



Burdur
Mehmet Akif Ersoy
Üniversitesi

MAKÜ|GMYO
GÖLHİSAR MESLEK YÜKSEKOKULU

İnternet Programcılığı I

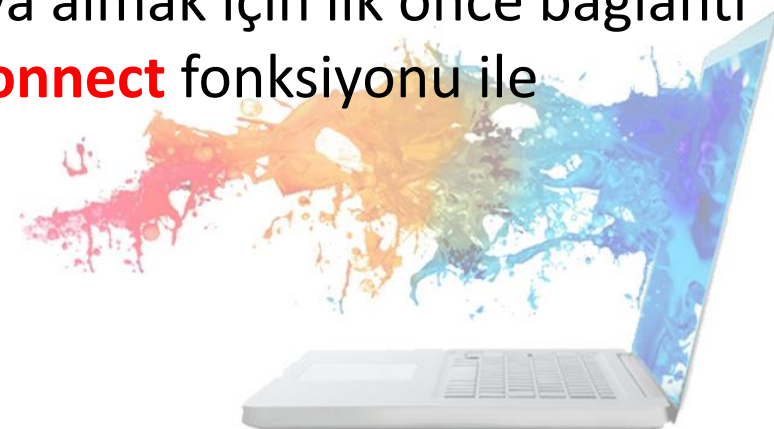
Öğr. Gör. Hüseyin Şengün
hsengun@mehmetakif.edu.tr

11. Hafta
VERİ TABANI İŞLEMLERİ-3



VERİ TABANINA BAĞLANMA

- PHP ile sadece PHP, HTML, JS veya CSS kodlanmıyor. Aynı zamanda SQL komutlarını kullanarak veritabanına erişebiliyorsunuz.
- PHP 7 sürümünden sonra artık **MySQLi** fonksiyonu zorunlu hale getirildi. **MySQL** fonksiyonu tamamen kaldırıldı. Bundan dolayı MySQL kullanmak istiyorsanız PHP'nin 5.6 sürümünü kullanmanız gerekmektedir. 5.6 sürümünde MySQLi desteği bulunmaktadır.
- PHP veri tabanına veri göndermek veya almak için ilk önce bağlantı kurması gerekir. Bu bağlantı **mysqli_connect** fonksiyonu ile gerçekleştirilir.





VERİ TABANINA BAĞLANMA

- Kullanım şekli:
`$baglanti = mysqli_connect("host","kullanici_adi","sifre","veritabani_adi");`
- Veri tabanımız PHP sayfaları ile aynı yerde ise “localhost” veya “127.0.0.1” yazabiliriz. Ancak farklı sunucuda ise sunucunun ip adresini veya domain adını(<https://www.siteadi.com>) yazmamız gerekir.



VERİ TABANINA BAĞLANMA

- Eğer veri tabanımızın şifresi yoksa araya hiçbir karakter yazmadan çift tırnak “ ” yazmamız yeterlidir.
- İnternette yayın yapmadığımız için genelde kendi bilgisayarımızdaki veri tabanına şifre koymayız ancak veri tabanımızı internette yayın yapan sunucuya koyduğumuzda güvenlik için veri tabanımıza şifre vermeyi unutmayın.

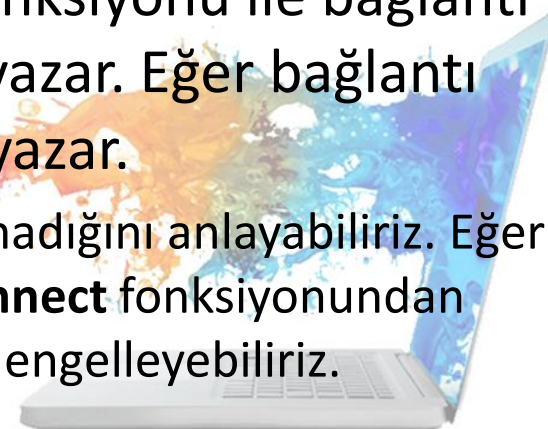




VERİ TABANINA BAĞLANMA

```
<?php
$baglanti = mysqli_connect("Sunucu Adresi","Kullanıcı Adı"," Şifre","Veritabanı");
if($baglanti)
{
echo("Veritabanı Bağlantısı Kuruldu!");
}
else
{
echo("Veritabanı Bağlantısı Kurulamadı!");
}
?>
```

- Yukardaki örnekte PHP'de **mysqli_connect()** fonksiyonu ile bağlantı yapılırsa ekrana Veritabanı Bağlantısı Kuruldu yazar. Eğer bağlantı yapılamazsa Veritabanı Bağlantısı Kurulamadı yazar.
- Bunu kullanarak veri tabanımıza bağlantı yapıp yapılamadığını anlayabiliriz. Eğer veri tabanı bağlantısında bir sorun yaşanırsa **mysqli_connect** fonksiyonundan dönen veriyi kullanarak sonraki kodlara geçiş yapmasını engelleyebiliriz.



Veri Tabanı Seçimi

- Veri tabanı sunucumuza bağlandık şimdi işlem yapacağımız veri tabanını seçmemiz gerekiyor. Bu işlem isteğe bağlıdır yani zorunlu bir işlem değildir. Ancak bu işlemin yapımı tavsiye edilir çünkü daha sonra göndereceğimiz her sorgunun hangi veri tabanını etkileyeceğini göndermek istemiyorsak bu işlemi gerçekleştirmeliyiz.
- Kullanım şekli: **mysqli_query(\$baglanti, "SELECT * FROM uyeler");**



Veri Tabanı Seçimi

```
<?php
```

```
$baglanti = mysqli_connect("localhost", "Kullanici", "Sifre", "yeni_veritabani");  
mysqli_query($baglanti, "SELECT * FROM uyeler");  
mysqli_close($baglanti);  
?>
```

- Yukardaki örnekte veri tabanımıza bağlantı yaptık ve daha önceden oluşturduğumuz “**yeni_veritabani**” ve bilgilerini yazdık. Bu bağlanma işlemini **\$baglanti** değişkenine atadık.
- **mysqli_query** ile sorgumuzu yazdık. Bu sorgu ile uyeler tablosudaki bütün alanları listelettik.
- **mysqli_close** ile bağlantı kapatıldı.



Sorgu Çalıştırma

- PHP'de sorgularımızı çalıştırmak için **mysqli_query** fonksiyonu kullanılır.
- Tabi bu fonksiyonu çalıştırmak için önce veri tabanına bağlantı yapılmazsa program hata verecektir. Bundan dolayı veri tabanı bağlantısını yapmayı unutmayınız.
- Bu fonksiyon yazdığımız bütün sorguları veri tabanına iletilir ve veri tabanından bir cevap döner.
- **mysqli_query(\$baglanti, "SELECT * FROM uyeler");**



Sorgu Çalıştırma

Örnek:

```
mysql_query($baglanti, "DELETE FROM uyeler WHERE sıra_no=5");
```

- Yukardaki örneği çalıştırdığımız da ekranda hiçbir şey gelmediğini ancak veri tabanımızdaki verileri kontrol ettiğimizde **uyeler** tablosundaki sıra_no kolonu beş “5” olan kayıttın silindiğini görürüz.



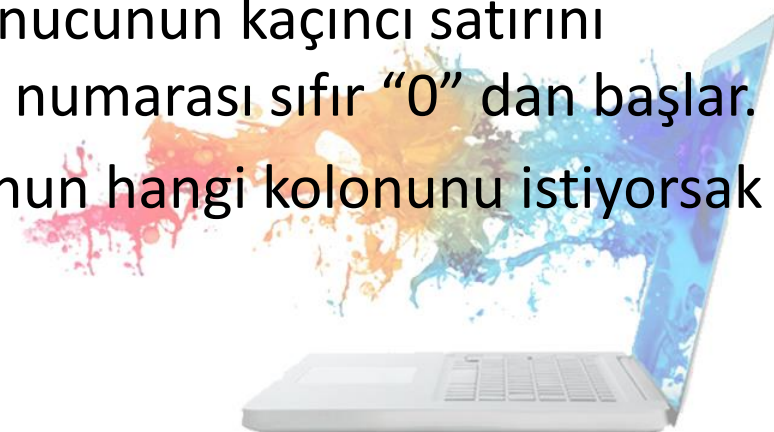
Veri Çekme işlemler

- Veri tabanından veri çekmek için “SELECT” kodu ile başlayan sorgular yazmalıyız. Sorgumuzu **mysqli_query()** fonksiyonu ile çalıştırsakta gelen sonuçları işlemek için farklı fonksiyonlar kullanmalıyız. Bu fonksiyonların en sık kullanılanları **mysqli_result** ve **mysqli_fetch_assoc** fonksiyonlarıdır.



Veri Çekme işlemler

- **mysqli_result()**
- Yazdığımız sorgu bir “SELECT” sorgusu ise bize bilgi döndürmesi gerekir. Bu bilgiyi PHP’de alma yollarından bir tanesi **mysqli_result()** fonksiyonudur.
- *Bu fonksiyon üç parametre alır:*
- **1. Sorgu:** Metin olan bir sorgu değil mysqli_query() fonksiyonundan dönen yanıt yazılır.
- **2. Satır numarası:** Yazdığımız sorgu sonucunun kaçınıcı satırını istiyorsak o satır numarası yazılır. Satır numarası sıfır “0” dan başlar.
- **3. Kolon adı:** Yazdığımız sorgu sonucunun hangi kolonunu istiyorsak o kolonun adı yazılır.

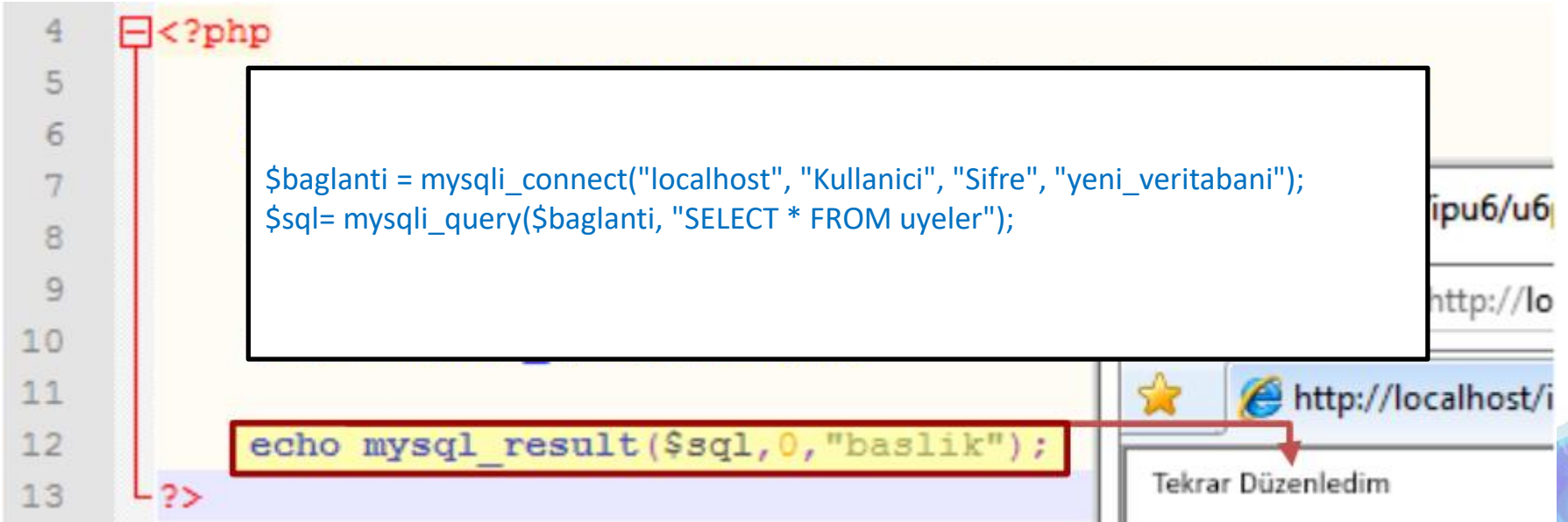


Veri Çekme işlemler

- **mysqli_result()**

Kullanım şekli: **mysqli_result**(sorgu değişkeni, satır numarası, kolon adı)

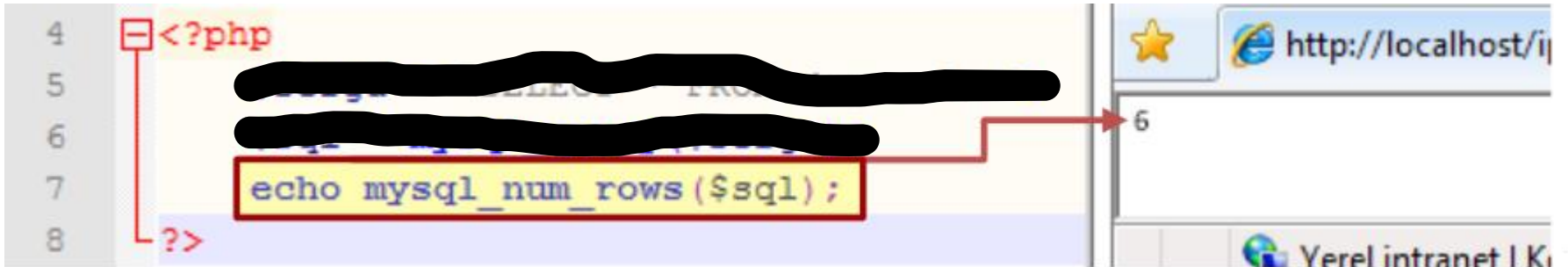
```
4 <?php
5
6
7 $baglanti = mysqli_connect("localhost", "Kullanici", "Sifre", "yeni_veritabani");
8 $sql= mysqli_query($baglanti, "SELECT * FROM uyeler");
9
10
11
12 echo mysqli_result($sql,0,"baslik");
13 ?>
```



Yukardaki örnekte mysqli_query() fonksiyonuna sorgumuza yazdık ve dönen cevabı \$sql adındaki değişkenimize atadık. Daha sonra mysqli_result() fonksiyonumuzu kullanarak “duyurular” tablosundaki “baslik” kolonunun ilk satırını ekrana yazdırıldı.

Veri Çekme işlemler

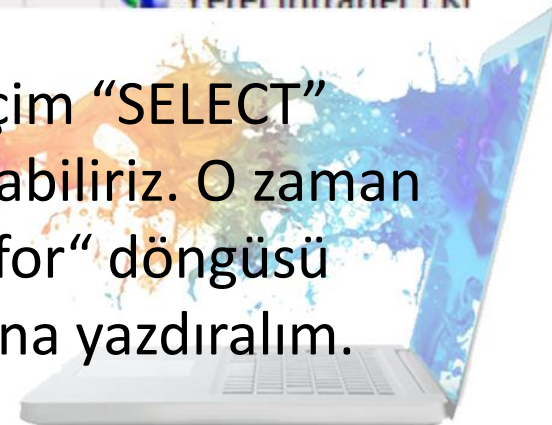
- Veri tabanından verileri çektik. Ancak yazdığımız sorgu sonucundan kaç satır geleceğini bilemeden `mysqli_result()` fonksiyonu ile bütün sonuçları elde edemeyiz. Seçim “SELECT” sorgularının kaç tane satır getireceği **`mysqli_num_rows()`** fonksiyonu kullanılarak bulunur.



```
4 <?php
5 [redacted]
6 [redacted]
7 echo mysqli_num_rows($sql);
8 ?>
```

http://localhost/i
6
Yerel intranet | Ki

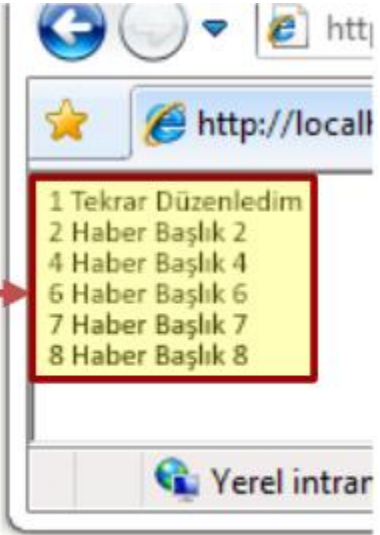
- Veri tabanından tek tek veri çekebiliyoruz ve seçim “SELECT” sorgusunun sonucunda kaç satır döneceğini bulabiliriz. O zaman şimdi de tablomuzda bulunan bütün satırların “for” döngüsü kullanarak “sira_no” ve “baslik” kolonlarını ekrana yazdıralım.



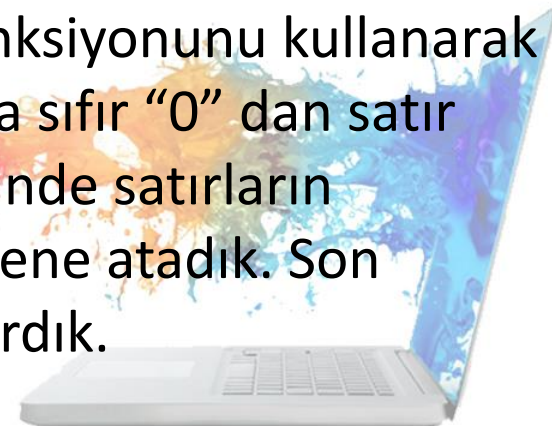


Veri Çekme işlemler

```
4 <?php
5
6 $baglanti = mysqli_connect("localhost", "Kullanici", "Sifre", "yeni_veritabani");
7 $sql= mysqli_query($baglanti, "SELECT * FROM uyeler");
8 $satisr_sayisi = mysqli_num_rows($sql);
9 for($i=0; $i<$satisr_sayisi; $i++)
10 {
11     $sira_no = mysqli_result($sql, $i, "sira_no");
12     $baslik = mysqli_result($sql, $i, "baslik");
13     echo $sira_no . " " . $baslik . "<br/>";
14 }
15 ?>
```



- Yukardaki örnekte sorgumuzu **mysqli_query()** fonksiyonu ile çalıştırdık. Kaç satır cevap geleceğini **mysqli_num_rows()** fonksiyonunu kullanarak **\$satisr_sayisi** adlı değişkenimize atadık. Daha sonra sıfır "0" dan satır sayısına kadar bir döngü kurduk. Döngünün içerisinde satırların "sira_no" ve "baslik" kolonlarındaki bilgileri değişkene atadık. Son olarak da bu değişkenlerdeki bilgileri ekrana yazdırdık.



Veri Çekme işlemler

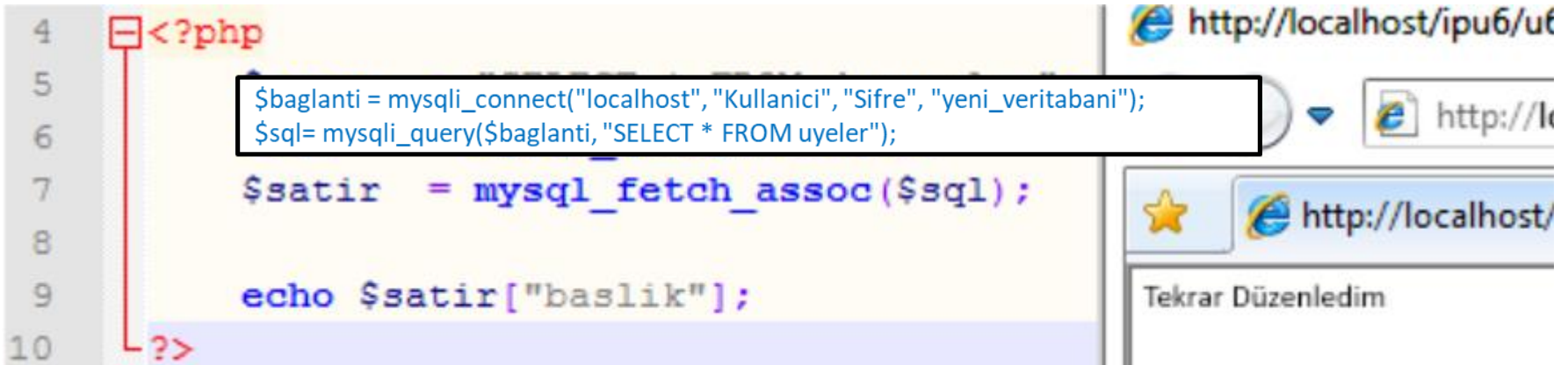
- **mysqli_fetch_assoc()**
- Veri tabanından diğer bir veri çekme fonksiyonu da **mysqli_fetch_assoc()** fonksiyonudur. Bu fonksiyonda veriler satır halinde diziye aktarılarak kullanılır.
- Kullanımı mysqli_result() fonksiyonuna göre daha kolaydır.
- Kullanım şekli: **mysqli_fetch_assoc(sorgu değişkeni)**



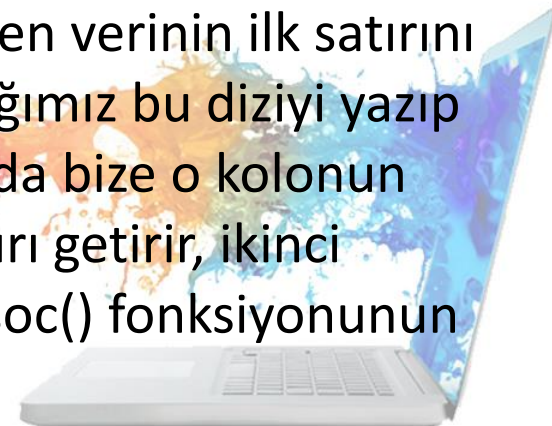
Veri Çekme işlemler

- **mysqli_fetch_assoc()**

```
4 <?php
5
6 $baglanti = mysqli_connect("localhost", "Kullanici", "Sifre", "yeni_veritabani");
7 $sql= mysqli_query($baglanti, "SELECT * FROM uyeler");
8
9 $saticir = mysqli_fetch_assoc($sql);
10
11 echo $saticir["baslik"];
```



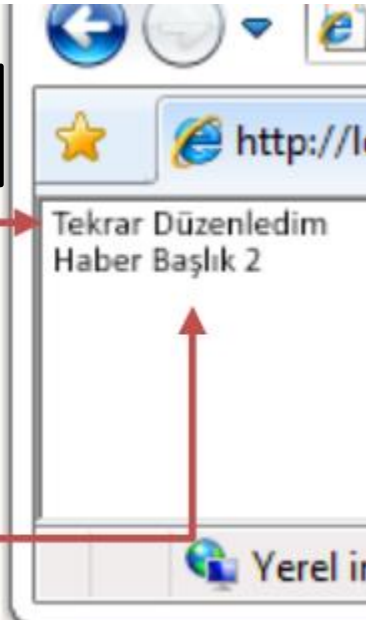
- Yukardaki örnekte sorgumuzu mysqli_query() fonksiyonunda çalıştırdıktan sonra mysqli_fetch_assoc() fonksiyonu sayesinde gelen verinin ilk satırını \$saticir adlı değişkene dizi olarak atadık. Verileri atadığımız bu diziyi yazıp dizinin index değeri olarak da kolon adını yazdığımızda bize o kolonun değerinin verir. Ancak veriyi getirirken sadece ilk satırı getirir, ikinci satırdaki veriyi getirmesi için tekrar mysqli_fetch_assoc() fonksiyonunun kullanılması gerekir.



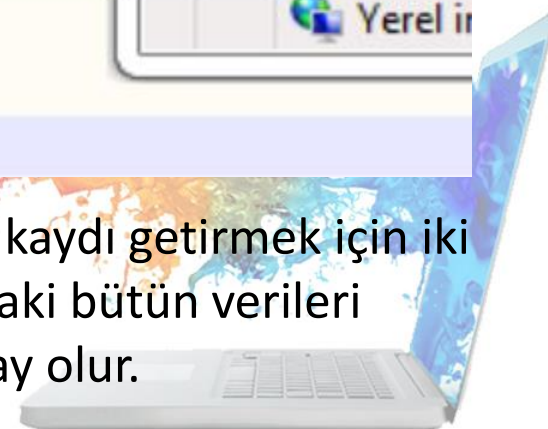
Veri Çekme işlemler

- **mysqli_fetch_assoc()**

```
5 <?php
6 $baglanti = mysqli_connect("localhost", "Kullanici", "Sifre", "yeni_veritabani");
7 $sql= mysqli_query($baglanti, "SELECT * FROM uyeler");
8
9 $saticir 1= mysql fetch assoc ($sql);
10 echo $saticir_1["baslik"];
11 echo "<br/>";
12
13 $saticir 2= mysql fetch assoc ($sql);
14 echo $saticir_2["baslik"];
15 echo "<br/>";
16 ?>
```



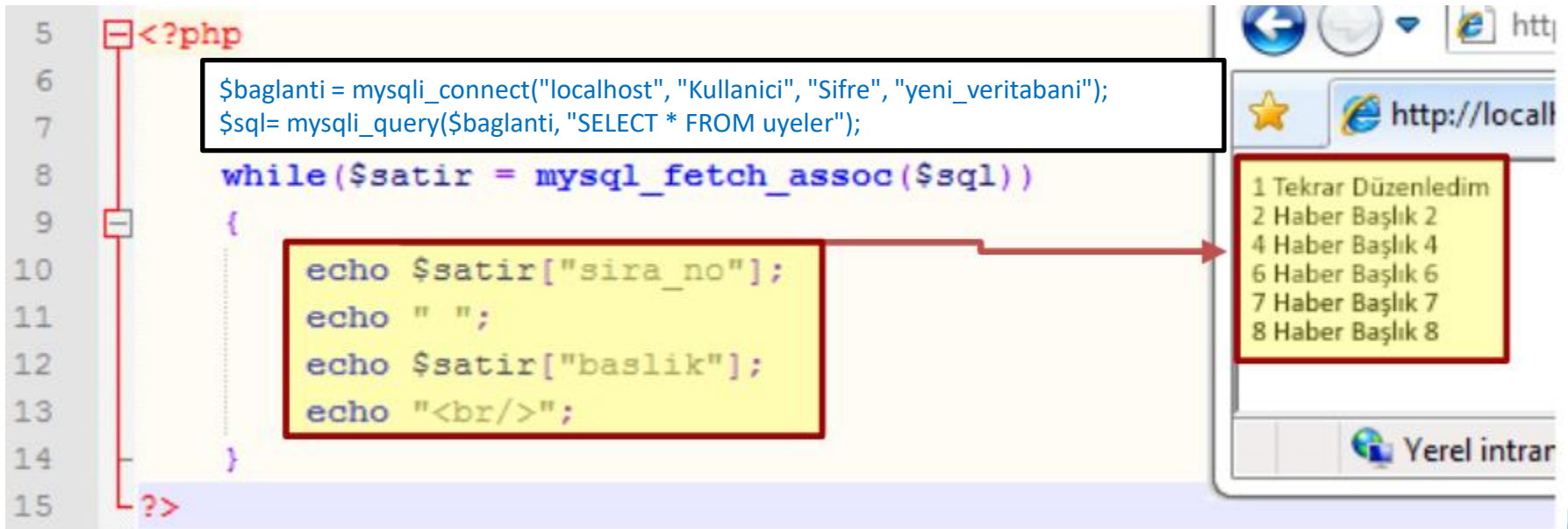
- Yukardaki örnekte “duyurular” tablosunda bulunan ilk iki kaydı getirmek için iki defa mysqli_fetch_assoc() fonksiyonunu kullandık. Tablodaki bütün verileri getirmek istiyorsak while döngüsü kullanmamız daha kolay olur.



Veri Çekme işlemler

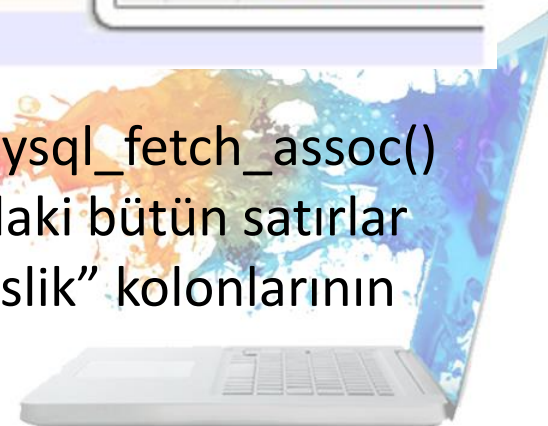
- **mysqli_fetch_assoc()**

```
5 <?php
6 $baglanti = mysqli_connect("localhost", "Kullanici", "Sifre", "yeni_veritabani");
7 $sql= mysqli_query($baglanti, "SELECT * FROM uyeler");
8 while($saticir = mysqli_fetch_assoc($sql))
9 {
10     echo $saticir["sira_no"];
11     echo " ";
12     echo $saticir["baslik"];
13     echo "<br/>";
14 }
15 ?>
```



1 Tekrar Düzenledim
2 Haber Başlık 2
4 Haber Başlık 4
6 Haber Başlık 6
7 Haber Başlık 7
8 Haber Başlık 8

- Yukardaki örnekte while döngüsünde koşul olarak `mysqli_fetch_assoc()` fonksiyonu kullandığımız için “duyurular” adlı tablodaki bütün satırlar sayısınca döngü oluşturduk. Ekranı “sira_no” ve “baslik” kolonlarının bütün satırlarındaki verileri yazdırdık.



Kaynaklar

- PHP Kılavuzu - <http://php.net/manual/tr/index.php>
- PHP Tutorials - <https://www.w3schools.com/php/default.asp>
- PHP Videolu Temel Dersler İçin: Youtube/Php Türkiye
- <https://php-mysql.org/>
- Atatürk Üniversitesi Açık Öğretim Fakültesi

