

# WEB PROGRAMLAMA II

Öğr. Gör. M. Mutlu YAPICI

Ankara Üniversitesi  
Elmadağ Meslek Yüksekokulu

# Ders İzlencesi

Hafta	Modüller/İçerik/Konular
1. Hafta	Oturum yönetimi
2. Hafta	Cookies kullanımı ve oturum yönetimi
3. Hafta	Session kullanımı ve oturum yönetimi
4. Hafta	Sayfalama ve Arama İşlemleri
5. Hafta	JavaScript, JQuery ve PHP
6. Hafta	AJAX ve PHP
7. Hafta	AJAX ve PHP
8. Hafta	ARA SINAV
9. Hafta	PHP'de Nesne Yönelimli Programlamaya Giriş(347)
10. Hafta	PHP de PDO
11. Hafta	
12. Hafta	
13. Hafta	
14. Hafta	

# Bu Ünitede Ele Alınan Konular

- PHP ve JAVASCRIPT
- PHP ve JQuery
- PHP ve AJAX
- AJAX ile Dinamik Veritabanı İşlemleri

# Ders Kazanımları

Bu bölümü Bitirdiğimizde,

- **PHP ve JAVASCRIPT**
- **PHP ve JQuery**
- **PHP ve AJAX**
- **AJAX ile Dinamik Veritabanı İşlemleri**

öğrenmiş olacaksınız.

# PHP ve AJAX

Buraya kadar php içerisinde java script ve JQuery kullanımının temel işlemlerini ve mantığını gördük. Görüldüğü üzere biz java script ile sayfanın tamamını güncellemeden istediğimiz HTML tagine yada alanına müdahale edebiliyor ve değiştirebiliyoruz. Ancak hala bu işlemi sunucu ile etkileşimli halde yapamıyoruz. Bunun için de AJAX teknolojisini kullanacağız.

Çalışma Mantığı kısaca şu şekilde, java Script kodu ile istemci tarafındaki isteği alıp ajax teknolojisinin yardımı ile sunucudaki bir PHP sayfasına göndereceğiz ve oradan gelen sonucu yani cevabı alıp tekrar java script ile sayfadaki HTML taglerine yazdıracağız. Böylece sayfanın tamamını yenilemeden sunucudan gelen verileri sayfaya eklemiş olacağız.

# PHP ve AJAX

**Öncelikle AJAX nedir? Kısaca bahsedelim.**

Asynchronous JavaScript and XML sözcüklerinin kısaltması olan **Ajax**, etkileşimli (interaktif) web uygulamaları yaratmak için kullanılan bir web programlama tekniğidir.

Temel amacı arka planda sunucuyla ufak miktarda veri değişimi sayesinde sayfayı daha hızlı güncelleyebilen web sayfaları yapmak, dolayısıyla kullanıcının istediği her anda bütün web sayfasını güncellemek derdinden kurtulmaktır. Bu da web sayfasının etkileşimini, hızını ve kullanılabilirliğini artırmak demektir.

# PHP ve AJAX

Aslında ajax **HTML / CSS/ DOM / JavaScript XML / XML/HttpRequest** gibi birçok bileşenin birleştirilmesiyle geliştirilmiş bir teknolojidir. Bir programlama dili değildir.

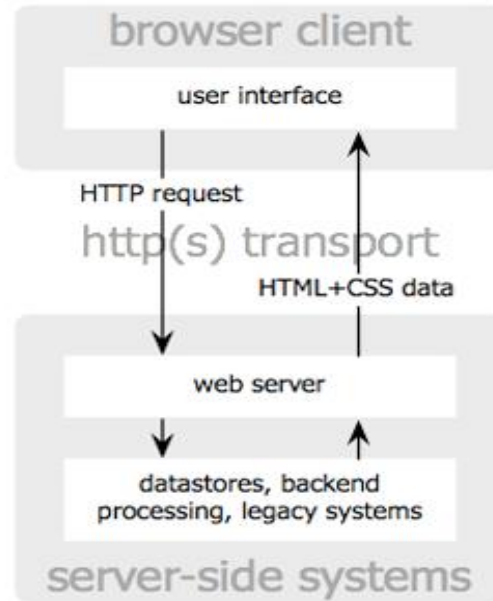
Kullanımı için iki farklı yol bulunmaktadır.

- 1) Java Script ile kendi ajax nesnemizi yaratıp kontrol etmek suretiyle
- 2) JQuery kütüphanesinde bizim için yaratılmış ajaz sınıfını kullanmak suretiyle.

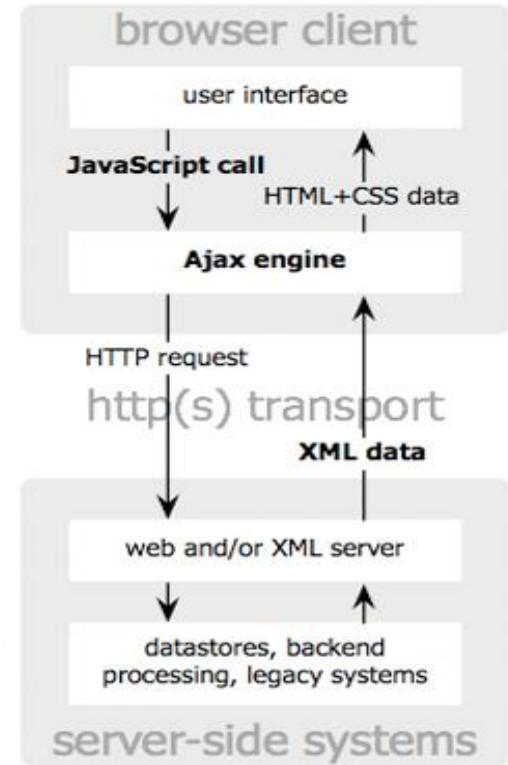
İki yöntemi de kullanarak AJAX teknolojisini kullanabiliriz. Biz de ikisine ait örnekleri inceleyeceğiz öncelikle 1. yöntemle başlayalım.

# AJAX Nasıl Çalışır?

AJAX çalışabilmek için sunucu ile istemci tarafındaki işlemleri sürekli takip eden ve gerçekleştiren bir **XMLHttpRequest** nesnesine ihtiyaç duyar.



classic  
web application model



Ajax  
web application model



# PHP ve AJAX

## Peki ama nedir bu XMLHttpRequest ?

- XMLHttpRequest (XHR), aslında JavaScript ile sunulmuş bir uygulama arayüzüdür.
- Web sitesinin eş zamanlı olarak arka planda çalışan olaylarını takip eder. XML CSS ve JavaScript ile sonucun anında görülmesini sağlar.
- XMLHttpRequest, istemci tarafında çalışır. Birden fazla asenkron Web sayfasına *postback* ve yenileme yapmadan istekte bulunabilir ve gelen cevapları istekte bulunanlara iletir.

# PHP ve AJAX

Madem AJAX için ilk olarak bu nesneyi yaratacağız, **XMLHttpRequest** nesnesi nasıl yaratılır bakalım.

Bu nesne tarayıcıdan tarayıcıya maalesef farklı şekillerde oluşturula biliniyor bu sebeple her tarayıcı için kontrol yaparak nesneyi oluşturmamız gerekiyor.

# PHP ve AJAX

```
<script language = "javascript" type = "text/javascript">
```

```
//Tarayıcılara Göre Kontrollü Nesne Oluşturma
```

```
function ajaxNesnesiOlustur(){
```

```
var HttpReq; //Oluşturduğumuz nesneyi bu değişkene yükleyeceğiz!
```

```
try { // Opera 8.0+, Firefox, Safari için
```

```
HttpReq = new XMLHttpRequest();
```

```
}catch (e) { try { // Internet Explorer İçin
```

```
HttpReq = new ActiveXObject("Msxml2.XMLHTTP");
```

```
}catch (e) { try{ // Internet Explorer İçin
```

```
HttpReq = new ActiveXObject("Microsoft.XMLHTTP");
```

```
}catch (e){
```

```
// Hala nesne oluşmuyorsa desteklemiyordur
```

```
alert("Tarayıcınız AJAX HttpRequest Nesnesini Desteklemiyor");
```

```
}
```

```
}
```

```
}
```

```
return HttpReq;
```

```
}
```

```
</script>
```

# PHP ve AJAX

**XMLHttpRequest** nesnesi oluşturacak bir fonksiyon yazdık, şimdi sırada bu nesneyi kullanarak sunucuya sorgu göndermek ve sonuçlarını almak gerekiyor.

Bunu yapabilmemiz için öncelikle **XMLHttpRequest** nesnesinin hangi özellikleri ve hangi olayları var öğrenmeliyiz ve ona göre bir fonksiyon yazmalıyız.

# PHP ve AJAX

## XMLHttpRequest nesnesi tanımlamaları:

- **open**("yöntem", "adres", "eşzamanlılık", "kullanıcı adı", "şifre")  
– Sorgu gönderilmeden önce, sorgunun gönderileceği adresi, gönderi metodunu ve diğer seçime bağlı nitelikleri belirler.

Yani sunucuda bizim işlemleri yaptıracağımız sayfayı ve formdan aldığımız verilerin hangi metod ile gideceğini belirlememizi sağlar

- **setRequestHeader**("etiket", "değer") – Gönderilen sorguya bir etiket/değer başlığını iliş­tirir.

`xmlHttp.setRequestHeader("Content-Type","application/x-www-form-urlencoded; charset=utf-8");`

- **send**("içerik") – Sorguyu gönderir. Eğer metod **POST** ise parametreler içine yazılır **GET** ise içine **null** yazılır.
- **abort**() – Geçerli isteği iptal eder.

# PHP ve AJAX

## XMLHttpRequest nesnesi özellikleri:

- **onreadystatechange** – Durum değişikliğini kontrol eder. Bir fonksiyona bağlantı için kullanılır.
- **readState** – Tam sayı değerleri durum bilgisini belirtir:  
**readyState** özelliği 1,2,3,4 gibi 4 değer alır 4 olduğunda gönderdiğimiz sorgu almamız için hazır demektir
  - 0 = uninitialized (başlatılmadı)
  - 1 = loading (yükleniyor)
  - 2 = loaded (yüklendi)
  - 3 = interactive (etkileşimli)
  - 4 = complete (tamamlandı)

# PHP ve AJAX

## XMLHttpRequest nesnesi özellikleri:

- **responseText** – Veriyi “string” biçiminde yollar. (HTML tagi içine yazılacak ve sunucudan gelen sonuçlardır bunlar.)
- **responseXML** – Veriyi XML olarak yollar. (HTML tagi içine yazılacak ve sunucudan gelen sonuçlardır bunlar.)
- **status** – Sunucunun durumunun sayısal değerini belirtir:
  - 404: Not found (Bulunamadı)
  - 200: OK (Tamam)
- **statusText** – Durumla gelen yazı değerini belirtir.

# PHP ve AJAX

Şimdi de bu XMLHttpRequest nesnesini kullanarak, sunucudaki bir PHP sayfasına sorgu gönderelim. Bunun için **open**("yöntem", "adres") ile verileri göndereceğimiz metodu ve sayfa adresini tanımlayıp, **setRequestHeader**("etiket", "değer") ile **xmlHttpRequest.setRequestHeader("Content-Type", "application/x-www-form-urlencoded; charset=utf-8");** header bilgilerini göndereceğiz daha sonrada **send**("içerik") ile verileri göndereceğiz.



# PHP ve AJAX

```
function AJAXKullan() {  
    //önce nesnemizi oluşturalım  
    var HttpReq =ajaxNesnesiOlustur();  
    if(!HttpReq)  
        return false;  
  
    ///şimdi bu nesneyi kullanarak AJAX bilgileirni tanımlayıp verileri gönderelim  
    HttpReq.open("POST", "ajaxicinPHP.php", true);  
    HttpReq.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');  
    HttpReq.send("adi=mutlu&sadi=YAPICI");///verileri gönderdik  
}
```

</script>

# PHP ve AJAX

Son olarak ta sunucudan gelen verileri alıp sayfamızda göstereceğiz.

Bunun için öncelikle XMLHttpRequest nesnemize ait durumu dinlemeli ve duruma göre işlem yapmalıyız. Bunun için XMLHttpRequest nesnemizin durumu değiştiğinde gerçekleşecek bir fonksiyona ihtiyacımız var. Bu sebeple nesnemizin durum değişikliğini bildiren olayına fonsiyon yazıp, durumun tamamlanıp tamamlanmadığına yani readystate in 4 olup olmadığına bakalım.

Daha sonra da nesnenin status özelliğine bakalım biliyorsunuz status 200 ise işlem tamam 404 ise sayfa bulunamadı demektir.

# PHP ve AJAX

Bir önceki kodda bulunan AJAXkullan() isimli fonksiyon içine devamına yazacağız.

```
HttpReq.send("adi=mutlu&sadi=YAPICI");//verileri gönderdik
```

```
////durum değişikliğini dinleyen bir fonksiyon tanımlayalım
```

```
HttpReq.onreadystatechange=function() {
```

```
    //////////////// readyState özelliği 1,2,3,4 gibi 4 değer alır
```

```
    //4 olduğunda gönderdiğimiz sorgu almamız için hazır demektir
```

```
    if(HttpReq.readyState==4)
```

```
    { alert(HttpReq.status);
```

```
        if (HttpReq.status == 200) ////eğer sayfa varsa sonuçları alalım
```

```
        { //herşey tamam olduğuna göre sonucu alıp sayfamızda gösterelim
```

```
            document.getElementById("div1").innerHTML=HttpReq.responseText;
```

```
        } if (HttpReq.status == 404) ////eğer sayfa yoksa hata verdirelim
```

```
        { //eğer hata varsa mesajı div içerisine yazalım
```

```
document.getElementById("div1").innerHTML="<h3>Ajax Verilerin Gönderildiği Sayfayı Bulamadı</h3>";
```

```
        }
```

```
    }
```

```
}
```

```
}
```

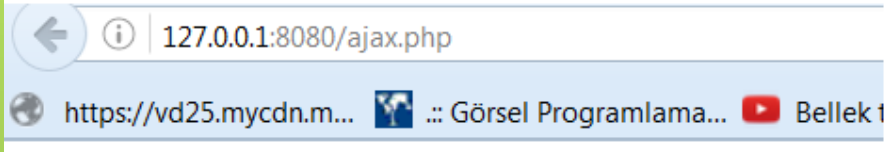
```
</script>
```

```
<div id="div1"> Butona Tıklanıldığında Veriler Bu Dive Yazılacak</div>
```

```
<button onclick="AJAXkullan()">AJAX'ı Kullanmak İçin Tıkla</button>
```

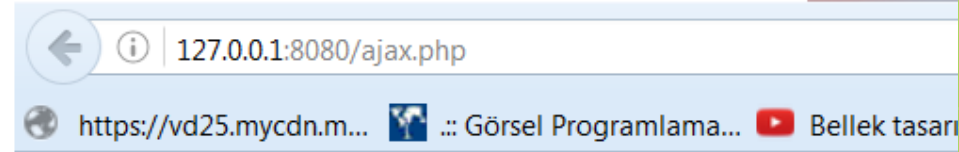
# PHP ve AJAX

Şimdi aşağıdaki ara yüzde butona tıkladığımızda AJAX fonksiyonumuzun çalışarak HTML div tagimizin içeriğini değiştirdiğini göreceksiniz.



Butona Tıklandığında Veriler Bu Dive Yazılacak

AJAX'ı Kullanmak İçin Tıkla



**Ajax Verilerin Gönderildiği Sayfayı Bulamadı**

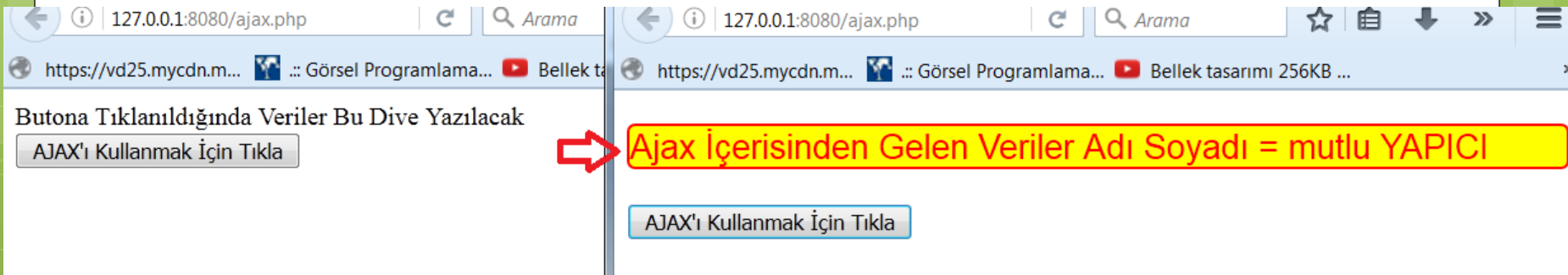
AJAX'ı Kullanmak İçin Tıkla

Tabi ki verilerimizi gönderdiğimiz **ajaxicinPHP.php** sayfası olmadığı için sonuç olarak nesnemizin **status** değeri **404** yani sayfa bulunamadı olarak döndü ve ekranda hata mesajını görüyoruz.

# PHP ve AJAX

Şimdi **ajaxicinPHP.php** sayfasını oluşturup kodlarımızı tekrar deneyelim.

**ajaxicinPHP.php** sayfasının içerisine yazacağımız herşey aslında sunucuda işlenip ekrana yazdırılacak değerleri oluşturacaktır. Örnek olarak sayfaya POST ile gönderdiğimiz ad ve soyad bilgisini alıp bunları bir paragraf tagi içerisinde birleştirip CSS ile biçimlendirerek gönderen kodları yazalım.



# PHP ve AJAX

ajaxicinPHP.php sayfasına ait kodlar.

```
1 <?php
2   $ad =$_POST["adi"];
3   $sadi =$_POST["sadi"];
4
5   echo<<<etiket
6 <style>
7   p{
8     color:red;
9     font-family:arial;
0     font-size:16pt;
1     background-color:yellow;
2     border:2px solid red;
3     border-radius:5px;
4   }
5 </style>
6 etiket;
7   echo   "<p>Ajax İçerisinden Gelen Veriler Adı Soyadı = $ad $sadi</p>";
8 >>
```

# PHP ve AJAX

Bu şekilde sunucu tarafında yapacağınız tüm işlemlere ait kodları AJAX ile ulaşacağınız PHP sayfasında yazmalısınız. Daha sonrada istemci tarafında AJAX nesnesi ile bu dosyaya istek gönderip sonuçları alarak sayfamızın istenen bölümünde java script sayesinde gösterebiliriz.

Bu aşamada kodlar biraz daha geliştirilip HttpRequest nesnesine ait status değerleri için gereken durumlar yazılabilir.

# PHP ve AJAX

Buraya kadar AJAX işlemlerini birinci yol olan Java Script ile yaptık. Şimdi de JQuery kütüphanesini kullanarak aynı işlemleri yapmayı deneyelim.

Jquery kütüphanesinde işlemler hazır fonksiyonlar halinde sunulduğu için biraz daha basitleştirilmiştir. Ama arka planda temelde aynı işlemler yapılmaktadır. JQuery de ajax sınıfına ait oldukça fazla parametre mevcuttur ancak biz temel birkaç tanesini göreceğiz.



# PHP ve AJAX

Öncelikle tabiki JQueryyi kullanabilmek için kütüphaneyi yüklemeliyiz. Daha sonra ajax sınıfına ulaşabilmek için `$.ajax({.....})` kodunu kullanıyoruz. Ve bu koda ait parametre değerlerini giriyoruz.

```
$.ajax({  
  type: "POST", /// Verilerin Gönderileceği Metod Tipi  
  url: "ajax.php", /// Verilerin Gönderileceği sayfanın adı  
  data: {ad:'Mutlu',sad:'YAPICI'}, /// Veriler  
  dataType: "HTML", /// Gelen Sonuç Verileri (Text mi HTML mi)  
  /// Veri gönderilmeden önce gerçekleşecek işlemler fonksiyonu  
  beforeSend: function(){ $('#div').HTML('<img src='load.gif'>') },  
  /// Sonuçlanınca gerçekleşecek işlemler fonksiyonu  
  success: function(cvp){ $('#div').HTML(cvp) },  
  /// Hata Oluştuğunda gerçekleşecek işlemler fonksiyonu  
  error: function(){ alert('Hata Meydana Geldi') },  
});
```

# PHP ve AJAX

## Önceki Örneğe ait JQuery AJAX kodları

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script>
$(document).ready(function() {
  $("button").click(function() {
    $.ajax({
      type:"POST", /// Verilerin Gönderileceği Metod Tipi
      url:"ajaxicinPHP.php", /// Verilerin Gönderileceği sayfanın adı
      data:{adi:"Mutlu",sadi:"YAPICI"}, /// Veriler
      dataType:"HTML", /// Gelen Sonuç Verileri (Text mi HTML mi)
      //// Veri gönderilmeden önce gerçekleşecek işlemler fonksiyonu
      beforeSend: function(){ $("#div1").html("<img src='load.gif'>") },
      //// Sonuçlanınca gerçekleşecek işlemler fonksiyonu
      success: function(cvp){ $("#div1").html(cvp) },
      //// Hata Oluştığında gerçekleşecek işlemler fonksiyonu
      error: function(){ alert("Hata Meydana Geldi") },
    });
  });
});
</script>
<div id="div1"> Butona Tıklanıldığında Veriler Bu Dive Yazılacak</div>
<button>AJAX'ı Kullanmak İçin Tıkla</button>
```

# KAYNAKLAR

- İnternet ortamı
- PHP ve AJAX Haydar TUNA
- A'dan Z'ye PHP Rıza ÇELİK

