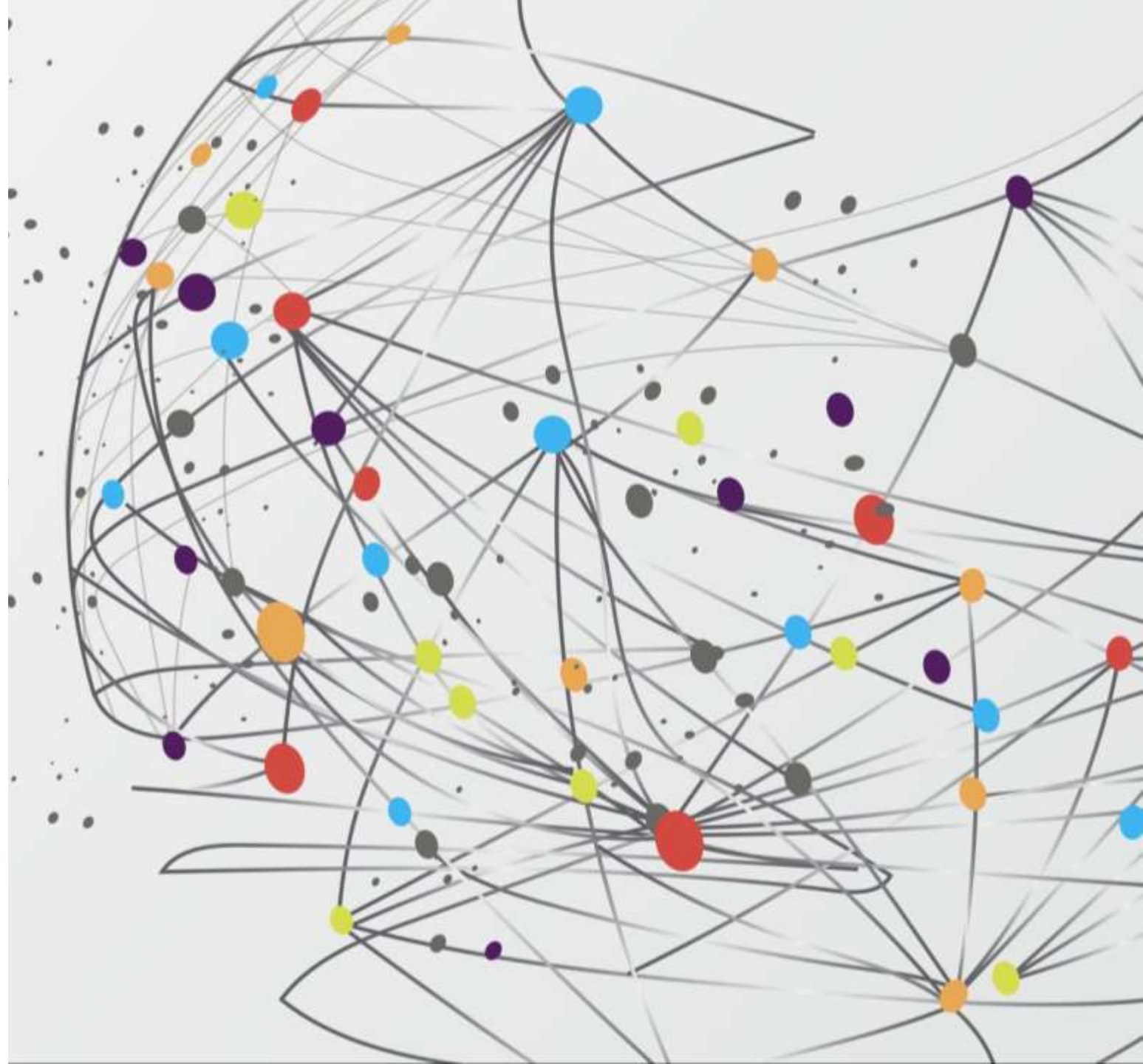

WEB TABANLI PROGRAMLAMA

BÖLÜM 6

XML VE ÖZELLİKLERİ
JSON VE ÖZELLİKLERİ

Prof. Dr. Turgay Tugay Bilgin

turgay.bilgin@btu.edu.tr



GENEL BAKIŞ...

6.1) XML Dilinin Özellikleri

6.2) XML ile HTML Arasındaki Farklar

6.3) XML Dilinin Avantajları

6.4) XML Dilinin Dezavantajları

6.5) XML Belgeleri Nasıl Çalışır?

6.6) XML Dilinde Etiketler

6.7) XML Belgeleri Oluşturma

6.8) XML Belgede Ağaç yapısı

6.9) XML Etiket Tanımlama Kuralları

6.9.1) Kapatma Etiket Kuralı

6.9.2) Büyük-Küçük Harf Duyarlılığı

6.9.3) Etiketlerin Harf ile Başlaması

6.9.4) Kök Eleman Zorunluluğu

6.10) Etiket Yerine Nitelik Kullanma

6.11) XML Dokümanlarında Stil Oluşturma

6.12) Özet

(6.1) XML DİLİNİN ÖZELLİKLERİ

- Extensible Markup Language (Geniřletilebilir İřaretleme Dili, kısaca **XML**), hem insanlar hem bilgisayarlar tarafından kolayca anlaşılabilen ve yorumlanabilen belgeler oluřturmayı saėlayan bir iřaretleme dilidir [9]. W3C konsorsiyumu tarafından tanımlanmıř standart bir dildir. Bu özelliėi ile veri saklamanın yanında farklı sistemler arasında veri alıřveriři yapmaya yarayan bir ara katman görevi de görür.

- XML belgelerinde veya XML içeriėini sembolize etmek için genellikle řekil 6.1'deki simge kullanılır.



řekil 6.1 XML belge logosu

-
- XML dilinin temel bazı özellikleri şunlardır:

→ XML dili HTML gibi bir metin işaretleme dilidir, programlama dili değildir, değişken tanımlama, döngü oluşturma gibi programlama dili özellikleri taşımaz.

→ XML dili veriyi farklı yazılım platformları arasında taşımak için geliştirilmiştir, verinin kullanıcıya sunumu için geliştirilmemiştir. Verinin sunumu için CSS veya XML dili için geliştirilmiş biçimlendirme dili (XSL) kullanılmaktadır.

→ XML dilinin önceden tanımlı etiketleri yoktur. Kullanıcılar kendi ihtiyaçlarına göre kendi etiketlerini kendileri tanımlarlar.

→ XML dili kendi kendini tanımlayıcı (self descriptive) bir yapıdadır. Etiketleri inceleyen bir insan veya etiketleri işleyen bir yazılım dokümanın içinde ifade edilen bilgiyi başka bir tanıma gerek kalmadan kolayca anlayabilir.

→ XML dili W3C konsorsiyumu tarafından tavsiye edilen bir veri aktarım dilidir.

(6.2) XML İLE HTML ARASINDAKİ FARKLAR

- XML dili HTML dilinin yerine geçmek üzere tasarlanmamıştır. HTML dilinin yaptığı işi yapmaz. XML dilinin geliştirilmesinde 2 temel probleme odaklanılmıştır :
 - XML dili verinin depolanması ve farklı platformlar arasında taşınması için geliştirilmiştir. XML dilinde odak noktası verinin içeriğidir.
 - HTML dili ise verinin görüntülenmesi ve sunumu için geliştirilmiştir. HTML dilinde odak noktası verinin nasıl görüntüleceğidir.

(6.3) XML DİLİNİN AVANTAJLARI

- XML dilinin en önemli avantajları aşağıda maddeler halinde verilmiştir:
 - XML dili hiyerarşik bir metin yapısına sahiptir.
 - XML, diğer internet teknolojileri ile birlikte çalışabilir.
 - XML dilinin lisans problemi yoktur, ticari veya ticari olmayan her türlü projede kullanılabilir.
 - XML, insanların okuyabildiği bir dildir.
 - XML belgelerini görüntülemek için özel bir yazılıma ihtiyaç yoktur, herhangi bir web tarayıcısında veya metin editöründe kolayca görüntülenebilir.

(6.4) XML DİLİNİN DEZAVANTAJLARI

- XML dili, birçok avantajına rağmen bazı dezavantajlara da sahiptir. En önemli dezavantajları:
 - XML belgeleri diskte fazlaca yer kaplayabilirler, bu durum işlemci ve bant genişliği açısından olumsuz bir durum olabilir.
 - XML sadece bir doküman dilidir, programlama dili değildir.
 - Her oluşturulan XML dökümanındaki etiketler kullanıcının kendi oluşturduğu etiketler olduğundan, başka uygulamalar tarafından anlaşılabilmesi zor olabilir.
 - XML belgeleri metin veri için harika sonuçlar verir ancak sayısal veri-lerin işlenmesini yavaşlatabilirler.

Dikkat ☞ *XML'in yerine geçebilecek başka veri aktarım yöntemi var mı?*

Günümüzde JSON (Javascript Object Notation) adı verilen veri aktarım standardı daha az yer kapladığı için XML diline göre daha fazla tercih edilmeye başlamıştır. JSON ile ilgili detaylı bilgi aşağıdaki web sitesinden elde edilebilir.

<http://www.json.org/json-tr.html>

(6.5) XML BELGELERİ NASIL ÇALIŞIR

- XML belgeleri aslında hiçbir şey yapmazlar. Web tarayıcı da XML belgeleri işleyemez, çünkü XML belgelerin içindeki etiketleri web tarayıcı yorumlayamaz, bunun yerine olduğu günü ekranda görüntüler. XML belgeleri yalnızca veriyi istediğimiz bir formatta başka uygulamaya aktarmayı sağlar. XML dilinin kullanımına bir örnek olması açısından Mehmet'in Volkan'a XML biçiminde mektup gönderdiğini düşünelim. Bu durumda kullanacağımız XML belgesi Kod 6.2'deki gibi olabilir.

```
<Zarf>
  <Kimden>Mehmet Bilgin</Kimden>
  <Kime>Volkan Tunalı</Kime>
  <Adres>Mimar Sinan Mah. Eflak Caddesi No:177
    Yıldırım BURSA</Adres>
  <Mektup>Nasılsın Volkan, umarım iyisindir,
    çalışmalarında başarılar.</Mektup>
</Zarf>
```

Kod 6.1. XML kullanımına örnek

- Kod 6.2'de görüldüğü gibi bu mektupta hangi bilgilerin gönderildiği ve bu bilgilerin nerede başlayıp nerede bittiği kendimizin tanımladığı etiketler sayesinde kolayca anlaşılmaktadır. Örneğin mesajı gönderen kişinin adı <Kimden>...</Kimden> etiketleri arasında bulunmakta, benzer şekilde alıcı da <Kime>...</kime> etiketleri arasında yer almaktadır. Adres bilgisi ve mektup içeriği de ilgili etiketlerden kolayca anlaşılmaktadır. Bu belgeyi alan bir yazılım, kendisine daha önce bu etiketlerin tanıtılmış olması durumunda ilgili alanlardaki bilgileri okuyarak veri tabanına kaydedebilir veya belirtilen adrese mektubu gönderebilir.

(6.6) XML DİLİNDE ETİKETLER

- Kod 6.2’de verilen örnekte göreceğiniz üzere XML dilinde kendi etiketlerinizi kendiniz tanımlarsınız. Örnekte <kimden>, <kime>, <Adres> ve <Zarf> gibi etiketleri biz kendi verimizi tanımlayacak şekilde tanımladık. Dolayısıyla bu etiketler herhangi bir uygulamaya tanıtılmadığı sürece bir işe yaramayacaktır. XML dilinde önceden belirlenmiş hiçbir etiket yoktur, bütün etiketler XML dokümanı yazarı tarafından tanımlanır.



(6.7) XML BELGELERİ OLUŞTURMA

- Kod 6.1’de verilen örnekte bir XML dosya oluşturulmuştu, fakat bir XML dosyanın W3C standartlarına uygun olabilmesi için ilk satırda XML sürümü zorunlu olarak bulunmalı ve opsiyonel olarak içindeki metnin hangi ülke dilinde yazıldığını belirten dil kodlaması tanımına da sahip olabilir. Bu tanımlama işlemi Kod 6.3’deki gibi gerçekleştirilir.

```
<?xml version="1.0" ?>
<Zarf>
    <Kimden>Mehmet Bilgin</Kimden>
    <Kime>Volkan Tunalı</Kime>
    <Adres>Mimar Sinan Mah. Eflak Caddesi No:177
        Yıldırım BURSA</Adres>
    <Mektup>Nasılsın Volkan, umarım iyisindir,
        çalışmalarında başarılar.</Mektup>
</Zarf>
```

Kod 6.2. W3C uyumlu XML tanımlama

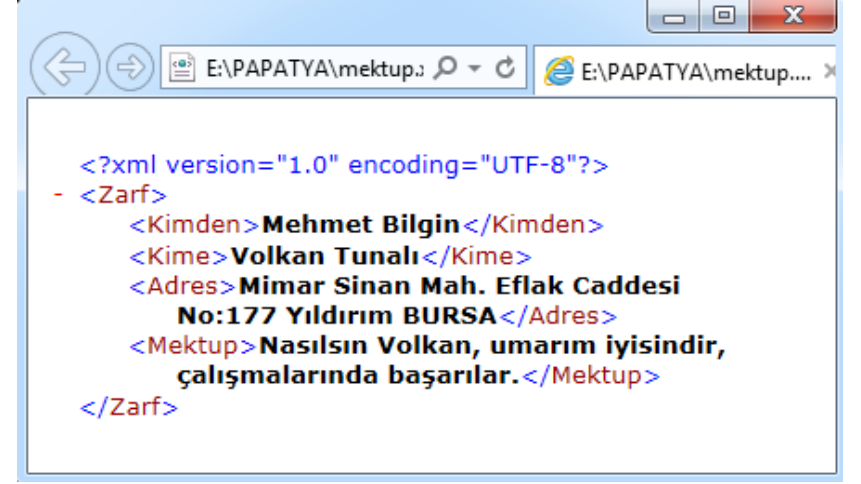
-
- Eğer dil kodlaması bilgisi de eklenecekse `version` niteliği sonrası eklenebilir. Bunun örneği Kod 6.3'te görülmektedir.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Zarf>
  <Kimden>Mehmet Bilgin</Kimden>
    <Kime>Volkan Tunalı</Kime>
    <Adres>Mimar Sinan Mah. Eflak Caddesi No:177
      Yıldırım BURSA</Adres>
    <Mektup>Nasılsın Volkan, umarım iyisindir,
      çalışmalarında başarılar.</Mektup>
</Zarf>
```

Kod 6.3. W3C uyumlu XML belgeye dil kodlaması ekleme

- Dil kodlaması olarak tüm ülke dilleri için genel olarak UTF-8 seçilebildiği gibi Türkçe içerikler için Türkçe dil kodlaması standardı olan ISO-8859-9 de yazılabilir. Örnekte UTF-8 tercih edilmiştir. Eğer hiçbir şey belirtilmezse web tarayıcı veya XML yorumlayıcı uygulamanın varsayılan dil ayarları ile görüntülenir.

- XML belgelerini görüntülemek için Notepad veya Notepad++ gibi basit metin editörleri kullanılabilse de işletim sistemleri genellikle bu belgeleri Web tarayıcılar ile açacaktır. Kod 6.4'deki örnek 3 farklı web tarayıcıda çalıştırıldığında elde edilen çıktı Şekil 6.2, Şekil 6.3 ve Şekil 6.3'deki gibi olacaktır. Şekil 6.2'de Microsoft Internet Explorer ile görüntülenen XML belgesinde sürüm bilgisi ekranda görünmekte ve kullanıcı tanımlı etiketler farklı bir renkte görünmektedir.



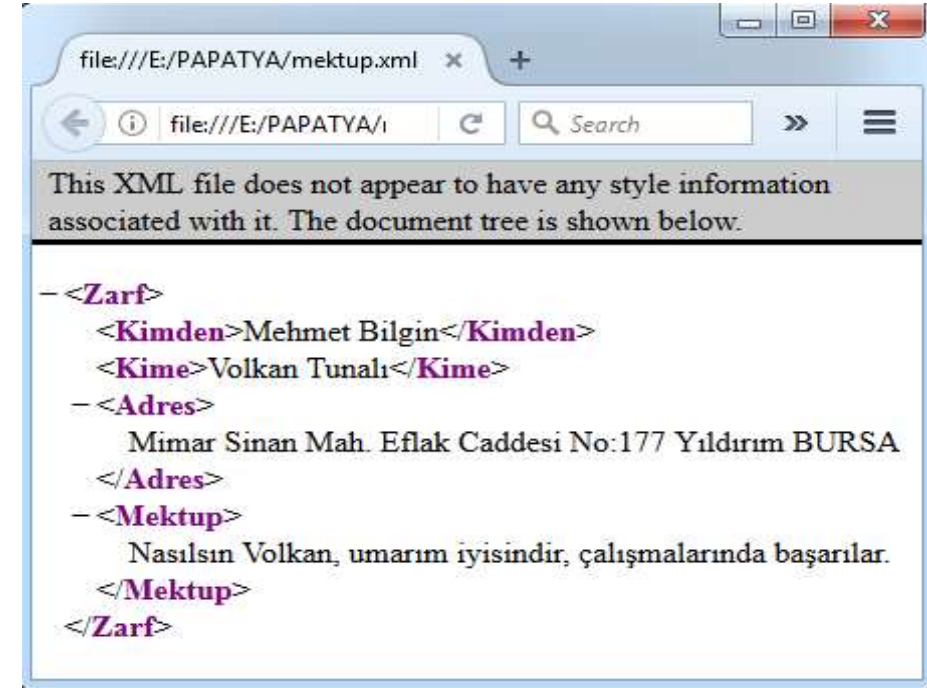
Şekil 6.2 XML belgenin Microsoft Internet Explorer ile görüntülenmesi

- Şekil 6.3'de Google Chrome ile görüntülenen XML belgesinde version bilgisi ekranda görünmüyor ve kullanıcı tanımlı etiketler farklı bir renkte görünmektedir. Ayrıca XML belgeye eşlik eden bir stil dosyası bulunmadığını belirten bir bilgi mesajı görüntülenmektedir. “XML dosyaları ile stil dosyası kullanımı” konusundan ünite sonunda bahsedilecektir.



Şekil 6.3 XML belgenin Google Chrome ile görüntülenmesi

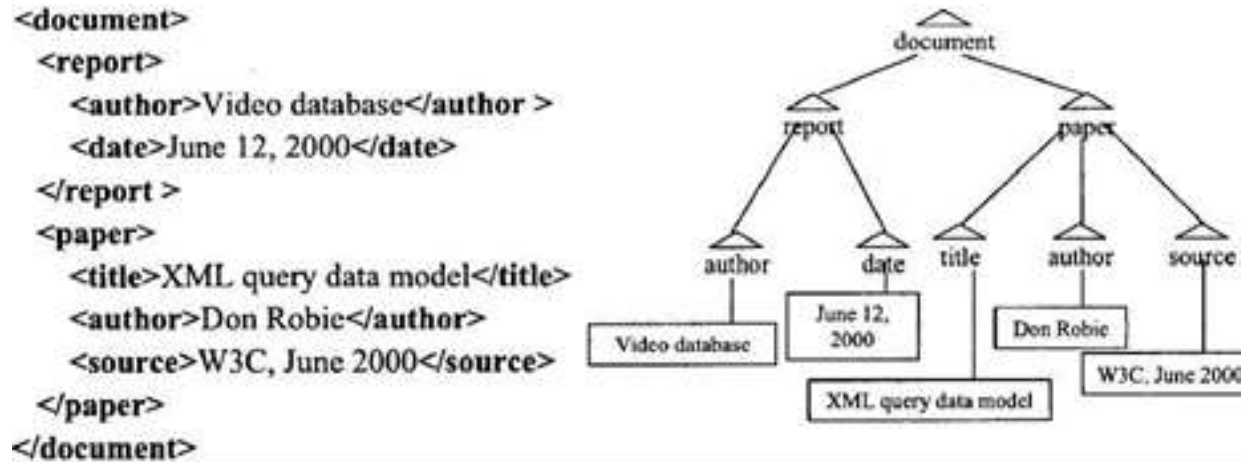
- Şekil 6.4’de Mozilla Firefox ile görüntülenen XML belgesinde de aynı şekilde sürüm bilgisi ekranda görünmüyor ve kullanıcı tanımlı etiketler farklı bir renkte görünmektedir. Ayrıca XML belgeye eşlik eden bir stil dosyası bulunmadığını belirten bir bilgi mesajı görüntülenmektedir.
- Microsoft Internet Explorer ve Mozilla Firefox tarayıcılarında bir XML etiketinin yanındaki küçük “-“ işaretine tıklayarak etikete bağlı alt etiketlerin gizlenmesini veya gösterilmesini sağlayabilirsiniz. Google Chrome tarayıcısında ise aynı işlevi gri ufak “aşağı ok” tuşu sunmaktadır.



Şekil 6.4 XML belgenin Mozilla Firefox ile görüntülenmesi

(6.8) XML BELGEDE AĞAÇ YAPISI

- XML belgelerde en dıştaki tüm etiketleri dıştan kapsayıcı elemana kök eleman (root element) adı verilir. Bu kök eleman diğer tüm elemanların ebeveyni (parent) durumdadır. Bu tasarıma XML belge ağacı (document tree) adı verilir. Ağaç, kök elemandan başlar ve en alt seviye elemana doğru iç içe ilerler. Bu durum Şekil 6.5'deki örnek XML belge ve buna karşılık gelen Ağaç yapısında açıkça görülmektedir.



Şekil 6.5 XML belgede ağaç yapısı [10]

-
- Şekil 6.5'deki örnekte <document> elemanı kök eleman, diğer bir ifadeyle <report> ve <paper> elemanlarının ebeveyni , <report> ve <paper> eleman-ları ise <document> elemanının çocukları (child) durumundadır.
 - XML belgelerindeki yapı genel hatlarıyla şu formül ile ifade edilebilir:

```
<kök>  
  <çocuk>  
    <torun>...</torun>  
  </çocuk>  
</kök>
```

- Ağaç yapısında bir de kardeş kavramı vardır. Şekil 6.5'de <report> ve <paper> aynı ebeveyne sahip olduklarından kardeş (sibling) olarak adlandırılırlar.

(6.9) XML ETİKETİ TANIMLAMA KURALLARI

- XML dilinde etiket tanımlarken bazı basit kurallara uymanız gerekir. Bu kurallara uymadan yazılmış XML dokümanları başta web tarayıcılar olmak üzere XML desteğine sahip birçok uygulamada hata oluşturacaktır. Uymanız gereken kurallar aşağıda verilmiştir.

6.9.1 Kapatma Etiketi Kuralı : Bütün XML etiketleri bir kapatma etiketine sahip olmalıdır. HTML dilinde `<hr/>` veya `
` gibi bazı etiketlerin kapatma etiketi yoktur. Fakat XML dilinde kendi oluşturduğunuz tüm etiketlerin kapatma etiketi olması zorunludur. Aksi halde XML belgenizi web tarayıcılarda çalıştırmaya kalktığınızda hata oluşacaktır.

Dikkat ✍ *XML belgelerin kurallara uygunluğu nasıl denetlenebilir?*

XML belgelerinizi hazırladıktan sonra, kurallara uygunluğunu denetlemek amacıyla Firefox Web tarayıcısı ile çalıştırmanızı öneririz. Hataların oluştuğu konumu gösterebilme yeteneği Firefox tarayıcısını diğerlerinden ayırmaktadır.


```
<?xml version="1.0" encoding="UTF-8" ?>

<Zarf>

    <Kimden>Mehmet Bilgin

    <Kime>Volkan Tunalı</Kime>

    <Adres>Mimar Sinan Mah. BURSA</Adres>

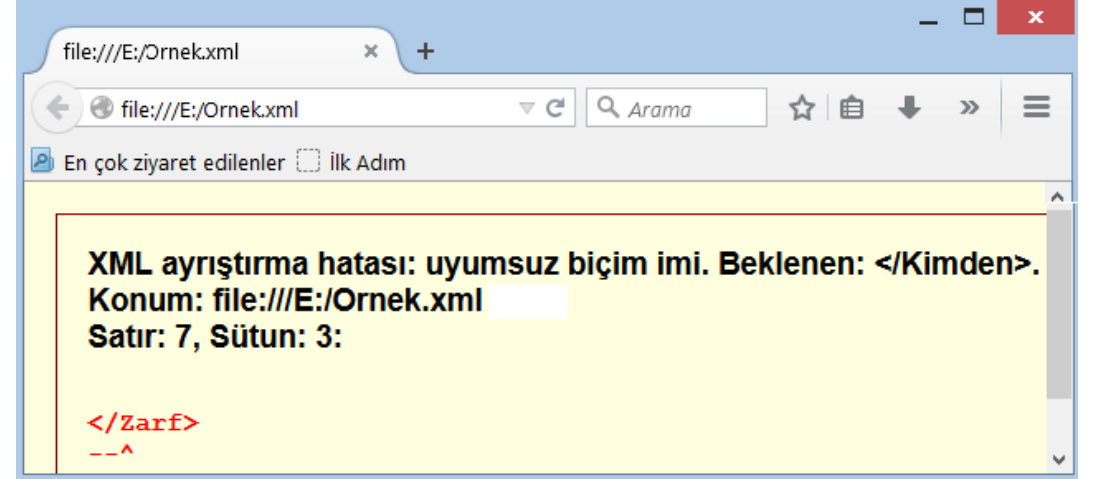
    <Mektup>Herkesel Selamlar.</Mektup>

</Zarf>
```

Kod 6.4. Hatalı XML belge örneđi

Hatırlatma XML belgelerinin ilk satırında yazılan `<?xml version="1.0" ?>` ifadesinin kapatma işareti yoktur, bunun dışındaki tüm kendi oluşturduğunuz etiketler kapatılmalıdır.

Yandaki hatalı XML dosya kodlarını **ornek.xml** adıyla kaydederek Firefox web tarayıcısında çalıştırdığınızda Şekil 6.6'daki görüntü elde edilir.



Şekil 6.6. XML belgede eksik etiket

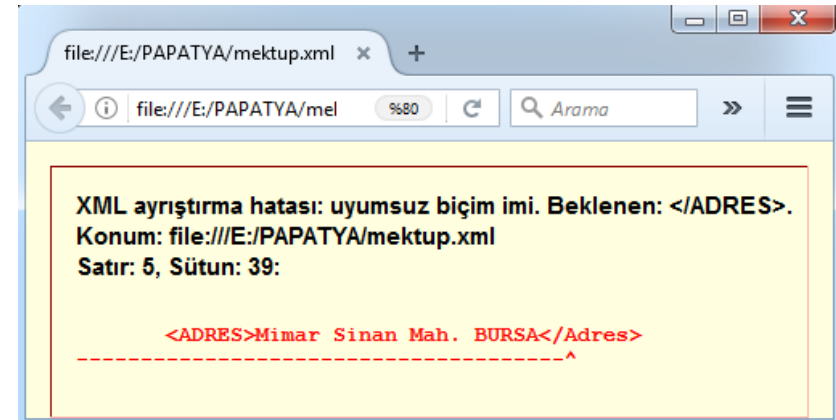
Kod 6.4'te `<Kimden>` etiketi kapatılmamıştır. Firefox ile çalıştırıldığında “**XML ayrıştırma hatası: uyumsuz biçim imi: Beklenen `</Kimden>`**” şekilde bir mesaj ile `</Kimden>` kapatma işaretinin eksik olduğunu bildirmektedir.

6.9.2 Büyük-Küçük Harf Duyarlılığı : XML etiketleri büyük-küçük harfe duyarlıdır. Bir etiketi nasıl başlattıysanız aynı şekilde kapatmalısınız. Başlatma ve bitiş etiketlerinde büyük-küçük harf uyumu aynı olduğu sürece ister büyük ister küçük harfli etiketler oluşturabilirsiniz.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Zarf>
    <Kimden>Mehmet Bilgin<Kimden>
    <Kime>Volkan Tunalı</Kime>
    <ADRES>Mimar Sinan Mah. BURSA</Adres>
    <Mektup>Herkes'e Selamlar.</Mektup>
</Zarf>
```

Kod 6.5. Hatalı XML belge örneği

Yandaki hatalı XML dosya kodlarını **ornek.xml** adıyla kaydederek Firefox web tarayıcısında çalıştırdığınızda Şekil 6.7'deki görüntü elde edilir.



Şekil 6.7 XML belgede büyük-küçük harf uyumsuzluğu

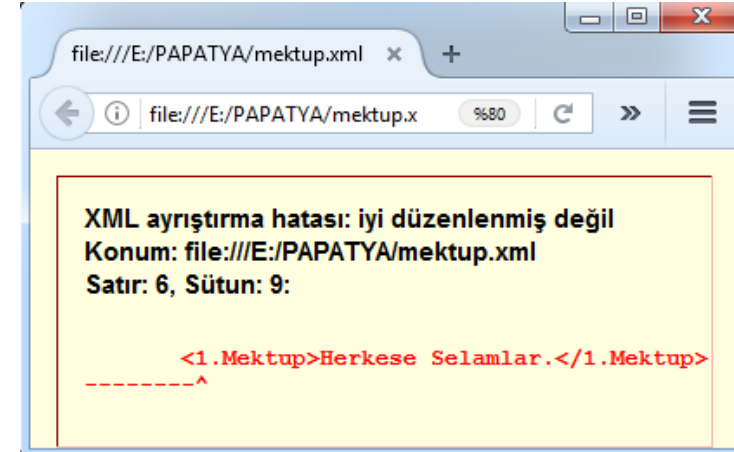
- Şekil 6.7'deki XML dosyada adres bilgisi <ADRES> etiketi ile başlanmış fakat </Adres> etiketi ile kapatılmıştır. Firefox ile çalıştırıldığında “XML ayrıştırma hatası: uyumsuz biçim imi: Beklenen </ADRES>” şekilde bir mesaj ile büyük-küçük harf uyumsuzluğu bildirmektedir.

6.9.3 Etiketlerin Harf ile Başlaması : XML etiketlerinin isimleri daima bir harf ile başlamalıdır. Sayı veya diğer semboller ile başlayan etiket isimleri geçersizdir.

Yukarıdaki hatalı XML dosya kodlarını **ornek.xml** adıyla kaydederek Firefox web tarayıcısında çalıştırdığınızda Şekil 6.8’deki görüntü elde edilir.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Zarf>
    <Kimden>Mehmet Bilgin<Kimden>
    <Kime>Volkan Tunalı</Kime>
    <Adres>Mimar Sinan Mah. BURSA</Adres>
    <1.Mektup>Herkese Selamlar.</1.Mektup>
</Zarf>
```

Kod 6.6. Hatalı XML belge örneği



Şekil 6.8 XML belgede etiket isimlendirme hatası

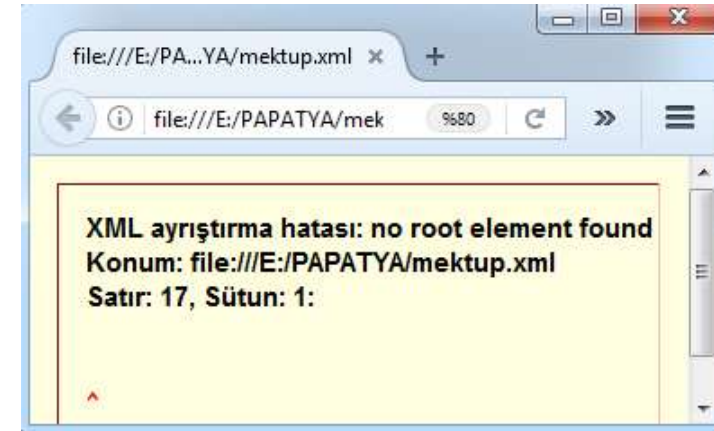
- Şekil 6.8’deki XML dosyada `<1.Mektup>` etiketi kullanılmıştır. Firefox ile çalıştırıldığında “XML ayrıştırma hatası: İyi düzenlenmiş değil” şekilde bir mesaj verilmekte ve problemin olduğu bölüm kırmızı renkli “_ _ _ _ _^” işareti ile gösterilmektedir.

6.9.4 Kök Eleman Zorunluluğu : XML dokümanı mutlaka bir kök elemana (root element) sahip olmalıdır. Kök elemana sahip olmayan XML dosyalar W3C standartlarına uygun değildir ve web tarayıcılarda hata oluşmasına sebep olurlar.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Kimden>Mehmet Bilgin<Kimden>
<Kime>Volkan Tunalı</Kime>
<Adres>Mimar Sinan Mah. BURSA</Adres>
<Mektup>Herkes e Selamlar.</Mektup>
```

Kod 6.7. Hatalı XML belge örneği

Yukarıdaki hatalı XML dosya kodlarını **ornek.xml** adıyla kaydederek Firefox web tarayıcısında çalıştırdığınızda Şekil 6.9'daki görüntü elde edilir.



Şekil 6.9 XML belgede eksik kök elemanı

- Şekil 6.9'daki XML dosyada tüm etiketleri dıştan kapsayıcı bir kök eleman bulunmuyor. Firefox ile çalıştırıldığında “XML ayrıştırma hatası: no root element found” şekilde bir mesaj verilmektedir. Bu mesaj kök eleman bulunmadığı için hata oluştuğunu bildirmektedir.

(6.10) ETİKET YERİNE NİTELİK KULLANMA

- Oluşturduğunuz XML belgelerinde her bir veri için ayrı bir etiket tanımlamak zorunda değilsiniz. Bazı verileri bir etiketin niteliği şeklinde de oluşturabilirsiniz. Örneğin, aşağıda soldaki XML dosyada cinsiyet verisi etiket olarak tanım-lanmış, sağdakinde ise personel adlı etiketin bir niteliği olarak tanımlanmıştır. Bu belgelerin her ikisi de doğru ve geçerlidir.

Kod 6.8. XML belgelerde etiket-nitelik farkı

<pre><?xml version="1.0" ?> <personel> <cinsiyet>erkek</cinsiyet> <ad>Ali</ad> <soyad>Yıldız</soyad> </personel></pre>	<pre><?xml version="1.0" ?> <personel cinsiyet="erkek"> <ad>Ali</ad> <soyad>Yıldız</soyad> </personel></pre>
--	--

- Ne zaman etiket, ne zaman nitelik kullanacağınıza dair kesin bir kural veya sınırlama yoktur, bu tamamen sizin tasarımınıza bağlıdır.

-
- **Hatırlatma** *↪ Nitelik kullanmak genellikle daha kısa belgeler oluşturduğundan biraz daha avantajlıdır.*



(6.11) XML DOKÜMANLARINDA STİL OLUŞTURMA

- HTML dili için daha fazla biçimlendirme ve daha güzel sunum sağlayan CSS gibi XML dilinde de XML belgelerinin sunumunu güzelleştirmeye yönelik XSLT (*Extensible Stylesheet Language Transformations - Genişletilebilir Biçimlendirme Dili Dönüşümleri*) adlı bir dil bulunmaktadır.
- XSLT diliyle hazırlanan biçim şablonları, HTML dilindeki içeriğe CSS eklenmesine benzer şekilde XML belgeye eklenir. XSLT dilinin tüm detayları konumuz kapsamı dışında olduğundan burada kısa bir örnek yapılarak nasıl çalıştığı anlatılacaktır.

Hatırlatma ↪ *XSLT dili ile ilgili detaylı bilgi W3C konsorsiyumu XSLT standartları sayfasından elde edilebilir.*

<https://www.w3.org/TR/xslt>

```

<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="tablostil.xsl"?>
<bilmuhbolumu>
  <ogrenci>
    <ad>Erkin</ad>
    <soyad>Beyaz</soyad>
    <sinif>3</sinif>
    <numara>1203098</numara>
    <dogumyili>1988</dogumyili>
  </ogrenci>
  <ogrenci>
    <ad>Tunç</ad>
    <soyad>Karalıoğlu</soyad>
    <sinif>1</sinif>
    <numara>1503044</numara>
    <dogumyili>1989</dogumyili>
  </ogrenci>
  <ogrenci>
    <ad>Ceyda</ad>
    <soyad>Şahoğlu</soyad>
    <sinif>4</sinif>
    <numara>1103023</numara>
    <dogumyili>1986</dogumyili>
  </ogrenci>
</bilmuhbolumu>

```

Kod 6.9. XML belgeye XSL eklenmesi

- Yukarıdaki **ogrenciler.xml** dosyası Bilgisayar Mühendisliği bölümü öğrencilerine ait Ad, Soyad, sınıf, Numara ve doğum yılı bilgileri bir XML dosyada listelenmiştir. XML dosyanın 2. Satırında `<xml-stylesheet>` etiketi ile “**tablostil.xsl**” adıyla bir biçim dosyası bu dosyaya eklenmiştir. XSLT kodları içeren dosyalar “.xsl” uzantılı olmak durumundadır. Aşağıdaki Oluşturulan XSLT biçim dosyası sayesinde XML çıktısı bir HTML tablo olarak görüntülenecektir. “**tablostil.xsl**” dosyası aşağıdaki gibidir.


```

<?xml version="1.0" ?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
  <h2>Bilgisayar Müh. Öğrencileri</h2>
  <table border="1">
    <tr bgcolor="#9acd32">
      <th style="text-align:left">Adı</th>
      <th style="text-align:left">Soyadı</th>
    </tr>
    <xsl:for-each select="bilmuhbolumu/ogrenci">
      <tr>
        <td> <xsl:value-of select="ad"/> </td>
        <td> <xsl:value-of select="soyad"/> </td>
      </tr>
    </xsl:for-each>
  </table>
  </body>
  </html>
</xsl:template>
</xsl:stylesheet>

```

Kod 6.10. Örnek XSL belge

- Bu XSLT dosyası, başlığına bakıldığında bir XML dosya gibi görünse de içinde HTML tablo kodları da olduğuna dikkat ediniz. Bu örnekte Bilgisayar mühendisliği öğrencilerinin yalnızca Adı ve Soyadı bilgileri 2 sütunlu bir tablo ile ekrana basılmaktadır. Bunun için <th> etiketi ile 2 adet sütun başlığı oluşturulmuştur. Sonrasında <xsl:for-each select="bilmuhbolumu/ogrenci"> satırı sayesinde bilmuhbolumu adlı kök elemandaki her bir öğrenci verisi otomatik olarak tekrarlanarak tabloya satır olarak ekleniyor. Öğrencinin hangi sütununun görüntüleneceğini ise <xsl:value-of select="ad"/> ve <xsl:value-of select="soyad"/> kodları ile bildirilmektedir.

ÖNEMLİ NOT: CHROME ve FIREFOX tarayıcıları 2018'den itibaren XSL komutlarının yerel bilgisayardan çalışmasını güvenlik sebebiyle engellemektedir. Bu örneği çalıştırabilmeniz için bir web sunucuya yüklemeniz gerekir.

- “ogrenciler.xml” dosyası web tarayıcıda çalıştırıldığında Şekil 6.10’daki görüntü elde edilir. Örnekte görüldüğü üzere XSLT biçim dosyaları sayesinde XML dosyaların kullanıcıya görsel sunumu gerçekleştirilmektedir.

Hatırlatma Web tarayıcıda sadece XML dosya çalıştırılmalıdır. XSLT dosyalar XML dosya tarafından çağırılır ve işletilirler. XSLT dosyalar web tarayıcıda tek başına çalıştırılmaya kalkışıldığında düz metin dosya gibi algılanırlar ve tüm kodları web tarayıcı ekranında görüntülenir.



Şekil 6.10 XML belgelerde XSL kullanımı

JSON



- **JSON** herhangi bir programlama dilinden bağımsız olan, XML alternatifi olarak üretilmiş, javascript tabanlı veri değişim formatıdır. JSON sayesinde hızlı ve küçük boyutlarda veri oluşturulabilmektedir.
- **JSON mu XML mi?**
 - JSON, XML'e göre daha küçük ve hızlıdır.
 - JSON okunabilirlik açısından daha kolay ve anlaşılabilir.
 - JSON, XML gibi geliştirilebilir bir yapıya **sahip değildir**.
 - JSON, XML'e göre daha az tekrar içerir.

JSON VE XML FARKI

```
{
  "ogrenciler": [
    {
      "numara": "12257000",
      "ad": "Ahmet Candan",
      "bolum": "Makine Muhendisligi"
    },
    {
      "numara": "12254000",
      "ad": "Mehmet Demir",
      "bolum": "Endustri Muhendisligi"
    },
    {
      "numara": "12253000",
      "ad": "Oguzhan Tasci",
      "bolum": "Bilgisayar Muhendisligi"
    }
  ]
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ogrenciler>
  <ogrenci>
    <numara>12257000</numara>
    <ad>Ahmet Candan</ad>
    <bolum>Makine Muhendisligi</bolum>
  </ogrenci>
  <ogrenci>
    <numara>12254000</numara>
    <ad>Mehmet Demir</ad>
    <bolum>Endustri Muhendisligi</bolum>
  </ogrenci>
  <ogrenci>
    <numara>12253000</numara>
    <ad>Oguzhan Tasci</ad>
    <bolum>Bilgisayar Muhendisligi</bolum>
  </ogrenci>
</ogrenciler>
```

(6.12) ÖZET

- XML, hem insanlar hem bilgisayarlar tarafından kolayca anlaşılabilen ve yorumlanabilen belgeler oluşturmayı sağlayan bir işaretleme dilidir. Veri saklamanın yanında farklı sistemler arasında veri alışverişi yapmaya yarayan bir ara katman görevi de görür. XML dili, HTML gibi bir metin işaretleme dilidir, programlama dili değildir. XML dilinin önceden tanımlı etiketleri yoktur. Kullanıcılar kendi ihtiyaçlarına göre kendi etiketlerini kendileri tanımlarlar. XML belgelerini görüntülemek için özel bir yazılıma ihtiyaç yoktur, herhangi bir web tarayıcısında veya metin editöründe kolayca görüntülenebilirler. Dezavantaj olarak ise XML belgeleri diskte fazlaca yer kaplayabilirler, bu durum işlemci ve bant genişliği açısından olumsuz bir durum olabilir. Bütün XML etiketleri bir kapatma etiketine sahip olmalıdır. XML etiketleri büyük-küçük harfe duyarlıdır. XML etiketlerinin isimleri daima bir harf ile başlamalıdır. XML dokümanı mutlaka bir kök elemana sahip olmalıdır. XML belgelerinin sunumunu güzelleştirmeye yönelik XSLT adlı bir stil tanımlama dili kullanılmaktadır.

KAYNAKÇA

[9] <https://tr.wikipedia.org/wiki/XML>

[10] <https://wiki.csc.calpoly.edu/csc560/wiki/Relative%20Region%20Coordinate>

[11] <https://tr.wikipedia.org/wiki/JSON>