

Php 2.dönem

Strlen() ile PHP'de Metin Uzunluğu Bulmak

- Bir string ifadenin uzunluğunu bulmak için kullanılır. İşlem sonucu geriye bir integer değer döndürür.
- Kullanım şekli: strlen(\$degisken)
- Örnek 1.
 - 1. \$webpage ="http://www.dpu.edu.tr";
 - 2. \$uzunluk= strlen(\$webpage);
 - 2. echo strlen(\$webpage);
- Örnek 2.

```
$kullaniciadi="deneyimlikullanici";  
    if (strlen($kullaniciadi) > 12)  
    {  
        echo "Kullanıcı Adı 12 Karakterden Uzun Olamaz<br>";  
        echo "Sizin denemiş olduğunuz kullanıcı adı".strlen($kullaniciadi)." karakterdir.  
<br>"; }  

```

Substr() ile PHP'de Bir Metinden Kesit Alma

- istediğimiz bir değişkenin içeriğinden **kesit** alabiliriz. **Kesit** diyerek neyi kastediyoruz. Örneğin bir değişkenin 2. karakterinden itibaren 4 karakteri elde etmek.
- **Substr** Fonksiyonu ile değişken metninin içeriğinden herhangi bir **kesiti** alıp başka bir değişkene atabiliriz.
 - Substr() Fonksiyonun Kullanımı;

```
$kesit=substr($degisken,"başlangıç indexi","kaç karakter alınacağı");
```

```
$webpage="http://www.dpu.edu.tr";  
$kesit=substr($webpage,11,3);  
echo $kesit; // ekrana "dpu" yazar.
```

Substr ve mb_substr örnek

Php deneme ve örnekler

örnek

deneme ve örnek

deneme ve örnek

- \$degisken = 'Php deneme ve örnekler';
- \$sonhali = substr(\$degisken, -8, 5); // "örnek"
- echo \$degisken."
";
- echo \$sonhali."
";
- echo substr(\$degisken, 3, 16)."
"; // "deneme ve örnek"
- echo mb_substr(\$degisken,3,16,'UTF-8');
- Not: Substr, Türkçe karakterleri (ç, ğ, ı, ö, ş, ü, Ç, Ğ, İ, Ö, Ş, Ü) 2 karakter olarak sayar.

explode(\$ayirici,\$metin)

- Verilen metni belirtilen ayırıcı karaktere göre parçalar ve bir dizi olarak gönderir. Örneğin bir cümledeki kelimeleri bir diziye aktaralım. Buradaki ayırıcımız boşluk karakteri olacaktır.
- <?php
- \$metin="Güzel gören güzel düşünür";
- \$dizi=explode(" ",\$metin); // cümlemiz boşluklardan bölünecek
- print_r(\$dizi); // her kelime dizide bir değer olacaktır
- ?>
-

- Programın çıktısı:

```
Array (
    [0] =>
        Güzel
    [1] =>
        gören
    [2] =>
        güzel
    [3] =>
        düşünür
)
```

Explode Örneği

- <?php
- \$dosya="veri.txt";
- \$metin= file_get_contents(\$dosya);
- \$dizi=explode(" ",\$metin); // cümlemiz boşluklardan bölünecek
- echo "<p style=padding:20px>";
- print_r(\$dizi); // her kelime dizide bir değer olacaktır
- echo "Kelime sayısı : ".count(\$dizi);
- echo "</p>";
- echo "<hr>";
- \$dizi=explode(".", \$metin); // cümlemiz noktaya göre bölünecek
- echo "<pre>";
- print_r(\$dizi); // her kelime dizide bir değer olacaktır
- echo "Cümle sayısı : ".count(\$dizi);
- echo "</pre>";
- echo "<hr>";
- \$dizi=explode(chr(10), \$metin); // cümlemiz enter tuşlarına göre bölünecek
- echo "<pre>";
- print_r(\$dizi); // her kelime dizide bir değer olacaktır
- echo "Paragraf sayısı : ".count(\$dizi);
- echo "</pre>";
- ?>

22.05.2022

```

Array ( [0] => Dünya [1] => Edebiyatında [2] => Gezi [3] => Yazısı. Dünya [4] => edebiyatında [5] => gezi [6] => yazısının [7] => ilk [8] => örnekleri [9] => sayılabilecek [10] => eserleri [11] => verenlerin [12] => başında [13] => Heredotos, [14] => Marco [15] => Polo, [16] => İbni [17] => Batuda [18] => gelir. Türk [19] => Edebiyatında [20] => Gezi [21] => Yazısı. Eski [22] => çağlarda [23] => özellikle [24] => keşif, [25] => ticaret, [26] => savaş [27] => amacıyla [28] => değişik [29] => geziler [30] => yapılmıştır. [31] => Eski [32] => Türk [33] => edebiyatında [34] => gezi [35] => yazısına [36] => 'seyahatname' [37] => denirdi. [38] => Türk [39] => edebiyatında [40] => en [41] => eski [42] => seyahatnameler. [43] => Timur'un [44] => oğlu [45] => Şahruh'un [46] => yanında [47] => bulunan [48] => Gıyasüddin [49] => Nakkaş [50] => ın [51] => yazdığı [52] => "Acâib'ül-Letâif" [53] => ve [54] => Ali [55] => Ekber [56] => Hatâî [57] => adlı [58] => bir [59] => tüccarın [60] => kaleme [61] => aldığı [62] => "Hitâî'nâme'dir. ) Kelime sayısı : 63

Array
(
    [0] => Dünya Edebiyatında Gezi Yazısı
    [1] =>
    Dünya edebiyatında gezi yazısının ilk örnekleri sayılabilecek eserleri verenlerin başında Heredotos, Marco Polo, İbni Batuda gelir
    [2] =>
    Türk Edebiyatında Gezi Yazısı
    [3] =>
    Eski çağlarda özellikle keşif, ticaret, savaş amacıyla değişik geziler yapılmıştır
    [4] => Eski Türk edebiyatında gezi yazısına 'seyahatname' denirdi
    [5] => Türk edebiyatında en eski seyahatnameler
    [6] => Timur'un oğlu Şahruh'un yanında bulunan Gıyasüddin Nakkaş ın yazdığı "Acâib'ül-Letâif" ve Ali Ekber Hatâî adlı bir tüccarın kaleme aldığı "Hitâî'nâme'dir.
    [7] =>
)
Cümle sayısı : 8

Array
(
    [0] => Dünya Edebiyatında Gezi Yazısı.
    [1] => Dünya edebiyatında gezi yazısının ilk örnekleri sayılabilecek eserleri verenlerin başında Heredotos, Marco Polo, İbni Batuda gelir.
    [2] => Türk Edebiyatında Gezi Yazısı.
    [3] => Eski çağlarda özellikle keşif, ticaret, savaş amacıyla değişik geziler yapılmıştır. Eski Türk edebiyatında gezi yazısına 'seyahatname' denirdi.
)
Paragraf sayısı : 4

```

implode(\$ayirici,\$dizi)

- Bu fonksiyon explode() fonksiyonunun aksine bir dizideki değerleri belirtilen ayırıcıyla birleştirerek bir araya getirir. Örneğimizde bir cümleyi oluşturan kelimelerin bulunduğu diziyi birleştirip cümlemizi kuralım.
- <?php
- \$kelimeler=array("Sükut","yalan","söylemekten","ve","başkalarını","çekiştirmekten","herhalde","evladır");
- \$cumle=implode(" ",\$kelimeler);
- echo \$cumle,"!";
- ?>
- Örnek: Ayların arasına tire – koyarak birleştirip ekrana yazdıralım.
- <?php
- \$aylar=array("Ocak","Şubat","Mart","Nisan","Mayıs","Haziran","Temmuz","Ağustos","Eylül","Ekim","Kasım","Aralık");
- \$metin=implode("-", \$aylar);
- echo \$metin;
- ?>

implode kullanımı örneği(dinamik checkbox)

- <?php
- if(isset(\$_POST['SİL']))
- {
- foreach (\$_POST['dersler'] as \$value) {
- echo \$value."
";
- }
- \$str="delete from deneme where ders_adı in(\"\".implode(\$_POST['dersler'],\"\",'').\"\\\")\"";
- echo "silme stringimiz : ".\$str;
- }
- else
- {
- echo "Silmek istediğiniz dersleri işaretleyip SİL butonuna basınız...";
-
- \$dersler=array("PHP","C#","HTML/CSS","Veri Tabanı","İşletim Sistemleri","Algoritma ve Prog.Giriş","Nesne Yön.Prg.");
- echo "<form action=\"\" method=POST>";
- foreach (\$dersler as \$value) {
- if (!empty(\$value))
- echo "<input type='checkbox' name='dersler[]' value='\$value'>".\$value."
";
- }
- echo "<input type='submit' value='SİL' name='SİL'>";
- echo "</form>";
- }
- ?>

Önceki slayttaki örneği checkbox yerine select elementi kullanarak yapınız.

- `<?php`
- `if(isset($_POST['SIL']))`
- `{`
- `foreach ($_POST['dersler'] as $value) {`
- `echo $value."
";`
- `}`
- `$str="delete from deneme where ders_adi in(\"".implode($_POST['dersler'],",","")."\\");`
- `echo "silme stringimiz : ".$str;`
- `}`
- `else`
- `{`
- `echo "Silmek istediğiniz dersleri işaretleyip SIL butonuna basınız...";`
- `$dersler=array("PHP","C#","HTML/CSS","Veri Tabanı","İşletim Sistemleri","Algoritma ve Prog.Giriş","Nesne Yön.Prg.");`
- `echo "<form action='' method=POST>";`
- `echo "<select name='dersler[]' size='20' multiple>";`
- `foreach ($dersler as $value) {`
- `if (!empty($value))`
- `echo "<option value='$value'>".$value."</option> ";`
- `}`
- `echo "</select>";`
- `echo "<input type='submit' value='SİL' name='SIL'>";`
- `echo "</form>";`
- `}`
- `?>`

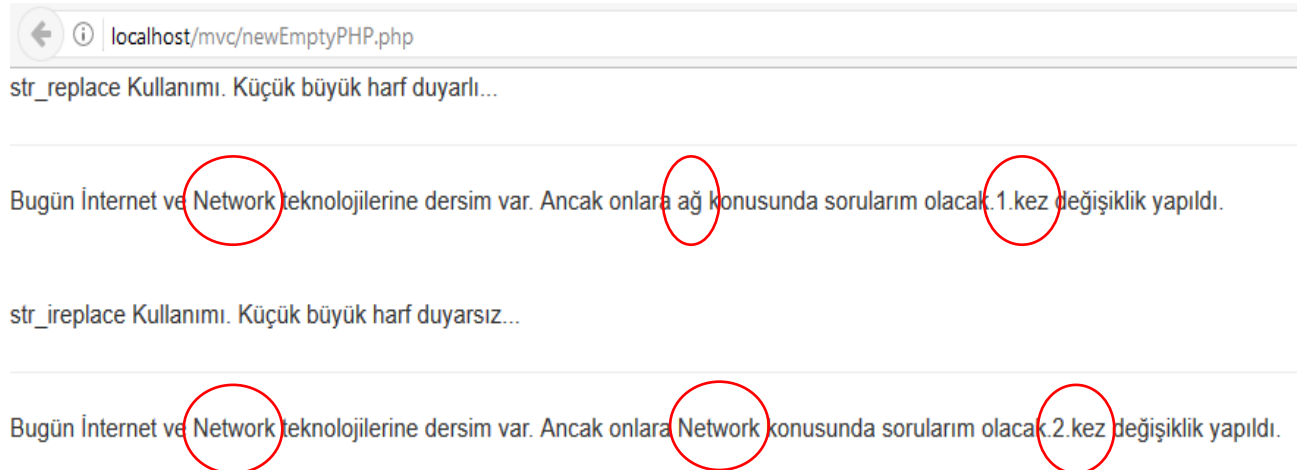
str_replace()

- Bu fonksiyon ile bir string içerisinde istenilen bir metni, bir başka metin ile değiştirebilirsiniz.
- **str_replace()** fonksiyonu büyük küçük harflere duyarlıdır (*case-sensitive*), büyük küçük harflere duyarsız (*case-insensitive*) fonksiyon için **str_ireplace()** fonksiyonunu kullanmalısınız. 2 fonksiyonun da kullanımları aynıdır. 4 parametre alır ancak genel olarak 3 parametre ile kullanılır.
- Kullanım Şekli
 - str_replace(find,replace,string,count)
 - str_ireplace(find,replace,string,count)
- **find:** Zorunludur. String ya da değişmesi istenen metin.
- **replace:** Zorunludur. Stringde ya da dizide değişecek yeni metin.
- **string:** Zorunludur. Üzerinde işlem yapılacak string veya dizi.
- **count:** İsteğe bağlıdır. Kaç kere değiştirme işlemi uygulandığını döndürür.
- **Not:** **str_replace()** fonksiyonu orijinal stringi veya diziyi bozmaz.

Str_replace() Örneği

- <?php
- \$metin="Bugün İnternet ve Ağ teknolojilerine dersim var. Ancak onlara ağ konusunda sorularım olacak.";
- \$eski_metin="Ağ";
- \$yeni_metin="Network";
- echo "str_replace Kullanımı. Küçük büyük harf duyarlı...";
- echo "<hr>";
- echo
str_replace(\$eski_metin,\$yeni_metin,\$metin,\$sayac);
- echo \$sayac.".kez değişiklik yapıldı.";
- echo "

";
- echo "str_ireplace Kullanımı. Küçük büyük harf duyarsız...";
- echo "<hr>";
- echo
str_ireplace(\$eski_metin,\$yeni_metin,\$metin,\$sayac);
- echo \$sayac.".kez değişiklik yapıldı.";
- ?>



Dizilerle Str_replace Kullanımı

- Birden fazla alanı değiştirmek istiyorsak bu sistemi kullanırız.
- \$degisecekyazi = 'Artık php öğrenmek çok zevkli';
- \$bul = array('php', 'çok', 'zevkli');
- \$degistir = array('html', 'fazla', 'kolay');
- \$degisen = str_replace(\$bul, \$degistir, \$degisecekyazi);
- echo \$degisen;
- // Ekrana
- // Artık html fazla kolay
- // yazısı çıkacak

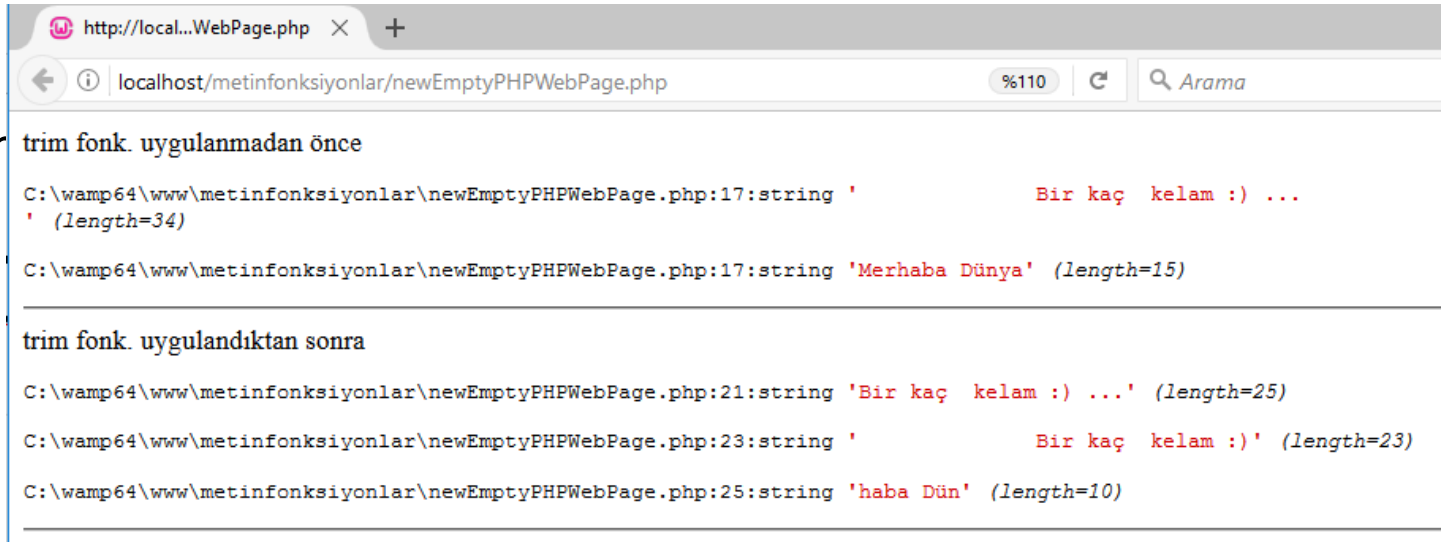
trim(\$string), ltrim(\$string), rtrim(\$string)

- Bu fonksiyonlar verilen string ifadenin başındaki ve sonundaki boşlukları ve belirteceğimiz karakterleri atmak için kullanılır.
- Form aracılığı ile gelen verilerde boşluklar olması mümkün ve bu bir sorun teşkil eder. Bunu önlemek için trim, ltrim ve rtrim çok sık kullanılan fonksiyonlardır.
- trim; string ifadenin hem başındaki hem sonundaki,
- ltrim başındaki,
- rtrim sonundaki boşlukları atar.
- Kullanım şekilleri;

trim	ltrim	rtrim
trim(string)	ltrim(string)	rtrim(string)
trim(string,charlist)	ltrim(string,charlist)	rtrim(string,charlist)

Trim(), ltrim() ve rtrim() Örnekleri

- <?php
- echo "trim fonk. uygulanmadan önce";
- \$metin=" \t\tBir kaç kelam :) ... \n";
- \$selam="Merhaba Dünya";
- var_dump(\$metin,\$selam);
- echo "<hr>";
- echo "trim fonk. uygulandıktan sonra";
- \$kırpık=trim(\$metin);
- var_dump(\$kırpık);
- \$kırpık=trim(\$metin," \n. ");
- var_dump(\$kırpık);
- echo "<hr>";
- ?>



```

http://local...WebPage.php X +
localhost/metinfonksiyonlar/newEmptyPHPWebPage.php %110 Arama

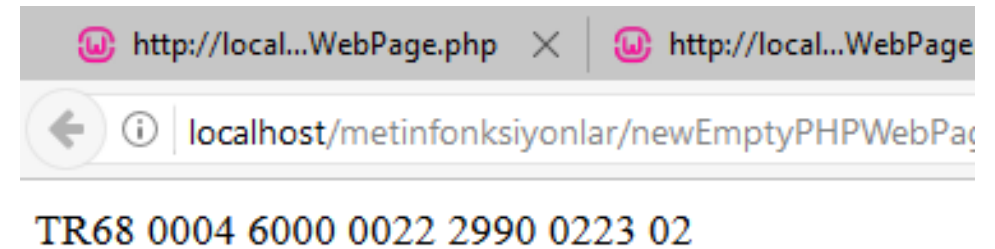
trim fonk. uygulanmadan önce
C:\wamp64\www\metinfonksiyonlar\newEmptyPHPWebPage.php:17:string '      Bir kaç kelam :) ...' (length=34)
C:\wamp64\www\metinfonksiyonlar\newEmptyPHPWebPage.php:17:string 'Merhaba Dünya' (length=15)

trim fonk. uygulandıktan sonra
C:\wamp64\www\metinfonksiyonlar\newEmptyPHPWebPage.php:21:string 'Bir kaç kelam :) ...' (length=25)
C:\wamp64\www\metinfonksiyonlar\newEmptyPHPWebPage.php:23:string '      Bir kaç kelam :)' (length=23)
C:\wamp64\www\metinfonksiyonlar\newEmptyPHPWebPage.php:25:string 'haba Dün' (length=10)

```

str_split() fonksiyonu

- Verilen metni verilen sayı kadar karakter gruplarına bölerek bir diziye dönüştürür. Eğer sayı belirtilmezse, metni harf harf bölerek bir diziye atar.
- Kullanım şekli;
 - `str_split($metin, $sayi);`
- `<?php`
- ```
$iban="TR680004600000222990022302";
```
- `$grup=str_split($iban,4);`
- `foreach($grup as $deger)`
- `echo $deger," ";`
- `?>`





# Strpos() fonksiyonu

- **strpos()** fonksiyonu bir stringin başka bir string içerisinde bulunduğu ilk konumu döndürür.
- Strpos kullanıldı. Büyük küçük harf duyarlı
  - \$email="senol@dpu.edu.tr";
  - \$indis=strpos(\$email,"@");
  - \$str= substr(\$email,0, \$indis);
  - echo \$indis." ".\$str;

5 senol
- Başka bir örnek. Stripos kullanıldı. Büyük küçük harf duyarsız
  - \$email="senol@dpu.edu.tr";
  - \$indis=stripes(\$email,"TR");
  - \$str= substr(\$email,0, \$indis);
  - echo \$indis." ".\$str;

14 senol@dpu.edu.

# str\_pad fonksiyonu

- Bir dizgeyi belli bir uzunlukta diğer dizgeyle doldurur.
- Kullanımı: **str\_pad** (girdi , dolgulu\_uzunluk [, dolgu\_dizgesi = " " [, dolgu\_türü = *STR\_PAD\_RIGHT* ]])
- *dolgu\_türü*: Seçimlik *dolgu\_türü* olarak **STR\_PAD\_RIGHT** (sağ), **STR\_PAD\_LEFT** (sol) veya **STR\_PAD\_BOTH** sabiti (her iki taraf) belirtilebilir. *dolgu\_türü* belirtilmezse **STR\_PAD\_RIGHT** değeri öntanımlıdır.

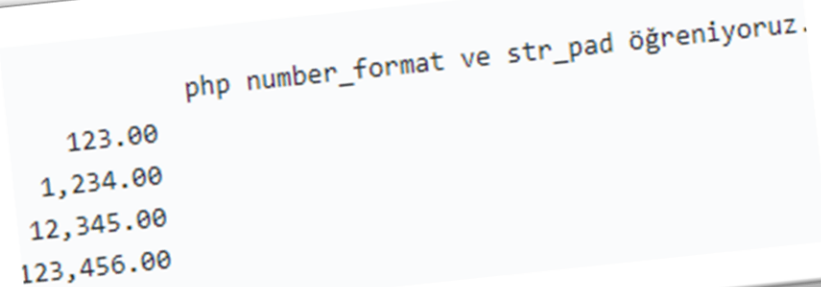
```
echo "php str_pad öğreniyoruz.
";
//tüm değerler sağa hizalı yazdırılır...
$tutarlar=[125.50,2578,1457800,129878,99];
foreach ($tutarlar as $key => $value) {
 echo str_pad($value, 10, " ", STR_PAD_LEFT). "
";
}
//number_format ve str_pad birlikte kullanımı
foreach ($tutarlar as $key => $value) {
 echo str_pad(number_format($value,2), ,15, " ", STR_PAD_LEFT);
 echo "
";
}
```

```
php str_pad öğreniyoruz.
str_pad kullanımı
125.5
2578
1457800
129878
99
str_pad ve numbe_format birlikte kullanımı
125.50
2,578.00
1,457,800.00
129,878.00
99.00
```

# number\_format()

- Sayıyı binlik bölümlere ayırır.
- Kullanımı:
- **number\_format** (sayı ,  
ondalık\_hane, ondalık\_ayracı,  
binlik\_ayracı )
- ```
<?php  
$sayi = 1234.56;  
$strformat = number_format($  
sayi, 2, ',', '.');  
echo $strformat;
```
- ```
// 1.234,56
?>
```

```
<?php
echo "php number_format ve str_pad öğreniyoruz.
";
echo str_pad(number_format(123 ,2), 10, " ",
STR_PAD_LEFT)."
";
echo str_pad(number_format(1234 ,2), 10, " ",
STR_PAD_LEFT)."
";
echo str_pad(number_format(12345 ,2), 10, " ",
STR_PAD_LEFT) ."
";
echo str_pad(number_format(123456 ,2), 10, " ",
STR_PAD_LEFT) ."
";
```



```
php number_format ve str_pad öğreniyoruz.
123.00
1,234.00
12,345.00
123,456.00
```

# Addslashes() kullanımı

- addslashes fonksiyonu kendisine parametre olarak verdiğimiz ifadedeki özel karakterlerin önüne ters bölü(\) ya da ters slash dediğimiz sembolü koyar.
- **Peki bu ne işimize yarayacak?**
- İçerisinde özel ifadeler(tek tırnak işareti gibi) barınan bir metin veri tabanı işlemleri yaparken bizim için hata üretebilir.
- <?php
- \$metin = "Tavşanlı'da hava nasıl?";
- echo addslashes(\$metin);
- //Çıktısı: Tavşanlı\'da hava nasıl?
- ?>

# stripslashes() kullanımı

- stripslashes() fonksiyonu bir string ifadedeki ters bölü (\) işaretini temizlemek için kullanılır.
- \$veri = "PHP\'nin yeni versiyonu!";
- echo stripslashes(\$veri);
- Ekran Çıktısı : PHP'nin yeni versiyonu!

# Php ile tarih ve saat işlemleri

- Tarih verilerini kullanmaya başlamadan önce yapmamız gereken bir takım ayarlar var. Bunlardan ilki doğru saat ve tarihe ulaşabilmek için hangi il, ülke ya da zaman diliminde olduğumuzu sunucuya belirtmek.
- Bunun için “**date\_default\_timezone\_set**” fonksiyonu ile bizim zaman dilimimizin belirtecini giriyoruz.
  - `date_default_timezone_set('Europe/Istanbul');` gibi.
- `Date()` fonksiyonunu, Tarih ve saati belirttiğimiz formatta elde etmek için kullanırız. Kullanılan fonksiyonların bazıları sonraki slaytta listelenmiştir.
- Kullanım şekli ;
  - `Date(format);`

# Date() fonksiyonu

- Bu fonksiyon bize tarihi, belirttiğimiz formatta gösterir. Aşağıda kullanılabilecek format karakterleri bulunmaktadır.

Harf	Açıklaması	Örnek
d	İki haneli şekilde günü verir	27
m	İki haneli şekilde ayı verir	07
Y	Dört haneli şekilde yılı verir	2013
H	İki haneli ve 24 saat formatında saati verir	19
i	İki haneli şekilde dakikayı verir	38
s	İki haneli şekilde saniyeyi verir	59
l	(küçük L) Haftanın gününün ingilizce adını verir	Monday
F	Ayın ingilizce adını verir	July
w	Haftanın kaçınıcı gününde olduğumuzu verir.	3

- Format için kullanacağımız diğer karakterler, “/”, “.”, yada “-” gibi isteğinize ve beğeninize göre eklenebilir.

# Date fonksiyonu örnekleri

- `date_default_timezone_set('Europe/Istanbul');`//zaman dilimini istanbul yaptık.
- `echo date("d")."<br>;"`//gün numarasını verir.
- `echo date("m")."<br>;"`//ay numarasını verir.
- `echo date("D")."<br>;"`//gün adını ingilizce kısa olarak yazar.
- `echo date("M")."<br>;"`//ay adını ingilizce kısa olarak yazar.
- `echo date("Y")."<br>;"`//Yılı verir.
- `echo date("l")."<br>;"`//gün adını ingilizce uzun olarak yazar.
- `echo date("F")."<br>;"`//ay adını ingilizce uzun olarak yazar.
- `echo date("d-m-Y")."<br>;"`//tarihi gün ay ve yıl formatında yazar
- `echo date("H:i:s")."<br>;"`//saati 24 saatlik dilim şeklinde dakika ve saniye ile yazar.
- `echo date("h:i:s")."<br>;"`//saati 12 saatlik dilim şeklinde dakika ve saniye ile yazar.
- `echo date("d-m-Y H:i:s")."<br>;"`//tarih ve saati yazar



# Date ile ilgili Örnekler...

- `echo "Yılın ".date('W')."nci haftasındayız <br>";`
- `echo "haftanın ".date('w')."nci günündeyiz<br>";`
- `echo "haftanın günü ".date('l')."<br>";`
- `echo "Yılın ".date('z')."nci günündeyiz<br>";`
- `echo "Ayın ".date('t')."gün sürecek<br>";`
- `echo "Ayın ".date('d')."günündeyiz <br>";`

# Zaman farkını gün ay yıl saat dakika ve saniye olarak hesaplar...

- `$bugun = time();`
- `$gecmis = mktime(15, 35, 59, 02, 25, 2017);`
- `$fark = $bugun - $gecmis;`
- `echo 'Bugün: ' . $bugun."---".date("d m Y H:i:s")."<br>";`
- `echo 'Geçmiş: ' . $gecmis."---".date("d m Y H:i:s",$gecmis)."<br>";`
- `echo 'Saniye farkı: ' . $fark."<br>";`
- `$dakika = $fark / 60;`
- `$saniye_farki = floor($fark - (floor($dakika) * 60));`
- `$saat = $dakika / 60;`
- `$dakika_farki = floor($dakika - (floor($saat) * 60));`
- `$gun = $saat / 24;`
- `$saat_farki = floor($saat - (floor($gun) * 24));`
- `$yil = floor($gun/365);`
- `$gun_farki = floor($gun - (floor($yil) * 365));`
- `echo $yil . ' yıl'."<br>";`
- `echo $gun_farki . ' gün'."<br>";`
- `echo $saat_farki . ' saat'."<br>";`
- `echo $dakika_farki . ' dakika'."<br>";`
- `echo $saniye_farki . ' saniye'."<br>";`

# İki tarih arasındaki farkı bulma

- `$ilktarih="22.02.2017";`//bu ilk kayıt tarihi olsun
- `$sontarih="30.05.2017";`//buda şu anki tarih olsun
- `$ilktarihstr=strtotime($ilktarih);` //ilk tarihi strtotime ile çeviriyorum
- `$sontarihstr=strtotime($sontarih);` //son tarihi strtotime ile çeviriyorum
- `$fark=round(($sontarihstr-$ilktarihstr)/86400);`//sondan ilki çıkarıp 86400 e bölüyoruz bu bize günü verecek
- `echo "Fark :".$fark." gün <br />";`
- `echo "<hr>";`

Fark :97 gün

- Not: bu örnekteki 86400 rakamı saniyeyi dakikaya dakikayı saate ve saati 24'e çarparsak gün sayısını verir.  $60*60*24 = 86400$ . Fark değişkeni saniye cinsinden olduğu için...

# Tarih işlevleri

- `$ys=strtotime('+1 day -1 hour');`//şu anki tarihe 1 gün ekleyip 1 saat çıkarır.
- `echo date('d.m.Y H:i:s', $ys);`
- `$ys=strtotime('+2 month +3 week');`//şu anki tarihe 2 ay ekleyip 3 hafta çıkarır.
- `echo date('d.m.Y H:i:s', $ys);`
- `$ys=strtotime('+1 year +2 month +3 week');`//şu anki tarihe 1 yıl 2 ay ve 3 hafta ekler
- `echo date('d.m.Y H:i:s', $ys);`

Tarih 22 Şubat 2017

23.02.2017 21:26:06 +1 gün, -1 saat

13.05.2017 22:26:06 +2 ay, +3 hafta

13.05.2018 22:26:06 +1 yıl, +2ay +3 hafta

# Setlocale() ve Strftime() Fonksiyonu

- **Setlocale ()** işlevi yerel ayar bilgilerini belirler. Yerel ayar bilgileri, dil, para, zaman ve bir coğrafi alana özgü diğer bilgilerdir.  
İpucu: Yerel ayar bilgileri, setlocale (LC\_ALL, NULL) ile sistem varsayılanına ayarlanabilir.
- **Strftime()** fonksiyonu ile gün ve ay isimlerini istediğimiz dilde çekebileceğiz. Peki bu PHP hangi dilde yazdırmak istediğimizi nasıl anlar. PHP'nin hangi dilde yazdırmamızı istediğini kodlarımızın başında bir tanımlama yaparak belirtiriz.
- Bu tanımla şu Şekildedir;
- `setlocale(LC_ALL,"turkish");`

# Ay ve günleri Türkçe yazdırma

- `setlocale(LC_ALL,"turkish");`
- `Echo "Bugün günlerden ".strftime("%A")."<br>";`
- `Echo "Saat ".strftime("%H:%M:%S")."<br>";`
- `Echo "Tarih ".strftime("%d.%B.%Y")."<br>";`

Harf	Açıklaması	Örnek
%A	Haftanın gününün adını verir.Uzun	Pazartesi
%B	Ayın adını verir.Uzun	Şubat
%a	Haftanın gününün adını verir.Kısa	Pzt
%b	Ayın adını verir.Kısa	Şub
%d	Günü 2 haneli yazar	05
%j	Yılın kaçıncı günü olduğunu yazar	001 den 366'ya

Bugün günlerden Çarşamba  
Saat 21:45:30  
Tarih 22 Şubat 2017

# Tarih bilgilerini Türkçe almak

- Önce **setlocale** fonksiyonu ile zaman işlemlerinde kullanacağımız dilimizi ifade ediyoruz. Sonrasında **strftime** fonksiyonu ile ifade ettiğimiz dilde zaman çıktısını alıyoruz.
  - `<?php`
    - `setlocale(LC_TIME,'turkish');`
    - `echo iconv('latin5','utf-8',strftime("%j %d %b %A %Y"))."<br>";`
    - `echo iconv('','utf-8',strftime(' %j %d %B %Y %A ', strtotime("now")))."<br>";`
    - `echo iconv('','utf-8',strftime('%j %d %B %Y %A ',strtotime("12-05-2017")))."<br>";`
  - `?>`

```
051 20 Şub Pzt 2017
051 20 Şubat 2017 Pazartesi
132 12 Mayıs 2017 Cuma
```

iconv fonksiyonu ile, karakter setini UTF-8 olarak ayarladık.

Harf	Açıklaması	Örnek
%A	Haftanın gününün adını verir.Uzun	Pazartesi
%B	Ayın adını verir.Uzun	Şubat
%a	Haftanın gününün adını verir.Kısa	Pzt
%b	Ayın adını verir.Kısa	Şub
%d	Günü 2 haneli yazar	05
%j	Yılın kaçınıcı günü olduğunu yazar	001 den 366'ya

# Dizi kullanarak Tarih bilgilerini Türkçe almak

- Aşağıdaki örnek, `setlocale(LC_TIME,'turkish');` ve `iconv` gibi fonksiyonları kullanmaya gerek kalmadan gün ve ay isimlerini Türkçe oluşturmaya yarar.
  - `$gunler=array("", "pazartesi", "salı", "çarşamba", "perşembe", "cuma", "cumartesi", "pazar");`
- `echo date("w").".gün :". $gunler[date("w")]."<br>;`
  - `$aylar = array("", 'Ocak', 'Şubat', 'Mart', 'Nisan', 'Mayıs', 'Haziran', 'Temmuz', 'Ağustos', 'Eylül', 'Ekim', 'Kasım', 'Aralık');`
- `echo date("m").".ay :". $aylar[floor(date("m"))]."<br>;`

```
1.gün :pazartesi
02.ay :Şubat
```



# İki tarih arasındaki çalışma günlerinin sayısını bulan php kodları 33

- `$ise_baslama_trh=strtotime("25-02-2017");`
- `$bugunku_trh=strtotime(date("d-m-Y",time()));`
- `setlocale(LC_ALL,"turkish");`
- `echo "ise_baslama_trh : ".$ise_baslama_trh."<br>";`
- `echo "bugunku_trh : ".$bugunku_trh."<br>";`
- `$is_gunleri=0; $gun_index=null;`
- `for ($index = $ise_baslama_trh; $index <= $bugunku_trh; $index+=86400)`
- `{`
- `$gun_index=date("w",$index);`
- `(($gun_index==6) || ($gun_index==0))?null:$is_gunleri++;`
- `echo date("d-m-Y",$index). "---". $gun_index. "----". iconv("", "utf-8", strftime("%A", $index)). "<br>";`
- `}`
- `echo "toplam iş günü : ".$is_gunleri;`
- `?>`

# Tarihleri kıyaslama(karşılaştırma) javascript

- <!DOCTYPE html>
- <html>
- <head>
- <script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
- </head>
- <body>
- <form>
- <input type="button" id="btn" name="btn" value="Ders Kayıt">
- <input type="text" id="lbl" name="lbl" value="" visible="false" size="40" >
- </form>
- </body>
- </html>
- <?php
- if (!\$\_POST)
- {
- if (strtotime("01/03/2017 11:00") < strtotime("01/03/2016 11:00"))
- {
- ?>
- <script>
- document.getElementById("btn").style.display='none';
- document.getElementById("lbl").value="Kayıt Tarihi henüz başlamadı...";
- </script>
- <?php
- }
- else
- {
- ?>
- <script>
- document.getElementById("btn").style.display='block';
- document.getElementById("lbl").value="Kayıt Tarihi BAŞLAMISHTIR...";
- </script>
- <?php
- }
- }

# Tarihleri kıyaslama(karşılaştırma) form elementleri 35

- <!DOCTYPE html>
- <html>
- <head>
- <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
- </head>
- <body>
- <form action="" method="POST">
- Kayıt Tarihi : <input type="datetime-local" id="kayit\_tarihi" name="kayit\_tarihi"></input><br>
- Şu anki Tarih : <input type="text" id="gecerli\_tarih" name="gecerli\_tarih" readonly value="<?php echo date("d.m.Y");?>"></input><br>
- <input type="submit" id="kontrol" name="kontrol" value="Kontrol"><br>
- <input type="button" id="btn" name="btn" value="Ders Kayıt" ><br>
- <input type="text" id="lbl" name="lbl" value="" size="40" ><br>
- </form>
- </body>
- </html>
- <?php
- if (isset(\$\_POST["kontrol"]))
- {
- if (strtotime(\$\_POST["kayit\_tarihi"]) > strtotime(\$\_POST["gecerli\_tarih"]))
- {
- ?>
- <script>
- document.getElementById("btn").style.display='none';
- document.getElementById("lbl").value="Kayıt Tarihi henüz başlamadı...";
- </script>
- <?php
- }
- else
- {
- ?>
- <script>
- document.getElementById("btn").style.display='block';
- document.getElementById("lbl").value="Kayıt Tarihi BAŞLAMISITIR...";
- </script>
- <?php
- }
- }

# Tarihleri kıyaslama(karşılaştırma) jquery

- <!DOCTYPE html>
- <html>
- <head>
- <script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
- </head>
- <body>
- <form>
- <input type="button" id="btn" name="btn" value="Ders Kayıt">
- <input type="text" id="lbl" name="lbl" value="" visible="false" size="40" >
- </form>
- </body>
- </html>
- <?php
- if (!\$\_POST)
- {
- if (strtotime("01/03/2017 11:00") > strtotime("01/03/2016 11:00"))
- {
- ?>
- <script>
- \$(document).ready(function(){
- \$("#btn").hide();
- \$("#lbl").val("Kayıt Tarihi henüz başlamadı...");
- });
- </script>
- <?php
- }
- }
- }

# Php Veri tabanı

# Mysql veritipleri

- **Metin alanları**

- **CHAR ()** 0 ile 255 karakter arası değer verebiliriz.
- **VARCHAR ()** 0 ile 255 karakter arası değer verebiliriz.
- **TINYTEXT** en fazla 255 karakter
- **TEXT** en fazla 65535 karakter
- **MEDIUMTEXT** en fazla 16777215 karakter
- **MEDIUMBLOB** en fazla 16777215 karakter
- **LONGTEXT** en fazla 4294967295 karakter

- **Zaman Alanları**

- **DATE** YYYY-MM-DD
- **DATETIME** YYYY-MM-DD HH:MM:SS

- **Sayısal Alanlar**

- **TINYINT ()** -128 ile 127 arası normal (0 ile 255 arası UNSIGNED)
- **SMALLINT ()** -32768 ile 32767 arası normal (0 ile 65535 arası)
- **MEDIUMINT ()** -8388608 ile 8388607 arası normal (0 ile 16777215 arası)
- **INT ()** -2147483648 ile 2147483647 arası normal (0 ile 4294967295 arası)
- **BIGINT ()** -9223372036854775808 ile 9223372036854775807 arası normal (0 ile 18446744073709551615 arası)
- **FLOAT** virgüllü sayı
- **DOUBLE (,)** virgüllü büyük sayı
- **DECIMAL (,)** string olarak virgüllü sayı

# PDO Nedir?

- **PDO (PHP Data Objects / PHP Veri Objeleri)** özetle; hafif ve tutarlı bir şekilde **veritabanına erişimi sağlayan bir arayüz**. Adından da anlayacağınız üzere **“Object Oriented Programming”** arayüzüne sahip, onlarca veritabanı sürücüsü destekliyor.
- PDO arayüzü tanımı bulunan her veritabanı sürücüsü, veritabanına özgü özellikleri sıradan eklenti işlevleri olarak ifade edebilir. Yalnız dikkat edin, PDO eklentisinin kendisini her veritabanı işlevi için kullanamazsınız; bir veritabanı sunucusuna erişmek için veritabanına özgü PDO sürücüsünü kullanmalısınız.
- PDO bir veri erişimli soyutlama katmanı sağlar, yani, kullandığınız veritabanından bağımsız olarak sorguları çalıştırmak ve sonuçları döndürmek için aynı işlevleri kullanabilirsiniz.

# Pdo'yu sunucuda etkinleştirme

- PDO yu kullanabilmek için php.ini dosyasındaki PDO'yu etkinleştirmelisiniz.
- Bununla birlikte veri tabanına has olan PDO sürücüsünü de etkinleştirmelisiniz.
- Yoğun biçimde MySql kullanıldığı için biz MySql eklentisini etkinleştireceğiz, bunun için php.ini ye şu satırları ekleyebilirsiniz.
- **[PHP\_PDO]**  
**extension=php\_pdo.dll**  
**[PHP\_MYSQL]**  
**extension=php\_mysql.dll**



# PDO Veritabanı Bağlantısı

- Genel olarak bağlantı olaylarını bir PDO sınıfını tanımlarken bir **DSN** yani “Data Source Name” ile belirtiyoruz. Hangi veritabanı sürücüsüne bağlanacağımızı ve bilgilerimizi DSN ile ifade ediyoruz. Diğer iki parametrede ise veritabanı kullanıcı adı ve şifremizi giriyoruz.
  - `$db = new PDO('mysql:host=localhost;dbname=test', $user, $pass);`
- Yada
  - `$dsn = 'mysql:host=localhost;dbname=test';`
  - `$user = 'dbuser';`
  - `$password = 'mypassword';`
  - `try {`
  - `$conn = new PDO($dsn, $user, $password);`
  - `} catch (PDOException $e) {`
  - `echo 'Connection failed: ' . $e->getMessage();`
  - `}`
- Elbette PDO ile bağlanırken try ~ catch yapısını kullanmak zorunda değildik ancak böyle bir kullanım çok daha kullanışlıdır. Çünkü PDO’nun hatalar için bir exception’ı var ve daha mantıklı hata mesajları üretiyor.
- **Not:** PDO sadece mysql’i değil bir çok veritabanı yapısını destekler

# PDO da karakter setini ayarlama;

- Bağlantıdan sonra karakter setini ayarlamak yerinde olacaktır. Bunun için;
  - `$conn = new PDO("mysql:host=localhost;dbname=test;charset=utf8", "root", "123456");`
- Yada
  - `$conn ->query("SET CHARACTER SET utf8");`
- Yada
  - `$conn >setAttribute(PDO::MYSQL_ATTR_INIT_COMMAND, 'SET NAMES utf8');`
- Yöntemlerinden birini kullanabiliriz.

# Hata/exception denetimini açmak

- **PDO::ATTR\_ERRMODE:** Hata yakalamayı sağlar.
- **PDO::ERRMODE\_EXCEPTION:** Hata içeriği hakkındaki bilgiyi gönderir.
- Kullanım Şekli;
  - `$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);`

# Bağlantının kapatılması

- Veritabanı bağlantısının başarıyla kurulmasıyla kodlarınıza PDO sınıfının bir örneği döner.
- PDO nesnesinin ömrü boyunca bağlantı etkin kalır.
- Bağlantıyı kapatmak için, nesneye gönderimli her şeyin silindiğinden emin olarak nesneyi yok etmeniz gerekir;
- bunu nesneyi tutan değişkene NULL atayarak yapabilirsiniz.
- Bunu siz yapmazsanız, PHP betiğiniz sonlandığında bunu sizin yerinize yapacaktır.

- `<?php`
- `try {`
- `$conn=new PDO('mysql:host=localhost;dbname=bilgisayarprg','root','');`
  - `//hata denetim modunu açıyoruz.`
- `$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);`
  - `// burada bağlantıyı kullanıyoruz.Örnek. Select işlemi`
- `$sonuc=$conn->query("select * from users",PDO::FETCH_ASSOC);`
- `Foreach ($sonuc as $value) { echo $value["userid"];}`
  - `// İşimiz bittiğine göre bağlantıyı kapatabiliriz.`
- `} catch (PDOException $exc) {`
- `echo $exc->getMessage();`
- `}`
- `finally {`
- `$conn=null;`
- `}`

# Tablodaki verileri listeleme 1.yöntem

- PDO::FETCH\_ASSOC: Bu yöntemle veriler sütun adlarına göre gelir.
- PDO::FETCH\_NUM : Bu yöntemde veriler sıra numaralarına göre gelir.Sütun ismi yer almaz.
- \$query = \$con->query("SELECT \* FROM projeler",PDO::FETCH\_ASSOC);
- foreach (\$query as \$value) {
- echo \$value["proje\_adi"]."--".\$value["proje\_detay"]."<br>";
- }

# Tablodaki verileri listeleme 2.yöntem

- Burada ilkinden farklı olarak PDO::FETCH\_ASSOC yerine FETCH\_NUM kullanıldı. Burada tablodan gelen sütun adları değil indis numaraları oldu. Foreach döngüsü içinde verilere erişirken \$value[1] şeklinde sütun adı değil indis no kullanıldı.
- \$query = \$con->query("SELECT \* FROM projeler",PDO::FETCH\_NUM);
- foreach (\$query as \$value) {
- echo \$value[1]."--".\$value[2]."<br>";
- }

# Tablodaki verileri listeleme 3.yöntem

46

## Yöntem 1

- \$conn=new PDO('mysql:host=localhost;dbname=bilgisayarprg','root','');
- \$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);
- foreach (\$conn->query("SELECT \* FROM projeler") as \$row) {
- echo \$row["proje\_adi"]."--  
".\$row["proje\_detay"]."<br>";
- }

## Yöntem 2

- \$conn=new PDO('mysql:host=localhost;dbname=bilgisayarprg','root','');
- \$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);
- foreach (\$conn->query("SELECT \* FROM projeler",PDO::FETCH\_ASSOC) as \$row) {
- echo \$row["proje\_adi"]."--  
".\$row["proje\_detay"]."<br>";
- }

# Tablodaki verileri listeleme 4.yöntem

- `$query = $con->query("SELECT * FROM projeler");`
- `$row=$query->fetchAll();`
- `foreach ($row as $value) {`
- `echo $value["proje_adi"]."--".$value["proje_detay"]."<br>";`
- `}`
- Buradaki örnekte FETCHALL kullanılmıştır. Bu fonksiyon sorgudan gelecek tüm satırları bir kerede getirir. Çok satır dönecek sorgularda FetchAll() yerine Fetch() kullanın. Fetch() fonksiyonu verileri satır satır getirir.

# Tablodan satır satır veri çekmek (FETCH)

- Fetch metodu sorgudan bir sonraki satırı getirir. Genellikle çok fazla satır dönmesini beklediğimiz sorgularda kullanılır.
- Fetch kullandığınızda foreach yerine while döngüsü kullanın.
  - `While($row=$sonuc->fetch(PDO::FETCH_ASSOC)) { }`
- Fetch ile gelen sonuçları elde etmek için ikinci bir farklı parametre kullanılır. Bu parametreler;
  - PDO::FETCH\_ASSOC
  - PDO::FETCH\_NUM
  - PDO::FETCH\_BOUND
  - Gibi
- Kullanım şekli;
  - `Fetch(PDO::FETCH_ASSOC)` //verileri sadece sütun adları ile çeker
  - `Fetch(PDO::FETCH_NUM)` //verileri sadece indis no ile çeker
  - `Fetch(PDO::FETCH_BOTH)` //verileri sütun adları ve indis noları ile çeker



# Tablodan tek bir satır getirme örneği

```

1. <?php
2. $conn = 'mysql:host=localhost;dbname=tavsanli;';
3. $user = 'senol'; $password = 'senol';
4. try { $db = new PDO($conn, $user, $password);
5. $db->query("SET CHARACTER SET utf8");
6. $id = 1;
7. $query = $db->query("SELECT * FROM ogrenci WHERE ono=1"); //1 nolu
 öğrenciye ait verileri select eder.
8. $query = $query -> fetch(PDO::FETCH_ASSOC); // PDO::FETCH_ASSOC: veriyi
 sütun ismi ile çeker.
9. if ($query)
10. { echo "Aranan Öğrenci adı :". $query['adi']."
";
11. echo "Aranan Öğrenci Soyadı :". $query['soyadi']."
";
12. echo "Aranan Öğrenci Doğum Yeri :". $query['dyeri']."
";
13. }
14. }
15. catch (PDOException $e) { echo 'Connection Hatası: ' . $e->getMessage();
 }

```

22.05.2022

Bu örnekte 1 nolu öğrenciye ait bilgiler istenmektedir. Ancak 1 nolu öğrenci yoksa ne olacak? İşte sol taraftaki kodlar; 1 nolu öğrenci yoksa hiçbir işlem yapmayacaktır. Kodlarımızın 16. satırından sonra aşağıdaki kodları ekleyelim.

```

else
{
 echo "Aradığınız numaraya ait bir
 öğrenci kayıtlı değil...";
}

```

10 satırda PDO::FETCH\_ASSOC yerine PDO::FETCH\_NUM kullanmış olsaydık 13, 14 ve 15. satırları aşağıdaki gibi yazmak gerekecekti; Yani sütunlara adları ile değil indis noları ile erişebilecektik.

```

echo "Aranan Öğrenci adı :". $query[0]."
";
 echo "Aranan Öğrenci Soyadı :".
 $query[1]."
";
 echo "Aranan Öğrenci Doğum Yeri :".
 $query[2]."
";

```

```

1. <?php ob_start(); ?>
2. <html>
3. <head> <meta charset="UTF-8"> <title></title> </head>
4. <body> <table> <form method="post">
5. <tr><td> Kullanıcı Adı:</td><td><input type="text" name="userid" value="<?php
 echo @$_COOKIE["c_userid"];?>"></td></tr>
6. <tr><td> Şifre </td><td><input type="text" name="password" value="<?php echo
 @$_COOKIE["c_password"];?>"></td></tr>
7. <tr><td> </td><td><input type="submit" name="giris" value="giris">
 </td></tr>
8. </form> </table>
9. <?php
10. if (isset($_POST["giris"]))
11. { Try {
12. $conn= new
 pdo("mysql:host=localhost;dbname=bilgisayarprg;charset=utf8","root","");
13. $sql="select * from users where userid='".$_$_POST["userid"]."' and
 password='".$_$_POST["password"]."'";
14. $sonuc=$conn->query($sql);
15. $sonuc=$sonuc->fetch(PDO::FETCH_ASSOC);
16. if ($sonuc){
17. setcookie("c_userid", $_POST["userid"], time()+86400);
18. setcookie("c_password", $_POST["password"], time()+86400);
19. Echo "bilgiler doğru";
20. }
21. else { echo "kullanıcı adı yada şifre yanlış"; }
22. } catch (PDOException $ex) { echo $ex->getMessage(); } }
 ?>
23. </body>
24. </html>

```

22.05.2022

- Aşağıdaki kodlar 13 ile 21 satırlar arasındaki kodların alternatifi olarak kullanılabilir
- \$conn= new  
pdo("mysql:host=localhost;dbname=bilgisayarprg;charset=utf8","root","");
- \$sql="select \* from users where userid=? and password=?";
- \$sonuc=\$conn->prepare(\$sql);
- \$sonuc->execute(array(\$\_POST["userid"],\$\_POST["password"]));
- \$row=\$sonuc->fetch(PDO::FETCH\_ASSOC);
- if (\$sonuc->rowCount()>0){
- if(isset(\$\_POST["hatirla"]))
- { setcookie("c\_userid", \$\_POST["userid"], time()+86400);
- setcookie("c\_password", \$\_POST["password"],  
time()+86400); }
- else
- { setcookie("c\_userid", "", time()-86400);
- setcookie("c\_password", "", time()-86400); }
- }
- else
- {
- echo "kullanıcı adı yada şifre yanlış";
- }

Bu örnekte kullanılan cookie(çerez)leri anlamak için 49-55 arası slaytları inceleyin.

# Cookie kullanımı

- Girilen sitelerin tarayıcılar(browserlar) üzerinde bıraktığı izlere **cookie**, Türkçesiyle de **çerez** denir. Bu izler siteye tekrar ziyaret edildiğinde yeniden gönderilerek bazı şeylerin yeniden hatırlanmasını sağlar.
- Çerezler (cookies), sonradan kullanmak için kullanıcının bilgisayarına sakladığımız verilerdir. Aynı veriyi almak için her seferinde sunucuda/veritabanında işlem yapmak yerine yükü azaltmaya da yarar.
- Farkedebileceğiniz en yaygın kullanımı, kullanıcı girişlerinde “beni hatırla” kutucukları sayesinde kullanıcı adı ve şifrenin saklanmasıdır.
- Çerez, genellikle bir kullanıcıyı tanımlamak için kullanılır. Çerez, sunucunun kullanıcının bilgisayarına yerleştirdiği küçük bir dosyadır. Aynı bilgisayar bir tarayıcıyla bir sayfa istediğinde, çerezi de gönderir. PHP ile çerez değerlerini oluşturabilir ve alabilirsiniz.

# PHP ile Çerezler Oluşturma

- - Bir çerez, `setcookie ()` işlevi ile oluşturulur.
- Yazım şekli;
  - `setcookie("isim", "deger", "geçerlilik_süresi");`
- Not: Sadece isim parametresi gereklidir. Diğer tüm parametreler isteğe bağlıdır.

# PHP Oluşturma / Çerez Alma

- Yandaki örnek, "ali akın" değeri ile "admin" adlı bir çerez oluşturur. Çerezin geçerlilik süresi 30 gün sonra ( $86400 * 30$ ) dolacak.
  - Ardından, "admin" çerezinin değerini alırsınız (genel değişken **`$_COOKIE`** 'yi kullanarak).
  - Çerezin ayarlanmış olup olmadığını öğrenmek için **`isset()`** işlevi de kullanılır:
- ```

<?php
• $cookie_adi = "admin";
• $cookie_value = "Ali Akın";
• setcookie($cookie_adi, $cookie_value,
  time() + (86400 * 30)); // 86400 = 1 day
• if(!isset($_COOKIE[$cookie_name])) {
•   echo "Cookie adı '" . $cookie_adi . "' set
  edilmemiş!";
• } else {
•   echo "Cookie Adı" . $cookie_name . "' set
  edilmiş!<br>";
•   echo "Cookie değeri: " .
    $_COOKIE[$cookie_adi];
• }
• ?>

```

Cookie'yi yeniden set etmek

- Bir cookie bilgisini değiştirmek için, cookie bilgisini setcookie () işlevini kullanarak ayarlayın :

```
• <?php
  $cookie_name = "admin"; //cookie'yi yeniden set ettik
  $cookie_value = "Ali Akın"; //cookie'yi yeniden set ettik
  setcookie($cookie_name, $cookie_value, time() + (86400 * 30));
?>
<html>
<body>
<?php
if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie Adı '" . $cookie_name . "' set edilmemiştir!";
} else {
    echo "Cookie '" . $cookie_name . "' Set edilmiştir!<br>";
    echo "Değeri: " . $_COOKIE[$cookie_name];
}
?>
</body>
</html>
```

Cookie (Çerez) Silme

- Bir tanımlama bilgisini silmek için, geçmişte bir son kullanma tarihi olan setcookie () işlevini kullanın:
- ```
<?php
setcookie("user", "", time() - 10);
?>
<html>
<body>

<?php
echo "Cookie 'user' silindi.";
?>

</body>
</html>
```

# Çerezlerin Etkin olup olmadığını Kontrol Edin

- Yandaki örnek, çerezlerin etkin olup olmadığını kontrol eden küçük bir komut dosyası oluşturur.
- İlk olarak, setcookie () işlevi ile bir test çerezi oluşturmayı deneyin, ardından \$\_COOKIE dizi değişkenini sayın:

```
<?php
setcookie("test_cookie", "test", time()
+ 3600, '/');
?>
<html>
<body>

<?php
if(count($_COOKIE) > 0) {
 echo "Cookies kullanılabilir.";
} else {
 echo "Cookies kullanılamaz.";
}
?>

</body>
</html>
```



# Session ve cookie arasındaki fark

- Bir çerez tanımladığımızda bu çerezin adı ve değeri tarayıcıya açıkça gönderilir ve bu değer istenilen süre boyunca tarayıcıda açık bir şekilde saklanır.
- Sessionda ise kaydedilen değer ve değerın adı tarayıcıya **gönderilmez**. Onun yerine **PHPSESSID** adında içinde uzun bir session id değeri bulunan bir çerez tanımlanır. Ve bizim oluşturduğumuz sessionlar sunucuda saklanır ve girenlerin gönderdiği PHPSESSID kimliğine göre kime ait oldukları bulunur ve yeniden düzenlenir.

# PDO Select ve Prepare Kullanımı

- Eğer SQL sorgularımızda dışarıdan verigirişi mevcut ise mutlaka prepare metodu kullanılmalıdır. Bu metot SQL injectionlardan kurtulmak için büyük bir güvenlik sağlar.
- Pdo da prepare() metodu çalıştırılmak için bir SQL deyimini hazırlar. Prepare metodu bindparam(), execute(), bindColumn(), bindValue() metotları ile beraber çalışır. SQL sorguları oluşturulurken dışarıdan gelen verileri soru işareti veya önünde iki nokta üst üste yazarak oluşturacağımız bir isim (örnek :adi , :eposta vb.) ile sorgumuzda uygun yerlere dahil edebiliriz.
- **Soru İşareti İle Dışarıdan Veri Alma**
- **Örnek :** Aşağıdaki örnekte \$\_GET ile alınan \$dyili değişkenindeki değere göre doğum yılı \$dyili değişkeninden daha büyük olan üye adları ve eposta adresleri listeleniyor.

```
<?php
$dyili=$_GET["dyili"];
$q = $pdo->prepare("SELECT * FROM uyeler where dyili>?");
$q->execute(array($dyili));
if ($d=$q->fetchAll()){
 foreach($d as $k=>$v) {
 echo $v["uyeadı"].'
';
 echo $v["eposta"].'
';
 }
}
?>
```

Burada sorgu hazırlanırken prepare metodu kullanılmıştır. Prepare metodunda dışarıdan gelmesi gereken veri için dyili>? şeklinde soru işareti değeri ile dışarıdan gelen verinin sorgumuzda dahil edileceği alan belirlenmiştir. \$q->execute(array(\$dyili)) sorgumuzu burada girilen dizi içerisindeki veriyi dahil etmiş ve \$q->fetchAll() ile sorguya uygun kayıtların listesi alınmıştır.

# prepare() ve execute() kullanımı

- Sql sorgumuzu hazırlayıp kullanabileceğimiz fonksiyondur. Değer alabilir ve değer döndürebilir. Hazırlanan sql sorgusu execute() fonksiyonu ile çalıştırılır. Aldığı değerler array() şeklinde gönderilir. İki şekilde değer alır. Birincisi “?” imi ile aldığı değerler ikincisi ise “:isim” şeklinde aldığı değerler. “:isim” şeklindeki atamalar daha anlaşılır olduğu için kullanılması mantıklıdır.
- Bir insert sorgusu yazarak tablomuza bir satır ekleyelim ve “?” imi ile nasıl değer alıyor bir görelim.

// insert sorgumuzu \$query isminde bir degiskene esitleyelim.

```
$query = $db->prepare("INSERT INTO example (userid,baslik,detay,gonderen) VALUES (?,?,?,?)");
```

// execute icinde array ile degerlerimizi insert sorgumuza yolladik ve kaydimizi ekledik.

```
$query->execute(array("ali","Makale icerigi","makale yazım kurallarını gönder","murat"));
```

// 2. bir kayıt eklemek istedigimde sql sorgumu tekrar yazmama gerek yok hemen degerlerimi degistirip// execute fonksiyonum ile baska bir icerik eklemesi yapabilirim ornegin ;

```
$query->execute(array("ayşe","projeler"," proje hazırlama hakkında bilgi ","admin"));
```

# Sorgularda parametre kullanmak

- Sorgularda bazen form üzerinden gelen bir yada birden fazla değere göre verileri listeletebiliriz.
- Örneğin; personel tablosundan erkeklerin ve pazarlama bölümünde çalışanları listelenmesini istersek;
  - `Select * from personel where cinsiyeti=1 and bolum='pazarlama'`
- Şeklinde bir sorgu yazarız. Ancak farklı zamanlarda farklı cinsiyet ve bölümdelikleri sorgulamak istediğimizde ne olacak. Çünkü yukarıdaki sorgu cümlesinde koşullarımız sabit verilmiş. İşte bu durumlarda sorgularımızı php ile şöyle oluşturabiliriz;
  - `Select * from personel where cinsiyeti=:cinsiyet and bolum=:bolum;`
- Yada
  - `Select * from personel where cinsiyeti=? and bolum=?;`
  - Sayfanın devamı sonraki slaytta.

# Sorgularda parametre olarak ? kullanımı

- \$conn=new  
PDO('mysql:host=localhost;dbname=bilgisayarprg;charset=utf8','root','');
- \$conn->setAttribute(PDO::ATTR\_ERRMODE,  
PDO::ERRMODE\_EXCEPTION);
- \$sonuc=\$conn->prepare("select \* from personel where cinsiyet=? and  
birim=?");
- \$sonuc->execute(array(1, 10));
- \$row=\$sonuc->fetchAll(PDO::FETCH\_ASSOC) ;
- foreach (\$row as \$value) {
  - echo \$value["ad"]."---".\$value["cinsiyet"]."---".\$value["birim"]."<br>";
- }

# Sorgularda parametre olarak *:parametre\_adi* kullanımı

- `$conn=new PDO('mysql:host=localhost;dbname=bilgisayarprg;charset=utf8','root','');`
- `$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);`
- `$sonuc=$conn->prepare("select * from personel where cinsiyet!=:cinsiyet and birim>:birim");`
- `$sonuc->execute(array(":cinsiyet"=>1,":birim"=>10));`
- `$row=$sonuc->fetchAll(PDO::FETCH_ASSOC) ;`
- `foreach ($row as $value) {`
- `echo $value["ad"]."---".$value["cinsiyet"]."---".$value["birim"]."<br>";`
- `}`

Sorgularda parametre olarak *:parametre\_adi* kullanımı (\$\_POST yada \$\_GET ile gelen veriler

- \$conn=new PDO('mysql:host=localhost;dbname=bilgisayarprg;charset=utf8','root','');
- \$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);
- \$sonuc=\$conn->prepare("select \* from personel where cinsiyet!=:cinsiyet and birim>:birim");
- \$sonuc->execute(array(":cinsiyet"=>\$\_POST["cinsiyet"],":birim"=>\$\_POST["birim"]));
- \$row=\$sonuc->fetchAll(PDO::FETCH\_ASSOC) ;
- foreach (\$row as \$value) {
- echo \$value["ad"]."---".\$value["cinsiyet"]."---".\$value["birim"]."<br>";
- }

# BindValue ile sorgulara parametre geçirme

- Sorgularda kullandığımız parametrelere, Bir değeri, SQL deyimindeki karşılık gelen ad veya soru işareti yer tutucusuna bağlar.
- <?php
  1. \$conn = new PDO("mysql:host=localhost;dbname=bp1;charset=utf8", 'root','');
  2. \$sonuc=\$conn->prepare("select \* from users where userid=:usr");
  3. \$sonuc->bindParam(":usr", \$user, pdo::PARAM\_STR);//parametre tipi de veriliyor.
  4. \$user='Ahmet';
  5. \$sonuc->execute();
  6. \$row=\$sonuc->fetch(PDO::FETCH\_ASSOC);
  7. echo \$row["adsoyad"]."--".\$row["userid"]."--".\$row["email"];
- Bu kodlarda 2.satırdaki :usr isimli parametreye 3.satırdaki bindparam ile 4.satırdaki \$user değişkenindeki değer aktarılıyor.



# Tablodan çok satır getirme (FETCHALL)

- **Fetchall()** metodu PREPARE() ve EXECUTE() metodları ile birlikte kullanılır.
- Bir sql sorgusuna ait bütün satırları döndüren bir dizi oluşturur. Sorgu sonucunda az sayıda satır dönmesini beklediğimiz durumlarda kullanırız.
- **PREPARE()**=Çalıştırılmak üzere bir sql deyimini hazırlar.
- **EXECUTE()**=prepare ile hazırlanmış olan sql deyimini çalıştırır. Aynı zamanda parametreleri sql'deki değişkenlere bağlar.
  - \$conn=new PDO('mysql:host=localhost;dbname=bilgisayarprg;charset=utf8','root','');
  - \$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);
  - \$sorgu=\$conn->prepare("select \* from personel ");
  - \$sorgu->execute();
  - \$row=\$sorgu->fetchAll(PDO::FETCH\_ASSOC);
  - Foreach (\$row as \$value) {
  - echo \$value["ad"]."---".\$value["cinsiyet"]."---".\$value["birim"]."<br>;
  - }

# While döngüsü ile verileri listeleme

- `$conn=new PDO('mysql:host=localhost;dbname=bilgisayarprg;charset=utf8','root','');`
- `$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);`
- `$sonuc=$conn->prepare("select * from personel ");`
- `$sonuc->execute();`
- `while ($row = $sonuc->fetch(PDO::FETCH_ASSOC) ) {`
- `echo $row["ad"]."<br>";`
- `}`

# Birden fazla satır çekme

```

• <html>
• <head>
• <meta charset="UTF-8">
• <title></title>
• </head>
• <body>
• <?php
• $conn = 'mysql:host=localhost;dbname=tavsanli;';
• $user = 'senol';
• $password = 'senol';
• try {
• $db = new PDO($conn, $user, $password);
• $db->query("SET CHARACTER SET utf8");
• $id = 1;
• $query = $db->query("SELECT * FROM ogrenci",PDO::FETCH_ASSOC);
• // $query = $query -> fetch(PDO::FETCH_ASSOC); // PDO::FETCH_ASSOC: veriyi sütun
• // ismi ile çeker.
• // print_r($query);
• if ($query)
• {
• echo '<border=1 width=400 >
•
• <tr>
•
• <th>Ad </th>

```

```

• <th>Soyad </th>
• <th>D.Yeri </th>
• </tr> <tbody>';
• foreach ($query as $value) {
• echo "<tr> <td>". $value["adi"]. "</td>";
• echo "<td>". $value["soyadi"]. "</td>";
• echo "<td>". $value["dyeri"]. "</td></tr>";
• }
• echo "</tbody></table>";
• }
• else
• {
• echo "Listelenecek kayıt bulunamadı...";
• }
• }
• catch (PDOException $e) {
• echo 'Connection Hatası: ' . $e->getMessage();
• }
• ?>
• </body>
• </html>

```

# PDO *Select* İşlemi (Tek satır Veri Çekmek)

- Formdan get metoduyla gelen id isimli parametreye göre tablodan veriyi listeler.
- Not: quote metodu, veriyi özel karakterlerden temizler.
- \$id = \$db->quote(\$\_GET['id']);
- //aşağıdaki satır tüm post ile gelen verileri özel karakterlerden temizler.
- \$\_POST= array\_map(array(\$conn, 'quote'),\$\_POST);
- \$query = \$db->query("SELECT \* FROM icerik WHERE id= ?");
- \$query = \$query -> fetch(PDO::FETCH\_ASSOC); // **PDO::FETCH\_ASSOC**: veriyi sütun ismi ile çeker.
- \$query ->bindParam(1, \$id );
- if ( \$query ){
- echo \$query['icerik\_baslik'];
- }

# Örnek : mdList.php (MasterDetails.php)

69

- <?php
- try{
- \$conn=new PDO("mysql:host=localhost;dbname=bp2;charset=utf8","root","");
- \$sql="select \* from users where userid in(select userid from duyurular)";
- \$sonuc = \$conn->query(\$sql);
- \$sonuc = \$sonuc->fetchAll(PDO::FETCH\_ASSOC);
- ?>
- <?php
- while (\$value=\$sonuc->fetch(PDO::FETCH\_ASSOC)){
- echo "<table border=1 width=1000>".
- "<tr><th width=100>Kullanıcı</th><th width=200>Duyuru Başlığı</th><th width=500>Duyuru Detay</th><th width=200>Duyuru Tarihi</th></tr>";
- echo "<tr><td colspan=4 align=center><strong>".\$value['userid']."</strong></td></tr>";
- \$duyurusql="select \* from duyurular where userid=" . \$value["userid"]."";
- \$duyurusonuc = \$conn->query(\$duyurusql,PDO::FETCH\_ASSOC);
- // \$duyurusonuc->execute(array(\$value["userid"]));
- \$duyurusonuc = \$duyurusonuc->fetchAll(PDO::FETCH\_ASSOC);
- if(\$duyurusonuc)
- {
- foreach (\$duyurusonuc as \$duyuruvalue) {
- echo "<tr><td>&nbsp;</td><td>".\$duyuruvalue["duyuru\_baslik"]."</td><td>".\$duyuruvalue["duyuru\_icerik"]."</td><td>".date("d/m/Y",strtotime(\$duyuruvalue["duyuru\_tarih"]))."</td></tr>";
- }
- echo "<tr><td colspan=4><strong>".\$value["userid"]."</strong> Kullanıcısına gönderilen Duyuru Sayısı : ".count(\$duyurusonuc);
- echo "</td></tr></table><br>";
- }
- }
- }
- }
- catch (PDOException \$e)
- {
- echo \$e->getMessage();
- }

# Kayıt güncelleme

- `$conn=new PDO("mysql:host=localhost;dbname=bp2;charset=utf8","root","");`
- `echo "<hr>kayıt güncelleme....";`
- `$upd="update users set durum=7 where id in(4,5) ";`
- `$update=$conn->prepare($upd);`
- `$update->execute();`
- `$row=$update->rowCount();`
- `echo "güncellenen kayıt sayısı : ".$row;`

## ? Soru işaretli Parametreye göre kayıt güncelleme(UnNamed Parameters)

1. \$conn=new PDO("mysql:host=localhost;dbname=bp2;charset=utf8","root","");
2. echo "kayıt sayısı : ".\$sorgu->rowCount()."<br>";
3. echo "<hr>kayıt güncelleme....";
4. \$upd="update users set durum=? where id in(?,?,?) ";
5. \$update=\$conn->prepare(\$upd);
6. \$update->execute(**array(8,2,3,4)**); yada: \$update-> execute([8,2,3,4])  
//6 nolu satır id değeri 2,3 ve 4 olanların durum sütununu 8 yapar
7. \$row=\$update->rowCount();//güncellenen kayıt sayısını verir
8. echo "güncellenen kayıt sayısı : ".\$row;

## : Parametreye göre kayıt güncelleme=İsimli Parametre (Named Parameters)

1. \$conn=new  
PDO("mysql:host=localhost;dbname=bp2;charset=utf8","root","");
  2. echo "kayıt sayısı : ".\$sorgu->rowCount()."<br>";
  3. echo "<hr>kayıt güncelleme....";
  4. \$upd="update users set durum= :drm where id in(:id1,:id2,:id3) ";
  5. \$update=\$conn->prepare(\$upd);
  6. \$update->execute(array(":drm"=>7,":id1"=>2,":id2"=>3,":id3"=>4));  
//6 nolu satır id değeri 2,3 ve 4 olanların durum sütununu 7 yapar
1. \$row=\$update->rowCount();
  2. echo "güncellenen kayıt sayısı : ".\$row;



# Bindvalue kullanımı

- \$conn=new PDO("mysql:host=localhost;dbname=bp2;charset=utf8","root","");
- echo "<hr>kayıt güncelleme....";
- \$upd="update users set durum= :**drm** where id in(:**id1**,:**id2**,:**id3**) ";
- \$update=\$conn->prepare(\$upd);
- \$update->bindValue(":**drm**", 9,PDO::PARAM\_INT);
- \$update->bindValue(":**id1**", 1,PDO::PARAM\_INT);
- \$update->bindValue(":**id2**", 5,PDO::PARAM\_INT);
- \$update->bindValue(":**id3**", 6,PDO::PARAM\_INT);
- \$update->execute();//id değeri 2,3 ve 4 olanların durum sütununu 8 yapar
- \$row=\$update->rowCount();
- echo "güncellenen kayıt sayısı : ".\$row;

- \$conn=new PDO("mysql:host=localhost;dbname=bp2;charset=utf8","root","");
- echo "<hr>kayıt güncelleme....";
- \$upd="update users set durum=? where id in(?,?,?) ";
- \$update=\$conn->prepare(\$upd);
- \$update->bindValue(**1**, 9,PDO::PARAM\_INT);
- \$update->bindValue(**2**, 1,PDO::PARAM\_INT);
- \$update->bindValue(**3**, 5,PDO::PARAM\_INT);
- \$update->bindValue(**4**, 6,PDO::PARAM\_INT);
- \$update->execute();//id değeri 2,3 ve 4 olanların durum sütununu 8 yapar
- \$row=\$update->rowCount();
- echo "güncellenen kayıt sayısı : ".\$row;

# Kayıt ekleme

- NOT: Bir sütun AUTO\_INCREMENT ("id" sütunu gibi) veya TIMESTAMP ("reg\_date" sütunundaki gibi) ise, SQL sorgusunda belirtilmesine gerek yoktur; MySQL otomatik olarak değeri ekleyecektir.
- try{
- \$ekle=\$conn->prepare("insert into duyurular values(null,'senol','pdo ile database','pdo ile database kullanımı ve örnekleri','2017/03/21',1,'pdo')");
- \$ekle->execute();
- }
- catch (PDOException \$e){
- echo \$e->getMessage();
- }

# Kayıt ekleme (parametre kullanarak)

- Bu örnekte tabloya veri eklerken formdan POST ile gelen veriler kullanılmıştır.
- `<form method=POST>`
  - `<input type=text name=ad >`
  - `<input type=text name=soyad >`
  - `<input type=text name=email >`
  - `<button type=submit name=ekle>Ekle</button>`
- `</form>`

```

1. if(isset($_POST['ekle'])) {
2. try{
3. $ekle=$conn->prepare("insert into users(ad,soyad,email)
 values(?,?,?)");
4. $ekle->execute($_POST['ad'], $_POST['soyad'],
 $_POST['email']);
5. if($ekle) //hata oluşursa $ekle değişkeni false değeri alır...
6. echo "kayıt başarılı";
7. else
8. echo "HATA!...";

9. }
10. catch (PDOException $e){
11. echo $e->getMessage();
12. }

```