



PHP Programlama

1. Bölüm

Tolga GÜYER

©2009-2010



İçindekiler

1.1 Programlama Dillerinin Sınıflandırılması

1.2 PHP'nin Tarihçesi

1.3 Bilgisayarımda Nasıl PHP Kodu Yazabilirim?

1.4 HTML'den Kaçış

1.5 İlk PHP Programım

1.6 Formlarla Çalışmak



Giriş

Bilgisayarlar, genel amaçlı makineler olmaları itibarı ile diğer bütün makinelerden ayrılırlar. Diğer bir deyişle bir bilgisayarın herhangi bir kullanıcı için anlamı, o kullanıcının bilgisayarı kullanım amacına göre şekillenecektir.

Giriş

Bilgisayarlar, onları farklı kılan bu özelliklerini “programlanabilme” yeteneklerine borçludurlar. Çeşitli alanlar için yazılmış paket programlar, bilgisayarların yaygın kullanımları yönünde önemli bir görevi yerine getirirler de, bilgisayarların kullanım alanlarındaki çeşitliliğin günümüzde oldukça artmış olması kurumlara, hatta kişiye özel yazılımların geliştirilmelerini kaçınılmaz kılmıştır.



Giriş

İşte programlama dillerinin devreye girdiği nokta da tam burasıdır.

Giriş

Başlangıçta, sadece kuruldukları bilgisayarlarda çalışan yerel uygulamalar şeklinde geliştirilen yazılımlar, daha sonra istemci/sunucu mimarisine taşınmıştır. Günümüzde ise bu mimarinin yerini, doğrudan internet tarayıcıları tarafından çalıştırılan internet uygulamaları almıştır.


Giriş

Bu uygulamaların yazılmaları için bir çok programlama dili ve ortam geliştirilmiştir. Bu dillerden birisi olan PHP, öğrenme kolaylığı ve açık kaynak koda sahip olma gibi özellikleri ile alanda kısa sürede kendisine önemli bir yer edinmiştir.




Giriş

Bu ders kapsamında, içinde bulunduğumuz paylaşım ortamında günden güne genişleyen PHP programlama dilinin temellerinin öğretilmesi ve ileri düzey PHP tekniklerine bir giriş yapılması amaçlanmıştır.




Programlama Dillerinin Sınıflandırılması

Programlama dillerinin bir çok açıdan sınıflandırması yapılabilir. Bu kesimde yer alan sınıflandırma, dilleri genel amaçlı ve alana özel dil olarak iki ana sınıf altında kategorize etmektedir. Bu başlıklar altında ise sırasıyla veri türü sistemi ve “paradigma” olarak da adlandırılan dilin modeli kategorilerine göre sınıflandırmalar yapılmıştır.




Programlama Dillerinin Sınıflandırılması

Bu sınıflandırmada amacımız, diller arasında bir “iyi-kötü” ayrımı yapmak değil, bir karşılaştırma olanağı sağlamak ve PHP dilinin diğer programlama dilleri arasındaki yerini netleştirmektedir.




Programlama Dillerinin Sınıflandırılması

Zira her dil, geliştirilecek uygulamanın özelliklerine bağlı olarak farklı nitelikleri ile tercih edilebilir ya da edilmeyebilir. Tabi burada, uygulamayı geliştirecek programcının mesleki tecrübesi de uygun programlama dilinin seçiminde önemli bir rol oynayacaktır. Zira bir programcı için her zaman en iyi programlama dili, kendisinin en iyi bildiği programlama dili olacaktır.



Programlama Dillerinin Sınıflandırılması

Programlama dilleri amaçlarına göre, “genel amaçlı” ve “alana özel dil” olarak iki kategoride gruplandırılmıştır. Genel amaçlı diller, ek fonksiyon kütüphanelerinin yardımıyla genişleyebilen yapılarının da sayesinde pek çok alanda uygulama geliştirmeye uygun araçlardır. PHP dili de bu kategori altında sınıflandırılmıştır.




Programlama Dillerinin Sınıflandırılması

Alana özel diller ise, belirli amaçları gerçekleştirmek üzere tasarlanmıştır. Örneğin SQL, veritabanlarında sorgulama işlemlerini gerçekleştirmek üzere geliştirilmiş yapısal bir dildir.

Programlama Dillerinin Sınıflandırılması

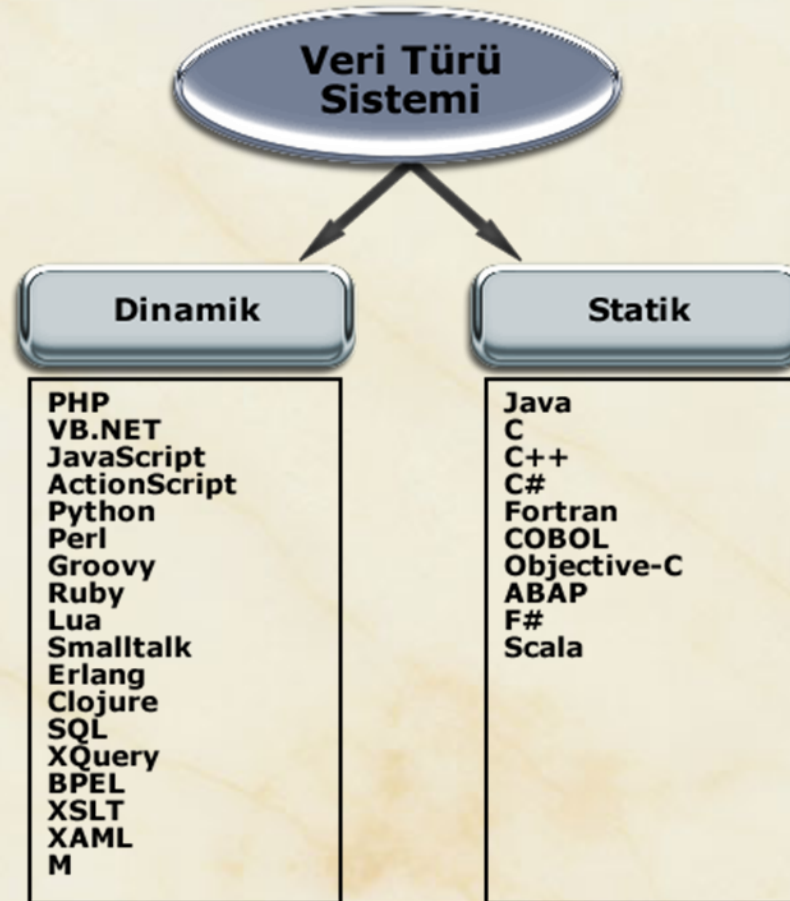





Programlama Dillerinin Sınıflandırılması

Programlama dilleri, veri türlü sistemlerine göre statik ve dinamik olarak iki farklı kategoride sınıflandırılabilir.


Programlama Dillerinin Sınıflandırılması






Programlama Dillerinin Sınıflandırılması

Statik sistemde değişkenlerin türleri, çalışma anında değil, derleme anında kontrol edilir. Dinamik sistemde ise tersine tür denetimi çalışma anında gerçekleştirilir. Bu durumda türler, değişkenlerle değil doğrudan değerlerle ilişkilendirilir.



Programlama Dillerinin Sınıflandırılması

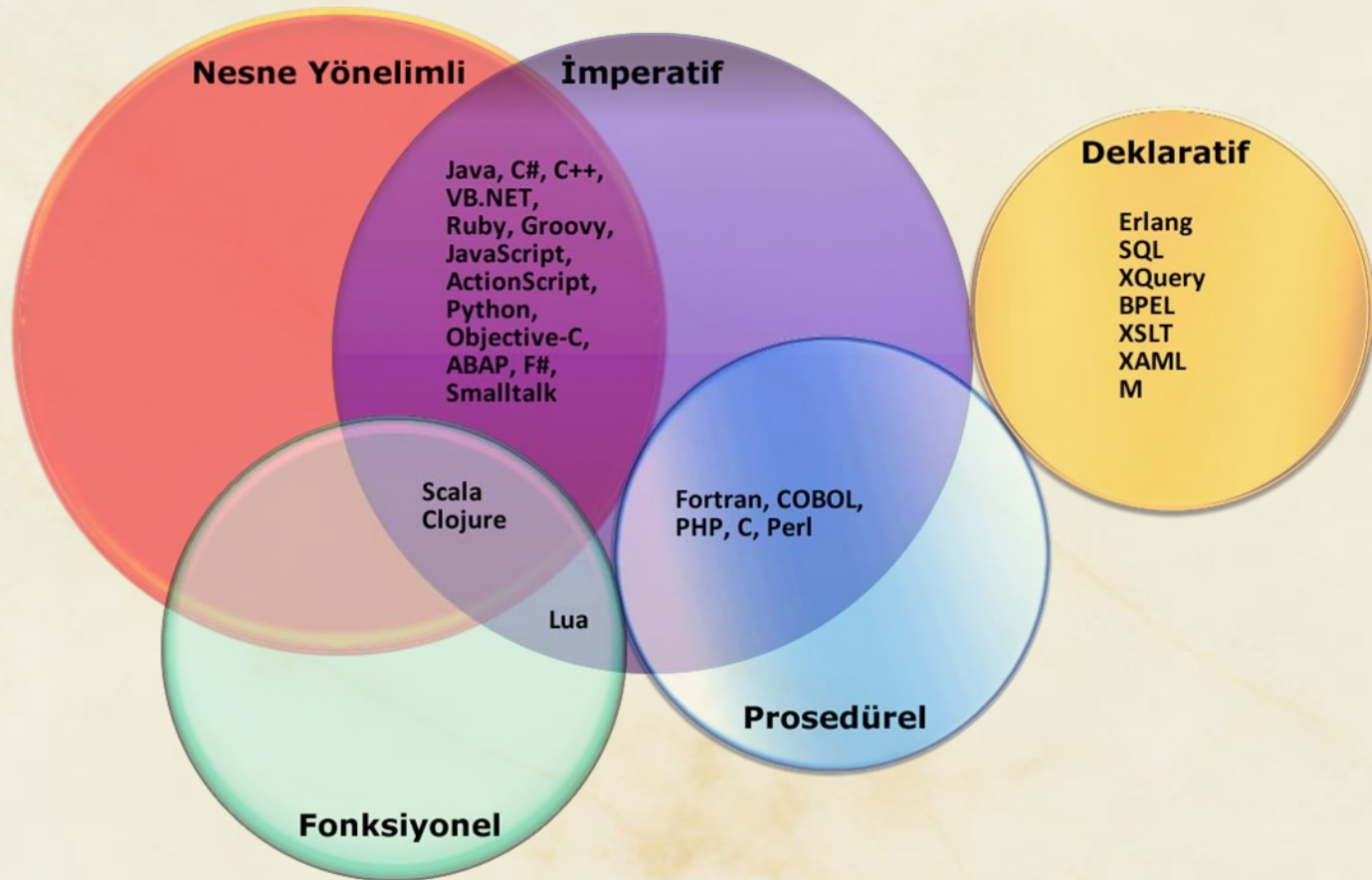
PHP programlama dili, dinamik veri türü sistemine sahiptir. Bu sistem, PHP'ye önemli bir üstünlük sağlamaktadır. Dinamik dillerde çalışma-anında girilen verilere bağlı olarak yeni veri türlerinin türetilmesi de mümkün olmaktadır.




Programlama Dillerinin Sınıflandırılması

Paradigma olarak da adlandırılan, dilin modeli olarak ifade edebileceğimiz sınıflandırma yöntemine göre beş farklı kategori ortaya çıkmaktadır. Ancak bu kategorilere göre Programlama dilleri ayrık olarak sınıflandırılmazlar, bir dil birden çok kategoriye dahil olabilir. Resimde dillerin bu kategorilere göre sınıflandırılması verilmiştir.


Programlama Dillerinin Sınıflandırılması






Programlama Dillerinin Sınıflandırılması

Zorunlu programlama olarak çevirebileceğimiz imperatif paradigmaya göre, bilgisayara bir işlemin gerçekleştirilmesi için gereken bütün komut dizeleri verilmektedir. İmperatif programlama, deklaratif, yani bildirimsel programlamanın tersi durum olarak betimlenebilir.




Programlama Dillerinin Sınıflandırılması

Deklaratif paradigmaya göre bilgisayara bir işlemin yapılması için gereken emir doğrudan verilir, işlemin nasıl gerçekleştirileceği ile ilgili olarak herhangi bir bilgi verilmez. Doğrudan sorgulama ifadelerinin yer aldığı SQL dili bu kategoriye güzel bir örnektir.




Programlama Dillerinin Sınıflandırılması

Prosedürel programlama, 1960'lı yıllardan sonra popüler olmuş bir akımdır. İmperatif programlamanın bir alt türü olarak düşünülebilir. Bu paradigmaya göre gerçekleştirilecek işlemler bilgisayara alt program olarak adlandırabileceğimiz prosedürler şeklinde bildirilir. Daha önceki modellerde yer alan sapma deyimleri, bu modellerle birlikte terk edilmiştir.



Programlama Dillerinin Sınıflandırılması

Fonksiyonel programlama dilleri ise tümüyle fonksiyonlar üzerine kurulmuş bir dil modeli olarak karşımıza çıkar. Hemen hemen bütün fonksiyonel programlama dilleri, temelleri 1930'lu yıllarda atılmış olan Lambda Analizi (Lambda Calculus) üzerine inşa edilmiştir.




Programlama Dillerinin Sınıflandırılması

Nesne yönelimli yaklaşımda, sınıflar altında gruplanan nesneler ve onların çeşitli etkileşimleri söz konusudur. Son yıllarda popülaritesi giderek artan bu yaklaşım, birçok programlama dili tarafından desteklenmeye başlanmıştır.

Programlama Dillerinin Sınıflandırılması



PHP Programlama dilinin bu sınıflandırmadaki yerine bakacak olursak, imperatif, yani zorunlu programlama modeli altında yer alan prosedürel paradigmaya dahil olduğunu görürüz.



Programlama Dillerinin Sınıflandırılması

Her ne kadar geliştirilmeye başlandığı yıllarda yapılandırılmış bir internet programlama dili olarak tasarlansa da PHP, en son sürümleri ile artık nesne yönelimli programlamaya büyük ölçüde uygun hale getirilmiştir. Dolayısıyla bu sınıflandırma içerisinde yakın bir gelecekte nesne yönelimli programlama dilleri kümesine dahil olacağını öngörebiliriz.

PHP'nin Tarihçesi



PHP, ilk olarak 1994 yılında Danimarka asıllı bir Kanadalı bilgisayar bilimcisi olan Rasmus Lerdorf tarafından tasarlanmıştır.

Lerdorf'un böyle bir kodlama sistemine ihtiyaç duyduğu konu ise oldukça ilginçtir: internete koyduğu kişisel özgeçmişine kimlerin göz attığını takip etmek.

PHP'nin Tarihçesi

Başka insanların da kullanmaya başlaması ile biraz daha geliştirilen dil, ilk olarak Personel Home Page Tools, yani kişisel ana sayfa araçları adıyla 1995 yılından itibaren yaygınlaşmaya başlamıştır. Başlangıçta basit bir gramer çözümleme (parsing) motoruna sahip olan dil, kişisel sitelere sayaç, ziyaretçi defteri gibi basit araçların yerleştirilmesi amacıyla kullanılmakta idi.

PHP'nin Tarihçesi

PHP gramer çözümleyicisi 1995 yılında PHP/FI Sürüm 2 adıyla yeniden yazılmıştır. FI, yine Lendorf tarafından başka bir proje kapsamında geliştirilen ve HTML formlarından veri elde etmek amacıyla kullanılan bir form yorumlayıcısı (Form Interpreter) yazılımıdır. Bu sistem de ikinci sürümü ile PHP ile tümleşik hale getirilmiştir.

PHP'nin Tarihçesi

1997 yılından itibaren PHP, Lendorf'un kişisel projesi olmaktan çıkmış, profesyonel bir ekibin ortak çalışması ile geliştirilmeye başlanmıştır. 1999 yılında bu ekibin çalışmalarının ilk ürünü olarak PHP3 sürümü ortaya çıkmış, bu sürümü ile PHP yaklaşık 1.000.000 internet sitesinde kullanılır hale gelmiştir.



PHP'nin Tarihçesi

Günümüzde IBM ve Oracle gibi bilişim sektörünün devlerinin de desteğini alan PHP, özellikle Zend firmasının çabaları ve GNU Genel Kamu Lisansına sahip olmasının sayesinde, 2006 yılının Kasım ayı itibarı ile tüm dünyada 25 milyondan fazla internet sitesinde kullanılmaya başlanmıştır.

Bilgisayarımda Nasıl PHP Kodu Yazabilirim?

Bilgisayarınızda PHP programlama dilini çalışır duruma getirmek için öncelikle yapmanız gereken, Apache sunucusunu kurmaktır. Apache sunucusu, PHP dilinde yazılmış kodların, istemciler tarafındaki internet tarayıcı yazılımları tarafından doğru bir şekilde görüntülenebilmeleri için gereken yorumlama ve gramer çözümleme işlemlerini gerçekleştirmektedir.



Bilgisayarımda Nasıl PHP Kodu Yazabilirim?

Apache sunucusunun çalışır duruma geldiği bir bilgisayar, hem yerel PHP uygulamalarını çalıştırabilir, hem de doğrudan internet üzerinden gelecek sayfa görüntüleme isteklerini karşılayacak duruma gelmiştir. Ancak aynı bilgisayarda başka bir web sunucusunun çalışır durumda olmadığından emin olunması gerekmektedir.

Bilgisayarımda Nasıl PHP Kodu Yazabilirim?

Peki kodları nerede yazacağım?

Aslında aynı HTML'de olduğu gibi PHP için de standart bir not defteri uygulaması, kod yazıp çalıştırmak için yeterlidir. Ancak sağladıkları kod tamamlama, kod renklendirme ya da çevrimiçi yardım gibi kolaylıklar nedeniyle programcılar tarafından kod yazma editörlerinin kullanılması daha çok tercih edilmektedir.



Bilgisayarımda Nasıl PHP Kodu Yazabilirim?

Bunun dışında gereksinim duyacağınız diğer bir yazılım, veritabanı uygulamaları için kullanacağımız MySQL veritabanı sistemidir.



Bilgisayarımda Nasıl PHP Kodu Yazabilirim?

Bu yazılımları standart ayarları ile bütün olarak kuran ve çalışır duruma getiren tümleşik kurulum sistemleri bulunmaktadır. Bunlardan ikisi, EasyPHP ve XAMPP dir. Her iki yazılım da internet üzerinden tam sürüm olarak ücretsiz olarak yayınlanmaktadır.



Bilgisayarımda Nasıl PHP Kodu Yazabilirim?

Tek bir kurulum dosyasını (setup) çalıştırıp kurulumu gerçekleştirmek suretiyle bilgisayarınızda PHP ve MySQL yazılımlarını çalışır duruma getirebilirsiniz. Biz, bu ders kapsamında XAMPP sistemini kullanacağız.

Bilgisayarımda Nasıl PHP Kodu Yazabilirim?



Apache Friends (Apache Dostları) adlı organizasyona ait web sitesinin XAMPP bölümünden indirebileceğiniz sistemi, kurulum dosyasını çalıştırarak bilgisayarınıza geçerli ayarlar ile kurabilirsiniz.



Bilgisayarımda Nasıl PHP Kodu Yazabilirim?

Yazılımların son sürümlerinin yanı sıra konuyla ilgili bir çok bilgi, belge ve bağlantıya da ulaşabileceğiniz sitenin internet adresi,

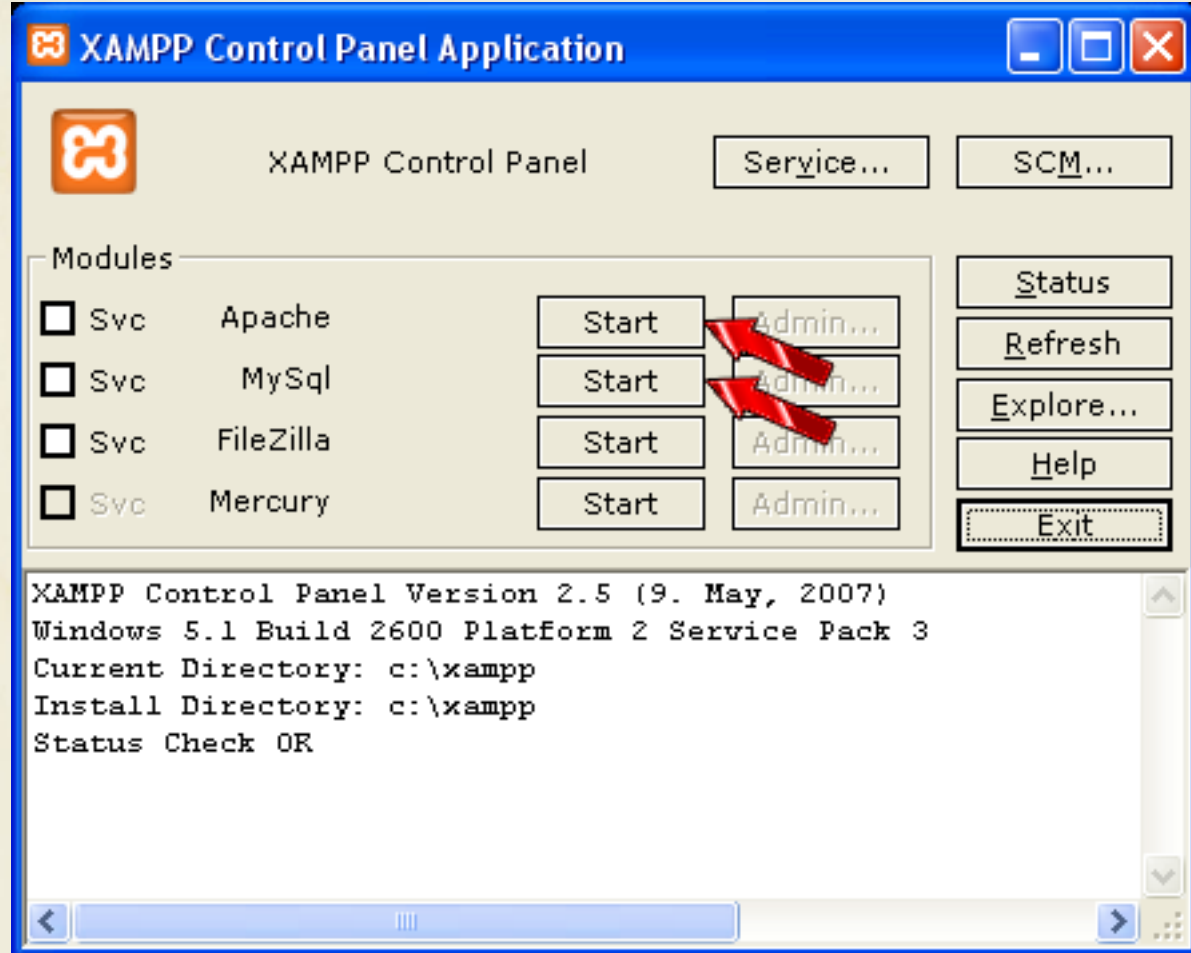
<http://www.apachefriends.org>

biçimindedir.

Bilgisayarımda Nasıl PHP Kodu Yazabilirim?

Kurulum gerçekleştikten sonra çalıştıracacağınız XAMPP Control Panel uygulaması, karşınıza resimdeki gibi bir pencere açacaktır.

Bilgisayarımda Nasıl PHP Kodu Yazabilirim?





Bilgisayarımda Nasıl PHP Kodu Yazabilirim?

Resimde gösterilen başlatma butonlarına basılması, sırasıyla Apache sunucusunun ve MySQL veritabanı sisteminin başlatılması için yeterli olacaktır. MySQL veritabanı sisteminin yönetimi, üçüncü bölümde detaylı olarak ele alınacaktır.

Bilgisayarımda Nasıl PHP Kodu Yazabilirim?

PHP Kodlarını Çalıştırma

Uygulamaları herhangi bir internet tarayıcısından çalıştırabilmeniz için, öncelikle Apache sunucusunun çalışıyor olması gerekir. Eğer uygulamanız veritabanı uygulaması ise, MySQL sisteminin de XAMPP kontrol panelinden başlatılmış olması gerekir.



Bilgisayarımda Nasıl PHP Kodu Yazabilirim?

Bu işlemler yapıldı ise, çalıştırmak istediğiniz PHP dosyasını/dosyalarını içeren yerel klasörü,

`C:\XAMPP\htdocs\`

klasörüne kopyaladıktan sonra, internet tarayıcısının adres kısmına

`localhost/<yerel klasör adı> /programın_adı.php`
yazmanız yeterli olacaktır.



Bilgisayarımda Nasıl PHP Kodu Yazabilirim?

Burada yerel klasör, geliştirdiğiniz uygulamaya ait bütün PHP dosyalarını içeren, internet adreslerinde geçerli olmayan karakterleri kullanmadan isimlendireceğiniz klasörü belirtmektedir.



HTML'den Kaçış

PHP dosyaları sadece PHP kodlamalarından oluşabileceği gibi, HTML kodları içersinde de uygun bildirim ile PHP kodlamasına geçilebilir. HTML'den kaçış olarak adlandırılan bu teknikte, web sayfasını oluşturan kodlamalar içersinde sadece bu bildirimle başlayan kodlar PHP gramer çözümleyicisi tarafından yorumlanarak çalıştırılırlar.



HTML'den Kaçış

Takip eden slaytta bu duruma bir örnek kodlama verilmiştir. **Kırmızıyla** belirtilen işaretler, HTML kodundan PHP kodlamasına geçilirken kullanılan etiket bildirimi göstermektedir.

HTML'den Kaçış

Anlaşılacağı gibi PHP çözümleyicisi, sayfayı oluşturan kodlardan hangilerini dikkate alıp, hangilerini dikkate almayacağını,

<?php

.....

?>

etiket simgeleri ile anlamaktadır. Dolayısıyla yazacağımız bütün PHP kodlarının bu iki simge arasında olması gerekmektedir.



HTML'den Kaçış

Bu etiketleme sistemi sadece HTML kodlarından kaçış amacıyla kullanılmaz, bütün PHP programları bu etiketler arasında yazılmak zorundadır.

İlk PHP Programım

Aşağıda ilk php uygulamamızı geliştirmek için izleyeceğimiz adımlar yer almaktadır:

1. XAMPP kontrol panelini çalıştır.
2. Apache sunucusunu, panelde yer alan Start butonuna basarak başlat.



İlk PHP Programım

3. Not Defteri uygulamasını açarak, aşağıdaki satırları yaz:

```
<?php
```

```
    echo “Merhaba Dünya!”;
```

```
?>
```

4. Bu dosyayı ilkprogram.php adıyla aşağıdaki klasöre kaydet:

```
C:\XAMPP\htdocs\
```



İlk PHP Programım

5. Bir internet tarayıcı uygulaması başlat.

6. Adres satırına,

`localhost/ilkprogram.php`

yaz ve Enter tuşuna bas.

Eğer ekranda Merhaba Dünya! yazısını görüyorsanız tebrikler; ilk PHP programınızı çalıştırmışsınız demektir...



Formlarla Çalışmak

Dinamik etkileşim araçlarından oluşan HTML formlarından elde edilen değerlerin işlenmesi, PHP'nin en önemli işlevsel kullanım alanlarından birisini oluşturmaktadır.

Bu kesimde HTML formları hakkında temel seviyede bilgi verilmesi amaçlanmıştır. Bu bilgiler, ilerleyen bölümlerde ele alacağımız MySQL&PHP uygulamaları konusuna da temel oluşturacaktır.

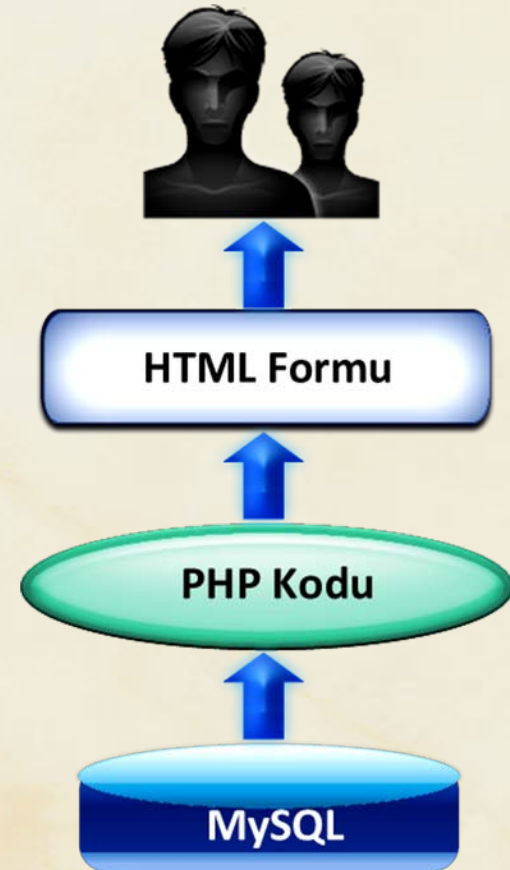


Formlarla Çalışmak

İnternet üzerinde çalışan veritabanı uygulamalarında formlar, kullanıcının doğrudan etkileşim kurduğu en üst katmanda yer alırlar. Dolayısıyla son derece dikkatli ve kullanıcı-dostu olarak tasarlanmaları gerekir.

Formlarla Çalışmak

İnternet üzerinde çalışan veritabanı uygulamalarında nesne hiyerarşisini resimdeki gibi gösterebiliriz:





Formlarla Çalışmak

HTML formları, üzerlerinde bulunan metin alanları, radyo düğmeleri, işaretleme kutuları ya da liste kutuları gibi çeşitli etkileşim nesneleri yoluyla kullanıcıdan bilgi alınmasını sağlayan araçlardır.



Formlarla Çalışmak

Bu formlar,

`<form>`

...

...

`</form>`

etiketleri arasında oluşturulur.

Formlarla Çalışmak

Bu durumda metin tabanlı bir verinin girilmesini sağlayan form, aşağıdaki gibi olacaktır:

```
<form>
```

```
    <p>Adınız: <input type="text"  
name="ad" ></p>
```

```
    <p>Soyadınız: <input type="text"  
name="soyad" ></p>  
</form>
```



Formlarla Çalışmak

Burada `<input type="text" name="ad">` bildirimi, alanın bir girdi (input) alanı, türünün ise metin kutusu (text) olduğunu söylemektedir.

“name” özelliği ise, nesnenin programatik adını göstermektedir.

Formlarla Çalışmak

Tasarladığımız bu formu, bilgileri gönderme işlemini sağlayacak bir komut düğmesi ile tamamlayalım:

```
<form>
```

```
    <p>Adınız: <input type="text"  
name="ad" ></p>
```

```
    <p>Soyadınız: <input type="text"  
name="soyad" ></p>
```

```
    <p><input type="submit" ></p>  
</form>
```

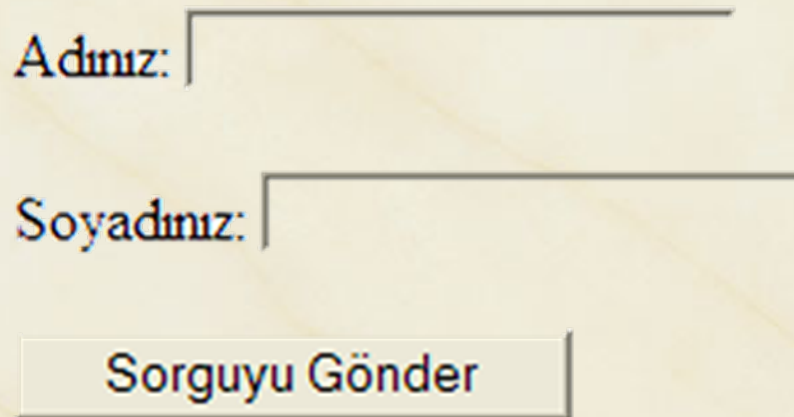


Formlarla Çalışmak

Burada `<input type="submit">` bildirimi, yine alanın bir girdi (input) alanı, ancak türünün gönderme butonu (submit) olduğunu söylemektedir.

Formlarla Çalışmak

Bu koda ait ekran görüntüsü aşağıdaki gibi olacaktır:



Adınız:

Soyadınız:

Formlarla Çalışmak

Gönderme butonuna basıldığında gerçekleşecek eylemin belirlenmesi için form etiketinin action özelliği kullanılır. Buraya yazılacak veri, herhangi bir script kodunun çalıştırılması ile ilgili olabileceği gibi, doğrudan bir php sayfasının adresi de yazılabilir.

Formlarla Çalışmak

```
<form action="verileri_al.php"  
method="post" >  
    <p>Adınız: <input type="text"  
name="ad" ></p>  
    <p>Soyadınız: <input type="text"  
name="soyad" ></p>  
    <p><input type="submit" ></p>  
</form>
```



Formlarla Çalışmak

Burada gönderme butonuna basıldığı takdirde, internet tarayıcı tarafından verileri_al.php dosyası yüklenecektir.

Formlarla Çalışmak

Bu değer boş bırakıldığı takdirde, yani,

```
<form action=" " method="post" >
    <p>Adınız: <input type="text"
name="ad" ></p>
    <p>Soyadınız: <input type="text"
name="soyad" ></p>
    <p><input type="submit" ></p>
</form>
```



Formlarla Çalışmak

...durumunda, gönderme butonu mevcut sayfanın yeniden yüklenmesini sağlayacaktır. Bu durumda çalışmasını istediğimiz php kodlarının da aynı sayfa üzerinde olması gerekmektedir.



Formlarla Çalışmak

Bilgilerin form üzerinden gönderilmesi için iki farklı yöntemden birisi seçilebilir.

Bunlardan ilki “post” , diğeri ise “get” yöntemidir. Gönderme yöntemi, form etiketinin method özelliği kullanılarak belirlenir.



Formlarla Çalışmak

Bilgilerin alınış yöntemi de, gönderme için seçilen yönteme göre değişecektir.

Öncelikle “post” metodunu inceleyelim.

Formlarla Çalışmak

Aşağıdaki HTML formunu göz önüne alalım:

```
<html>
<body>
<form action="verileri_al.php" method="post">
    <p>Adınız: <input type="text" name="ad"></p>
    <p>Soyadınız: <input type="text" name="soyad"
></p>
    <p><input type="submit"></p>
</form>
</body>
</html>
```

Formlarla Çalışmak

Bu kodu taşıyan dosyayı,
“verileri_gonder.html” adıyla kaydedelim.
Şimdi, action özelliğinde belirtilen
“verileri_al.php” dosyasının içereceği kodları
yazalım.

```
<?php
    $kullaniciAdi = $_POST['ad'];
    $kullaniciSoyadi = $_POST['soyad'];
    echo "Merhaba $kullaniciAdi $kullaniciSoyadi";
?>
```



Formlarla Çalışmak

İnternet tarayıcısı tarafından görüntülenecek dosya, “verileri_gonder.html” adıyla kaydettiğimiz form dosyası olacaktır. Bu form üzerinde yer alan veri gönderme butonuna basıldığında internet tarayıcısı tarafından “verileri_al.php” dosyası yüklenecek ve PHP çözümleyicisi tarafından çalıştırılacaktır.

Formlarla Çalışmak

Sonuç olarak ekranda aşağıdaki gibi bir görüntü olacaktır:

```
Merhaba Tolga Güyer
```



Formlarla Çalışmak

Kullanıcı tarafından verileri_gonder.html dosyasında girilen bilgiler, post metodu kullanılarak verileri_al.php dosyasına gönderilmiştir ve kullanıcı gönderilen verileri ancak bu PHP kodları çalıştırıldığında görebilmiştir. Veriler, \$_POST fonksiyonu kullanılarak alınmıştır. Parametre olarak da form üzerindeki alanın adı kullanılmıştır. Get metodunda ise durum biraz farklıdır.

Formlarla Çalışmak

Aynı örneği bu defa get metodunu kullanarak yapalım:

```
<html>
<body>
<form action="verileri_al.php" method="get">
    <p>Adınız: <input type="text" name="ad"></p>
    <p>Soyadınız: <input type="text" name="soyad"
></p>
    <p><input type="submit"></p>
</form>
</body>
</html>
```



Formlarla Çalışmak

Bu durumda HTML dosyasında değişen kısım form etiketinin method özelliğinin değeri olmuştur. PHP dosyasında ise verilerin alınmasında kullanılan `$_POST` fonksiyonu yerine `$_GET` fonksiyonu kullanılmıştır.

Formlarla Çalışmak

```
<?php
    $kullaniciAdi = $_GET['ad'];
    $kullaniciSoyadi = $_GET['soyad'];
    echo "Merhaba $kullaniciAdi $kullaniciSoyadi";
?>
```




Formlarla Çalışmak

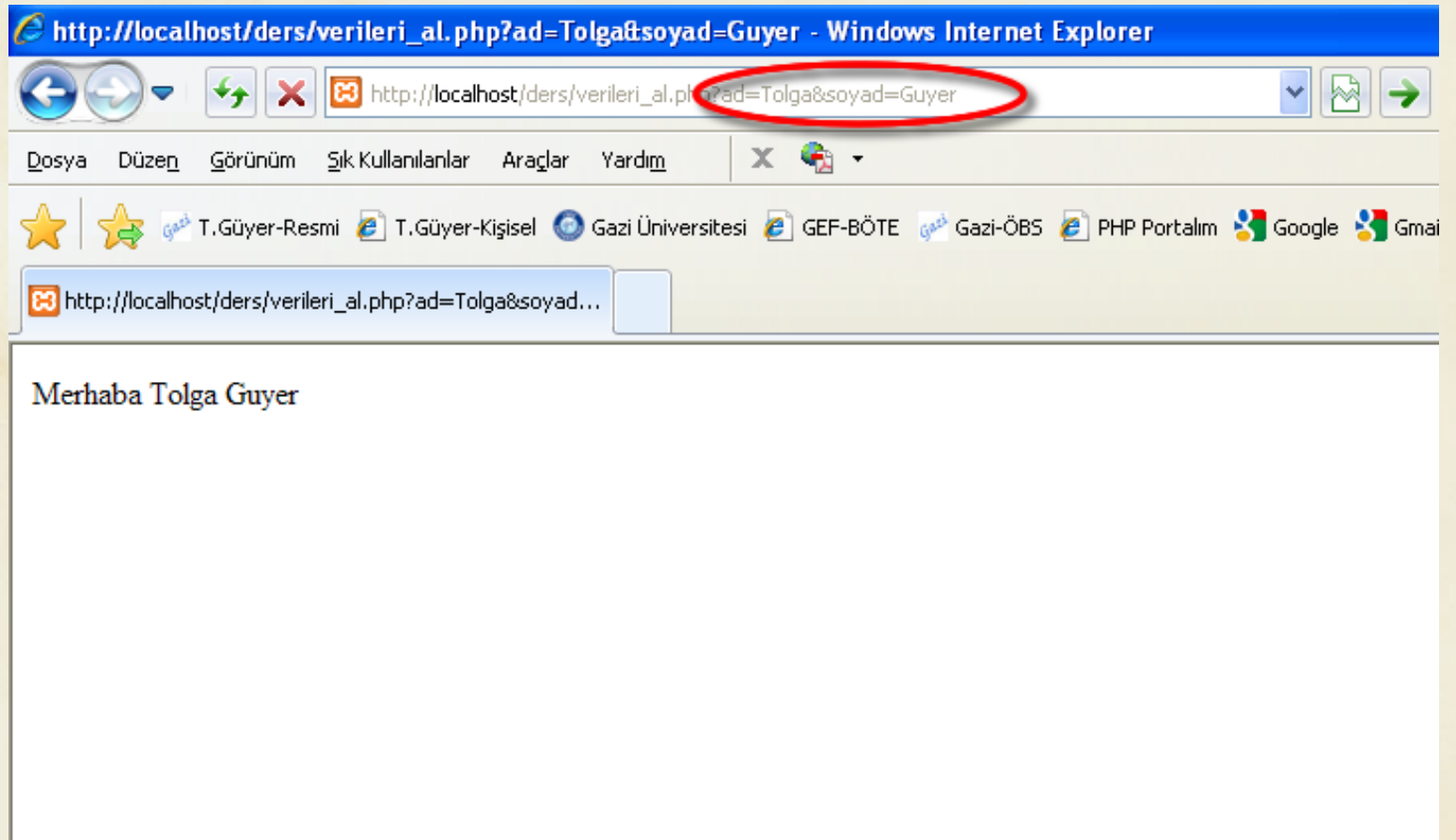
Aynı amaçla kullanılan bu iki yöntem arasındaki en önemli fark, get metodunda verilerin gönderiliş biçiminden kaynaklanmaktadır. Bu durumda veriler, doğrudan internet tarayıcısının adres çubuğu kullanılarak aktarılırlar.



Formlarla Çalışmak

HTML formu doldurulup gönderme butonuna basıldığı anda, internet tarayıcısında resimdeki gibi bir durum oluşacaktır.

Formlarla Çalışmak





Formlarla Çalışmak

Resimde de görüldüğü gibi internet tarayıcısının adres çubuğunda yer alan

`http://.../verileri_al.php?ad=Tolga&soyad=Guyer`

adres bilgisi, gönderilen parametreleri de içermektedir.



Formlarla Çalışmak

Dolayısıyla gizlilik içeren bilgilerin get metodu kullanılarak gönderilmesi önerilmez. Bu şekilde gönderilecekse, verilerin şifrelenmesi ve kullanılmadan önce şifrelerinin çözülmesi bu güvenlik problemini ortadan kaldıracaktır.



Formlarla Çalışmak

Ayrıca parametre sayısı çok fazla olduğunda adres bilgisi çok uzun olacağından ve bu bilgi için 1 Kb üst sınırı bulunduğundan, bu durumda post metodu tercih edilmelidir.

Diğer yandan post metodu kullanıldığında bilgiler HTTP tarafından paketlenerek gönderildiğinden, get metoduna oranla daha yavaş bir transfer olacağı söylenebilir.



Formlarla Çalışmak

Aynı örneği tek dosya üzerinde çalışacak şekilde tasarlayalım. Bu durumda hem HTML formu, hem de PHP kodu aynı dosya üzerinde olacaktır ve gönderme butonuna basıldığında bu dosya tekrar yüklenecektir.

Formlarla Çalışmak

```
<html>
<body>
<form action="" method="post">
<p>Adınız: <input type="text" name="ad"></p>
<p>Soyadınız: <input type="text" name="soyad"></p>
<p><input type="submit" name="dugme"></p>
</form>

<?php
    if(isset($_POST["dugme"])){
        $kullaniciAdi = $_POST['ad'];
        $kullaniciSoyadi = $_POST['soyad'];
        echo "Merhaba $kullaniciAdi $kullaniciSoyadi";
    }
?>
</body>
</html>
```



Formlarla Çalışmak

Örneğin bu biçiminde, aynı sayfa tekrar yüklendiğinden, ilk yüklenişinde çalışmasını istemediğimiz bölüm bir koşul ile kontrol altına alınmıştır. Şöyle ki, bu koşulda yer alan “isset” fonksiyonu, “dugme” adlı butona basıldığında sağlanmaktadır. Dolayısıyla sayfanın ilk yüklenişinde **kırmızı renkli kod** otomatik olarak çalışmayacaktır.



Formlarla Çalışmak

Koşullu ifadeler ve çalışma biçimleri, ikinci bölümde detaylı olarak ele alınacaktır.