

PHP ile Yazılım Geliřtirme Atölyesi

Sinan Ürün

Bilgisayar Mühendisi ve Öğretmeni



BİLGİ İŞLEM
DAİRESİ
BAŞKANLIĞI

 **Tech
Istanbul**



Online



2, 9, 16, 23 Temmuz
18.00

Php ile Yazılım Geliştirme Atölyesi



Sinan ÜRÜN
Bilgisayar Mühendisi ve Öğretmeni

Atölyenin Amacı

- Bu 4 haftalık PHP atölyesinin amacı, katılımcılara PHP programlama dilinin temellerini öğretmek ve dinamik web uygulamaları geliştirebilmeleri için gerekli becerileri kazandırmaktır.
- Eğitim, başlangıç ve orta seviye geliştiricilere hitap eder ve şu hedeflere odaklanır:

Atölyenin Amacı

- **PHP Temellerini Anlama:** PHP'nin ne olduğu, nasıl çalıştığı ve web geliştirmedeki rolünü kavrama.
- **Pratik Programlama Becerileri:** Değişkenler, veri tipleri, operatörler, koşullu ifadeler, döngüler ve diziler gibi temel programlama yapılarını öğrenme.
- **Fonksiyonlar ve Form İşlemleri:** Kendi fonksiyonlarını yazma, hazır fonksiyonları kullanma ve GET/POST yöntemleriyle form verilerini işleme.
- **Oturum ve Veritabanı Yönetimi:** Kullanıcı oturumlarını yönetme, çerezlerle çalışma ve MySQL ile CRUD (Oluşturma, Okuma, Güncelleme, Silme) işlemleri gerçekleştirme.
- **Uygulamalı Öğrenme:** Gerçek dünya senaryolarına dayalı mini projeler ve egzersizlerle teorik bilgiyi pratiğe dökme.

Hedef Kitle

- Web geliştirme öğrenmek isteyen yeni başlayanlar, temel programlama bilgisi olanlar ve PHP ile dinamik web uygulamaları geliştirmek isteyenler.

Beklenen Hedef Çıktılar

- Katılımcılar, PHP ile temel bir web uygulamasını sıfırdan geliştirebilecek.
- Veritabanı entegrasyonu ve kullanıcı yönetimi gibi gerçek dünya senaryolarını uygulayabilecek.
- PHP'nin temel yapı taşlarını öğrenerek ileri seviye web geliştirme için sağlam bir temel oluşturacak.

Eğitmeni Tanıyalım - Sinan ÜRÜN



Bilgisayar Mühendisi ve Öğretmeni



Yazılım Mühendisliği Yüksek Lisans



15 Yıl Yazılım ve Bilişim Teknolojileri Alanında
Eğitmenlik



BTK Akademi Eğitmenliği



Udemy ve Youtube Video Eğitimi

içerik

- **Gün 01 - Php Temelleri ve Veri Yapıları**

- Php Nedir? Nasıl Çalışır ?
- Php ve Gerekli Programların Kurulumu
- Temel Yazdırma İşlemleri
- Değişken tanımlama - Sabit
- Veri Tipleri - Tip Dönüşümleri
- Operatörler
- String İşlemleri

- **Gün 02 - Php'de Programlama Yapıları**

- Koşullu ifadeler (if...else....Elseif - switch case)
- Döngüler (for....while....foreach)
- Diziler
- Diziler ile işlemler

İçerik

- **Gün 03 - Php Fonksiyonlar ve Form işlemleri**
 - Fonksiyon işlemleri (tanımlama, fonksiyon çağırma vb)
 - Hazır fonksiyonlar
 - Form işlemleri (Get ve Post işlemleri)
- **Gün 04 - Php'de Oturum Yönetimi Ve Veri Tabanı İşlemleri**
 - Oturum ve cookie tanımlama
 - Veri tabanı oluşturma
 - Veri tabanı CRUD işlemleri

Php ile Web Geliştirme Atölyesi

1. Gün : PHP Temelleri ve Veri Yapıları

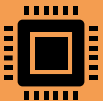
Php Nedir ? Nasıl Çalışır?



PHP: Hypertext Preprocessor, sunucu tarafında çalışan açık kaynaklı betik dili.



Kullanım Alanları: Web siteleri (WordPress, Joomla), e-ticaret, form işleme.



Avantajlar:

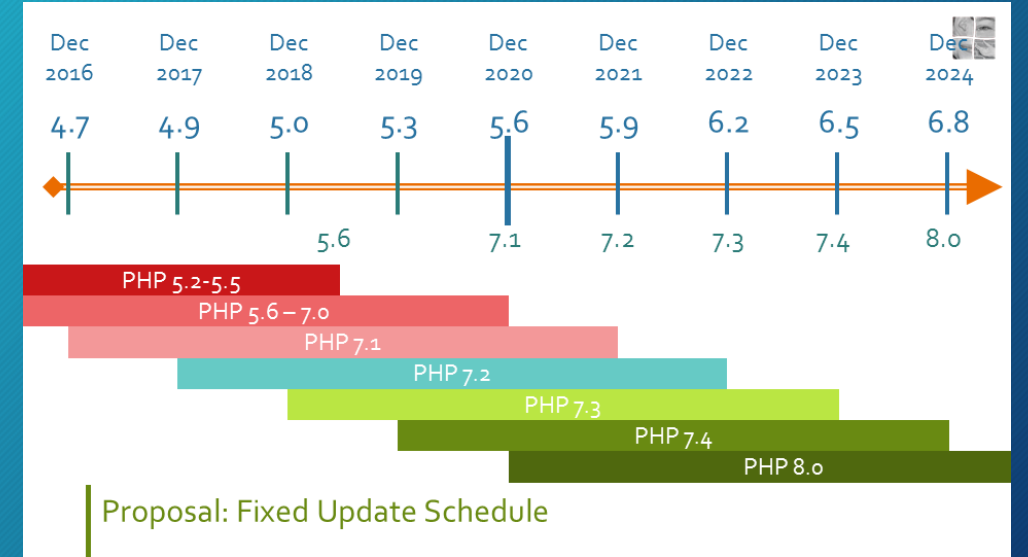
Ücretsiz ve açık kaynak.
Geniş topluluk desteği.
HTML ile kolay entegrasyon.

Php Nedir ? Nasıl Çalışır ?



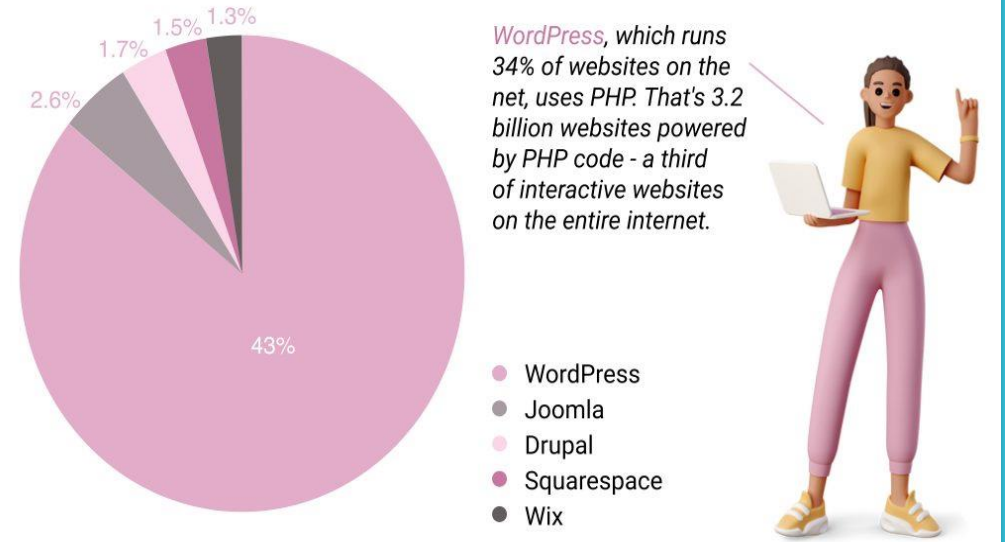
Php Nedir ? Nasıl Çalışır ?

- PHP, açılımıyla Hypertext Preprocessor ya da eski açılımıyla Personal Home Page, internet için üretilmiş, sunucu taraflı, çok geniş kullanımlı, genel amaçlı, içerisine HTML gömülebilen betik ve programlama dilidir.
- İlk kez 1995 yılında Rasmus Lerdorf tarafından yaratılan PHP'nin geliştirilmesi bugün PHP topluluğu tarafından sürdürülmektedir.
- Ocak 2013 itibarıyla 244 milyondan fazla web sitesi PHP ile çalışırken 2.1 milyon web sunucusunda PHP kurulumu bulunmaktadır.

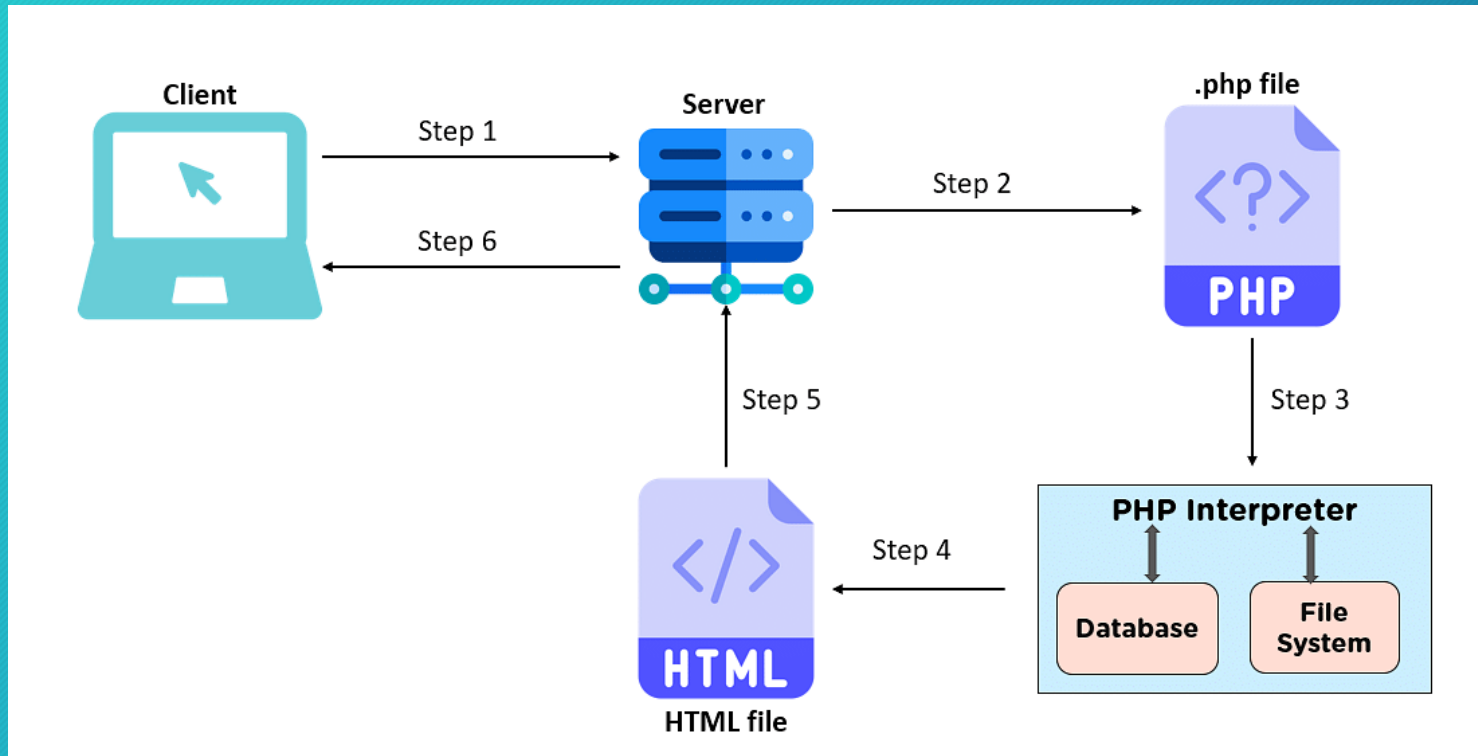


Php Öldü Mü ?

- PHP, MediaWiki,WordPress,Joomla!, Drupal,Moodle, eZ Publish, eZ Platform ve SilverStripe dahil olmak üzere Web içerik yönetim sistemlerinde kullanılır.
- 28 Ocak 2024 tarihi itibarıyla (PHP 8.3'ün yayınlanmasından 2 ay sonra), dilin belirlenebildiği web sitelerinin %76,5'inde sunucu tarafı programlama dili olarak PHP kullanılmaktadır;
- PHP 7, dilin en çok kullanılan sürümüdür; PHP kullanan web sitelerinin %58,8'i bu sürümü kullanırken, %23,9'u PHP 8 ve %17,1'i PHP 5 kullanmaktadır.



Php Nedir ? Nasıl Çalışır?



İlk Php Kodu

php

```
<?php  
echo "Merhaba, PHP!";  
?>
```

Output

Merhaba PHP!

- Bu kod tarayıcıda
- "Merhaba, PHP!" yazdırır.
- Php kodları
 <?php
 kod(lar)
 ?>
- Şeklinde yazılır

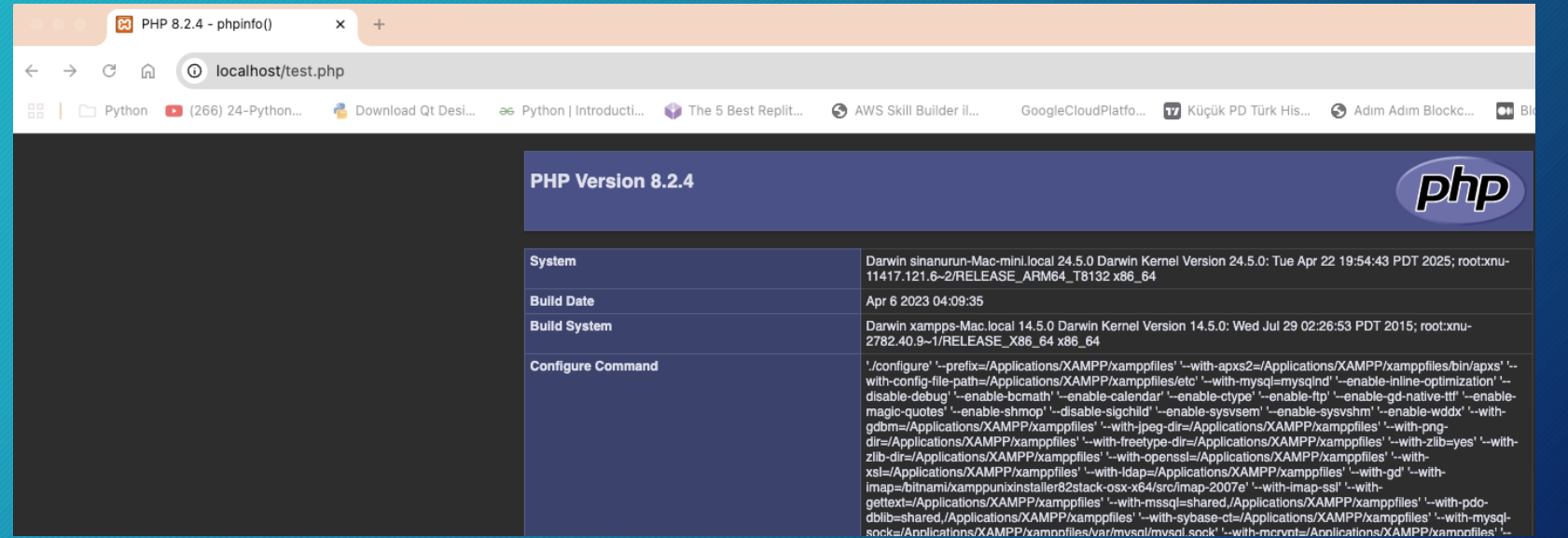
Php ve Gerekli Programların Kurulumu

- Gerekli Araçları:
 - Php : Betik Dilini Çalıştırır
 - Web Sunucusu : Apache veya Nginx
 - Veritabanı Sunucusu : Mysql (Maria DB) ileride kullanılacak
 - Kolay Çözüm : Xampp



XAMPP Kurulumu

- apachefriends.org'dan XAMPP indir.
- Kurulum sihirbazını takip et (Windows/Mac/Linux).
- htdocs klasörüne test.php oluştur:
- test.php'de phpinfo() komutunu çalıştır.
- Çıktıyı görüntüle



The screenshot shows a web browser window with the address bar displaying 'localhost/test.php'. The page content is the PHP 8.2.4 phpinfo() output, which includes the PHP logo and a table of system information.

PHP Version 8.2.4	
System	Darwin sinanurun-Mac-mini.local 24.5.0 Darwin Kernel Version 24.5.0: Tue Apr 22 19:54:43 PDT 2025; root:xnu-11417.121.6~2/RELEASE_ARM64_T8132 x86_64
Build Date	Apr 6 2023 04:09:35
Build System	Darwin xampps-Mac.local 14.5.0 Darwin Kernel Version 14.5.0: Wed Jul 29 02:26:53 PDT 2015; root:xnu-2782.40.9~1/RELEASE_X86_64 x86_64
Configure Command	./configure '--prefix=/Applications/XAMPP/xamppfiles' '--with-apxs2=/Applications/XAMPP/xamppfiles/bin/apxs' '--with-config-file-path=/Applications/XAMPP/xamppfiles/etc' '--with-mysql=mysqlnd' '--enable-inline-optimization' '--disable-debug' '--enable-bcmath' '--enable-calendar' '--enable-ctype' '--enable-ftp' '--enable-gd-native-ttf' '--enable-magic-quotes' '--enable-shmop' '--disable-sigchild' '--enable-sysvsem' '--enable-sysvshm' '--enable-wddx' '--with-gdbm=/Applications/XAMPP/xamppfiles' '--with-jpeg-dir=/Applications/XAMPP/xamppfiles' '--with-png-dir=/Applications/XAMPP/xamppfiles' '--with-freetype-dir=/Applications/XAMPP/xamppfiles' '--with-zlib=yes' '--with-zlib-dir=/Applications/XAMPP/xamppfiles' '--with-openssl=/Applications/XAMPP/xamppfiles' '--with-xsl=/Applications/XAMPP/xamppfiles' '--with-ldap=/Applications/XAMPP/xamppfiles' '--with-gd' '--with-imap=bitnami/xamppunixinstaller82stack-osx-x64/src/imap-2007e' '--with-imap-ssl' '--with-gettext=/Applications/XAMPP/xamppfiles' '--with-mssql=shared,/Applications/XAMPP/xamppfiles' '--with-pdo-dblib=shared,/Applications/XAMPP/xamppfiles' '--with-sybase-ct=/Applications/XAMPP/xamppfiles' '--with-mysql-sock=/Applications/XAMPP/xamppfiles/var/mysql/mysql.sock' '--with-mcrypt=/Applications/XAMPP/xamppfiles' '--

Temel Yazdırma İşlemleri

- echo: Hızlı, birden fazla parametre alır, dönüş değeri yok.
- print: Tek parametre, dönüş değeri 1.
- print_r(): Diziler için, hata ayıklama.
- var_dump(): Değişken tipi ve değerini detaylı gösterir.

Örnek Yazdırma Kodları

```
test.php
1  <?php
2  echo "Merhaba, Dünya!<br>";
3  echo "Birinci ", "İkinci ", "Üçüncü"; // Çoklu parametre
4  print "<br>Bu print ile yazıldı.";
5  $dizi = ["elma", "armut", "çilek"];
6  echo "<br>Dizi (print_r): ";
7  print_r(value: $dizi);
8  echo "<br>Dizi (var_dump): ";
9  var_dump(value: $dizi);
10 ?>
```

localhost/test.php

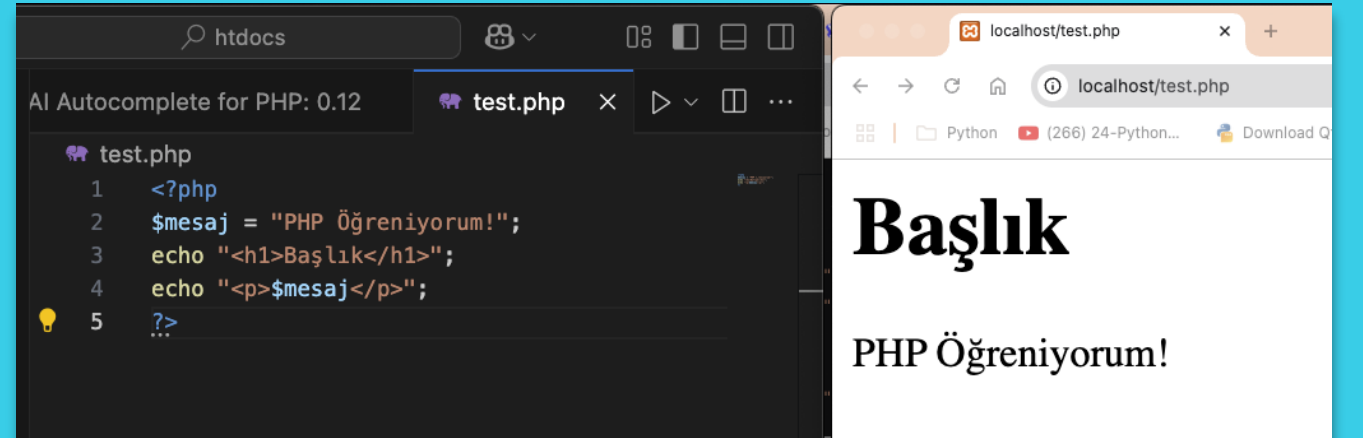
Merhaba, Dünya!
Birinci İkinci Üçüncü
Bu print ile yazıldı.
Dizi (print_r): Array ([0] => elma [1] => armut [2] => çilek)
Dizi (var_dump): array(3) { [0]=> string(4) "elma" [1]=> string(5) "armut" [2]=> string(6) "çilek" }

Php Yazdırma Komutları Farkı

ÖZELLİK	ECHO	PRINT	PRINT_R()	VAR_DUMP()	↓
Dizi/Nesne Yazdırır mı?	✗	✗	✓	✓	
Return Değeri	Yok	1 (true)	Opsiyonel	Yok	
Okunabilirlik	Basit	Basit	Orta	Detaylı	
Kullanım Amacı	Genel çıktı	Basit çıktı	Debug	Detaylı debug	

Php kodları ile HTML yazdırılabilir

- Html kodlarını birer metinsel ifade gibi çıktı olarak tarayıcıda görüntüleyebiliriz.
- Php kodlarının çıktıları zaten html olduğu için html kodlarını da kolaylıkla uygun formatta tarayıcıda sunabilir.
- PHP, HTML etiketleriyle birleştğinde dinamik içerik üretir.



The image shows a side-by-side comparison of a PHP script and its rendered output. On the left, a code editor window titled 'test.php' displays the following code:

```
1 <?php
2 $mesaj = "PHP Öğreniyorum!";
3 echo "<h1>Başlık</h1>";
4 echo "<p>$mesaj</p>";
5 ?>
```

On the right, a web browser window shows the output of this script at the URL 'localhost/test.php'. The browser displays a large heading 'Başlık' and a paragraph 'PHP Öğreniyorum!' below it.

Yorum - Comment

- Php dilinde yazılan kodlarda 3 farklı yöntem ile yorum satırları oluşturulabilir
- Tek satırlık yorumlar : //
- Tek satırlık yorumlar : #
- Çok satırlı yorumlar : /* */
- // yorum satırı
- # yorum satırı
- /*
yorum
satırları
*/

Değişken Tanımlama

- \$ sembolü ile başlar: \$degisken.
- Harf veya alt çizgi ile başlar, rakam/special karakter içeremez.
- Büyük-küçük harf duyarlı: \$isim ≠ \$Isim.

Doğru	Yanlış
\$ad	\$1ad
\$kullanici_adi	\$kullanici adi

Değişken Tanımlama Kuralları

KURAL	AÇIKLAMA
Başlangıç karakteri	<code>\$</code> ile başlar. Örnek: <code>\$isim</code> , <code>\$yas</code>
Harf veya alt çizgiyle başlamalı	İlk karakter harf (<code>a-z</code> , <code>A-Z</code>) veya alt çizgi (<code>_</code>) olabilir. Sayıyla başlayamaz. ✗ <code>\$Sisim</code> ha
İzin verilen karakterler	Harfler, rakamlar ve alt çizgi (<code>_</code>) kullanılabilir.
Büyük/küçük harf duyarlılığı	PHP büyük/küçük harfe duyarlıdır . <code>\$isim</code> ≠ <code>\$isim</code>
Değer ataması	<code>=</code> operatörü ile yapılır. Örnek: <code>\$isim = "Ali";</code>
Veri türüne gerek yok	PHP dinamik tipli bir dildir. Değişkenlere tür belirtmeye gerek yoktur.
Geçici olarak boş bırakılabilir mi?	Evet, <code>null</code> atanabilir: <code>\$degisken = null;</code>
Özel karakterler	Boşluk, tire (<code>-</code>), nokta (<code>.</code>) gibi özel karakterler kullanılamaz.
Anahtar kelimelerle isimlendirme	PHP anahtar kelimelerle (örn. <code>if</code> , <code>for</code> , <code>echo</code>) isim verilemez. ✗ <code>\$if = 5;</code> hatalı.

Veri Tipleri ve Tip Dönüşümleri

- **Skaler:** string, integer, float, boolean.
- **Bileşik:** array, object.
- **Özel:** null, resource.
- `gettype()`: Değişkenin tipini döndürür.
- `is_int()`, `is_string()`: Tip doğrulama.
- `var_dump()`: Detaylı bilgi.

Veri Tipleri ve Tip Dönüşümleri

- **Otomatik:** PHP bağlama göre dönüştürür.
- **Manuel:** (int), (string), (float).

Operatörler

- Aritmetik operatörler
- Atama operatörleri
- Karşılaştırma operatörleri
- Artırma/Azaltma operatörleri
- Mantıksal operatörler
- Dize operatörleri
- Dizi operatörleri
- Koşullu atama operatörleri

Sabitler Constants

- Yürütülen program süresinde değeri değişmeyecek verileri tutmak için kullanılan yapılardır.
- `define(isim, deger);`
- `define("zam_orani", 1.5);`
 - `zam_orani` adında bir sabit tanımlanmış oldu
- Ayrıca aşağıdaki gibi de tanımlanabilir
- `const zam_orani = 1.5;`

Php ile Web Geliştirme Atölyesi

2. Gün

Koşullu İfadeler

- PHP'de koşullu durumlar (conditional statements), belirli kod bloklarını yalnızca belirli şartlar sağlandığında çalıştırmak için kullanılır.
- Blok koşul durumunun gerçekleşmesi yani true - Doğru durumu arayacak ve ona göre blok çalıştıracaktır.

Php Koşullu İfadeler - If ifadesi

- Bir koşul doğruysa (true) belirli bir kod bloğunu çalıştırır.

```
php
1  if (koşul) {
2      // Koşul doğruysa bu kod çalışır
3  }
```

```
php
1  $yas = 18;
2
3  if ($yas >= 18) {
4      echo "Ehliyet alabilirsiniz.";
5  }
```


Php Koşullu İfadeler - If..else ifadesi

- Bir koşul doğruysa (true) if kod bloğunu çalıştırır değilse else bloğuna ait kod bloğunu çalıştırır.

```
php
1 if (koşul) {
2     // Doğruysa burası çalışır
3 } else {
4     // Yanlışsa burası çalışır
5 }
```

```
php
1 $yas = 16;
2
3 if ($yas >= 18) {
4     echo "Ehliyet alabilirsiniz.";
5 } else {
6     echo "Henüz ehliyet alamazsınız.";
7 }
```

Php Koşullu İfadeler - If..elseif..else ifadesi

- Birden fazla koşulu kontrol etmek için kullanılır.
- Bir koşul doğruysa (true) if kod bloğunu çalıştırır değilse elseif diğer bir koşulu kontrol eder, eğer elseif blok(ları) doğru koşulu sağlamaz ise else bloğuna ait kod bloğunu çalıştırır.

php

```
1 if (koşul1) {  
2     // Koşul1 doğruysa  
3 } elseif (koşul2) {  
4     // Koşul2 doğruysa  
5 } else {  
6     // Hiçbiri doğru değilse  
7 }
```

```
1 $not = 75;  
2  
3 if ($not >= 90) {  
4     echo "Notunuz: AA";  
5 } elseif ($not >= 80) {  
6     echo "Notunuz: BA";  
7 } elseif ($not >= 70) {  
8     echo "Notunuz: BB";  
9 } else {  
10     echo "Kaldınız.";  
11 }
```

Ternary Operatör (Üçlü Koşul Operatörü)

- koşul ? doğruysa_değer : yanlışsa_değer;

php

```
1 $yas = 18;  
2  
3 echo ($yas >= 18) ? "Erişkin" : "Çocuk";
```

```
1 $puan = 60;  
2 $sonuc = ($puan >= 50) ? "Geçti" : "Kaldı";  
3  
4 echo $sonuc; // Geçti
```

php

```
1 $ad = $_GET['ad'] ?? 'Misafir';  
2 echo "Hoş geldin, $ad"; // Eğer $_GET['ad'] varsa onu, yoksa 'Misafir' kullanır
```

switch İfadesi

- Aynı değişkenin farklı değerlerine göre farklı işlemler yapmak için kullanılır. Özellikle birçok if...elseif yazmaktan kaçınmak için idealdir.

```
php
1 switch (değişken) {
2     case değer1:
3         // Kod bloğu
4         break;
5     case değer2:
6         // Kod bloğu
7         break;
8     default:
9         // Hiçbiri uymazsa
10 }
```

```
1 $gun = "pazar";
2
3 switch ($gun) {
4     case "pazartesi":
5         echo "Bugün Pazartesi.";
6         break;
7     case "salı":
8         echo "Bugün Salı.";
9         break;
10    case "pazar":
11        echo "Bugün Pazar.";
12        break;
13    default:
14        echo "Diğer günler...";
15 }
```


Php'de Döngüler - for Döngüsü

- Belirli bir sayıda işlem yapmak için kullanılır.

```
php
1 for (başlangıç; koşul; artış/azalış) {
2     // Koşul doğru olduğu sürece çalışacak kod
3 }
```

php

```
1 v for ($i = 1; $i <= 5; $i++) {
2     echo "Sayı: $i <br>";
3 }
```

"Bu döngü 1'den 5'e kadar olan sayıları ekrana yazdırır: "

```
1 Sayı: 1
2 Sayı: 2
3 Sayı: 3
4 Sayı: 4
5 Sayı: 5
```

Php'de Döngüler - while Döngüsü

- Bir koşul doğru olduğu sürece kod bloğunu çalıştırır.

php

```
1 while (koşul) {  
2     // Koşul doğru olduğu sürece çalışacak kod  
3 }
```

php

```
1 $i = 1;  
2 while ($i <= 5) {  
3     echo "Sayı: $i <br>";  
4     $i++;  
5 }
```

Php'de Döngüler - do...while Döngüsü

- Koşul yanlış olsa bile en az bir kez çalışır. Önce çalışır, sonra koşula bakar.

```
php
1 do {
2     // Kod burada en az bir kez çalışır
3 } while (koşul);
```

```
php
1 $i = 6;
2 do {
3     echo "Sayı: $i <br>";
4     $i++;
5 } while ($i <= 5);
```

Php'de Döngüler - foreach Döngüsü

- Dizilerde (array) ve nesnelerde (objects) gezinmek için kullanılır.

php

```
1 foreach ($dizi as $anahtar => $değer) {  
2     // Her eleman için çalışacak kod  
3 }
```

php

```
1 $renkler = array("Kırmızı", "Yeşil", "Mavi");  
2  
3 foreach ($renkler as $renk) {  
4     echo "Renk: $renk <br>";  
5 }
```

php

```
1 $kullanici = array("Ad" => "Ahmet", "Yaş" => 25, "Şehir" => "İstanbul");  
2  
3 foreach ($kullanici as $ozellik => $deger) {  
4     echo "$ozellik: $deger <br>";  
5 }
```


Php'de Döngüler - break (Döngüden Çıkış)

- break: Döngüyü tamamen durdurur.

```
php
1 v for ($i = 1; $i <= 10; $i++) {
2     if ($i == 5) break;
3     echo "$i ";
4 }
```

"Çıktı: 1 2 3 4 "

Php'de Döngüler - continue (Döngüde Atlama)

- continue: O turu atlar, döngü devam eder.

```
php
1 ✓ for ($i = 1; $i <= 5; $i++) {
2     if ($i == 3) continue;
3     echo "$i ";
4 }
```

"Çıktı: 1 2 4 5 "

Php'de Dizi - Dizi Nedir ? Türleri Nelerdir?

- Bir değişkenin içinde birden fazla değer tutmamızı sağlar.
- Php'de 3 farklı türde dizi yapısı bulunur:
 - Sayısal Diziler (indexed arrays)
 - İlişkisel Diziler (Associative Arrays)
 - Çok Boyutlu Diziler (Multidimensional Arrays)

php

```
1 $renkler = array("Kırmızı", "Yeşil", "Mavi");
```


Php'de Dizi - Sayısal(Indexed) Dizi

- Elemanlar otomatik veya manuel olarak sayısal anahtarlarla atandığı dizi tipidir.
- Tüm elemanlar bir index numarası ile bellekte tutulur.
- İki sarklı şekilde tanımlanabilir:

php

```
1 $sehirler = array("İstanbul", "Ankara", "İzmir");  
2 // Veya kısa yol (PHP 5.4+)  
3 $sehirler = ["İstanbul", "Ankara", "İzmir"];
```


Php'de Dizi - İlişkisel (Associative) Diziler

- Anahtarlar kullanıcı tarafından belirlenen dizilerdir (string ya da sayı olabilir).

php

```
1 $kullanici = array("ad" => "Ali", "yas" => 25);  
2 // Kısa yol:  
3 $kullanici = ["ad" => "Ali", "yas" => 25];
```

Php'de Dizi - Çok Boyutlu(Multidimensional) Diziler

- Dizinin içinde başka bir dizi barındırır. Tablo gibi yapıları düşünebilirsiniz

```
php
1 $ogrenciler = [
2     ["Adem", 20],
3     ["Zeynep", 22],
4     ["Mehmet", 21]
5 ];
6
7 echo $ogrenciler[1][0]; // Zeynep
```

Php ile Web Geliştirme Atölyesi

3. Gün

Php'de Fonksiyon - Fonksiyon Nedir?

- Fonksiyon, belirli bir görevi yerine getirmek için yazılmış olan ve gerektiğinde tekrar tekrar çağırılabilen kod bloklarıdır.
- Eğer belirli bir işlevi veya kod bloğunu farklı yerlerde tekrar tekrar kullanma ihtiyacı oluyor ise kod tekrarını önlemek adına kodlar bir blok haline getirilerek bu bloğa bir fonksiyonalite kazandırılır. Bu duruma fonksiyon oluşturma diyebiliriz.

Php'de Fonksiyon - Neden Kullanılır?

- Kod tekrarını önler : Aynı işlemi birçok yerde yapmak istersen her seferinde aynı kodu yazmak zorunda kalmazsın.
- Modülerlik sağlar : Uygulamayı küçük parçalara ayırmanı sağlar, bu da kodun okunabilirliğini artırır.
- Bakım ve güncelleme kolaylığı sağlar : Bir hata varsa sadece o fonksiyonda düzeltme yaparsın.
- Test edilebilirliği artırır : Fonksiyonlar bağımsız olarak test edilebilir.

Php'de Fonksiyon - Nasıl Tanımlanır?

php

```
1 function fonksiyonAdi($parametre1, $parametre2, ...) {  
2     // Yapılacak işlemler  
3     return $sonuc; // Opsiyonel  
4 }
```

Php'de Fonksiyon - Parametresiz ve Dönüşsüz

php

```
1 function merhabaDe() {  
2     echo "Merhaba Dünya!";  
3 }  
4  
5 merhabaDe(); // Çıktı: Merhaba Dünya!
```


Php'de Fonksiyon - Parametrelili ve Dönüşsüz

php

```
1 function selamVer($isim) {  
2     echo "Merhaba $isim";  
3 }  
4  
5 selamVer("Ahmet"); // Çıktı: Merhaba Ahmet
```


Php'de Fonksiyon - Dönüslü Return

php

```
1 function topla($a, $b) {  
2     return $a + $b;  
3 }  
4  
5 $sonuc = topla(3, 5);  
6 echo $sonuc; // Çıktı: 8
```

Php'de Fonksiyon - Varsayılan Değerli Parametreler

php

```
1 function mesajGoster($mesaj = "Varsayılan mesaj") {  
2     echo $mesaj;  
3 }  
4  
5 mesajGoster(); // Çıktı: Varsayılan mesaj  
6 mesajGoster("Özel mesaj"); // Çıktı: Özel mesaj
```

Php'de Fonksiyon - Dinamik Sayıda Parametre Almak (Variadic Functions)

php

```
1 function sayilariTopla(...$sayilar) {  
2     return array_sum($sayilar);  
3 }  
4  
5 echo sayilariTopla(1, 2, 3, 4); // Çıktı: 10
```

Php'de Fonksiyon - Referans ile Parametre Geçme (& operatörü)

php

```
1 function arttir(&$sayi) {  
2     $sayi++;  
3 }  
4  
5 $a = 5;  
6 arttir($a);  
7 echo $a; // Çıktı: 6
```


Php'de Fonksiyon - Anonim Fonksiyonlar (Closure)

php

```
1 $sayHello = function($isim) {  
2     echo "Merhaba $isim";  
3 };  
4  
5 $sayHello("Ayşe"); // Çıktı: Merhaba Ayşe
```

Php'de Fonksiyon - Callback Fonksiyonlar

php

```
1 function islem($deger, $fonksiyon) {  
2     $fonksiyon($deger);  
3 }  
4  
5 islem("PHP", function($text) {  
6     echo strtoupper($text); // Çıktı: PHP  
7 });
```

Php'de Fonksiyon - Global Değişken

php

```
1  $a = 10;  
2  
3  function degerOku() {  
4      global $a;  
5      echo $a;  
6  }  
7  
8  degerOku(); // Çıktı: 10
```

Php'de Fonksiyon - Static Değişken

php

```
1 function sayac() {  
2     static $sayi = 0;  
3     echo $sayi;  
4     $sayi++;  
5 }  
6  
7 sayac(); // Çıktı: 0  
8 sayac(); // Çıktı: 1  
9 sayac(); // Çıktı: 2
```


Php'de Hazır Fonksiyonlar - Sayısal İşlemler

- `echo rand(1, 10);` // 1 ile 10 arasında rastgele sayı üretir
- `echo round(3.6);` // Çıktı: 4
- `echo abs(-5);` // Çıktı: 5

Php'de Hazır Fonksiyonlar - Tarih Zaman İşlemleri

- `echo date("d/m/Y H:i:s");` // Şu anki tarih ve saat bilgisi
- `echo time();` // Şu andaki zaman damgası

Php'de Dosya İşlemleri

php

```
1 $dosya = fopen("ornek.txt", "r"); // r = read modunda aç
2 if ($dosya) {
3     while (($satir = fgets($dosya)) !== false) {
4         echo $satir . "<br>";
5     }
6     fclose($dosya);
7 }
```

php

```
1 if (file_exists("rapor.pdf")) {
2     echo "Dosya mevcut.";
3 } else {
4     echo "Dosya bulunamadı.";
5 }
```

php

```
1 $dosya = fopen("yazilar.txt", "w"); // w = write modunda aç (eski içerik silinir)
2 fwrite($dosya, "Bu bir yazıdır.\n");
3 fclose($dosya);
```

php

```
1 $dosya = fopen("yazilar.txt", "a"); // a = append (eklemeye izin verir)
2 fwrite($dosya, "Bu eklenen satırdır.\n");
3 fclose($dosya);
```


Php'de Form İşlemleri - post get

- Formlar, kullanıcıdan veri almak için kullanılan HTML öğeleridir.
- Verilerin sunucuya gönderilmesi ve PHP tarafında işlenmesi oldukça yaygın bir durumdur.
- Özellikle POST ve GET yöntemleriyle veri gönderimi yapılır.

Php'de Form İşlemleri - HTML Formu

- HTML'de <form> etiketi ile kullanıcıdan veri toplanır. Kullanıcı bu alanlara bilgi girer (örneğin ad, e-posta, şifre vs.) ve bu bilgiler bir sunucuya gönderilir.

```
<form action="kaydet.php" method="post">  
  Adınız: <input type="text" name="ad"><br>  
  Şehriniz: <input type="text" name="sehir"><br>  
  <input type="submit" value="Gönder">  
</form>
```



Adınız:

Şehriniz:

Php'de Form - post ve get Karşılaştırması

ÖZELLİK	POST	GET
Veri URL'de görünür mü?	Hayır	Evet
Güvenlik	Daha güvenli (veri URL'de görünmez)	Daha az güvenli
Karakter Sınırı	Yok (büyük veriler için uygun)	Sınırlı (URL uzunluğu sınırlıdır)
Cache edilebilir mi?	Hayır	Evet
Favorilere eklenebilir mi?	Hayır	Evet
Kullanım Alanları	Giriş formu, kayıt formu, ödeme vs.	Sayfa filtreleme, arama vs.

Php Formdan Gelen Veriyi Almak

php

```
1 <?php
2 v if ($_SERVER['REQUEST_METHOD'] === 'POST') {
3     $ad = $_POST['ad'];
4     $sehir = $_POST['sehir'];
5
6     echo "Merhaba $ad<br>";
7     echo "Şehriniz: $sehir";
8 }
9 ?>
```

php

```
1 <?php
2 v if ($_SERVER['REQUEST_METHOD'] === 'GET') {
3     $ad = $_GET['ad'];
4     $sehir = $_GET['sehir'];
5
6     echo "Adınız: $ad<br>";
7     echo "Şehriniz: $sehir";
8 }
9 ?>
```

Php'de Formdan Gelen Veriyi Temizlemek

php

```
1 $ad = htmlspecialchars($_POST['ad']);  
2 echo "Hoş geldiniz, $ad";
```

php

```
1 $email = filter_var($_POST['email'], FILTER_VALIDATE_EMAIL);  
2 if ($email) {  
3     echo "Geçerli e-mail: $email";  
4 } else {  
5     echo "Geçersiz e-mail!";  
6 }
```


Php ile Web Geliştirme Atölyesi

4. Gün

Cookies (Çerezler)

- Cookies, kullanıcı bilgilerini tarayıcıya saklamamızı sağlar.
- Tarayıcıda belirli bir süre kalır.
- Genellikle kullanıcı oturum açma durumu, dil tercihi gibi şeyleri saklamak için kullanılır.

Cookies (Çerezler) Tanımlama

php

```
1 // setcookie( isim, değer, süresi, path )  
2 setcookie("kullaniciAdi", "Ahmet", time() + (60*60*24*7), "/"); // 7 gün geçerli
```

- `time() + saniye cinsinden süre` → cookie'nin ne kadar süre tutulacağını belirtir.
- `"Ahmet"` → cookie'nin değeri
- `path` → hangi dizinde kullanılacağı (/ tüm site için)

Cookies (Çerezler) Okuma

php

```
1 if(isset($_COOKIE['kullaniciAdi'])) {  
2     echo "Cookie değeri: " . $_COOKIE['kullaniciAdi'];  
3 } else {  
4     echo "Cookie bulunamadı.";  
5 }
```


Cookies (Çerezler) Silme

php

```
1 setcookie("kullaniciAdi", "", time() - 3600, "/");
```

Session (Oturum Yönetimi)

- Session'lar sunucu tarafında saklanır.
- Güvenlik açısından cookie'lere göre daha iyidir.
- Kullanıcı giriş yaptıysa, bu bilgi session'da tutulabilir.

php

```
1 session_start(); // Her zaman en başta olmalı!
```

Session (Oturum Yönetimi) Tanımlama

php

```
1 $_SESSION['kullaniciAdi'] = "Mehmet";  
2 $_SESSION['email'] = "mehmet@example.com";
```

Session (Oturum Yönetimi) Veri Alma

php

```
1 if(isset($_SESSION['kullaniciAdi'])) {  
2     echo "Hoş geldin, " . $_SESSION['kullaniciAdi'];  
3 }
```


Session (Oturum Yönetimi) Bitirme

php

```
1 session_destroy(); // Tüm session silinir  
2 unset($_SESSION['kullaniciAdi']); // Belirli bir değeri silmek için
```

" ⚠ `session_start()` her dosyada kullanılmalıdır! "

Session (Oturum Yönetimi) Bitirme

php

```
1 session_destroy(); // Tüm session silinir
2 unset($_SESSION['kullaniciAdi']); // Belirli bir değeri silmek için
```

⚠ `session_start()` her dosyada kullanılmalıdır! "

Özellik	Oturum (Session)	Çerez (Cookie)
Depolama Yeri	Sunucu tarafında (genellikle geçici dosyalar, veritabanı vb.)	Kullanıcının tarayıcısında
Depolanan Veri	Kullanıcıya özel kimlik (session ID) ile ilişkilendirilmiş veriler	Anahtar-değer çiftleri şeklinde küçük miktarda veri
Veri Boyutu	Teorik olarak sınırsız (sunucu kaynaklarına bağlı)	Sınırlı (genellikle 4 KB civarı)
Ömür	Tarayıcı kapatıldığında veya oturum süresi dolduğunda sona erer (yapılandırılabilir)	Belirlenen süreye kadar veya tarayıcı kapatılana kadar (kalıcı veya oturum çerezleri)
Güvenlik	Daha güvenli, veri sunucuda saklandığı için kullanıcı tarafından doğrudan erişilemez	Daha az güvenli, kullanıcı tarafından görülebilir ve değiştirilebilir (ancak şifreleme ile geliştirilebilir)
Performans	Sunucu tarafında ek işlem yükü oluşturabilir	Her HTTP isteği ile sunucuya gönderildiği için küçük bir bant genişliği kullanımı yaratır
Kullanım Alanı	Kullanıcı girişi, alışveriş sepeti, kullanıcı tercihleri gibi oturum boyunca geçerli bilgiler	Beni hatırla özelliği, tema tercihleri, son ziyaret edilen sayfalar gibi kalıcı veya yarı kalıcı bilgiler
Erişim	Sadece sunucu tarafından erişilebilir (PHP kodu aracılığıyla)	Hem sunucu hem de istemci tarafında (JavaScript aracılığıyla) erişilebilir (güvenlik açığı yaratabilir)
Kontrol	Sunucu tarafında tam kontrol (oturum başlatma, sona erdirmeye, veri silme)	Kullanıcı tarafından reddedilebilir veya silinebilir

Php'de Veritabanı İşlemleri

- Veritabanı, verilerin düzenli, yapılandırılmış ve organize bir şekilde saklandığı, yönetildiği ve erişildiği elektronik bir sistemdir.
- Bir kütüphanedeki kitapların veya bir mağazadaki ürünlerin belirli bir düzene göre listelendiği ve kolayca bulunabildiği bir kataloğa benzetebiliriz.
- Dijital ortamda ise veritabanları, bu bilgileri çok daha hızlı ve etkili bir şekilde depolama, sorgulama, güncelleme ve silme imkanı sunar.
- CRUD: Create, Read, Update, Delete işlemleridir. Bu işlemleri yapmak için PHP ile MySQLi kullanacağız.

Php'de Veritabanına Bağlanma (MySQLi):

php

```
1 $host = "localhost";
2 $kullanici = "root";
3 $sifre = "";
4 $veritabani = "test_db";
5
6 $baglanti = new mysqli($host, $kullanici, $sifre, $veritabani);
7
8 if ($baglanti->connect_error) {
9     die("Bağlantı hatası: " . $baglanti->connect_error);
10 }
```

Php'de Veritabanına Ekleme (MySQLi):

php

```
1 $sql = "INSERT INTO kullanicilar (ad, email) VALUES ('Ayşe', 'ayse@example.com')";
2 if ($baglanti->query($sql) === TRUE) {
3     echo "Kayıt başarıyla eklendi.";
4 } else {
5     echo "Hata: " . $baglanti->error;
6 }
```

Php'de Veritabanına Okuma (MySQLi):

php

```
1 $sql = "SELECT id, ad, email FROM kullanicilar";
2 $sonuc = $baglanti->query($sql);
3
4 if ($sonuc->num_rows > 0) {
5     while($satir = $sonuc->fetch_assoc()) {
6         echo "ID: " . $satir["id"]. " - Ad: " . $satir["ad"]. " - Email: " . $satir["email"]. "<br>";
7     }
8 } else {
9     echo "Kayıt bulunamadı.";
10 }
```

Php'de Veritabanına Güncelleme (MySQLi):

php

```
1 $sql = "UPDATE kullanicilar SET ad='Fatma' WHERE id=1";
2 if ($baglanti->query($sql) === TRUE) {
3     echo "Kayıt güncellendi.";
4 } else {
5     echo "Hata: " . $baglanti->error;
6 }
```


Php'de Veritabanına Silme (MySQLi):

php

```
1 $sql = "DELETE FROM kullanicilar WHERE id=1";
2 if ($baglanti->query($sql) === TRUE) {
3     echo "Kayıt silindi.";
4 } else {
5     echo "Hata: " . $baglanti->error;
6 }
```

Php ile Web Geliştirme Atölyesi
Dinlediğiniz için teşekkürler

Sinan ÜRÜN
Bilgisayar Mühendisi ve Öğretmeni