

Ağ Programlama

Web Uygulama Mimarisi



Konular

- ✓ Web Uygulama Mimarisi
- ✓ Temel Bilgiler
 - ✓ İnternet
 - ✓ URI, URN, URL (Universal Resource Locator) ?
 - ✓ Web
- ✓ HTML (Hyper Text Markup Language)
- ✓ HTML5
- ✓ CSS (Cascading Style Sheets)
- ✓ CSS Kuralları
- ✓ Java Script
- ✓ Kaynaklar

Web Uygulama Mimarisi

<https://github.com/kamranahmedse/developer-roadmap>

- ✓ HTML (XHTML HTML5)
- ✓ CSS (Cascading Style Sheets)
- ✓ Client Side Scripting (İstemci Tarafı)
 - ✓ Java Script (dinamik, etkileşim, olaya yanıt, formların kontrolü) (Jquery, [Dojo](#), Ext js, Angular JS, Backbone, React ...)
- ✓ Server Side Scripting (Sunucu Tarafı)
 - ✓ nodeJS, php, jsp, servlet, cgi, asp, Cold Fusion ...
(daha dinamik, veritabanı sorgulama, gelişmiş güvenlik ve erişim kontrolü v.s.)
 - ✓ Web Uygulaması Geliştirme Çatıları
- ✓ SQL / noSQL veritabanı yönetim sistemleri
- ✓ XML, JSON, AJAX

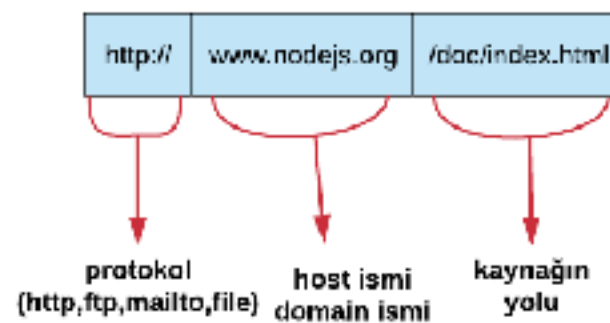
Temel Bilgiler

İnternet?

- ✓ İlk olarak 1960 larda Amerikada askeri amaçlı olarak kuruldu.
- ✓ İnternet e bağlı her bilgisayar IP olarak adlandırılan bir mantıksal adrese sahiptir. (194.27.72.1)
- ✓ İnternet üzerinde çalışan uygulamalar haberleşmek için soket adresi (ip:port) kullanırlar.

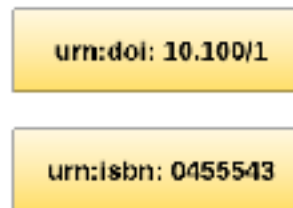
Temel Bilgiler (URI, URL, URN)

URL (Universal Resource Locator): Yerini ve nasıl erişeceğini gösterir.



URI (Universal Resource Identifier)

URL (Universal Resource Name): Kaynağın adını tanımlar.



Not: URL ile IP adresi arasındaki dönüşüm DNS (Domain Name System) Sunucular tarafından yapılır.

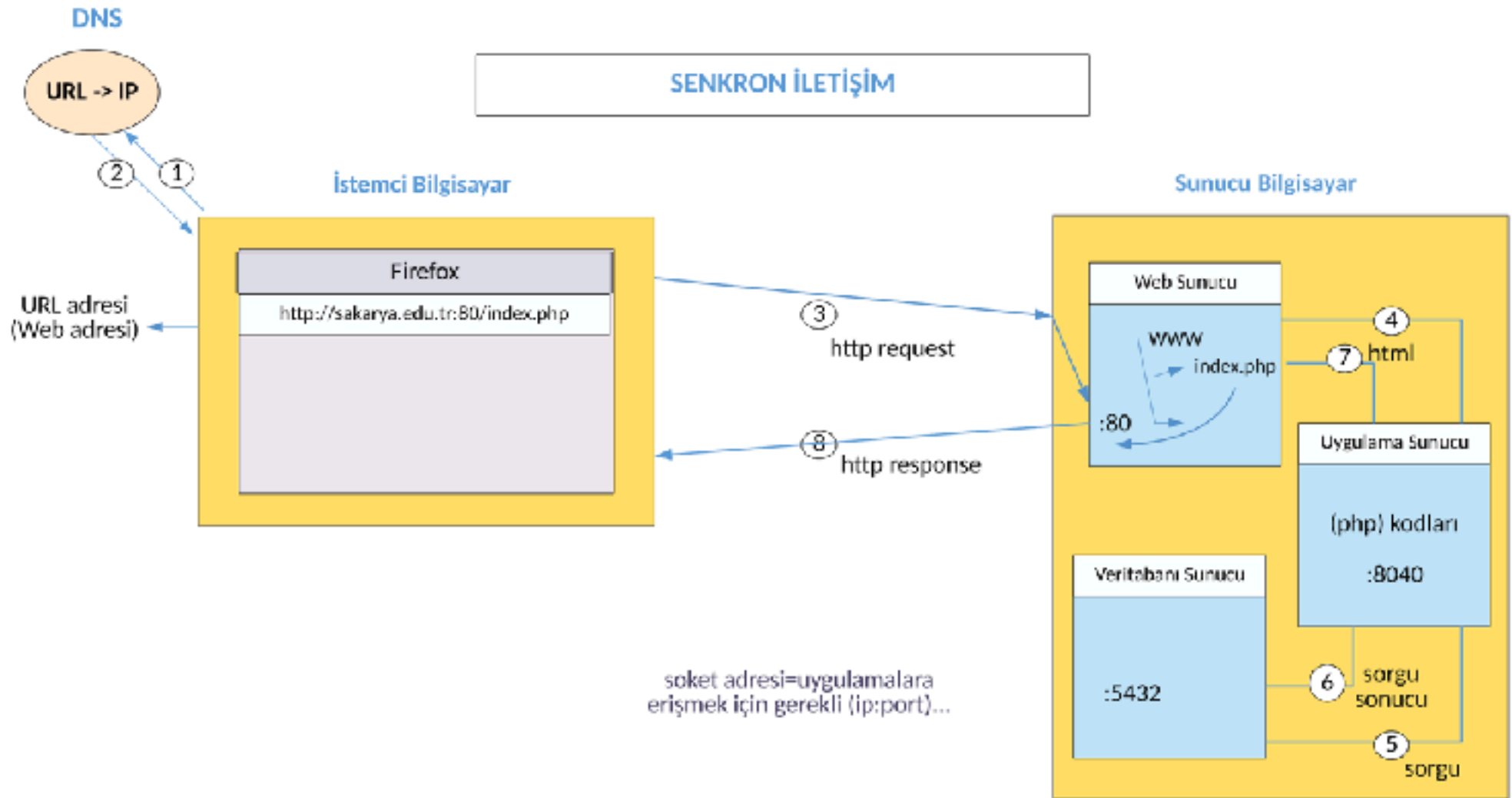
Temel Bilgiler

Web Nedir?

- ✓ İnternet üzerinde çok sayıda servis vardır (www (web), ftp, telnet, mail, irc ...)
- ✓ www= world wide web
- ✓ web servisinde haberleşmek için HTTP (Hyper Text Transfer Protocol) protokolü kullanılır.

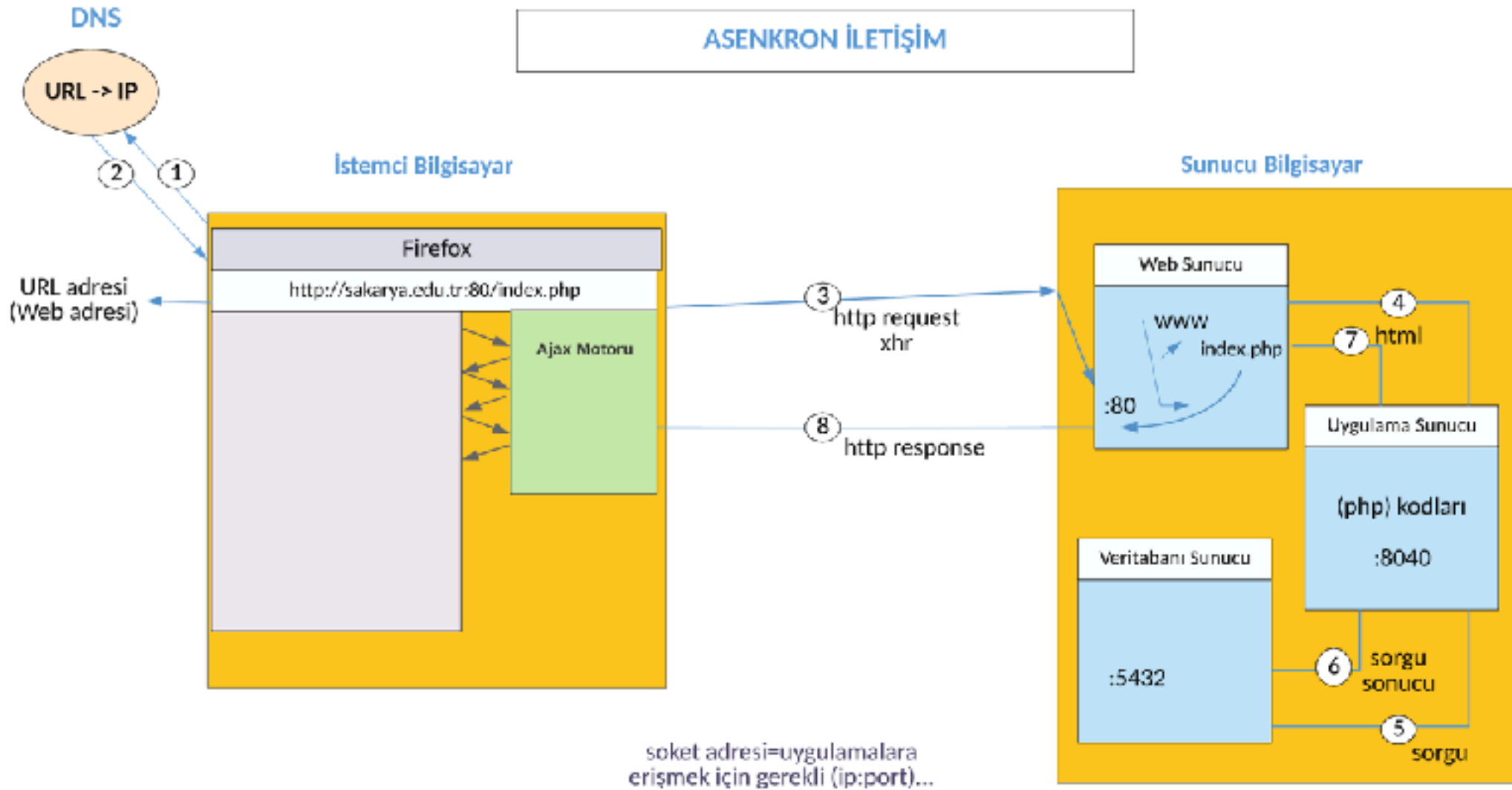
Temel Bilgiler

Web Uygulamaları Nasıl Çalışır (Senkron İletişim)?



Temel Bilgiler

Web Uygulamaları Nasıl Çalışır (Asenkron İletişim)?



HTML Belgelerinin Genel Yapısı



CSS (Cascading Style Sheets)

CSS html elementlerinin/sayfalarının **nasıl gösterileceğini** tanımlayan kurallardan oluşur.

- Html, web sayfalarının **içeriklerini ve anahatlarını** belirlerken CSS kuralları sayfanın **nasıl sunulacağıyla** ilgili kuralları tanımlar.
- Html 4.0 dan sonra eklenmiştir günümüzde çok yaygın olarak kullanılan bir standard haline gelmiştir.
- **External (Harici) Style Sheets** işleri kolaylaştırır(Tek bir dosyada yapılacak değişiklik tüm siteyi etkileyebilir. Oysa etiket özelliği ile biçileme işleri karmaşık hale dönüştürür.)
- External Style Sheets ayrı **CSS dosyalarında** saklanır. (.css)

Çoklu Stil Tanımlamaları

Öncelik Sıralaması

Öncelik 4 (en yüksek) den 1 (en düşük) e doğrudur

- 1- Browser default
- 2- External style sheet

```
<head>
  <link rel="stylesheet" type="text/css" href="Bicim/Bicim.css">
</head>
```

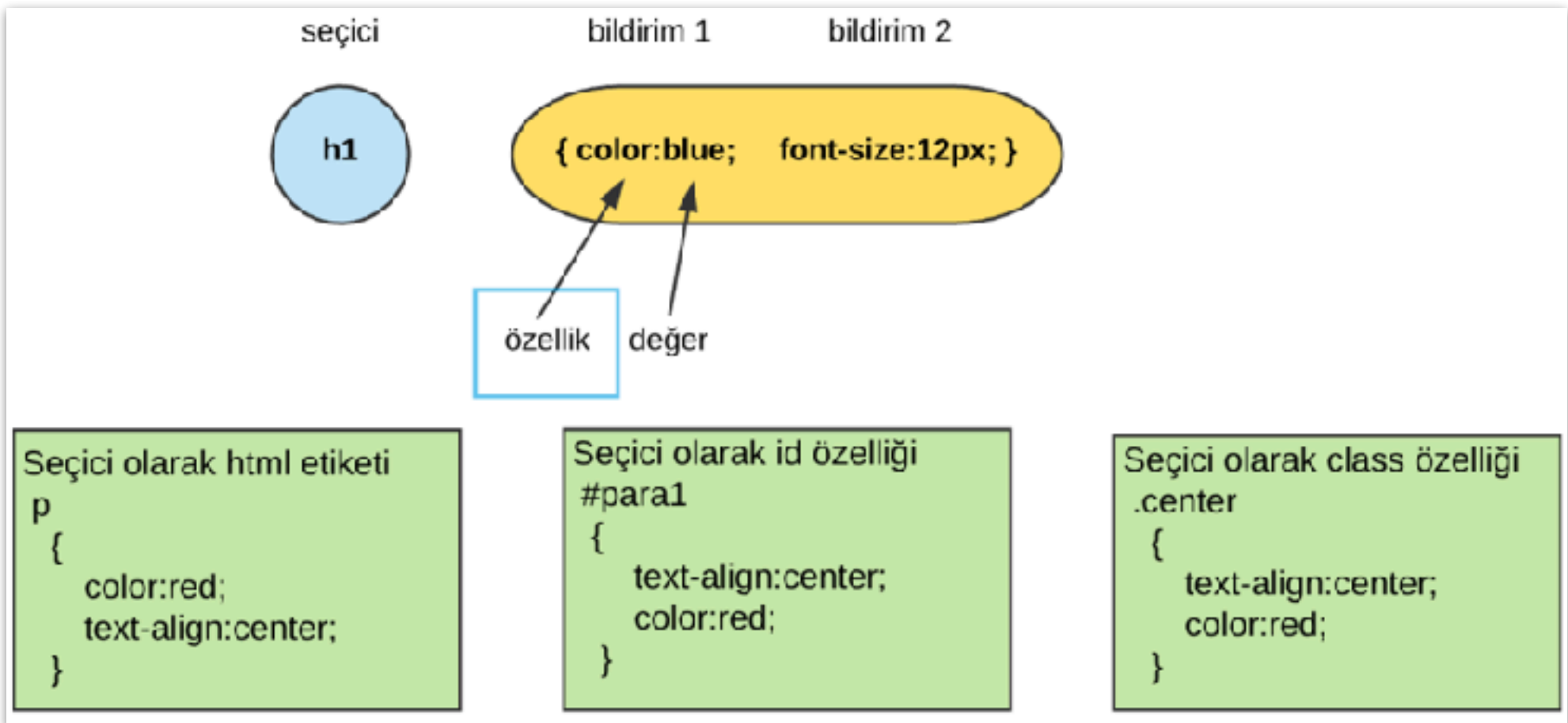
3- Internal style sheet (inside the <head> tag)

```
<head>
  <style>
    p {color:black;}
    h2 {margin-left:30px;}
    body {background-image:url("resimler/zemin.png");}
  </style>
</head>
```

4- Inline style (inside an HTML element)

```
<h2 style="color:red;margin-left:30px">Merhaba Dünya</h2>
```

CSS Kural Yapısı



JQuery

- ✓ İstemci tarafı Java Script çatısıdır.
- ✓ Java Script ile yapılan işleri daha az kod yazarak ve daha basit şekilde ele almak için geliştirilmiştir. (write less, do more)
- ✓ Yaygın olarak kullanılır. (Google, IBM, Microsoft, v.b. gibi şirketler tarafından kullanılır)
- ✓ JQuery nin kullanılabilesi için www.jquery.com sitesinden indirilip web sayfalarına aşağıdaki gibi eklenmelidir.

```
<script src="JS/jquery-1.9.1.js"></script>
```

CDN (Content Delivery Network)

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
```

JQuery Yazım Kuralları

Genel JQuery sözdizimi

`$(selector).action()`

```
$( "p" ).show( "1500" ); // p etiketli elementleri göster
$( ".mesaj1" ).hide();   // class özelliği "mesaj1" olan
                           // elementleri gizler
$( "#soyadi" ).focus(); // id özelliği soyadi olan elemente
odaklanır
```

Tüm JQuery ifadeleri aşağıdaki ifadenin(belge yüklendi) içerisinde yer almalıdır. Böylece henüz oluşturulmayan elementler için herhangi bir işlemin yapılamaması garanti edilmiş olur.

```
$(document).ready(function()
{
    //JQuery
} );
```

Olaylar (Events)

Fare Olayları	Klavye Olayları	Form Olayları	Belge/Pencere Olayları
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

```
$(document).ready(function(){  
    $("#kaydet").click(function(){  
        alert("Kayıt eklenmiştir!");  
    });  
});
```

AJAX Nedir?

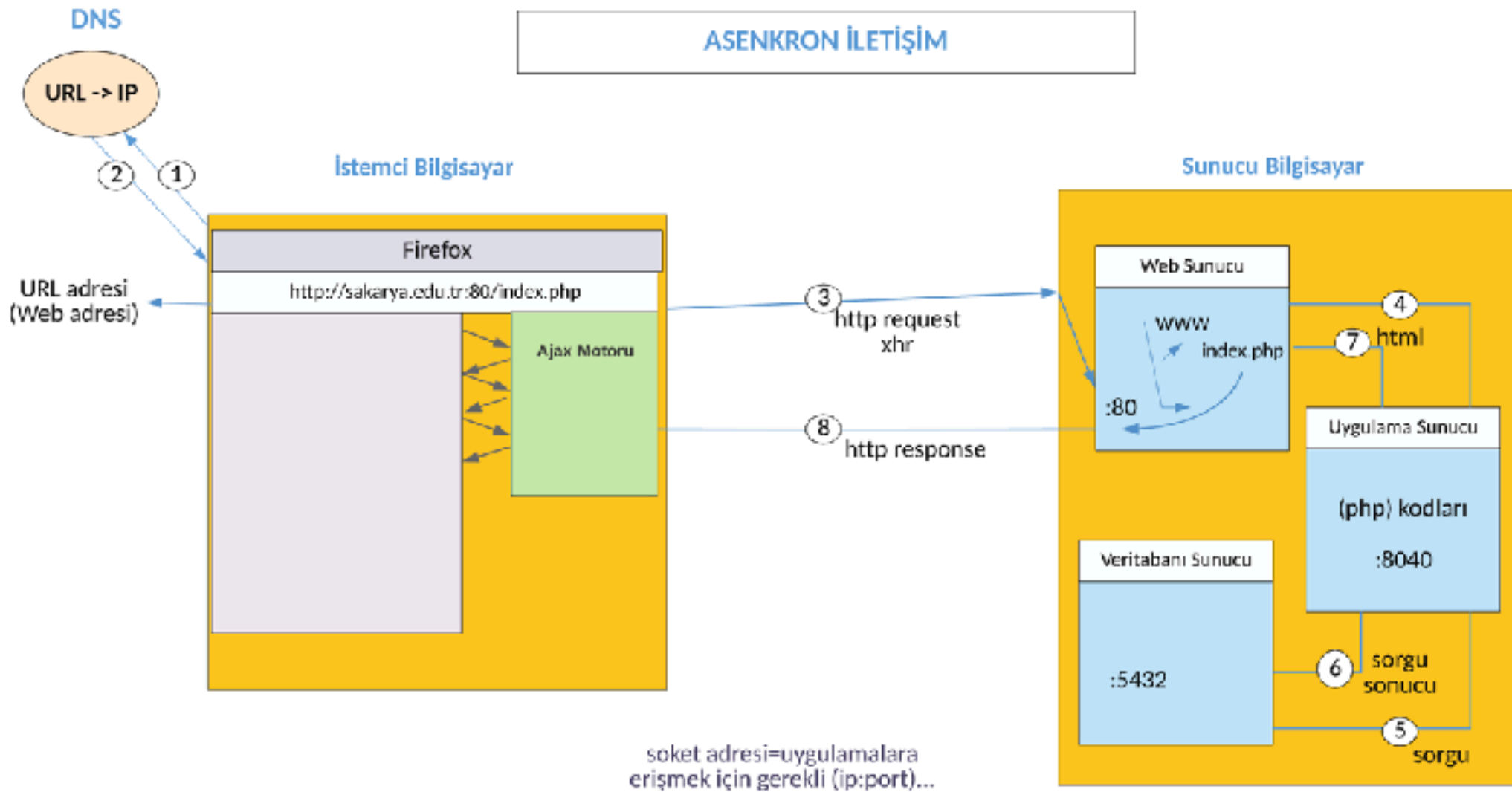
✓ AJAX = Asynchronous(Asenkron)JavaScript and XML.

✓ İstemci sunucu arasındaki haberleşme, istemci tarafında oluşturulan bir nesne (AJAX motoru) tarafından yürütülür.

✓ Web sayfasının tamamının yeniden yüklenmesine gerek kalmadan yalnızca gerekli olan kısmının güncellenmesini sağlar. (Ağ trafiğinin azalması, daha az işlemci gücüne ihtiyaç duyulması, hız artışı, daha esnek, daha modüler uygulamalar)

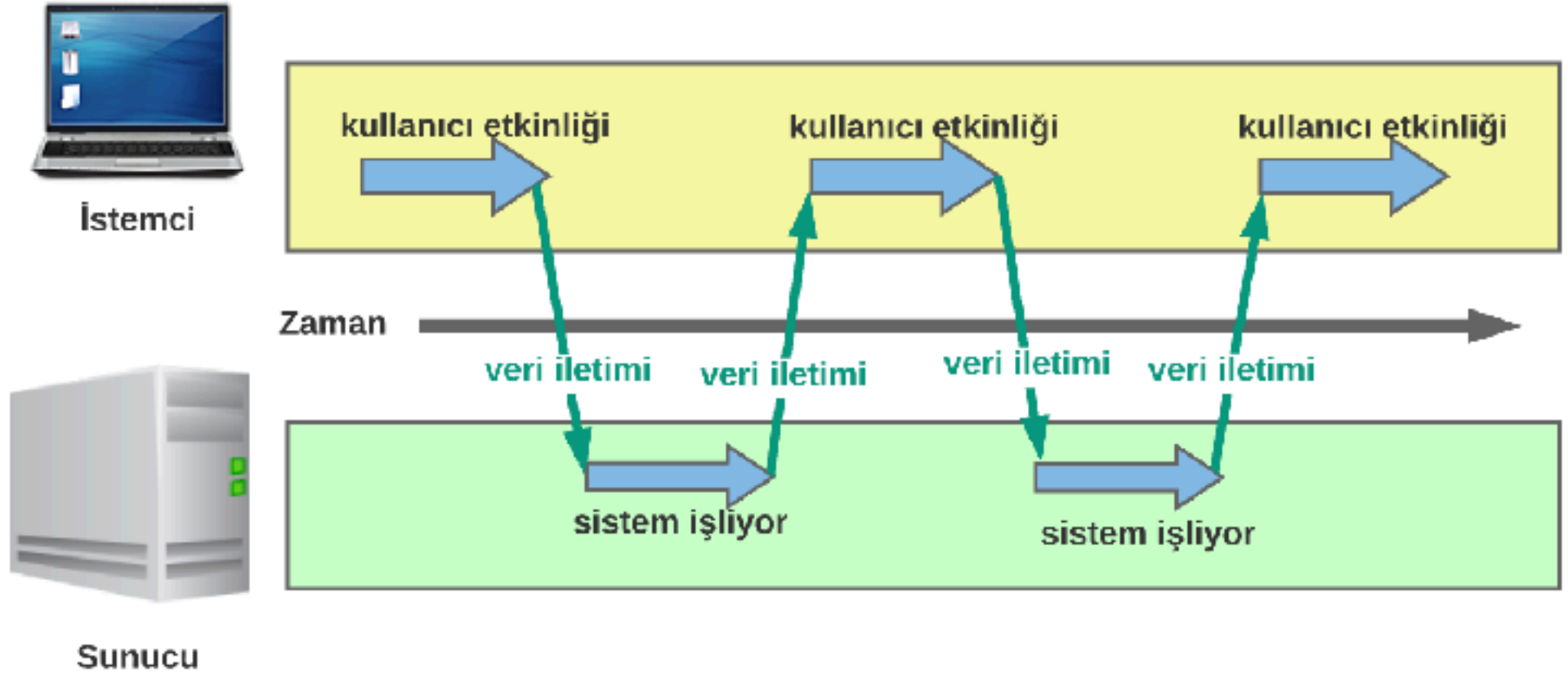
AJAX ile daha hızlı, daha etkileşimli ve dinamik web sayfaları oluşturulabilir. ((yazarken kaydet, yazarken ara ...))

AJAX - Nasıl Çalışır



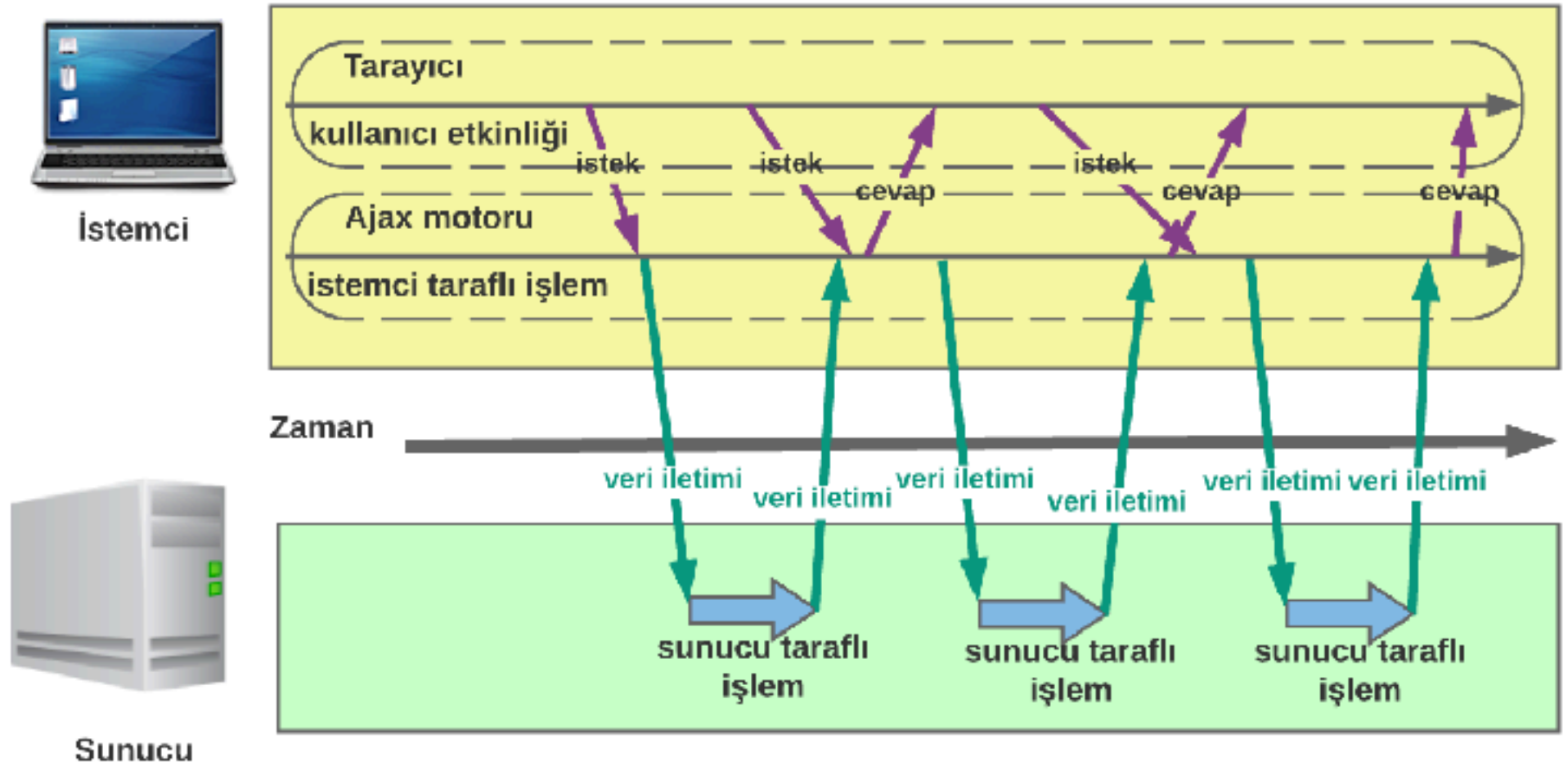
AJAX - Nasıl Çalışır

Klasik Web Uygulama Modeli (Senkron)



AJAX - Nasıl Çalışır

AJAX Web Uygulama Modeli (Asenkron)



AJAX

Ajax içerisinde aşağıdaki teknolojiler kullanılır:

- XMLHttpRequest sınıfından oluşturulan nesne (JS ile istemci tarafında tanımlanır ve istemci-sunucu arasındaki haberleşmeyi sağlar)
- JavaScript/DOM (Belge nesne modeli)
- CSS (Stil Şablonu)
- XML, JSON (Genellikle veri transferinde kullanılan veri gösterim biçimleri)

XMLHttpRequest Nesnesi

- ✓Günümüz tarayıcılar ının tamamı XMLHttpRequest nesnesini destekler.
- ✓XMLHttpRequest nesnesi istemci-sunucu arasındaki iletişimi sağlar

➤XMLHttpRequest Nesnesi Oluşturma

- ✓Modern Browserlarda XMLHttpRequest sınıfı bulunmaktadır.
- XMLHttpRequest Nesnesi (Ajax Motoru) oluşturmak için :

```
variable=new XMLHttpRequest();
```

```

<script>
function listele()
{
    if (window.XMLHttpRequest)
    {
        // Firefox, Chrome, Opera, IE7+, Safari v.s.
        xmlhttp = new XMLHttpRequest(); // readystate:0
        //Ajax engine oluşturuluyor... sunucuyla olan tüm iletişim bu nesne üzerinden
        //asenkron olarak gerçekleştirilecek...
    }

    xmlhttp.onreadystatechange = function() //istek durumunu tutan readystate her değiştiğinde
    //bu fonksiyon tetikleniyor. readystate 0-4 arası değer alıyor. 4->istek sonucu
    //istemci tarafından alındı.
    {
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200)// status ile sunucunun isteğe
        //verdiği yanıtın durumu tutuluyor. 200->kaynak bulundu ve istek başarılı olarak karşılandı
        // 404 -> kaynak bulunamadı, 304 -> get isteği ise ve değişiklik yoksa önbellekten al,
        //500-> dahili sunucu hatası (sunucuda beklenmeyen bir durumla karşılaşıldı)
        // 403 -> yetkisiz erişim
        {
            document.getElementById("ortaForm").innerHTML = xmlhttp.responseText;
            //istek sonucu gelen veriler xmlhttp.responseText değişkenine aktarılıyor ve
            //dom ile bu veriler sayfa yeniden güncellenmeden ortaform id değerine sahip
            //div içerisinde gösteriliyor.
        }
    }
    xmlhttp.open("GET", "1Ajax1.php", true);//istek ayarları yapılıyor. readystate:1
    xmlhttp.send();//istek gönderiliyor. readystate:2
}
</script>

```

➤ Onreadystatechange

- ✓ readyState değeri her değiştiğinde onreadystatechange fonksiyonu çalıştırılır.
- ✓ readyState özelliği XMLHttpRequest sınıfından oluşturulan nesnenin durum bilgisini tutar.
- ✓ XMLHttpRequest nesnesinin 3 tane önemli özelliği vardır.

Özellik	Tanım
onreadystatechange	readyState özelliği değiştiği zaman otomatik olarak çağırılacak fonksiyonu saklar.
readyState	XMLHttpRequest in durumunu tutar. 0 ile 4 arasında değişir. 0: XMLHttpRequest sınıfından nesne oluşturulduğu zaman. 1: istek oluşturulduğunda. open () çağrıldı 2: istek gönderildi. send() çağrıldı 3: sunucu tarafında istek işletiliyor ve yanıt hazırlanıyor 4: yanıt hazır. istemci tarafından alındı.
status	200: Sayfa başarıyla gönderildi 404: Sayfa bulunamadı. 403: Erişim izni yok. 500: Dahili sunucu hatası 304: Değiştirilmemiş, önbellekten alınabilir.

- ✓ readyState 4 ve status 200 olduğunda yanıt hazırdır.

Kaynaklar

✓ www.w3schools.com