Tek haneli sayıların önüne sıfır konarak ay sayısı: "01"- "12"

- n - Tek haneli sayıların önüne sıfır konmadan ay sayısı: "1"-"12"
- M - Kısaltılmış ay adı: "Jan","Feb", ..
- s - Saniye: "00"-"59"
- S - İngilizce "'nci" anlamında 2 karakter ek: "th","nd"
- t - Belirtilen ayın toplam gün sayısı: "28"-"31"
- w - Haftanın kaçınıcı günü olduğu bilgisi: "0" Pazar, "6" Cumartesi
- Y - Dört haneli yıl: "2000"
- y - İki haneli yıl "02"
- z - Yılın kaçınıcı günü olduğu bilgisi: "0"-"365"

**Örnek:** Şimdiki tarihi ekranda 05/06/2010 şeklinde gösterelim.

```
<?php
echo date("d/m/y");
?>
```

**Örnek:** Yarını 26 April 2010 Monday biçiminde ekrana yazdıralım.

```
<?php
echo date("d F Y l",time()+24*60*60);
?>
```

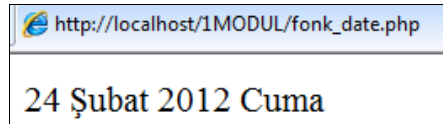
**Örnek:** 18.06.2010 tarihini ekrana 18 June 2010, Friday biçiminde yazdıralım.

```
<?php
echo date("d F Y, l",mktime(0,0,0,6,18,2010));
?>
```

**Örnek:** Tanımlı olan tarihi ekrana Türkçe olarak 1 Mart 1996 Cuma biçiminde yazdıralım.(Tarihimiz 2012-02-24 olsun.)

```
1 <?php
2 $tarih="2012-02-24";
3 $tarih=explode("-", $tarih); //Dizi olarak parçalara ayırır
4 $zaman=mktime(0,0,0,$tarih[1],$tarih[2],$tarih[0]);
5 $tarih=date("d F Y l", $zaman);
6 $ingilizce_gun=array("Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
7 "Saturday", "Sunday"); //dizi tanımlaması yaptık
8 $turkce_gun=array("Pazartesi", "Salı", "Çarşamba", "Perşembe", "Cuma", "Cumartesi",
9 "Pazar"); //dizi tanımlaması yaptık
10 $ingilizce_ay=array("January", "February", "March", "April", "May", "June", "July",
11 "August", "September", "October", "November", "December"); //dizi tanımlaması
12 $turkce_ay=array("Ocak", "Şubat", "Mart", "Nisan", "Mayıs", "Haziran", "Temmuz",
13 "Ağustos", "Eylül", "Ekim", "Kasım", "Aralık"); //dizi tanımlaması yaptık
14 $tarih=str_replace($ingilizce_ay, $turkce_ay, $tarih); //dizi elmanları
15 $tarih=str_replace($ingilizce_gun, $turkce_gun, $tarih); // yer değiştirdi...
16 echo $tarih;
17 ?>
```

Yukarıdaki işlem gerçekleştiğinde ekrana aşağıdaki şekilde bir tarih gelecektir.



### 3.7.2. Mail ( ) Fonksiyonu

Php'de mail göndermemize yarayan bir fonksiyondur. Mail fonksiyonunu Kullanabilmek için sisteminizde SMTP server kurulu olması gerekmektedir.

**SMTP: Elektronik** posta gönderme protokolü (Simple Mail Transfer Protocol), bir e-posta göndermek için sunucu ile istemci arasındaki iletişim şeklini belirleyen protokoldür. E-posta almak için ise **POP3** ya da **IMAP** protokolü kullanılır.

Localhost ortamında mail gönderme işlemi yapamayız. Web sitesi yaparken alacağınız hostta bu özelliklerin olması gerekir ki bu fonksiyon çalışsın.

Fonksiyonun kullanım şekli şu şekildedir

**<?php**

Mail (alıcı, konu, mesaj);

**?>**

**Örnek:** mtegm@meb.gov.tr adresine konusu “Bayram Mesajı”, içeriği “Bayramınız kutlu olsun.” şeklinde bir mesaj gönderelim.

```
1 <?php
2 $kime="mtegm@meb.gov.tr";
3 $konu="Bayram Mesajı";
4 $mesaj="Bayramınız Mübarek Olsun";
5 mail($kime,$konu,$mesaj)
6 or die("Mesaj Gönderilemedi");
7 echo "Mesajınız Gönderildi";
8 ?>
```

Yukarıdaki örnekte daha önce kullanmadığımız die ( ) komutunun Türkçe karşılığı “ölmek” olup “ işlem gerçekleşmezse” manasına gelir. Programdaki görevi mail gönderilememişse parantez içindeki ifadeyi (“Mesaj Gönderilemedi.”) ekrana yazmaktır.

### 3.8. Dosya ve Dizin İşlemleri

Web sayfamıza dosya ve dizin oluşturmak için değişik komutlar kullanılır. Örneğin bir resim albümü scripti yazılacaksa; albüm isimleri dizin, içerisindeki resimler ise dosya işlemleri konusuna girer.

#### 3.8.1. file\_exists ( )

Bir dosya veya klasörün var olup olmadığını kontrol eder. Dosya varsa true, yoksa false değerini döndürür.

```
1 <?php
2 $dizin="resimler"; //Kontrol edilecek dizin
3 if(file_exists($dizin)) //Kontrol yapılıyor
4 { //Dizin mevcutsa
5 echo "Dizin var.";
6 }
7 else //Dizin mevcut değilse
8 {
9 echo "Dizin yok";
10 }
11 ?>
```

Bu örneği uygularken local hostunuzun içerisinde “resimler” adında bir tane klasör oluşturun. Bu programla oluşturduğunuz klasörün aynı dizinde olmasına dikkat edin. İlk çalıştırdığınızda “Dizin Var.” mesajı ekrana gelecektir. Oluşturduğunuz resimler klasörünü silip tekrar çalıştırdığınızda ise “Dizin yok.” mesajını alacaksınız.

#### 3.8.2. is\_file ( ) / is\_dir ( )

Bazen dosyalar uzantısız şekilde klasör gibi adlandırılabilir. Bu gibi durumlarda sorguladığımız nesnenin klasör mü yoksa dosya mı olduğunu **is\_file ( )** ve **is\_dir ( )** fonksiyonları ile anlarız.

- **is\_file ( )**: Dosya ise true değerini döndürür.
- **is\_dir ( )**: Klasör ise true değerini döndürür.

```
1 <?php
2 $dosya_veya_klasor_adi = "test";
3 if(is_file($dosya_veya_klasor_adi)) //Dosya mı?
4 {
5     echo "Bu bir dosyadır";
6 }
7 if(is_dir($dosya_veya_klasor_adi)) //Klasör mü?
8 {
9     echo "Bu bir klasördür";
10 }
11 ?>
```

### 3.8.3. mkdir ( )

Klasör oluşturmak için **mkdir ( )** fonksiyonu kullanılır. Oluşturulan klasörün izin yetkisi parametre kısmında belirtilir. Varsayılan değer **0700**'dür.

```
1 <?php
2 $klasor_adi = "test";
3 if(file_exists($klasor_adi))//Böyle bir dosya var mı
4 {
5     echo "Klasör zaten var!";
6     exit();//İşlemi durdur
7 }
8 $olustur=mkdir($klasor_adi,0700);
9 if($olustur)
10 {
11     echo "Klasör oluşturuldu.";
12 }
13 else
14 {
15     echo "Klasör oluşturulamadı!";
16 }
17 ?>
```

Yukarıdaki örnekte öncelikle klasörün var olup olmadığı **file\_exists ( )** ile kontrol edilir. Aynı isimde klasör varsa **exit ( )** fonksiyonu ile işlem durdurulur. Klasör daha önce oluşturulmuşsa program ekrana “Klasör zaten var!” mesajını yazdırır. Eğer klasör yoksa **mkdir()** fonksiyonu ile oluşturmasını sağlar ve klasörün oluşturulma durumunu ekrana yazdırır. Klasör silmek için ise **rmdir(“silinecek\_klasor”)** fonksiyonu kullanılır.

```
1 <?php
2 $klasor_adi = "test";
3 if(!file_exists($klasor_adi))
4 {
5     echo "Silinecek klasör yok!";
6     exit();//İşlemi durdur
7 }
8 $klasor_sil = rmdir($klasor_adi);
9 if($klasor_sil)
10 {
11     echo "Klasör silindi.";
12 }
13 else
14 {
15     echo "Klasör silinemedi!";
16 }
17 ?>
```



### 3.8.4. Touch ( )

Yeni bir dosya oluşturmak için Php'nin **touch ( )** fonksiyonu kullanılır.

```
1 <?php
2 $dosya_adi="dosya.txt";
3 if (file_exists($dosya_adi))//Dosyayı kontrol et
4 {
5     echo "Dosya zaten var!";
6 }
7 else
8 {
9     touch($dosya_adi); // Dosya oluştur
10    echo "Dosya oluşturuldu.";
11 }
12 ?>
```

Her dosyanın bir uzantısı vardır. Klasörlerin bir uzantısı yoktur. Touch ( ) komutu ile mkdir( ) komutları birbirinden tamamen farklıdır. Birisi dosya diğeri klasör oluşturmaktadır.Bu nedenle touch ( ) komutu ile oluşturduğumuz dosyayı tanımlarken bir uzantı eklemek gerekir.

### 3.8.5. Fwrite ( )

Dosyaya veri yazdırmak için kullanılan fonksiyondur. Bir dosyaya veri yazdırmadan önce bilinmesi gereken hususlar vardır. Bunlar dosyanın hazır hale getirilmesi, yani dosyanın açılması ve işlemiz bittiği zaman dosyanın kapatılması işlemleridir.

- Dosyayı açmak için **fopen()**,
- Veri yazdırma işlemi için **fwrite()**,
- Dosya ile işlemiz bittiğinde kapatma işlemi için **fclose()** fonksiyonu kullanılır.

```
1 <?php
2 $dosya="dosya.txt";
3 $dosya=fopen($dosya,"a");//Dosya açıldı
4 $veri="Bu bilgiyi yaz";
5 fwrite($dosya, $veri); //Dosyaya bilgi yazıldı
6 echo "Veri yazdırıldı.";
7 fclose($dosya);
8 ?>
```

Yukarıdaki örnekte 3.satırda dosya.txt dosyasını içine bilgi girişi yapmak için açıyoruz. Burada şuna dikkat edilmelidir:

Bizim daha önceden dosya.txt dosyamızı oluşturmuş olmamız gerekir.Var olmayan bir dosya açılmaz.

Örneğimizin 5.satırında fwrite() deyiminin ilk parametresinde dosya ad, ikinci parametresinde ise dosya veri yazılmıştır

### 3.8.6. Fget ( )


Dosyadan veri okumak için kullanılan fonksiyondur. Okuma işlemine başlamadan önce **fopen( )** fonksiyonu ile dosyamızı açarız. **fopen( )** fonksiyonunda ilk parametrede dosya yolu, ikinci parametrede dosyaya erişim izni belirtilir. Son olarak **while** döngüsü yardımı ile txt dosyasındaki veriler ekrana yazdırılır.

```
1 <?php
2 $dosya=fopen("dosya.txt","r");//Dosya açılıyor
3 if(!$dosya) //Dosya kontrolü yapılıyor
4 {
5     echo"Böyle bir dosya yok.";
6     exit(); //Dosya yoksa çıkış
7 }
8 while(!feof($dosya)) //Dosyanın sonuna kadar oku
9 {
10     $satir=fgets($dosya);//Veriyi bitinceye kadar aktar
11     echo("$satir");
12 }
13 fclose($dosya);
14 ?>
```

Dosya işlemleri, dosya silme ve dosya listeleme olarak devam etmektedir. Fakat modül ders saati dikkate alınarak bu kadarla iktifa edip detaylı bilgi isteyen arkadaşlarımızı değişik kaynaklara yönlendirebiliriz.

## UYGULAMA FAALİYETİ

Aşağıda verilen işlem basamaklarını takip ederek konuyu daha da pekiştirelim.

İşlem Basamakları	Öneriler
➤ Birer adet string ve integer değişken tanımlayınız.	➤ Değişken atamasına bakınız.
➤ “pi” adında ve değeri 3.14 olan bir sabit tanımlayınız.	➤ Sabitler ve kullanımı
➤ İki farklı sayı belirleyip bunları değişkenlere atayınız. Bu iki değişken arasında 4 aritmetik işlemi (toplama, çıkarma, çarpma, bölme) yapınız ve sonuçları ekrana yazdırınız.	➤ Aritmetik operatörleri kullanınız. Echo ➤ komutunu kullanınız.
➤ “Php” ve “Öğreniyorum.” sözcüklerini iki farklı değişkene atayınız.	➤ Atama operatörünü kullanınız.
➤ Bu iki sözcüğü birleştirip tek bir değişkene atayınız ve ekrana yazdırınız.	➤ String birleştirme operatörünü kullanınız.
➤ 10 sayısını bir değişkene atayıp değerini birer birer azaltıp her azaltmada değerini ekrana alt alta yazdırınız.	➤ Azaltma operatörünü kullanınız. Alt satıra geçmek için   tagını kullanınız.
➤ İki sayıdan küçüğünü bulan programı yazınız.	➤ If deyimi yardımıyla
➤ Üç kişinin yaşlarını kıyaslayıp en büyüğünün yaşını ekrana yazdıran programı yazınız.	➤ İç içe if deyimi ve mantıksal operatörler
➤ 1’ den 10 ‘a kadar olan sayıları aralarına + işareti koyarak yazan ve toplamlarını = işareti ile yazan php kodunu yazınız.	➤ For döngüsünü kullanınız. ➤ Ekran çıktısı ➤ 1+2+3+4+5+6+7+8+9+10=55
➤ Ekrana ilk çizgi boyutu 100 pixel ikinci çizgi boyutu 200 pixel olacak şekilde yüzer yüzer artan alt alta 10 tane çizgi çizin. Son çizgi 1000 pix genişlikte olacak.	➤  , <width> html etiketlerini ve FOR döngüsünü kullanınız. Aşağıdaki şekildeki gibi... (width=\$i) 
➤ İki sayıyı toplayan fonksiyonu yazınız ve program içinde çağırıp kullanınız.	➤ Geriye değer döndüren fonksiyonları kullanabilirsiniz.
➤ Sınıfınız için bir dizi tanımlayıp (EML11A) 5 arkadaşınızın ismini bu dizi içine atayıp sonra alt alta ekrana yazdırınız.	➤ Dizi atama uygulamasına bakınız.
➤ Konusu, içeriği ve kime gideceği belli olan bir mail gönderme programı yazınız	➤ Mail fonk.

➤ Resimler adında bir dizin oluşturunuz.	➤ mkdir() –Klasör Oluşturma
➤ Bu dizinin içine araba.jpg ve agac.jpg adında iki tane dosya oluşturunuz.	➤ touch()-Dosya Oluşturma
➤ Ayrı bir defter.txt dosyasını oluşturunuz.	➤ touch()-Dosya Oluşturma
➤ Defter.txt dosyasının içerisine “Bilgi ekledim.” yazısını yazınız.	➤ Fwrite() –Dosyaya Veri yazdırma Komutları:

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız becerileri Evet, kazanamadığınız becerileri Hayır kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri		Evet	Hayır
1	İki farklı değişkene integer ve string değer ataması yaptınız mı?		
2	İki ayrı değişkene birer sayı değeri atayıp toplama, çarpma işlemi yaptınız mı?		
3	İki string ifadeyi tek bir değişkende birleştirdiniz mi?		
4	Üç farklı sayıyı if deyimi yardımıyla kıyaslayabildiniz mi?		
5	And, or mantıksal operatörleri if deyimi içinde kullandınız mı?		
6	İç içe if deyimi kullanarak çarpım tablosu hazırlayabildiniz mi?		
7	İstenilen satır ve sütun sayısı kadar for deyimi yardımıyla tablo oluşturabildiniz mi?		
8	Switch deyimi yardımıyla iki sayıya, yapılan seçime göre toplama, çıkarma, çarpma ve bölme işlemi yapan program yaptınız mı?		
9	While deyimi yardımıyla alfabenin harflerini yan yana yazdırabildiniz mi?		
10	Dizileri kullanarak haftanın günlerini bir dizi içine atıp daha sonra alt alta yazdırabildiniz mi?		
11	Bir kenarı verilen karenin çevresini ve alanını hesaplatan fonksiyon tanımlayabildiniz mi?		
12	Şimdiki zamanı ekrana 20-12-2012 şeklinde yazdırabildiniz mi?		
13	Konusu-içeriği ve kime gideceği belli olan mail gönderme programı yazabildiniz mi?		
14	Program yardımıyla, değişkene ismi atanmış bir klasör oluşturdunuz mu?		
15	Klasör içerisine defter.txt adında bir dosya oluşturabildiniz mi?		
16	Bu dosya içine veri ekleme komutlarını girip “Sevgili anneciğim” cümlesini yazmayı denediniz mi?		

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Aşağıdaki değişken tanımlamalarda verilenlerden hangisi doğrudur?  
A) \$Uzun Kenar=30  
B) Yas=35  
C) \$Alinin\_yasi=45  
D) \$pidegeri=3,14
2. Aşağıdaki atama operatörlerinden hangisi yanlıştır?  
A) \$Alan=\$a2 \* \$a3  
B) \$Ad\_Soyad="Talha UZUN"  
C) \$Yas Farki=\$yas1-\$yas2  
D) \$memleket="ELAZIĞ";
3. And operatörünün karşılığı aşağıdakilerden hangisidir?  
A) &&  
B) ||  
C) ==  
D) !
4. Php de ekrana "Merhaba Dünya" nasıl yazdırılır?  
A) echo " Merhaba Dünya ";  
B) "Merhaba Dünya ";  
C) echo Merhaba Dünya  
D) Response.Write("Merhaba Dünya")
5. if (\$yas>18) and (\$cinsiyet=="erkek") satırının anlamı nedir?  
A) yaşı 18 den büyük veya cinsiyeti erkek ise  
B) yaşı 18 ve cinsiyeti erkek değil ise  
C) yaşı 18 den küçük ve erkek ise  
D) yaşı 18 den büyük ve cinsiyeti erkek ise
6. Aşağıdaki program çalıştırıldığında ekran çıktısı nasıl olur?(15p)  

```
<?php
for($i=0;$i<=5;$i=$i+1)
{echo $i ;}
?>
```

  
A) 5  
B) 0-1-2-3-4-5  
C) 1-3-5  
D) 0-1-2-3-4

7. mail(\$salici, \$ABC,\$mesaj,\$headers) yandaki mail fonksiyonunda \$ABC neyi ifade eder?
- A) Alıcı
  - B) İçerik
  - C) Konu
  - D) Ip Adresi
8. Aşağıdakilerden hangisi bir dosya oluşturma komutudur?
- A) Fwrite
  - B) Touch
  - C) Mkdir
  - D) Rmdir

## 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. Apache web sunucusu hangi portu kullanarak çalışır?  
A) 23  
B) 21  
C) 80  
D) 127
2. Aşağıdakilerden hangisi çalışmakta olan Apache web sunucusunu yeniden başlatır?  
A) Monitor Apache Servers  
B) Restart  
C) Start  
D) Stop
3. Localhost hangi IP adresini kullanır?  
A) 127.0.0.1  
B) 192.168.1.1  
C) 10.0.0.1  
D) 85.168.5.1
4. Aşağıdakilerden hangisi PHP kodunun yazım stillerinden biri değildir?  
A) `<?php ...?>`  
B) `<php>...</php>`  
C) `<script language="php">....</script>`  
D) `<?...?>`
5. Php'de yorum satırı eklemek için aşağıdakilerden hangisi kullanılır?  
A) !  
B) \*  
C) //  
D) /
6. Fiziksel adresi **C:\AppServ\www\deneme\index.php** olan bir dosya tarayıcıda nasıl çalıştırılır?  
A) `http://www.localhost.com/deneme/index.php`  
B) `http://localhost:deneme/index.php`  
C) `http://deneme/index.php`  
D) `http://localhost/deneme/index.php`
7. Aşağıdaki değişken tanımlamalarından hangisi yanlıştır?  
A) `$sayi= 10`  
B) `$sayı= 10`  
C) `$sayi1= 10`  
D) `$SAYI= 10`



8. Aşağıdaki operatörlerden hangisi büyük eşit karşılaştırma operatörüdür?

- A) >=
- B) <=
- C) =>
- D) =<

```
$a=10;
```

```
++$a;
```

```
Printf ++$a;
```

9. Yukarıdaki kod parçası çalıştırıldığında ekran çıktısı ne olur?

- A) 9
- B) 10
- C) 11
- D) 12

10. Aşağıdakilerden hangisi bir dosya veya klasörün var olup olmadığını kontrol eder?

- A) file\_exists
- B) is\_file
- C) is\_dir
- D) mkdir

## 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	B
3	A
4	C
5	D

## ÖĞRENME FAALİYETİ-2'NİN CEVAP ANAHTARI

1	Doğru
2	Yanlış
3	Doğru
4	Yanlış
5	Doğru
6	Yanlış

## ÖĞRENME FAALİYETİ-3'ÜN CEVAP ANAHTARI

1	C
2	C
3	A
4	A
5	D
6	B
7	C
8	C

## MODÜL DEĞERLENDİRMENİN CEVAP ANAHTARI

1	C
2	B
3	A
4	B
5	C
6	D
7	B
8	A
9	D
10	A

## KAYNAKÇA

- <http://www.php.net/>(02-02-2012/11:30)
- <http://www.mysql.com/>(06-02-2012/08:55)
- <http://httpd.apache.org/>(06-02-2012/09:20)
- <http://www.easyphp.org/>(07-02-2012/15:10)
- <http://www.phpdosyasi.com>(02-15-2012/16:30)
- ALTINKAYA Muhittin, Yahya DEMİRCAN, **MySQL ve Programlamaya Giriş**, Ankara, 2005.